

Rethinking Fano's Inequality in Ensemble Learning

Terufumi Morishita¹ Gaku Morio^{*1} Shota Horiguchi^{*1} Hiroaki Ozaki¹ Nobuo Nukaga¹

Abstract

We propose a fundamental theory on ensemble learning that evaluates a given ensemble system by a well-grounded set of metrics. Previous studies used a variant of Fano's inequality of information theory and derived a lower bound of the classification error rate on the basis of the accuracy and diversity of models. We revisit the original Fano's inequality and argue that the studies did not take into account the information lost when multiple model predictions are combined into a final prediction. To address this issue, we generalize the previous theory to incorporate the information loss. Further, we empirically validate and demonstrate the proposed theory through extensive experiments on actual systems. The theory reveals the strengths and weaknesses of systems on each metric, which will push the theoretical understanding of ensemble learning and give us insights into designing systems.

1. Introduction

Ensemble learning has had great success in various fields of machine learning. Bagging (Breiman, 1996) trains diverse models from artificial datasets built by random sub-sampling on the original one. It is common to train models with different weight initializations (Lakshminarayanan et al., 2017) or models with different network architectures (Qummar et al., 2019; Morishita et al., 2020b). While models are usually combined by voting on predictions, other methods focus on how to combine them cleverly (Omari & Figueiras-Vidal, 2015; Morio et al., 2020a). Stacking (Wolpert, 1992) trains meta-estimators that make final predictions from model predictions as their inputs. Mixture of Experts (Jacobs et al., 1991; Shazeer et al., 2017) focuses more on the models that are best specialized for a given dataset instance.

^{*}Equal contribution ¹Hitachi, Ltd. Research and Development Group, Kokubunji, Tokyo, Japan. Correspondence to: Terufumi Morishita <terufumi.morishita.wp@hitachi.com>.

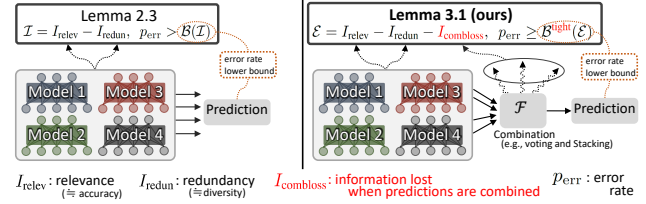


Figure 1: Previous framework (Brown, 2009; Zhou & Li, 2010) (left) and ours (right).

It has been widely believed that accurate and diverse models lead to better performance for ensemble systems. Guided by this intuition, many heuristical metrics have been proposed to measure accuracy and diversity (Kohavi et al., 1996; Skalak et al., 1996; Cunningham & Carney, 2000; Shipp & Kuncheva, 2002). However, these metrics lack theoretical grounding, and indeed, Kuncheva & Whitaker (2003) empirically showed that there are no connections between the metrics and system performance through a broad range of experiments. Turning to theoretical viewpoints, Geman et al. (1992) decomposed the squared error loss used in regression tasks into the bias and covariance of models. Bias here corresponds to accuracy and covariance diversity. For classification tasks, Tumer & Ghosh (1995) showed that the error rate reductions obtained by unweighted voting is a decreasing function of models' correlations, indicating that diverse models lead to better performance.

While the theory of Tumer & Ghosh (1995) deals with classification tasks under a limited setting, Brown (2009); Zhou & Li (2010) first derived accuracy and diversity in a general setting. Using Fano's inequality of information theory, they derived a lower bound to the error rate of a given system. Then, they decomposed the lower bound into relevance I_{relev} and redundancy I_{redun} (Lemma 2.3, illustrated in Figure 1). I_{relev} is the information theoretical version of accuracy and I_{redun} diversity. Their framework is promising as a fundamental theory of ensemble learning since it derives well-believed metrics in a general setting. However, the validity of the framework has not been examined much from both theoretical and empirical perspectives. Theoretically, we find that the framework rests on implicit assumptions used by a variant of Fano's inequality, which generally do not hold in ensemble learning. As a result, the framework fails

in capturing important aspects of ensemble learning. Empirically, the experiments of the studies were not extensive enough to justify the framework. In particular, they did not check whether the framework can predict representative phenomena in ensemble learning.

In this paper, we rethink the theoretical framework from both perspectives. We first revisit the theory (Sections 2 and 3). We argue that the framework does not take into account the information lost when multiple model predictions are combined into a single final prediction. We call this information loss *combination loss*. To address the issue, we propose a generalized framework that incorporates the third metric of combination loss I_{combloss} based on original Fano’s inequality (Lemma 3.1, illustrated in Figure 1). We also solve the issue of the previous framework producing a loose lower bound when the number of classes is small.

Next, we turn to empirical viewpoints. We first validate the proposed framework in Sections 4 and 5. In contrast to the previous studies, (i) we directly check whether the framework can predict phenomena in ensemble learning, (ii) we use various ensemble systems (Table 2), and (iii) we use various tasks (Table E.10). Additionally, to be modern and realistic, we use state-of-the-art DNNs such as BERT (Devlin et al., 2019), and the tasks are chosen from widely-used benchmarks such as GLUE (Wang et al., 2018). Extensive experiments reveal that the previous framework can not predict phenomena such as the performance ranking of ensemble systems (Figure 2) and performance scaling behavior (Figure 3), ignoring combination loss. These results refute the previous framework. In contrast, the proposed framework justifies itself by predicting all these phenomena. Finally, we demonstrate the proposed framework (Section 6). We analyze DNN ensemble systems and answer *why* a system performs well or badly through its strengths and weaknesses in terms of the three metrics (Table 4). Such analysis pushes the theoretical understanding of ensemble learning and gives us insights into designing systems. In summary,

- We propose a fundamental theoretical framework that measures a given ensemble system from a well-grounded set of metrics: relevance I_{relev} , redundancy I_{redun} , and combination loss I_{combloss} . The metrics are tied to the bound on the performance of a system. The framework applies to any ensemble system.
- We validate the framework through extensive experiments on DNN ensemble systems.
- We demonstrate the framework. We analyze the DNN ensemble systems and answer why a system performs well or badly as follows:
 1. Systems with models that simply differ in the training seeds perform well because the models

are accurate (large I_{relev}) and combinable (small I_{combloss}).

2. Heterogeneous systems, which use various types of DNNs, also perform well. While some DNNs are inaccurate (small I_{relev}), DNNs are diverse (small I_{redun}). Further, such systems should perform the best among all the systems when DNNs are combined by meta-estimators.
3. Bagging-based systems do not perform that well. Their models are diverse (small I_{redun}) but inaccurate (small I_{relev}) and uncombinable (large I_{combloss}).
4. Systems with models with randomly chosen hyperparameters do not perform that well. The models are diverse (small I_{redun}) but inaccurate (small I_{relev}).
5. Meta-estimators generally push the performance of the systems by combining models smartly to reduce I_{combloss} . Further, meta-estimators benefit systems such as 2 and 4 the most since the amount of information of the true label is unevenly distributed on models of varied accuracies and such information is recovered well by meta-estimators. Finally, a simple estimator such as logistic regression should be enough on strong DNNs.

- We plan to release our code as open source.

2. Conventional framework based on variant of Fano’s inequality

2.1. Fano’s inequality

Let $Y \in \{1, 2, \dots, Y_{\max}\}$ be a discrete stochastic variable representing the input and $\mathbf{O} \in \mathbb{R}^m$ be m stochastic variables representing an observation after a noisy channel. We want to recover Y from \mathbf{O} by using the reconstruction function $\mathcal{F} : \mathbf{O} \mapsto \hat{Y} \in \{1, 2, \dots, Y_{\max}\}$. Note that $Y \Rightarrow \mathbf{O} \Rightarrow \hat{Y}$ forms a Markov chain. Fano’s inequality relates the information lost in a noisy channel to the error rate when recovering the input as follows.

Lemma 2.1 (Fano’s inequality (Fano, 1961)). *For any function \mathcal{F} , the following holds:*

$$\mathcal{H}_2(p_{\text{err}}) + p_{\text{err}} \log_2(Y_{\max} - 1) \geq H(Y | \hat{Y}), \quad (1)$$

where $p_{\text{err}} = \Pr[\hat{Y} \neq Y] \in [0, 1]$ is the reconstruction error rate, $\mathcal{H}_2(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)$ binary cross entropy, and $H(Y | \hat{Y})$ conditional entropy (C.2).

From the Markovness, the amount of information carried by \hat{Y} is never more than that carried by \mathbf{O} ; thus, the right-hand side of (1) is lower bounded as

$$H(Y | \hat{Y}) \geq H(Y | \mathbf{O}). \quad (2)$$

Since the binary cross entropy never exceeds one, the left side of Lemma 2.1 is upper bounded as

$$\begin{aligned} \mathcal{H}_2(p_{\text{err}}) + p_{\text{err}} \log_2(Y_{\max} - 1) &\leq 1 + p_{\text{err}} \log_2(Y_{\max} - 1), \\ &< 1 + p_{\text{err}} \log_2(Y_{\max}) \quad (3) \end{aligned}$$

From (1)–(3), we obtain the following well-known variant of Fano's inequality:

Lemma 2.2 (An error rate lower bound (Fano, 1961)).

$$p_{\text{err}} > \frac{H(Y | \mathbf{O}) - 1}{\log_2 Y_{\max}}.$$

2.2. Error rate lower bound of ensemble systems

In ensemble learning, Y denotes a label on a given instance, and $\mathbf{O} = \{O_1, O_2, \dots, O_N\}$ is the output from N models. Note that the output from i -th model $O_i \in \mathbb{R}^{Y_{\max}}$ can be a predicted label ($Y_{\max} = 1$) or class probabilities ($Y_{\max} \geq 2$). \mathcal{F} denotes a model combination method such as voting or Stacking. Lemma 2.2 gives a lower bound of the classification error rate p_{err} of an ensemble system.

Brown (2009) decomposed the lower bound into relevance and redundancy, the formulation of which was later simplified by Zhou & Li (2010) as follows:

Lemma 2.3 (Zhou & Li, 2010).

$$p_{\text{err}} > \mathcal{B}(\mathcal{I}(\mathbf{O}, Y)) := \frac{H(Y) - \mathcal{I}(\mathbf{O}, Y) - 1}{\log_2 Y_{\max}}. \quad (4)$$

$\mathcal{I}(\mathbf{O}, Y)$ is defined as follows:

$$\mathcal{I}(\mathbf{O}, Y) := I_{\text{relev}}(\mathbf{O}, Y) - I_{\text{redun}}(\mathbf{O}, Y), \quad (5)$$

$$I_{\text{relev}}(\mathbf{O}, Y) := \sum_{i=1}^N I(O_i; Y),$$

$$I_{\text{redun}}(\mathbf{O}, Y) := I_{\text{multi}}(\mathbf{O}) - I_{\text{multi}}(\mathbf{O} | Y),$$

where H denotes entropy (C.1), I denotes mutual information (C.3), and I_{multi} denotes multi-information (C.4) to (C.5), a multivariate generalization of mutual information.

Since $H(Y)$ and Y_{\max} are constants given a machine learning task, the important term in (4) is $\mathcal{I}(\mathbf{O}, Y)$ defined in (5), which denotes the amount of unique information on Y carried by \mathbf{O} . The first term $I_{\text{relev}}(\mathbf{O}, Y)$ is the *relevance*, whose component $I(O_i; Y)$ denotes the amount of information on Y given by O_i . It can be seen as the accuracy of the model i from the information theoretical point of view. The second term $I_{\text{redun}}(\mathbf{O}, Y)$ is the *redundancy*, which indicates how strongly the model outputs $\mathbf{O} = \{O_1, O_2, \dots, O_N\}$ are correlated with each other. In other words, it describes the amount of redundant (duplicated) information. Overall, Lemma 2.3 reveals that an ensemble system should include accurate (large I_{relev}) and diverse (small I_{redun}) models to get a small lower bound for the error rate $\mathcal{B}(\mathcal{I})$.

3. Proposed framework based on original Fano's inequality

3.1. Error rate lower bound with better properties

To derive Lemma 2.2, which is the basis of Lemma 2.3, two bounds, (2) and (3), are used. However, in an ensemble learning context, both are not tight, so Lemma 2.3 would not give a good approximation of the lower bound.

The problem with relying on (2) is that the existence of a perfect reconstruction function \mathcal{F} is implicitly assumed. In the information theoretical context, using the noisy-channel coding theorem (Shannon, 1948), we can construct a smart reconstruction function \mathcal{F} so that the information lost by \mathcal{F} is zero as $H(Y | \hat{Y}) - H(Y | \mathbf{O}) \rightarrow 0$. Thus, the equality in (2) holds. On the other hand, in the ensemble learning context, we usually use a simple function such as voting or a meta-estimator trained on a limited amount of data as \mathcal{F} . Therefore, the information loss $H(Y | \hat{Y}) - H(Y | \mathbf{O})$ caused by combining the outputs from multiple models \mathbf{O} into a single prediction \hat{Y} should also be taken into account. We refer to this loss as *combination loss*.

The problem with relying (3) is that an exponentially large number of classes is assumed, i.e., $Y_{\max} \gg 1$. In information theory, Y is assumed to be a sequence of symbols (e.g., bits). Suppose that the sequence length $L \gg 1$ and that there are C types of symbols; Y_{\max} becomes exponentially large as $Y_{\max} = C^L$. Then, the second term of the left-hand side of (3) is approximated as $p_{\text{err}} \log_2(Y_{\max} - 1) \approx p_{\text{err}} L \log_2 C \gg 1$. Since the first term ($\mathcal{H}_2(p_{\text{err}}) \leq 1$) becomes negligible, it can be safely replaced with its upper bound (i.e. 1) without loosening the inequality much. On the other hand, in the ensemble learning context, the number of classes Y_{\max} can be small; thus, simply neglecting $\mathcal{H}_2(p_{\text{err}})$ produces a loose bound. For example for binary classification problems, the bound by Lemma 2.2 is *always negative* as $\frac{H(Y | \mathbf{O}) - 1}{\log_2 Y_{\max}} \leq 0$ because $H(Y | \mathbf{O}) \leq 1$ when $Y_{\max} = 2$.

To address these two problems, we lower bounded the error rate using the original Fano's inequality (Lemma 2.1) directly:

Lemma 3.1 (Decomposition of error rate lower bound into three metrics). *Let $\mathcal{U}(p) = \mathcal{H}_2(p) + p \log_2(Y_{\max} - 1)$ and $\mathcal{U}'(p) = \frac{d\mathcal{U}}{dp}(p)$, and let $p_0 \in [0, 1]$ be the approximate error rate. Then, for any p_0 , the error rate p_{err} is bounded as*

$$\begin{aligned} p_{\text{err}} &\geq \mathcal{B}_{p_0}^{\text{tight}}(\mathcal{E}(\mathbf{O}, Y, \hat{Y})) \\ &:= p_0 + \frac{\mathcal{U}'(p_0)}{4} \left\{ 1 - \sqrt{1 - 8 \frac{H(Y) - \mathcal{E}(\mathbf{O}, Y, \hat{Y}) - \mathcal{U}(p_0)}{\mathcal{U}'(p_0)^2}} \right\}, \end{aligned} \quad (6)$$

Table 1: Extreme toy ensemble systems on imaginary binary classification task for discussing combination loss (Section 3.2). Each row shows predicted labels on instance from dataset. $\mathbf{O} = \{O_1, \dots, O_5\}$: model predictions, $\hat{Y} = \mathcal{F}(\mathbf{O})$: ensemble prediction, and Y : ground-truth label. Red **0/1** shows wrong ensemble predictions. Orange **0/1** shows correct but neglected model predictions. Blue **0** shows correct prediction recovered by weighted voting. Tables 1b and 1c use the same \mathbf{O} .

(a) \hat{Y}_{vote} : voting on \mathbf{O} .			(b) \hat{Y}_{vote} : voting on \mathbf{O} .			(c) $\hat{Y}_{w.\text{vote}}$: just using O_1 .			(d) \hat{Y}_{vote} : voting on \mathbf{O} , $\hat{Y}_{w.\text{vote}}$: on O_3-5			
\mathbf{O}	\hat{Y}_{vote}	Y	\mathbf{O}	\hat{Y}_{vote}	Y	\mathbf{O}	$\hat{Y}_{w.\text{vote}}$	Y	\mathbf{O}	\hat{Y}_{vote}	$\hat{Y}_{w.\text{vote}}$	Y
11111	1	1	11100	1	1	11100	1	1	11100	1	0	1
11111	1	1	11111	1	1	11111	1	1	11111	1	1	1
11111	1	1	10011	1	1	10011	1	1	000 11	0	1	1
...
00000	0	0	01101	1	0	01101	0	0	111 00	1	0	0
11111	1	0	00000	0	0	00000	0	0	00000	0	0	0
00000	0	0	00011	0	0	00011	0	0	00 011	0	1	0
...

where the ensemble strength $\mathcal{E}(\mathbf{O}, Y, \hat{Y})$ is given by

$$\begin{aligned} \mathcal{E}(\mathbf{O}, Y, \hat{Y}) &:= I_{\text{rele}}(\mathbf{O}, Y) - I_{\text{redun}}(\mathbf{O}, Y) \\ &\quad - I_{\text{combloss}}(\mathbf{O}, Y, \hat{Y}), \\ I_{\text{combloss}}(\mathbf{O}, Y, \hat{Y}) &:= H(Y|\hat{Y}) - H(Y|\mathbf{O}). \end{aligned} \quad (7)$$

Proof. In Lemma 2.1, we expand $\mathcal{H}_2(p_{\text{err}})$ by using strong convexity and solve for p_{err} . Appendix D.1 shows the proof.

Lemma 3.1 differs from Lemma 2.3 in that (i) the ensemble strength \mathcal{E} (7) includes the third metric of combination loss, and (ii) the bound function is tighter¹: $\mathcal{B}_{p_0}^{\text{tight}}(E) \geq \mathcal{B}(E)$, which is the result of removing the large Y_{max} assumption.

Since $\mathcal{E} = \mathcal{I} - I_{\text{combloss}}$ holds, \mathcal{E} denotes the amount of unique information on Y carried by \mathbf{O} that can be extracted when a combination \mathcal{F} is applied to \mathbf{O} . $\mathcal{B}_{p_0}^{\text{tight}}$ is still a decreasing function of \mathcal{E} when $p_0 \in [0, \frac{Y_{\text{max}}-1}{Y_{\text{max}}}]$, where $\frac{Y_{\text{max}}-1}{Y_{\text{max}}}$ denotes the error rate of a random-guessing system on a balanced label dataset. Thus, Lemma 3.1 reveals that an ensemble system should include accurate (large I_{rele}) and diverse (small I_{redun}) models and keep I_{combloss} small in order to have a small lower bound.

3.2. What kind of systems produce combination loss?

To clarify in what kind of ensemble systems combination loss becomes apparent, four toy ensemble systems on an imaginary binary classification task are shown in Table 1. The systems differ in terms of models $\mathbf{O} = \{O_1, O_2, O_3, O_4, O_5\}$ or combination function \mathcal{F} . Although the systems examined here are extremely simplified and the claims here are hypothetical, they can illustrate certain aspects of empirical behaviors of ensemble systems as discussed in Section 6.

Table 1a shows the case where the outputs from each model in \mathbf{O} are perfectly correlated, i.e., there is no diversity be-

tween models. Information theoretically, the system has large redundancy I_{redun} . In this case, simple voting \hat{Y}_{vote} does not lose any information carried by \mathbf{O} , so the combination loss is trivially zero.

Tables 1b and 1c show the cases where the models differ in accuracy, among which O_1 performs best. Information theoretically, the amount of information on Y given by the models is unevenly distributed on the models, and especially concentrated on O_1 . Note that the same model set is shown in both tables. If naive voting is used for model combination (Table 1b), it produces a prediction error **1** even though some of the models (O_1 and O_4 in this case) give correct predictions **0**. These correct but neglected minorities are the source of combination loss. On the other hand, if weighted voting that focuses more on the best model (i.e., O_1) is used (Table 1c), it will succeed in recovering the correct prediction, **0**.

Table 1d shows the case where the models' outputs are diverse but have the same accuracy. Information theoretically, information on Y given by \mathbf{O} is uniformly distributed on all the models. In this case, weighted voting will not help much in recovering the correct predictions compared with simple voting, since there are no better models to be focused on.

From the discussion above, it is expected that (i) models' redundancy decreases combination loss, and (ii) smart combination functions help reduce combination loss, especially when the accuracies of models are varied.

4. Experiments

We empirically validate and demonstrate Lemma 3.1. To this end, we built various ensemble systems and measured their error rates, error rate lower bounds, and the three metric values. To be modern and realistic, we built ensemble systems on top of state-of-the-art DNNs, specifically pre-trained language models such as BERT (Devlin et al., 2019).

¹If p_0 is not far from the lower bound values (Appendix D.3)

Table 2: Ensemble methods used in this study. We built 16 ensemble systems using all combinations of model generation and combination methods. Note that Stacking has three variations (i.e., LogR, SVM and RForest). All generation methods train N (≤ 30) models using different seed for each model. Seed affects random aspects of training, e.g., weight initialization or hidden units dropped when using dropout. See Section 4.2 for details.

Type	Method	Description
Model Generation	Random-HyP	Train models with different hyperparameters randomly sampled around the best value.
	Bagging	Train models using different dataset instance sets. Each set contains instances randomly sampled from the original dataset.
	Random-Seed	Train models that differ only in the seed of fine-tuning.
	Hetero-DNNs	Train models from $L(=5)$ types of DNNs. M models from each type so that $L \times M = N$.
Model Combination	Voting	Take a majority vote on labels predicted by models.
	Stacking (LogR SVM RForest)	Use meta-estimators that make prediction from outputs of models as inputs. We used two-layered stacking with a single meta-estimator, which takes predicted labels as inputs. We trained logistic regression (LogR), Support Vector Machine (Platt, 1999) with RBF kernel (SVM) and Random Forest (Breiman, 2001) (RForest) as meta-estimators.

We used various tasks from the GLUE and SuperGLUE benchmarks (Wang et al., 2018; 2019). These benchmarks include challenging tasks from different domains of NLP and are commonly used to compare state-of-the-art models.

Below, we briefly describe these setups. For reproducibility, we show the details in Appendix E and release the code.

4.1. Models

We fine-tuned the following five types of language models on downstream tasks: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), ALBERT (Lan et al., 2020), and BART (Lewis et al., 2020).

4.2. Ensemble systems

To build an ensemble system, we must specify a model generation method (i.e., how to train models that produce \mathbf{O}) and a combination method (i.e., \mathcal{F}). We used well-established methods that can be used with DNNs (Table 2). These methods are commonly used with DNNs in a wide range of domains (Kumar et al., 2016; Liu et al., 2017; Qummar et al., 2019; Ma & Chu, 2019), especially in competitions where the highest performance is required (Szegedy et al., 2015; Yan et al., 2015; Atwood et al., 2020; Morishita et al., 2020a; Morio et al., 2020b). We built 16 systems using all the combinations of generation and combination methods.

For later convenience, we define the baseline system s_0 in each task, which is a single DNN (i.e., no-ensemble) that performs the best among DNNs: ELECTRA for the MRPC/Boolq/SST and RoBERTa for the other tasks.

Random-Seed, Random-HyP and Bagging used a single DNN type the same as s_0 . Hetero-DNNs used $L=5$ DNN

types.

4.3. Estimation of metric values and lower bound

We estimated the three metric values (I_{rele} , I_{redun} , and I_{combloss}) and the other quantities appearing in Lemmas 2.3 and 3.1 on the basis of the observed frequency distribution of the labels (\mathbf{O} , \hat{Y} , Y). Then, we computed the lower bounds by Lemmas 2.3 and 3.1. All such operations were done on *test sets* to discuss generalization performance ².

To tackle the count sparsity of high-dimensional variables $\mathbf{O} = \{O_i \mid 1 \leq i \leq N, O_i \in \{1, 2, \dots, Y_{\max}\}\}$, we used the trick of $\text{MTI}_{k=3}$ introduced by Zhou & Li (2010).

We set the approximate error rate p_0 in (6) as the error rate of the baseline s_0 . Below, we simply denote $\mathcal{B}_{p_0}^{\text{tight}}$ as $\mathcal{B}^{\text{tight}}$.

4.4. Tasks

We used eight classification tasks with moderately-sized datasets for computational reasons: Boolq (Clark et al., 2019), CoLA (Dolan & Brockett, 2005), Cosmos QA (Khot et al., 2018), MNLI (Williams et al., 2018), MRPC (Dolan & Brockett, 2005), SciTail (Khot et al., 2018), SST (Socher et al., 2013), and QQP.

4.5. Computational resources / experimental runs

A single run of experiments required about 200 GPUs ($V100$) \times 1 day. We ran the experiments three times.

²This makes sense since our aim is not to “predict” these quantities on unseen test set from the training/valid sets, but to interpret the test performance by decomposing it into the three metric values.

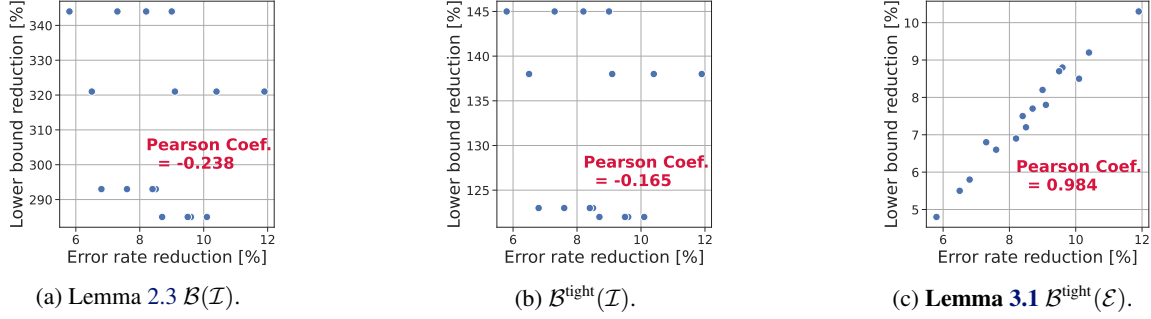


Figure 2: Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in figures shows quantity of specific ensemble system s , and quantity is average over eight tasks. See Table 4a for real value of each point. We used 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. Baseline values in (8) and (9) are: $\text{ER}(s_0)$: 15.5 %, $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.8 %, $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.8 %, and $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.0 %.

5. Validation of framework through its predictive power to ensemble phenomena

We show that we can predict various phenomena observed on actual ensemble systems using Lemma 3.1. We show the results aggregated over the eight tasks here and those for each task in Appendix K. The discussions here are valid for all tasks, showing their significance.

Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$ differs from Lemma 2.3 $\mathcal{B}(\mathcal{I})$ in two ways, i.e., it has a tightened bound function $\mathcal{B}^{\text{tight}}$ and ensemble strength with combination loss \mathcal{E} . To separate contribution of each, we analyze three types of lower bounds hereafter: $\mathcal{B}(\mathcal{I})$, $\mathcal{B}^{\text{tight}}(\mathcal{I})$, and $\mathcal{B}^{\text{tight}}(\mathcal{E})$.

5.1. Effect of bound function $\mathcal{B}^{\text{tight}}$

First, as theoretically expected, the lower bound $\mathcal{B}^{\text{tight}}(\mathcal{I})$ was tighter than Lemma 2.3 $\mathcal{B}(\mathcal{I})$, for example for the baseline system s_0 , $\mathcal{B}^{\text{tight}}(\mathcal{I}_{s_0}) = 2.8\%$ and $\mathcal{B}(\mathcal{I}_{s_0}) = -2.0\%$ (average of eight tasks). Tables K.15 to K.22 show these lower bounds for eight tasks.

5.2. Correlation between error rate and lower bound

The error rate lower bound denotes the best-case error rate. Thus, a system with a smaller lower bound has higher chance of having a smaller error rate (Brown, 2009; Zhou & Li, 2010). Guided by this intuition, we measured the correlation between the error rates and lower bounds of the ensemble systems.

Figure 2 plots the following normalized versions of the error

Table 3: Pearson correlation coefficients between error rate reduction and lower bound reduction. In each task, we used the 16 ensemble systems described in Section 4.2, and each system used $N = 15$ models.

Task	Lower bound type		
	Lemma 2.3 $\mathcal{B}(\mathcal{I})$	$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$
Boolq	0.341	0.330	0.910
CoLA	-0.211	-0.210	0.991
CosmosQA	-0.324	-0.320	1.000
MNLI	0.226	0.216	0.961
MRPC	0.332	0.252	0.989
QQP	-0.131	-0.076	0.998
SciTail	-0.237	-0.191	0.966
SST	-0.242	-0.252	0.998
average ³	-0.238	-0.165	0.984

rate and lower bound for each ensemble system s :

$$\text{ErrorRateReduction}(s) = \frac{\text{ER}(s_0) - \text{ER}(s)}{\text{ER}(s_0)} \times 100 [\%], \quad (8)$$

$$\text{LowerBoundReduction}(s) = \frac{\text{LB}(s_0) - \text{LB}(s)}{|\text{LB}(s_0)|} \times 100 [\%]. \quad (9)$$

s_0 denotes the single DNN baseline defined in Section 4.2. $\text{ER}(s)$ denotes the error rate (i.e., $100\% - \text{accuracy}$) and $\text{LB}(s)$ the lower bound. Note that Pearson correlation coefficient is invariant under this transformation.

Neither the lower bound reduction by Lemma 2.3 $\mathcal{B}(\mathcal{I})$ nor that by $\mathcal{B}^{\text{tight}}(\mathcal{I})$ correlated with the error rate reduction, as shown in Figures 2a and 2b. In addition, Lemma 2.3 $\mathcal{B}(\mathcal{I})$ predicted the same lower bound reduction value for different

³The correlation coefficient between the averaged error rate reductions and lower bound reductions. The average is taken over the eight tasks.

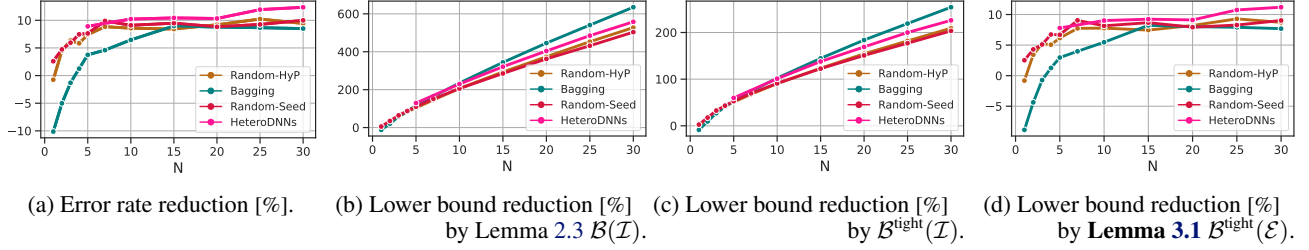


Figure 3: Change in error rate reduction and lower bound reduction when number of models N was changed. Each value is an average of eight tasks. Ensemble systems used SVM model combination.

systems that share the same model generation method. This behavior of the lower bounds can be seen from the points on the same horizontal lines in Figure 2a. This behavior is theoretically expected: since Lemma 2.3 $\mathcal{B}(\mathcal{I})$ does not include I_{combloss} , it does not consider model combination methods. This behavior was also observed on $\mathcal{B}^{\text{tight}}(\mathcal{I})$ for the same reason.

By contrast, the lower bound reduction by Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$ was very strongly correlated with the error rate reduction, as shown in Figure 2c. Strong correlations were observed for all eight tasks (Table 3) and also for different N s (Tables G.11 to G.14). These results justify Lemma 3.1 and show that $\mathcal{B}^{\text{tight}}(\mathcal{E})$ can be used for comparing systems. These results also show the importance of combination loss given that the only difference between $\mathcal{B}^{\text{tight}}(\mathcal{E})$ and $\mathcal{B}^{\text{tight}}(\mathcal{I})$ is combination loss.

5.3. Predicting error rate scaling curve

Figure 3 shows the change in error rate reduction and lower bound reductions when the number of models N was changed.

Both Lemma 2.3 $\mathcal{B}(\mathcal{I})$ (Figure 3b) and $\mathcal{B}^{\text{tight}}(\mathcal{I})$ (Figure 3c) could not predict the shape of the error rate reduction curve (Figure 3a), especially the saturation over $N \gtrsim 15$. By contrast, Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$ (Figure 3d) could predict such phenomena. The results again justify Lemma 3.1 and show the importance of combination loss.

Refer to Appendix H for more detailed discussions, where we examine the scaling property of each metric values.

6. Analysis of ensemble systems by framework

We demonstrate how we can reveal the strengths and weaknesses of the systems on the basis of the metrics in Lemma 3.1. The results here are summarized in Section 1. We show the results aggregated over the eight tasks here and those for each task in Appendix K. The discussions here are

valid for all tasks, showing their significance.

6.1. Justification of three metrics for ensemble system analysis

Table 4 shows the statistics of the ensemble systems. First, the ranking of the lower bound reduction by $\mathcal{B}^{\text{tight}}(\mathcal{E})$ in Table 4a matches the ranking of \mathcal{E} in Table 4b. This is theoretically expected because $\mathcal{B}^{\text{tight}}$ is a decreasing function. Thus, \mathcal{E} can be used for comparing systems, instead of $\mathcal{B}^{\text{tight}}(\mathcal{E})$. Furthermore, since \mathcal{E} is decomposed into the three metrics (I_{relev} , I_{redun} , I_{combloss}) as in (7), the three metrics can be used to analyze ensemble systems.

Below, we use *per-model* metrics $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$ for intuitive understanding.

6.2. Analysis of model generation methods

Random-Seed and Hetero-DNNs systems performed the best or second best in each column of Table 4a (i.e. among the systems with the same combination method). Looking into the per-model relevance i_{relev} in Table 4b, Random-Seed had the largest i_{relev} in each column. i_{relev} denotes the average accuracy of the models. Indeed, the ranking of i_{relev} coincided with the ranking of the average error rate shown as “avg” in Table 5. Random-Seed had the most accurate models because it used only the best DNN type (cf. Hetero-DNNs), all the dataset instances (cf. Bagging), and only the best hyperparameter (cf. Random-HyP).

On the per-model redundancy i_{redun} , Hetero-DNNs had a value smaller than that of Random-Seed (i.e., it had more diverse models), benefitting from the diverse DNN types.

For per-model combination loss i_{combloss} ⁴, Random-Seed had the smallest value in the voting column. We attribute this to it having lowest diversity (i.e., the largest i_{redun}), similarly to Table 1a. However, the meta-estimators (LogR,

⁴The magnitude of i_{combloss} is smaller than those of i_{relev} and i_{redun} . However, i_{relev} and i_{redun} are strongly correlated, and thus, i_{combloss} is not negligible compared with $i_{\text{relev}} - i_{\text{redun}}$, as shown. Thus, combination loss is significant.

Table 4: Statistics of ensemble systems described in Section 4.2. Rows and columns list model generation and combination methods of Table 2, respectively. Each cell shows quantity of specific system s . Each quantity is average over eight tasks. Each system contains $N = 15$ models. Color shows rank within *each column* (brighter is better).

(a) Error rate and lower bound reductions. Baseline values used in (8) and (9) were $\text{ER}(s_0)$: 15.5 %, $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.8 %, $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.8 %, and $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.0 %.

	Error rate reductions [%]				Lower bound reductions by $\mathcal{B}^{\text{tight}}(\mathcal{E})$ [%]			
	Voting	LogR	SVM	RForest	Voting	LogR	SVM	RForest
Random-HyP	6.8 \pm 1.4	8.5 \pm 0.9	8.4 \pm 1.2	7.6 \pm 0.7	5.8 \pm 1.4	7.2 \pm 1.0	7.5 \pm 1.1	6.6 \pm 0.7
Bagging	7.3 \pm 2.0	8.2 \pm 1.9	9.0 \pm 1.9	5.8 \pm 2.0	6.8 \pm 2.1	6.9 \pm 2.0	8.2 \pm 2.1	4.8 \pm 2.0
Random-Seed	9.6 \pm 1.2	10.1 \pm 0.7	9.5 \pm 0.7	8.7 \pm 0.2	8.8 \pm 1.2	8.5 \pm 0.7	8.7 \pm 0.8	7.7 \pm 0.1
Hetero-DNNs	6.5 \pm 1.4	11.9 \pm 0.8	10.4 \pm 1.5	9.1 \pm 1.9	5.5 \pm 1.4	10.3 \pm 0.8	9.2 \pm 1.5	7.8 \pm 1.9

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (i_{\text{relev}} - i_{\text{redun}} - i_{\text{combloss}}) \times N$ holds. For intuitive understanding, all values are normalized by ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is raw value.

	Ensemble strength \mathcal{E}				Per-model metric values						
	Voting	LogR	SVM	RForest	i_{relev}	i_{redun}	i_{combloss}				$i_{\text{relev}} - i_{\text{redun}}$
							Voting	LogR	SVM	RForest	
Baseline (s_0)	100 (the raw value is 0.478)				100	0	0	0	0	0	100
Random-HyP	105.0 \pm 1.4	107.4 \pm 1.0	107.5 \pm 1.3	105.2 \pm 0.7	89.4 \pm 0.9	74.5 \pm 0.9	7.96 \pm 0.36	7.80 \pm 0.34	8.00 \pm 0.37	7.94 \pm 0.29	15.0 \pm 1.3
Bagging	105.3 \pm 1.9	105.7 \pm 1.8	108.0 \pm 1.1	103.1 \pm 1.4	90.1 \pm 0.3	73.5 \pm 0.3	9.56 \pm 0.08	9.54 \pm 0.03	9.40 \pm 0.05	9.71 \pm 0.05	16.6 \pm 0.4
Random-Seed	109.2 \pm 1.1	108.5 \pm 0.9	108.8 \pm 1.3	107.7 \pm 1.0	100.0 \pm 0.0	84.9 \pm 0.3	7.79 \pm 0.21	7.84 \pm 0.26	7.82 \pm 0.19	7.89 \pm 0.23	15.1 \pm 0.3
Hetero-DNNs	104.5 \pm 1.1	110.9 \pm 0.8	110.6 \pm 1.0	107.8 \pm 1.7	86.0 \pm 0.4	69.9 \pm 0.2	9.16 \pm 0.24	8.73 \pm 0.26	8.75 \pm 0.29	8.94 \pm 0.19	16.1 \pm 0.4

SVM, and RForest) reduced i_{combloss} more on Hetero-DNNs than on Random-Seed. This pushed the performance of Hetero-DNNs to the highest among all the systems. Regarding this phenomenon, Hetero-DNNs can be analogous to Tables 1b to 1c and Random-Seed to Table 1d: since Hetero-DNNs uses various DNN types of varied accuracies, the amount of information on Y is concentrated more on better models compared with Random-Seed. Thus, Hetero-DNNs benefitted more from the meta-estimators, which focused on these models and recovered the information to reduce i_{combloss} , similarly to the transition from Tables 1b to 1c. This phenomenon did not occur in Random-Seed since it uses models of similar accuracies, similarly to Table 1d.

Indeed, we can see the information concentration and how the meta-estimator handled such information more directly. To this end, we propose an auxiliary metric of n -model concentration Conc_n^N (Appendix J) which measures the degree to which the amount of information given by N models $\mathbf{O} = \{O_1, \dots, O_N\}$ is concentrated on the top- n models $\Omega_n^{N, \max}$:

$$\text{Conc}_n^N(\mathbf{O}, Y) = \frac{I(\Omega_n^{N, \max}; Y) - I(\Omega_n^{N, \min}; Y)}{I(\mathbf{O}; Y)} \in [0, 1], \quad (10)$$

$$I(\Omega_n^{N, \max/\min}; Y) = \max/\min_{\{i_1, i_2, \dots, i_n\} \in \Omega_n^N} I(\{O_{i_1}, O_{i_2}, \dots, O_{i_n}\}; Y).$$

Table 5 shows $\text{Conc}_{n=3}^{N=15}$ for each model generation method. Intuitively, Conc_n^N and the standard deviation of

model error rates, which denotes the variety in accuracies, were strongly correlated. Hetero-DNNs had a larger Conc_n^N and Random-Seed a smaller one, as expected. Table 6 shows that the meta-estimator for Hetero-DNNs distributed weight W_t to each DNN type t in accordance with its error rate. Overall, we can see a clear analogy of Hetero-DNNs to Tables 1b and 1c, and of Random-Seed to Table 1d.

Bagging and Random-HyP performed the third or fourth best in each column of Table 4a. Similarly to Hetero-DNNs, they had a smaller i_{relev} and i_{redun} compared with Random-Seed (Table 4b). The smaller i_{relev} is attributed to Bagging using smaller subsets of training instances and Random-HyP using randomly sampled non-optimal hyperparameters, which degraded model accuracies. The smaller i_{redun} is due to the diverse instance sets of Bagging and the diverse hyperparameters of Random-HyP.

Bagging had the largest i_{combloss} in each column, and more importantly, the meta-estimators (LogR, SVM and RForest) could not reduce i_{combloss} as much as they could on Hetero-DNNs and Random-HyP. This phenomenon should be due to the Bagging's smaller Conc_n^N (Table 5), which is the result of models of similar accuracies, similarly to Table 1d. Such models were generated because Bagging used dataset sub-sets of the same size.

Table 5: The information concentration metric $Conc_{n=3}^{N=15}$. See (10). Color shows rank (brighter is better) in each column. Values are averages over eight tasks.

Model generation	$Conc_{n=3}^{N=15}$	Error rates of models [%]	
		avg.	std.
Baseline (s_0)	0	16.1 \pm 0.9	-
Random-HyP	0.28 \pm 0.02	17.3 \pm 0.1	3.4 \pm 0.2
Bagging	0.08 \pm 0.00	17.1 \pm 0.0	0.8 \pm 0.0
Random-Seed	0.08 \pm 0.00	15.5 \pm 0.1	0.7 \pm 0.1
Hetero-DNNs	0.20 \pm 0.00	18.1 \pm 0.0	2.3 \pm 0.0

Table 6: Logistic regression meta-estimator weight W_t distributed to each DNN type t . $N=15$ models are generated by Hetero-DNNs (i.e., 3 models per DNN type). Values are averages over eight tasks. See Appendix F.2 for details.

DNN t	Average error rate of models [%]	W_t
RoBERTa	15.1 \pm 0.3	0.49
ELECTRA	17.0 \pm 0.1	0.40
BART	17.9 \pm 0.1	0.25
BERT	18.7 \pm 0.1	0.24
ALBERT	20.4 \pm 0.1	0.21

6.3. Analysis of model combination methods

Stacking (i.e., LogR, SVM, and RForest) generally outperformed voting in each row of Table 4a. This is due to the smaller i_{combloss} since $i_{\text{rele}}v$ and i_{redun} are the same in each row. Simply, the meta-estimators combined the models better to reduce i_{combloss} .

Interestingly, the simple meta-estimator of LogR performed on par with or better than the complex ones of SVM and RForest. We estimate that the DNN’s predictions were so good that simple combinations were enough, and complex ones were superfluous.

7. Other considerations

The limitations of the study are listed in Appendix A. Ethical matters and social impacts are discussed in Appendix B.

8. Conclusion

We proposed a novel and fundamental theoretical framework that measures a given ensemble system on the basis of a well-grounded set of metrics. We also validated and demonstrated the framework through experiments on DNN ensemble systems. In the future, we will analyze a broader range of systems, including rec/ent DNN ensemble systems optimized in an end-to-end manner. We will also incorporate combination loss into ensemble systems as an optimization target (i.e., as a loss-term) for better performance.

Acknowledgement

We thank the three anonymous reviewers and the meta-reviewer, who gave us insightful comments and suggestions. Computational resources of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) were used. We thank Dr. Masaaki Shimizu at Hitachi for the convenience of additional computational resources. We thank Dr. Naoaki Okazaki, professor at Tokyo Institute of Technology, for the keen comments.

References

- Atwood, J., Halpern, Y., Baljekar, P., Breck, E., Sculley, D., Ostyakov, P., Nikolenko, S. I., Ivanov, I., Solovyev, R., Wang, W., et al. The inclusive images competition. In *The NeurIPS’18 Competition*, pp. 155–186. Springer, 2020.
- Breiman, L. Bagging predictors. *Machine Learning*, 24(2): 123–140, 1996.
- Breiman, L. Random forests. *Machine Learning*, 45 (1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL <http://dx.doi.org/10.1023/A%3A1010933404324>.
- Brown, G. An information theoretic perspective on multiple classifier systems. In *International Workshop on Multiple Classifier Systems*, pp. 344–353, 2009.
- Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2924–2936, 2019.
- Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.
- Cunningham, P. and Carney, J. Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning*, pp. 109–116. Springer, 2000.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.
- Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *International Workshop on Paraphrasing*, 2005.

- Fano, R. M. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.
- Geman, S., Bienenstock, E., and Doursat, R. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.
- Khot, T., Sabharwal, A., and Clark, P. SciTail: A textual entailment dataset from science question answering. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Kohavi, R., Wolpert, D. H., et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pp. 275–83, 1996.
- Kumar, A., Kim, J., Lyndon, D., Fulham, M., and Feng, D. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 21(1):31–40, 2016.
- Kuncheva, L. I. and Whitaker, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.
- Liu, W., Zhang, M., Luo, Z., and Cai, Y. An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors. *IEEE Access*, 5:24417–24425, 2017.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- Ma, S. and Chu, F. Ensemble deep learning-based fault diagnosis of rotor bearing systems. *Computers in Industry*, 105:143–152, 2019.
- Morio, G., Morishita, T., Ozaki, H., and Miyoshi, T. Hitachi at SemEval-2020 task 10: Emphasis distribution fusion on fine-tuned language models. In *Workshop on Semantic Evaluation*, pp. 1658–1664, 2020a.
- Morio, G., Morishita, T., Ozaki, H., and Miyoshi, T. Hitachi at SemEval-2020 task 11: An empirical study of pre-trained transformer family for propaganda detection. In *Workshop on Semantic Evaluation*, pp. 1739–1748, 2020b.
- Morishita, T., Morio, G., Horiguchi, S., Ozaki, H., and Miyoshi, T. Hitachi at SemEval-2020 task 8: Simple but effective modality ensemble for meme emotion recognition. In *Workshop on Semantic Evaluation*, pp. 1126–1134, 2020a.
- Morishita, T., Morio, G., Ozaki, H., and Miyoshi, T. Hitachi at semeval-2020 task 7: Stacking at scale with heterogeneous language models for humor recognition. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 791–803, 2020b.
- Omari, A. and Figueiras-Vidal, A. R. Post-aggregation of classifier ensembles. *Information Fusion*, 26:96–102, 2015.
- Phang, J., Yeres, P., Swanson, J., Liu, H., Tenney, I. F., Htut, P. M., Vania, C., Wang, A., and Bowman, S. R. jiant 2.0: A software toolkit for research on general-purpose text understanding models. <http://jiant.info/>, 2020.
- Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large-Margin Classifiers*, pp. 61–74. 1999.
- Qummar, S., Khan, F. G., Shah, S., Khan, A., Shamshirband, S., Rehman, Z. U., Khan, I. A., and Jadoon, W. A deep learning ensemble approach for diabetic retinopathy detection. *IEEE Access*, 7:150530–150539, 2019.
- Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

- Shipp, C. A. and Kuncheva, L. I. Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2):135–148, 2002.
- Skalak, D. B. et al. The sources of increased accuracy for two proposed boosting algorithms. In *Integrating Multiple Learned Models Workshop*, volume 1129, pp. 1133, 1996.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <https://aclanthology.org/D13-1170>.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- Tumer, K. and Ghosh, J. Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. *IEEE Trans. Neural Networks*, 1995.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Black-boxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Williams, A., Nangia, N., and Bowman, S. 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)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL <http://aclweb.org/anthology/N18-1101>.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.
- Wolpert, D. H. Stacked generalization. *Neural Networks*, 5: 241–259, 1992.
- Yan, J., Yu, Y., Zhu, X., Lei, Z., and Li, S. Z. Object detection by labeling superpixels. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5107–5116, 2015. doi: 10.1109/CVPR.2015.7299146.
- Zhou, Z.-H. and Li, N. Multi-information ensemble diversity. In *Multiple Classifier Systems*, pp. 134–144, 2010.

A. Limitations

The study has the following limitations:

- As stated in Section 1, the framework Lemma 3.1 deals with classification tasks.
- As stated in Section 3.2, the claims made on the toy ensemble systems of Table 1 are hypothetical rather than theoretically driven, although they have explained certain aspects of the experimental results as discussed in Section 6.

B. Ethics and social impacts

Ensemble learning is a generic technology to boost the performance of machine learning models. This study provides a theoretical framework on ensemble learning for evaluating a given ensemble system by a set of specific metrics. The framework enables us to reveal the strengths and weaknesses of ensemble systems on each metric, which will give us insights into the designing of ensemble systems. Thus, this study should ultimately lead to the better performance of machine learning models.

While it is possible that inappropriate use of improved machine learning models poses negative effects on society, we believe that this study does not directly pose negative effects on society.

C. Definitions

We show the definitions of information theoretical quantities used in this study. In the below, we assume that \mathbf{S} and \mathbf{T} denote sets of discrete stochastic variables:

$$\begin{aligned}\mathbf{S} &= \{S_1, S_2, \dots, S_L\}, L \in \mathbb{N}, \\ \mathbf{T} &= \{T_1, T_2, \dots, T_M\}, M \in \mathbb{N},\end{aligned}$$

where S_i and T_i are discrete stochastic variables. We denote s_i, t_i as the values of S_i, T_i , and p as the probability distribution function.

Definition C.1 (Entropy of \mathbf{S}).

$$H(\mathbf{S}) = - \sum_{s_1, \dots, s_L} p(s_1, \dots, s_L) \log_2 p(s_1, \dots, s_L). \quad (\text{C.1})$$

Definition C.2 (Conditional entropy of \mathbf{T} given \mathbf{S}).

$$\begin{aligned}H(\mathbf{T}|\mathbf{S}) &= - \sum_{s_1, \dots, s_L} \sum_{t_1, \dots, t_M} p(s_1, \dots, s_L, t_1, \dots, t_M) \\ &\quad \times \log_2 p(t_1, \dots, t_M | s_1, \dots, s_L). \quad (\text{C.2})\end{aligned}$$

Definition C.3 (Mutual-information between \mathbf{S} and \mathbf{T}).

$$I(\mathbf{S}; \mathbf{T}) = H(\mathbf{T}) - H(\mathbf{T}|\mathbf{S}). \quad (\text{C.3})$$

Definition C.4 (Multi-information of \mathbf{S}).

$$I_{\text{multi}}(\mathbf{S}) = \sum_{s_1, \dots, s_L} p(s_1, \dots, s_L) \log_2 \frac{p(s_1, \dots, s_L)}{p(s_1) \dots p(s_L)}. \quad (\text{C.4})$$

Definition C.5 (Conditional multi-information of \mathbf{T} given \mathbf{S}).

$$\begin{aligned}I_{\text{multi}}(\mathbf{T}|\mathbf{S}) &= \sum_{s_1, \dots, s_L} \sum_{t_1, \dots, t_M} p(s_1, \dots, s_L, t_1, \dots, t_M) \\ &\quad \times \log_2 \frac{p(t_1, \dots, t_M | s_1, \dots, s_L)}{p(t_1 | s_1, \dots, s_L) \dots p(t_M | s_1, \dots, s_L)}. \quad (\text{C.5})\end{aligned}$$

For the interpretation of (C.4) to (C.5), see (Brown, 2009; Zhou & Li, 2010).

D. About Lemma 3.1

D.1. Full proof

$$\begin{aligned}H(Y|\mathbf{O}) &+ \underbrace{H(Y|\hat{Y}) - H(Y|\mathbf{O})}_{\text{combination loss}} = H(Y|\hat{Y}), \\ &\leq \underbrace{\mathcal{H}_2(p_{\text{err}})}_{\text{combination loss}} + p_{\text{err}} \log_2(Y_{\text{max}} - 1) =: \mathcal{U}(p_{\text{err}}), \\ &\leq \underbrace{\mathcal{H}_2(p_0) + \mathcal{H}'_2(p_0)(p_{\text{err}} - p_0) - \frac{m}{2}(p_{\text{err}} - p_0)^2}_{=: \hat{\mathcal{H}}_2(p_{\text{err}})} \\ &\quad + p_{\text{err}} \log_2(Y_{\text{max}} - 1), \\ &=: \hat{\mathcal{U}}_{m, p_0}^{\text{tight}}(p_{\text{err}}). \quad (\text{D.1})\end{aligned}$$

The first inequality follows from Fano's inequality Lemma 2.1. In the second inequality, we used strong concavity of binary cross entropy function $\mathcal{H}_2(p_{\text{err}})$ to upper

bound it by another quadratic function $\hat{\mathcal{H}}_2(p_{\text{err}})$ tangent to $\mathcal{H}_2(p_{\text{err}})$ at $p_{\text{err}} = p_0$. (D.1) holds for any $p_0 \in [0, 1]$ and $m \leq 4$.

m represents the curvature of $\hat{\mathcal{H}}_2(p_{\text{err}})$. Setting $m = 4$ produces the most curved quadratic function $\hat{\mathcal{H}}_2(p_{\text{err}})$, and hence the tightest upper bound of $\mathcal{H}_2(p_{\text{err}})$. Then, decomposing $H(Y|\mathbf{O})$ of the left-hand side as Lemma 2.3 and solving (D.1) for p_{err} derives Lemma 3.1.

The choice of p_0 of Lemma 3.1 is discussed in Appendix D.2.

D.2. Which choice of p_0 is preferable for ensemble system comparison

Lemma 3.1 discloses lower bounds that depend on p_0 . For fair comparisons of ensemble systems, we must first choose

and fix a specific value of p_0 from $[0, 1]$. Any choice of p_0 is ok since it does not change the ranking of lower bounds. In our experiments, we chose the baseline error rate as our p_0 due to the following reason.

As stated in Appendix D.1, we approximated the binary cross entropy function $\mathcal{H}_2(p_{\text{err}})$ as a quadratic function $\hat{\mathcal{H}}_2(p_{\text{err}})$ tangent to $\mathcal{H}_2(p_{\text{err}})$ at $p_{\text{err}} = p_0$. Thus, the approximation error $\hat{\mathcal{U}}_{m=4, p_0}^{\text{tight}}(p_{\text{err}}) - \mathcal{U}(p_{\text{err}})$ is the smallest when $p_0 \sim p_{\text{err}}$, where $p_{\text{err}} = \mathcal{B}_{p_0}^{\text{tight}}(E)$ is the actual lower bound obtained by ensemble strength E of each of the ensemble systems. This means that we should choose a value of p_0 that is similar to the error rate lower bounds of the target ensemble systems due to the following reason.

Since we do not know the error lower bounds of the systems before we choose p_0 and solve $p_{\text{err}} = \mathcal{B}_{p_0}^{\text{tight}}(E)$, it is a bit complicated to tune the value of p_0 , although it is possible. Thus, in the experiments of this study, we chose the baseline error rate as our p_0 rather than tuning p_0 . The baseline error rate is expected to be similar to the error rates of the ensemble systems, and hence it should not be much different from the lower bounds of the systems.

D.3. Comparison between tightness of Lemma 3.1 and Lemma 2.3

Lemma 3.1 differs from Lemma 2.3 in the lower bound functions. That is, Lemma 3.1 uses $\mathcal{B}_{p_0}^{\text{tight}}(E)$ while Lemma 2.3 uses $\mathcal{B}(E)$. In this section, we show that the bound function $\mathcal{B}_{p_0}^{\text{tight}}(E)$ is tighter (i.e. larger) than $\mathcal{B}(E)$ if E is in a specific range in which $\mathcal{B}_{p_0}^{\text{tight}}(E)$ is not much different from p_0 . Hereafter we assume $p_0 \leq \frac{Y_{\max}-1}{Y_{\max}}$, $\mathcal{B}_{p_0}^{\text{tight}}(E) \leq \frac{Y_{\max}-1}{Y_{\max}}$, and $\mathcal{B}(E) \leq \frac{Y_{\max}-1}{Y_{\max}}$, where $\frac{Y_{\max}-1}{Y_{\max}}$ means an error rate of a random guessing system on a balanced label dataset.

Firstly, we show how the two lemmas are derived.

Lemma 3.1 is derived using (D.1) as:

1. Set $m = 4$. That is, we use $\hat{\mathcal{U}}_{m=4, p_0}^{\text{tight}}(p_{\text{err}})$ for the upper bound function.
2. Solving for p_{err} derives Lemma 3.1 $p_{\text{err}} \geq \mathcal{B}_{p_0}^{\text{tight}}(E)$.

Lemma 2.3 is derived in a similar way as:

$$H(Y|\mathbf{O}) + \underbrace{H(Y|\hat{Y}) - H(Y|\mathbf{O})}_{\text{combination loss}} \leq \hat{\mathcal{U}}_{m, p_0}^{\text{tight}}(p_{\text{err}}),$$

$$\leq \hat{\mathcal{U}}_{m, p_0}^{\text{tight}}(p_{\text{err}}) + p_{\text{err}} \log_2 \frac{Y_{\max}}{Y_{\max} - 1} =: \hat{\mathcal{U}}_{m, p_0}(p_{\text{err}}). \quad (\text{D.2})$$

1. Set $m = 0, p_0 = \frac{1}{2}$. That is, we use $\hat{\mathcal{U}}_{m=0, p_0=\frac{1}{2}}(p_{\text{err}})$ for the upper bound function.

2. Loosen the left-hand side as $H(Y|\mathbf{O}) + \underbrace{H(Y|\hat{Y}) - H(Y|\mathbf{O})}_{\text{combination loss}} \geq H(Y|\mathbf{O})$, that is, ignore the combination loss.

3. Then, solving for p_{err} derives Lemma 2.3 $p_{\text{err}} \geq \mathcal{B}(E)$.

Viewing these, we can immediately show that if we use $p_0 = \frac{1}{2}$ for Lemma 3.1, it's bound function is tighter as $\forall E, \mathcal{B}_{p_0=\frac{1}{2}}^{\text{tight}}(E) \geq \mathcal{B}(E)$. This follows from $\hat{\mathcal{U}}_{m=4, p_0=\frac{1}{2}}^{\text{tight}}(p_{\text{err}}) \leq \hat{\mathcal{U}}_{m=0, p_0=\frac{1}{2}}^{\text{tight}}(p_{\text{err}}) + p_{\text{err}} \log_2 \frac{Y_{\max}}{Y_{\max}-1} = \hat{\mathcal{U}}_{m=0, p_0=\frac{1}{2}}(p_{\text{err}})$. We also point out that Lemma 2.3 poses the following assumptions which may lead to loose bound; (i) $m = 0$. This means that the upper bound function $\hat{\mathcal{U}}_{m=0, p_0=\frac{1}{2}}(p_{\text{err}})$ is a line. (ii) The existence of positive term $p_{\text{err}} \log_2 \frac{Y_{\max}}{Y_{\max}-1}$.

When we use $p_0 \neq \frac{1}{2}$, that is more general, the tightness $\mathcal{B}_{p_0}^{\text{tight}}(E) \geq \mathcal{B}(E)$ holds in limited ranges of E . As stated in Appendix D.2, the approximation error produced by $\hat{\mathcal{U}}_{m, p_0}^{\text{tight}}(E)$ is the smallest if $\mathcal{B}_{p_0}^{\text{tight}}(E) \sim p_0$. Thus, roughly speaking, $\mathcal{B}_{p_0}^{\text{tight}}(E) \geq \mathcal{B}(E)$ holds if $\mathcal{B}_{p_0}^{\text{tight}}(E) \sim p_0$ and $\mathcal{B}_{p_0}^{\text{tight}}(E) \leq \mathcal{B}(E)$ holds if $\mathcal{B}_{p_0}^{\text{tight}}(E)$ and p_0 differ much in their values.

We can discuss the details as follows. The tightness condition on ensemble strength E is given by:

$$\mathcal{B}_{p_0}^{\text{tight}}(E) \geq \mathcal{B}(E). \quad (\text{D.3})$$

Let lower bound function $\bar{p}_{p_0}(E) = \mathcal{B}_{p_0}^{\text{tight}}(E)$. Solving (D.3) for E can derive the range of $\bar{p}_{p_0}(E)$ where the tightness holds:

$$\bar{p}_{p_0}(E) \leq \min(p_0 + \Delta_{p_0}^+, p_0 + \Delta_{p_0}^-), \quad (\text{D.4})$$

$$\max(p_0 + \Delta_{p_0}^+, p_0 + \Delta_{p_0}^-) \leq \bar{p}_{p_0}(E), \quad (\text{D.5})$$

where

$$\Delta_{p_0}^{\pm} := \tau(p_0) \left[1 \pm \sqrt{1 - \frac{1}{2} \frac{1 - \mathcal{H}_2(p_0) - \log \frac{Y_{\max}-1}{Y_{\max}}}{\tau(p_0)^2}} \right],$$

$$\tau(p_0) := \frac{1}{4} \left[\frac{d\mathcal{H}_2}{dp}(p_0) - \log \frac{Y_{\max}}{Y_{\max}-1} \right].$$

We assumed $1 - \frac{1}{2} \frac{1 - \mathcal{H}_2(p_0) - \log \frac{Y_{\max}-1}{Y_{\max}}}{\tau(p_0)^2} \geq 0$. Otherwise, the tightness (D.3) always holds.

We proceed by specifying p_0 . Firstly, suppose p_0 is mildly small: $p_0 \leq \frac{Y_{\max}-1}{2Y_{\max}-1}$. Then, we can show $\tau(p_0) \geq 0$. Thus, $\Delta_{p_0}^- \leq \Delta_{p_0}^+$ holds, and (D.4) becomes:

$$\bar{p}_{p_0}(E) \leq p_0 + \Delta_{p_0}^-. \quad (\text{D.6})$$

Additionally, we can show that $\Delta_{p_0}^- \geq 0$. Thus, (D.6) discloses that if the lower bound $\bar{p}_{p_0}(E)$ is not much larger than p_0 , the tightness (D.3) holds. Especially, if $\bar{p}_{p_0}(E) \leq p_0$ the tightness holds. This condition applies to the experiments of this study. We have also directly shown that $\mathcal{B}_{p_0}^{\text{tight}}(\mathcal{I}) > \mathcal{B}(\mathcal{I})$ in Table 4a.

If p_0 is large $p_0 \geq \frac{Y_{\max}-1}{2Y_{\max}-1}$, we can show $\tau(p_0) \leq 0$. Thus, $\Delta_{p_0}^- \geq \Delta_{p_0}^+$ holds, and (D.5) becomes:

$$\bar{p}_{p_0}(E) \geq p_0 + \Delta_{p_0}^-. \quad (\text{D.7})$$

Additionally, we can show that $\Delta_{p_0}^- \leq 0$. Thus, (D.7) discloses that if the lower bound $\bar{p}_{p_0}(E)$ is not much smaller than p_0 , the tightness (D.3) holds.

E. Details of experimental setup

E.1. Models

DNN types: Table E.7 shows the five types of pre-trained language models used in this study. Pre-trained language models are essentially large neural networks with self-attention layers that are trained on huge text corpora in an unsupervised manner. These models are shown to obtain state-of-the-art performance when fine-tuned on downstream tasks (Liu et al., 2019; Lan et al., 2020; Clark et al., 2020; Devlin et al., 2019; Lewis et al., 2020). In addition, since they differ in terms of model architecture and pre-training method, they should produce strong diversity, and hence, are suitable for ensembles.

Fine-tuning procedures: We trained each DNN on each downstream task following the standard practice of language model fine-tuning (see Devlin et al. (2019) for example) as follows.

We added a new softmax layer on top of the embedding layers of the DNNs. We preprocessed the input text by the following steps: (i) we tokenized the input text with a DNN-type-specific tokenizer, (ii) if the text included more than two sentences, we added DNN-type-specific “separator” tokens between sentences, (iii) we tensorized each token into a one-hot vector using DNN-type-specific vocabulary.

We trained these models on the training sets of the tasks. TValidation sets were used only during the preliminary experiments to adjust some hyperparameters (shown below). Please refer to Appendix F.4 for the details of the dataset splitting strategy.

We used the hyperparameters shown in Table E.8 to fine-tune all of the DNN types. The values were chosen on the basis of the original papers (Liu et al., 2019; Lan et al., 2020; Clark et al., 2020; Devlin et al., 2019; Lewis et al., 2020) and our preliminary experiments. Note that language models require only a few epochs for convergence.

Some of the ensemble methods in Table 2 use different seeds for fine-tuning to produce diverse DNN models. In our study, seeds affect (i) the initial weights of the softmax layer, (ii) the hidden units dropped by dropout, and (iii) the shuffling order of the training instances.

Implementations: We implemented the fine-tuning of DNNs described here using the *jiant* library (Phang et al., 2020) (v2.2.0⁵), which in turn utilizes Hugging Face’s Transformers library (Wolf et al., 2020). *Jiant* enables us to fine-tune various types of pre-trained language models on various NLP tasks. See our code for details.

⁵github hash: 961bd577f736449956ddb2c15dcfce68bbb75e59

E.2. Ensemble systems

For the random hyperparameter sampling of Random-HyP, we sampled the fine-tuning learning rate since it affect the resulting model the most. We sampled the learning rate around the best value of $3e-5$, i.e., from $[1e-5, 1e-4]$, as shown in Table E.8.

The baseline system s_0 was single DNN (i.e. no-ensemble) that performed the best among DNNs. These baselines are shown as bold in Table E.7

We implemented the model generation methods in Table 2 by ourselves.

We implemented the model combination methods in Table 2 using scikit-learn⁶. For the training of Stacking meta-estimators, we used the hyperparameters shown in Table E.9. We tuned some of the hyperparameters using scikit-learn’s GridSearchCV with 5-fold cross validation. Appendix F gives other details on Stacking ensemble used in this study.

E.3. Estimation of metric values and lower bound

Trick of MTI

In our experiments, we estimated the three metric values on the basis of the frequency distribution observed for the datasets. We used the trick of MTI introduced by (Zhou & Li, 2010), which approximates quantities appearing in the three metrics which depend on high-dimensional stochastic variables \mathbf{O} . Please refer to (Zhou & Li, 2010) for more details.

We repeat the three terms of Lemma 3.1 below:

$$\begin{aligned} I_{\text{relev}}(\mathbf{O}, Y) &= \sum_{i=1}^N I(O_i, Y), \\ I_{\text{redun}}(\mathbf{O}, Y) &= I_{\text{multi}}(\mathbf{O}) - I_{\text{multi}}(\mathbf{O}|Y), \\ I_{\text{combloss}}(\mathbf{O}, Y, \hat{Y}) &= H(Y|\hat{Y}) - H(Y|\mathbf{O}). \end{aligned}$$

Looking at above, it can be seen that some terms (i.e. $I_{\text{multi}}(\mathbf{O})$, $I_{\text{multi}}(\mathbf{O}|Y)$ and $H(Y|\mathbf{O})$) depend on high-dimensional variable $\mathbf{O} = \{O_1, \dots, O_N\}$, where N is the number of models. Since N can be as large as 30 in our experiments, these terms might not be estimated reliably due to the count sparsity for the limited amount of dataset instances.

Thus, we use the trick of MTI introduced by Zhou & Li (2010), which approximates the quantities by replacing \mathbf{O}

⁶<https://scikit-learn.org/stable/>

Table E.7: DNNs used in study and their error rates for each task. Convention of “variant” follows Huggingface’s transformer library (Wolf et al., 2020). **Bold** shows best model in each task, which is used as baseline s_0 stated in Section 4.2.

DNN type	variant	avg.	Boolq	CoLA	CosmosQA	MNLI	MRPC	QQP	SciTail	SST
RoBERTa (Liu et al., 2019)	base	15.5 ± 0.3	24.1 ± 0.6	15.6 ± 0.2	28.2 ± 0.5	18.7 ± 1.2	13.6 ± 0.5	14.1 ± 0.8	4.2 ± 0.2	5.8 ± 0.7
ELECTRA (Clark et al., 2020)	base-discriminator	17.3 ± 0.3	23.1 ± 1.3	17.0 ± 0.5	29.8 ± 0.7	22.6 ± 1.0	13.3 ± 0.7	18.5 ± 0.9	7.1 ± 0.2	5.7 ± 0.5
BART (Lewis et al., 2020)	base	17.9 ± 0.2	25.5 ± 1.3	20.9 ± 0.5	30.1 ± 0.5	22.3 ± 0.8	15.8 ± 0.7	15.9 ± 1.2	4.7 ± 0.3	8.3 ± 0.5
BERT (Devlin et al., 2019)	base-uncased	18.7 ± 0.1	26.0 ± 0.8	17.2 ± 0.4	34.1 ± 0.6	26.3 ± 0.2	17.1 ± 0.7	16.5 ± 0.7	4.5 ± 0.2	8.0 ± 0.7
ALBERT (Lan et al., 2020)	base-v1	20.4 ± 0.1	25.3 ± 2.2	19.5 ± 0.2	43.2 ± 0.1	27.1 ± 0.3	14.5 ± 0.4	18.8 ± 0.8	4.9 ± 0.1	9.6 ± 0.3

Table E.8: Hyperparameters used for fine-tuning of DNNs.

hyperparameter	value
learning rate	3e-5 ([1e-5, 1e-4] for the random sampling of Random-HyP)
optimizer	Adam (Kingma & Ba, 2015) ($\epsilon = 1e - 8$) with linear warmup (data size proportion=0.1), described in (Devlin et al., 2019).
gradient clipping	1.0
gradient accumulation steps	1
epochs	5
dropout	DNN specific values (follows jiant (Phang et al., 2020))
training batch size	16
inference batch size	32
number of softmax layer	1

Table E.9: Meta-estimator hyperparameters. Hyperparameter names follow scikit-learn. Most of the hyperparameters are set as default values of scikit-learn (version 0.22.2).

meta-estimator	hyperparameter	value / search range
logistic regression	C	[1e-2, 3e-2, 1e-1, 3e-1, 1e0]
	penalty	L2
	solver	liblinear
	max_iter	1000
	multi_class	auto
	random_state	0
SVM	C	[1e-2, 3e-2, 1e-1, 3e-1, 1e0]
	max_iter	-1
	decision_function_shape	ovr
	random_state	0
Random Forest	ccp_alpha	[0.0, 0.03, 0.1, 0.3]
	random_state	0
	criterion	gini
	max_depth	None

with its smaller subset Ω as follows:

$$I_{\text{multi}}(\mathbf{O}) = \sum_{i=1}^N I(O_i; O_{1:i-1}) \geq \sum_{i=1}^N \max_{\Omega_k^{i-1}} I(O_i; \Omega_k^{i-1}), \quad (\text{E.1})$$

$$I_{\text{multi}}(\mathbf{O}|Y) = \sum_{i=1}^N I(O_i; O_{1:i-1}|Y) \geq \sum_{i=1}^N \max_{\Omega_k^{i-1}} I(O_i; \Omega_k^{i-1}|Y) \quad (\text{E.2})$$

$$H(Y|\mathbf{O}) \leq \min_{\Omega_k^N} H(Y|\Omega_k^N), \quad (\text{E.3})$$

where $\Omega^i = \{X_1, \dots, X_i\}$, and Ω_k^{i-1} is a subset of size k .

The first equalities of (E.1) and (E.2) were proved by (Zhou & Li, 2010). The last inequality in each equation is understood as follows. By replacing \mathbf{O} with its subset Ω_k^i , we lose some amount of information carried by \mathbf{O} . Thus, this transformation might make mutual information in (E.1) and (E.2) smaller than the original value and the entropy in (E.3) larger. However, if we find Ω_k^i , which contains the largest amount of information (corresponding to max and min operations in each equation), the difference from the original value (i.e., approximation error) is the smallest.

Zhou & Li (2010) empirically showed that the method works well to produce almost an exact value. In our experiments, we used $k = 3$ ($\text{MTI}_{k=3}$).

On the choice of p_0

We set the approximate error rate p_0 in (6) as the error rate of the baseline s_0 defined in Section 4.2. We state the reason for this in Appendix D.2.

E.4. Tasks

Table E.10 details the eight tasks used in this study.

Table E.10: Tasks used in this study. Majority of tasks are from GLUE benchmark (Wang et al., 2018) (shown as *) and SuperGLUE benchmark (Wang et al., 2019) (shown as †). Both benchmarks are commonly used to compare state-of-the-art models such as pre-trained language models. All datasets are publicly available.

task	dataset size	# classes (Y_{\max})	description
Boolq† (Boolean Question) (Clark et al., 2019)	9.5k	2	We are required to choose yes or no about a given question on a given passage. The questions are the ones naturally occurring in Google search engine, rather than the ones artificially built. Answering the questions often requires query for complex, non-factoid information, and difficult entailment-like inference.
CoLA* (Corpus of Linguistic Acceptability) (Dolan & Brockett, 2005)	8.5k	2	We are required to judge linguistic acceptability (i.e., grammatical or non grammatical) of given text such as “What did Bill buy potatoes?”. The text are drawn from books and journal articles on linguistic theory. Answering the questions requires the rich grammatical knowledge from the local word dependencies such as subject-verb-object order to the non-local dependencies.
Cosmos QA (Khot et al., 2018)	25k	4	After reading a short narrative passage, we are required to answer a question about the passage (such as “What’s a possible reason the writer needed someone to dress him every morning?”) by choosing one answer from four possible candidates. The passages are taken from blogs on the web and personal narratives. Understanding the narrative requires common sense such as inference on causes and effects of events, even when they are not mentioned explicitly in the texts.
MNLI* (Multi-Genre Natural Language Inference) (Williams et al., 2018)	400k (10k used)	3	Given two pieces of text, we are required to answer the relationship of the one piece to the other piece from three choices: “entails”, “neutral”, “contradicts”. The dataset is composed of texts from various distinct genres of written English. The pairs are such as “At 8:34, the Boston Center controller received a third transmission from American 11” and “The Boston Center controller got a third transmission from American 11.” Answering the question requires total ability of natural language understanding, e.g., handling lexical entailment, quantification, coreference, tense, belief, modality, and lexical and syntactic ambiguity.
MRPC* (Microsoft Research Paraphrase Corpus) (Dolan & Brockett, 2005)	3k	2	We are required to judge whether given two sentences are semantically equivalent. The sentences are automatically extracted from online news sources and twitter. Pairs are such as: “Charles O. Prince, 53, was named as Mr. Weill’s successor.” “Mr. Weill’s longtime confidant, Charles O. Prince, 53, was named as his successor.”. Recognizing such paraphrase is a fundamental skill needed for various tasks in NLP.
QQP* (Quora Question Pairs) ⁷	300k (10k used)	2	We are required to determine whether a pair of questions are semantically equivalent. The questions are taken from the social Q&A website Quora. The skill is used by question-answering system to recognize the semantically same questions of different linguistic expressions.
SciTail (Khot et al., 2018)	23k	4	We are required to answer a given scientific question such as “Which of the following best explains how stems transport water to other parts of the plant?” by choosing one answer from four candidates. We have access to the additional relevant text. The questions are the ones naturally arising in the web rather than ones artificially created.
SST* (Stanford Sentiment Treebank) (Socher et al., 2013)	50k (10k used)	2	We are required to predict a sentiment label (i.e., positive or negative) of a given sentence. The sentences are taken from movie reviews. The task requires the understanding of compositionality of language.

F. Stacking ensemble

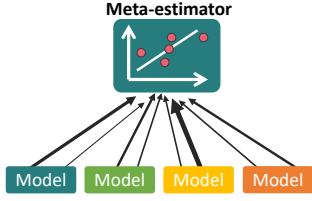


Figure F.4: Stacking ensemble used in this study.

Here, we give details of the stacking ensemble used in this study.

F.1. Architecture:

Figure F.4 illustrates the stacking ensemble used in this study. We used the two-layered stacking ensemble where the first-layer models are fine-tuned DNNs, and the second-layer model (i.e. the meta-estimator) is another classification model. For the meta-estimator, we used logistic regression, Support Vector Machine (Platt, 1999) with RBF kernel, and Random Forest (Breiman, 2001). For the inputs of the meta-estimator, we used class labels predicted by the models.

In the below, we show the details of the logistic regression meta-estimator case. The meta-estimator estimates the probability for a given instance i belonging to class c $p_{i,c} \in [0, 1]$ from class labels predicted by N models $\hat{\mathbf{y}}_i = \{\hat{y}_i^1, \hat{y}_i^2, \dots, \hat{y}_i^N\}$, $\hat{y}_i^n \in \{0, 1\}$ as:

$$p_{i,c} = \frac{1}{1 + \exp(-l_{i,c})},$$

$$l_{i,c} = w_c^0 + \sum_{m=1}^N w_c^m \hat{y}_i^m.$$

The class with the largest $p_{i,c}$ is chosen as the final answer. The meta-estimator is trained using “meta-feature dataset” $D_{\text{meta}} = \{(\hat{\mathbf{y}}_1, y_1), (\hat{\mathbf{y}}_2, y_2), \dots, (\hat{\mathbf{y}}_{|D|}, y_{|D|})\}$, where y_i denotes the groundtruth label. Details of the meta-estimator training are shown in E.2 and F.4.

F.2. Weight distribution of Table 6

The DNN-type-wise weight sum mentioned in Table 6 is calculated as follows:

$$W_t = \sum_{m \in M_t} |w_{c=1}^m|,$$

where m denote the index of model, t a specific DNN type and M_t the set of indexes of models from DNN type t . Note that since our study used binary classification tasks, it suffices to look $c = 1$.

F.3. Meta-estimator training

Hyperparameters The hyperparameters of the meta-estimators (i.e., logistic regression and the SVM used by Stacking ensemble) are shown in Table E.9.

Implementation: We implemented the model combination methods in Table 2 using scikit-learn⁸.

F.4. Dataset splitting

In order to train meta-estimator of Stacking, we must take cross-validation based dataset splitting strategy (Wolpert, 1992). In the below, we describe the data splitting strategy, which is illustrated in Figure F.5. Note that the same data splitting strategy was used for voting-based systems for fair comparisons.

Training of stacking meta-estimators requires “meta-feature dataset” $D_{\text{meta}} = \{(\hat{\mathbf{y}}_1, y_1), (\hat{\mathbf{y}}_2, y_2), \dots, (\hat{\mathbf{y}}_{|D|}, y_{|D|})\}$, as stated in Appendix F. Here, $\hat{\mathbf{y}}_i = \{\hat{y}_i^1, \hat{y}_i^2, \dots, \hat{y}_i^N\}$ where $\hat{y}_i^m \in \{0, 1\}$ denotes the label predicted by model m on instance i . y_i denotes the groundtruth label of the same instance i . To prevent overfitting of meta-estimators, the model predictions $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots\}$ must be *label-leak-free*. Thus, the model predictions are usually obtained using n -fold cross-validation as follows.

Meta-feature dataset construction For each model m , we use n -fold cross-validation to obtain its label-leak-free predictions. Specifically:

1. Choose model m .
2. Divide the dataset $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{|D|}, y_{|D|})\}$ into n sets.
3. One of them (i.e. base-test- i) is set aside for testing later.
4. Train the model m on the rest sets (i.e. base-train- i).
5. Apply the trained model m to the test set (i.e. base-test- i) to get label-leak-free predictions.
6. Repeat 3-5 for i to collect label-leak-free predictions on whole the dataset $\{\hat{y}_1^m, \hat{y}_2^m, \dots, \hat{y}_{|D|}^m\}$ where \hat{y}_i^m denotes a label prediction by model m on the instance i , as stated in F.
7. Repeat 1-6 for m to collect label-leak-free predictions by all the models: $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_{|D|}\}$. Then, we concatenate the predictions D_{meta} . Then, we merge the predicted labels $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_{|D|}\}$ and the groundtruth labels $\{y_1, \dots, y_{|D|}\}$ into the meta-feature dataset $D_{\text{meta}} = \{(\hat{\mathbf{y}}_1, y_1), (\hat{\mathbf{y}}_2, y_2), \dots, (\hat{\mathbf{y}}_{|D|}, y_{|D|})\}$.

⁸<https://scikit-learn.org/stable/>

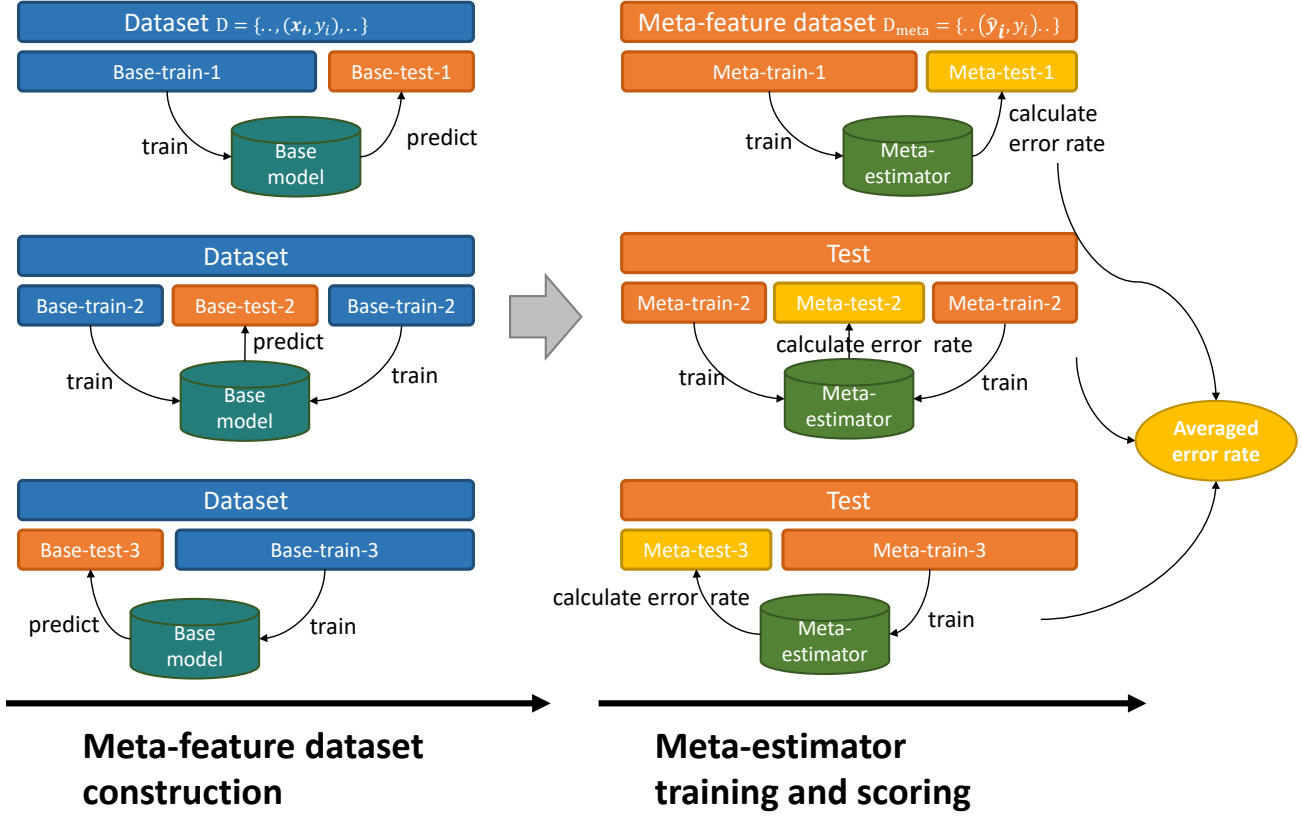


Figure F.5: Our dataset splitting strategy (Appendix F.4) with 3-split case.

Note that, since the test set (i.e. base-test- i) is never used by model training, the predictions on the test set are label-leak-free.

In this study, we used $n = 5$.

5. Repeat 2-4 for i to get error rates on the test-sets, then calculate the average of them.

In this study, we used $l = 4$.

Meta-estimator training and scoring :

Some of the datasets used in this study are small, as shown in Table E.10. The official test-sets of the datasets are also small. For example, the test sets of RTE dataset includes only 277 instances. We supposed that the performance measurements conducted on such small test-sets might not be so reliable. Thus, we conducted the following l -fold cross-validation to train and score the meta-estimators:

1. Divide D_{meta} into l sets.
2. One of them (i.e. meta-test- i) is set aside for testing.
3. Train a meta-estimator on the rest sets (i.e. meta-train- i).
4. Apply the meta-estimator to the test sets (i.e. meta-test- i) and calculate its error rate.

G. Pearson correlation coefficients between error rate reductions and lower bound reductions for various number of models N

Tables G.11 to G.14 show the Pearson correlation coefficients between the error reductions and lower bound reductions of the ensemble systems in each task. Each table shows the results of different N , which is the number of models used by the ensemble systems.

See Section 5.2 for the discussion of such correlations.

Table G.11: $N = 10$. Pearson correlation coefficients between error rate reduction and lower bound reduction. In each task we used the 16 ensemble systems described in Section 4.2.

Task	Lower bound type		
	Lemma 2.3 $\mathcal{B}(\mathcal{I})$	$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$
Boolq	0.413	0.377	0.869
CoLA	-0.259	-0.245	0.993
CosmosQA	-0.188	-0.174	1.000
MNLI	-0.275	-0.385	0.955
MRPC	0.218	0.218	0.983
QQP	-0.359	-0.330	0.999
SciTail	-0.076	-0.092	0.944
SST	0.286	0.357	0.998
average ⁹	-0.482	-0.431	0.975

Table G.12: $N = 15$. Pearson correlation coefficients between error rate reduction and lower bound reduction. In each task we used the 16 ensemble systems described in Section 4.2.

Task	Lower bound type		
	Lemma 2.3 $\mathcal{B}(\mathcal{I})$	$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$
Boolq	0.341	0.330	0.910
CoLA	-0.211	-0.210	0.991
CosmosQA	-0.324	-0.320	1.000
MNLI	0.226	0.216	0.961
MRPC	0.332	0.252	0.989
QQP	-0.131	-0.076	0.998
SciTail	-0.237	-0.191	0.966
SST	-0.242	-0.252	0.998
average ¹⁰	-0.238	-0.165	0.984

⁹The correlation coefficient between the averaged error rate reductions and lower bound reductions. The average is taken over the eight tasks.

Table G.13: $N = 20$. Pearson correlation coefficients between error rate reduction and lower bound reduction. In each task we used the 16 ensemble systems described in Section 4.2.

Task	Lower bound type		
	Lemma 2.3 $\mathcal{B}(\mathcal{I})$	$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$
Boolq	0.323	0.311	0.915
CoLA	-0.324	-0.320	0.995
CosmosQA	-0.510	-0.512	1.000
MNLI	-0.190	-0.192	0.976
MRPC	-0.235	-0.199	0.964
QQP	0.411	0.390	0.999
SciTail	-0.286	-0.307	0.958
SST	0.032	0.024	0.997
average ¹⁰	-0.452	-0.425	0.985

Table G.14: $N = 30$. Pearson correlation coefficients between error rate reduction and lower bound reduction. In each task we used the 16 ensemble systems described in Section 4.2.

Task	Lower bound type		
	Lemma 2.3 $\mathcal{B}(\mathcal{I})$	$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$
Boolq	0.158	0.146	0.940
CoLA	-0.215	-0.213	0.994
CosmosQA	-0.592	-0.588	1.000
MNLI	-0.048	-0.050	0.976
MRPC	-0.471	-0.498	0.974
QQP	0.187	0.231	0.999
SciTail	-0.379	-0.377	0.954
SST	0.213	0.208	0.996
average ¹⁰	-0.330	-0.288	0.990

H. Behavior of ensemble quantities when number of models N is changed

In this section, we examine the behavior of the ensemble quantities when the number of models is changed (Figure G.6). Most importantly: (i) both Lemma 2.3 $\mathcal{B}(\mathcal{I})$ (Figure G.6b) and $\mathcal{B}^{\text{tight}}(\mathcal{E})$ (Figure G.6c) could not predict the shape of error rate reduction curve (Figure G.6a), especially the saturation over $N \gtrsim 15$. (ii) by contrast Lemma 3.1 (Figure G.6d) could predict the phenomena. This success is attributed to the ensemble strength which consider combination loss (Figure G.6j).

Figure G.6e shows the per-model relevance $i_{\text{relev}} = I_{\text{relev}}/N$, that denotes the average amount of information on Y conveyed by a single model or average accuracy of the models. All the systems kept it nearly constant, since their model training procedures do not change with respect to N .

Figure G.6f shows the per-model redundancy $i_{\text{redun}} = I_{\text{redun}}/N$, which denotes the average amount of information on Y conveyed by a single model that is redundant to the other models. In all of the systems, it increased to about the same as i_{relev} . It increased because as more models come into an ensemble system, it becomes more difficult for a new model to output a “novel” prediction distribution compared with those of the existing models. As a result, new models eventually become totally redundant as $i_{\text{redun}} \sim i_{\text{relev}}$.

$i_{\text{relev}} - i_{\text{redun}}$ (Figure G.6g), the average amount of *unique* information conveyed by a single model, converged to nearly zero. Because of this diversity saturation, the increase in the $\mathcal{I} = N \times (i_{\text{relev}} - i_{\text{redun}})$ slowed at large scale (Figure G.6h). However, their saturation speed was smaller than the observed one (Figure G.6a). As a result, both lower bound reductions of Lemma 2.3 $\mathcal{B}(\mathcal{I})$ (Figure G.6b) and $\mathcal{B}^{\text{tight}}(\mathcal{E})$ (Figure G.6c) could not predict the saturation behavior.

Figure G.6i shows the combination loss I_{combloss} . I_{combloss} increased in proportion to the increase of \mathcal{I} , since I_{combloss} represents the amount of information lost from \mathcal{I} (Appendix I gives the intuition behind this increase). Overall, $\mathcal{E} = \mathcal{I} - I_{\text{combloss}}$ saturated at the large scale (Figure G.6j). Thus, the lower bound reduction by Lemma 3.1 (Figure G.6d) produced by \mathcal{E} succeeded in detecting the observed saturation behavior (Figure G.6a).

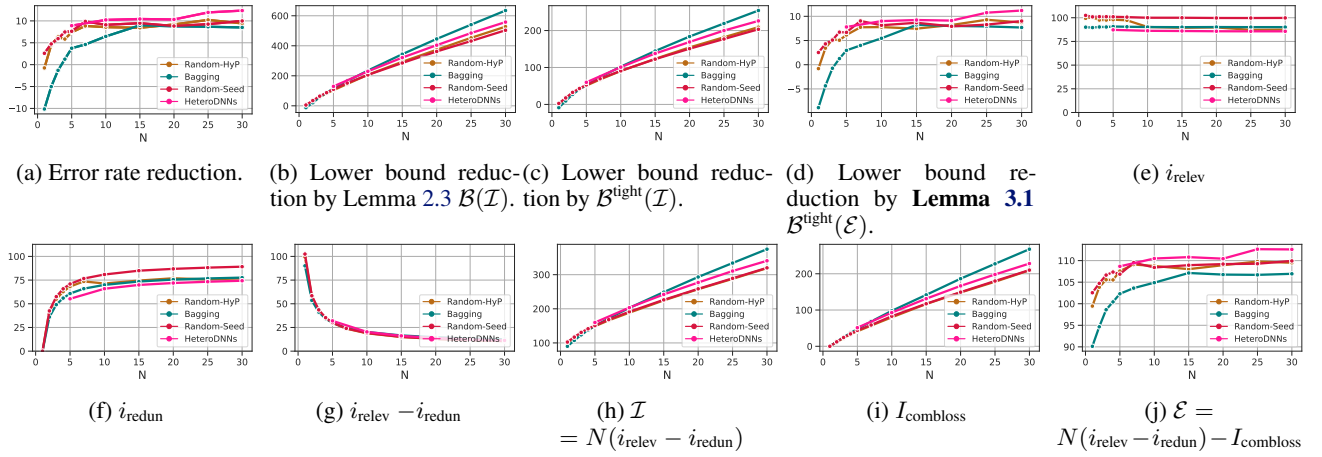


Figure G.6: The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$.

I. On increase in combination loss with respect to N

We describe the reason for the increase in combination loss I_{combloss} with respect to number of models N observed in Figure G.6i.

In particular, we discuss that the fact that I_{combloss} increases with N does not contradict the fact that a larger N leads to better performance.

I.1. Information theoretical view

From information theoretical viewpoints:

1. Since incoming models bring us more information, \mathcal{I} , which denotes the total amount of information carried by the models, increases with N , as shown in Figure G.6h.
2. Since I_{combloss} represents the amount of information lost from \mathcal{I} when a combination function \mathcal{F} is applied, I_{combloss} generally increases as \mathcal{I} increases. This is shown in Figure G.6i. This fact is not counter-intuitive, since, for example, if the information loss ‘‘rate’’ is constant as c , $I_{\text{combloss}} = c \times \mathcal{I}$ increases at the same speed as \mathcal{I} .
3. Since the growth of \mathcal{I} is faster than that of I_{combloss} , $\mathcal{E} = \mathcal{I} - I_{\text{combloss}}$, which denotes the total amount of information remaining after the combination, also increases, as shown in Figure G.6j.
4. Since \mathcal{E} represents the performance of an ensemble system, increasing \mathcal{E} leads to better performance.

As seen, the fact 2 that I_{combloss} increases with N does not contradict the fact 3-4 that a larger N leads to better performance.

I.2. Viewing through neglected minority model predictions

In Section 3.2, we discussed that the source of I_{combloss} is neglected but correct model predictions. We can also discuss I_{combloss} from this view as follows:

1. The number of neglected minority predictions on a misclassified dataset instance increases as the number of total predictions on the instance increases. Since the latter is roughly proportional to N , the former is also roughly proportional to N .
2. The total number of misclassified dataset instances, which denotes error rate, decreases *more slowly* than linearly with N . This is empirically known, for example as shown in Figure G.6a.

3. The total number of neglected minority predictions in a dataset, which is the source of I_{combloss} , is roughly estimated as [the number of neglected minority predictions on a misclassified dataset instance] \times [the total number of misclassified dataset instances]. From 1 and 2, this quantity increases roughly linearly with N .

As seen, the fact 3 that I_{combloss} increase with N does not contradict with the fact 2 that error rate decrease with N .

J. Measurements of information concentration

To observe this directly, we defined *n-model concentration* (Conc_n^N) which measures the degree of concentration on top- n models as a value in $[0, 1]$:

$$\text{Conc}_n^N(\mathbf{O}, Y) = \frac{I(\Omega_n^{N, \max}; Y) - I(\Omega_n^{N, \min}; Y)}{I(\mathbf{O}; Y)} \in [0, 1],$$

$$I(\Omega_n^{N, \max/\min}; Y) = \max/\min_{\{i_1, i_2, \dots, i_n\} \in \Omega_n^N} I(\{O_{i_1}, O_{i_2}, \dots, O_{i_n}\}; Y),$$

where I is mutual information defined by (C.3) and Ω_n^N is all possible combinations of n integers from $[1, N]$. Since the amount of information on Y carried by a subset $\{O_{i_1}, \dots, O_{i_n}\}$ can never be more than that of a full set \mathbf{O} , $I(\Omega_n^{N, \max/\min}; Y) \leq I(\mathbf{O}; Y)$. This leads to $\text{Conc}_n^N(\mathbf{O}, Y) \in [0, 1]$. The Conc_n^N takes 1 when all the information carried by \mathbf{O} can be reconstructed by top- n O_i and bottom- n O_i s having no information (i.e. $I(\Omega_n^{N, \max}; Y) = I(\mathbf{O}; Y)$ and $I(\Omega_n^{N, \min}; Y) = 0$). The Conc_n^N are small when the amount of information on top- n O_i is similar to that of bottom- n O_i (i.e. $I(\Omega_n^{N, \max}; Y) \sim I(\Omega_n^{N, \min}; Y)$).

K. Results of each task

Below, we show the experimental results of the eight tasks. The discussion in Sections 5 and 6 holds in each task, that is:

- $\mathcal{B}^{\text{tight}}$ generate lower bound tighter than \mathcal{B} . This is discussed in Section 5.1.
- The lower bound reduction by Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$ is strongly correlated to the error rate reductions, while those of Lemma 2.3 $\mathcal{B}(\mathcal{I})$ and $\mathcal{B}^{\text{tight}}(\mathcal{I})$ are not. This is discussed in Section 5.2.
- The lower bound reduction by Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$ successfully predicts the shape of error rate reduction curve when the number of models N is changed, while those of Lemma 2.3 $\mathcal{B}(\mathcal{I})$ and $\mathcal{B}^{\text{tight}}(\mathcal{I})$ do not. This is discussed in Section 5.3.
- The strengths and weaknesses of ensemble systems in terms of the three metrics. This is discussed in Section 6.

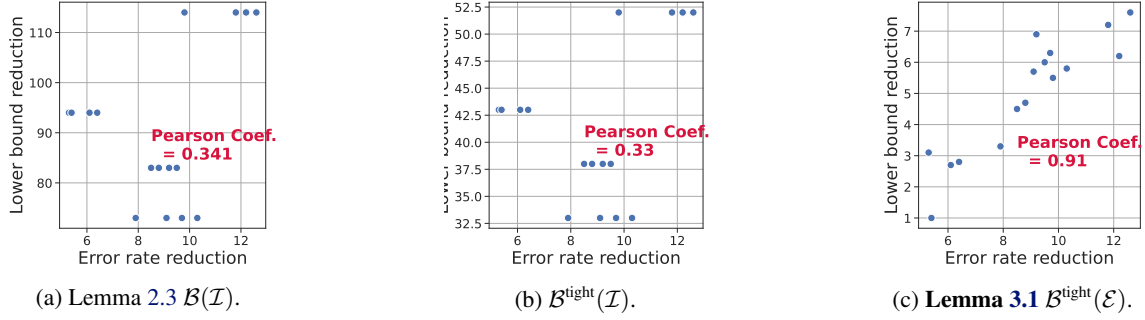


Figure K.7: **Boolq task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.15 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 24.1 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 3.1 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 3.1 %. $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.0 %.

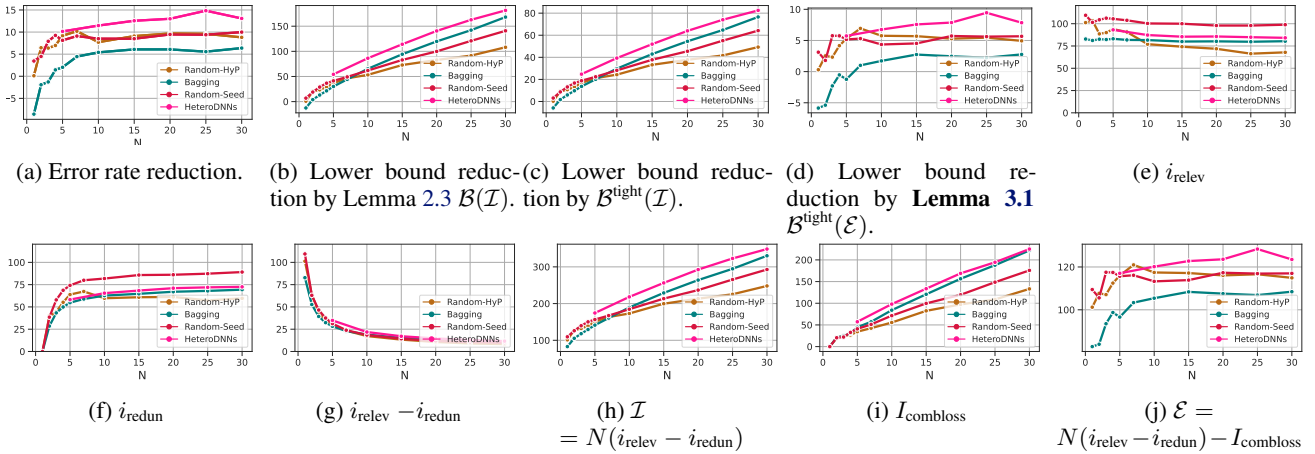


Figure K.8: **Boolq task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as $i_{\{\text{relev, redun, combloss}\}} = I_{\{\text{relev, redun, combloss}\}}/N$.

Table K.15: **Boolq task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 24.1 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 3.1 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 3.1 %. $LB(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.0 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	7.9 \pm 1.2	9.6 \pm 2.0	9.1 \pm 2.0	10.3 \pm 1.4	3.3 \pm 1.2	6.3 \pm 1.8	5.7 \pm 1.0	5.8 \pm 1.7	33 \pm 2	73 \pm 5
Bagging	5.3 \pm 2.0	6.4 \pm 3.0	6.2 \pm 2.4	5.4 \pm 1.4	3.1 \pm 1.7	2.8 \pm 2.7	2.7 \pm 1.8	1.0 \pm 1.3	43 \pm 2	94 \pm 3
Random-Seed	9.2 \pm 1.9	9.5 \pm 1.8	8.5 \pm 3.1	8.8 \pm 2.9	6.9 \pm 1.7	6.0 \pm 1.7	4.5 \pm 3.1	4.7 \pm 2.5	38 \pm 1	83 \pm 3
Hetero-DNNs	9.8 \pm 0.3	11.8 \pm 1.6	12.6 \pm 0.4	12.2 \pm 1.5	5.5 \pm 0.4	7.2 \pm 1.3	7.6 \pm 0.4	6.2 \pm 1.4	52 \pm 2	114 \pm 7

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	LogR	$\hat{i}_{\text{combloss}}$ SVM	RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.182)				100	0	0	0	0	0	100
Random-HyP	110.1 \pm 3.8	119.2 \pm 6.0	117.2 \pm 3.5	117.7 \pm 5.6	74.2 \pm 3.2	60.9 \pm 2.8	5.97 \pm 0.32	5.36 \pm 0.47	5.50 \pm 0.48	5.46 \pm 0.55	13.3 \pm 4.3
Bagging	109.5 \pm 5.3	108.8 \pm 8.5	108.4 \pm 5.7	103.2 \pm 3.9	80.0 \pm 2.4	64.7 \pm 2.5	7.97 \pm 0.12	8.02 \pm 0.41	8.04 \pm 0.28	8.39 \pm 0.07	15.3 \pm 3.4
Random-Seed	120.8 \pm 5.6	118.1 \pm 5.8	113.9 \pm 9.8	114.3 \pm 8.0	100.0 \pm 0.0	85.8 \pm 0.3	6.18 \pm 0.27	6.36 \pm 0.17	6.64 \pm 0.41	6.61 \pm 0.27	14.2 \pm 0.3
Hetero-DNNs	116.4 \pm 1.6	121.4 \pm 3.1	122.8 \pm 1.8	118.7 \pm 4.9	85.4 \pm 2.0	68.4 \pm 1.5	9.33 \pm 0.65	8.99 \pm 0.92	8.90 \pm 0.63	9.18 \pm 0.41	17.1 \pm 2.5

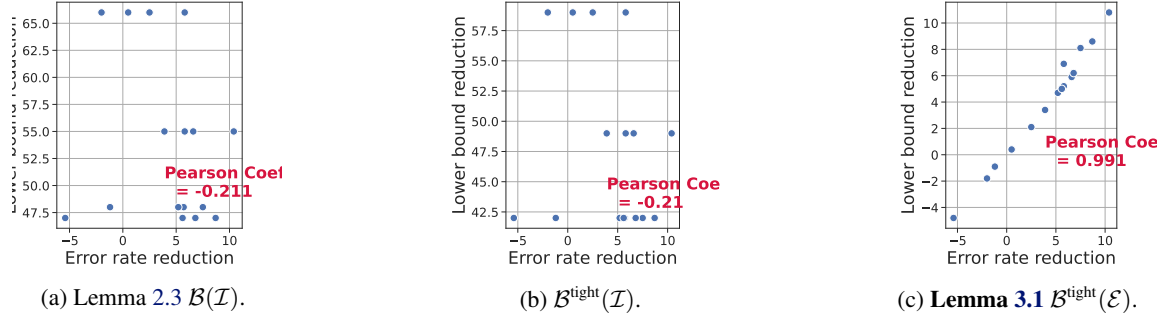


Figure K.9: **CoLA task.** Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.16 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 15.6 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.6 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.6 %. $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.1 %.

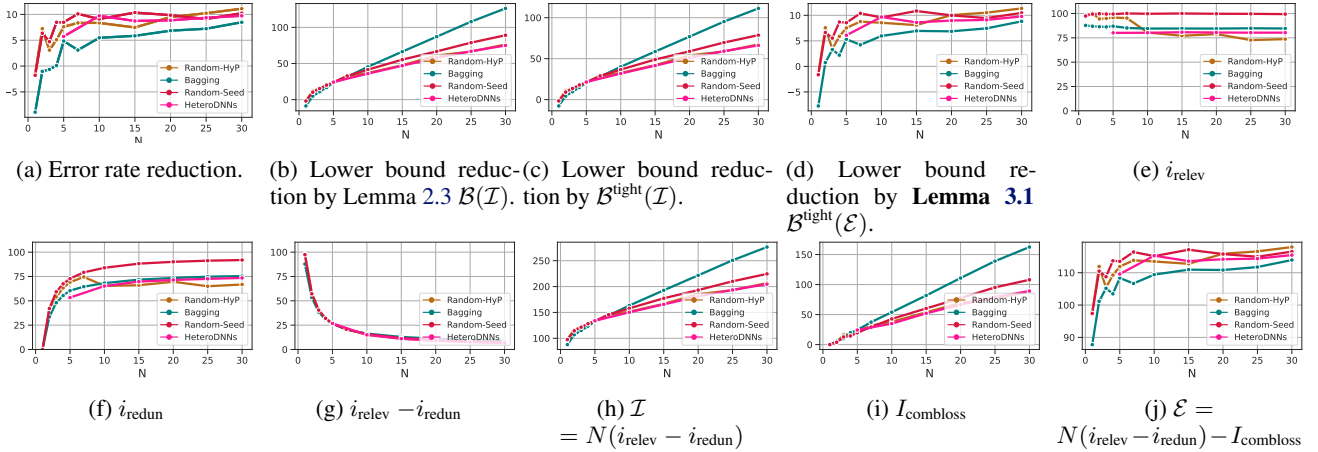


Figure K.10: **CoLA task.** The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as $i_{\{\text{relev, redu, combloss}\}} = I_{\{\text{relev, redu, combloss}\}}/N$.

Table K.16: **CoLA task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 15.6 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.6 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.6 %. $LB(s_0)$ by $\mathcal{B}(\mathcal{I})$: -2.1 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	-1.2 \pm 1.3	5.6 \pm 2.2	7.5 \pm 1.6	5.2 \pm 2.5	-0.9 \pm 1.2	5.1 \pm 2.1	8.1 \pm 1.6	4.7 \pm 2.3	42 \pm 3	48 \pm 3
Bagging	0.5 \pm 1.5	2.5 \pm 1.5	5.9 \pm 1.2	-2.0 \pm 2.3	0.4 \pm 1.3	2.1 \pm 1.4	6.9 \pm 1.6	-1.8 \pm 2.1	59 \pm 3	66 \pm 3
Random-Seed	3.9 \pm 0.5	6.6 \pm 1.7	10.4 \pm 0.5	5.8 \pm 1.0	3.4 \pm 0.4	5.9 \pm 1.6	10.8 \pm 1.1	5.2 \pm 0.9	49 \pm 3	55 \pm 3
Hetero-DNNs	-5.4 \pm 3.0	6.8 \pm 0.9	8.7 \pm 1.0	5.6 \pm 2.0	-4.8 \pm 2.5	6.2 \pm 0.9	8.6 \pm 0.7	5.0 \pm 1.9	42 \pm 1	47 \pm 1

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (i_{\text{relev}} - i_{\text{redun}} - i_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	i_{relev}	i_{redun}	Voting	LogR	SVM	RForest	$i_{\text{relev}} - i_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.252)				100	0	0	0	0	0	100
Random-HyP	98.6 \pm 1.8	108.1 \pm 3.2	112.8 \pm 2.6	107.4 \pm 3.6	77.1 \pm 2.6	66.0 \pm 2.3	4.53 \pm 0.38	3.89 \pm 0.52	3.58 \pm 0.25	3.94 \pm 0.55	11.1 \pm 3.5
Bagging	100.6 \pm 2.0	103.4 \pm 2.2	110.9 \pm 2.5	97.1 \pm 3.2	84.6 \pm 0.5	71.7 \pm 0.2	6.14 \pm 0.16	5.96 \pm 0.29	5.45 \pm 0.44	6.37 \pm 0.12	12.9 \pm 0.5
Random-Seed	105.3 \pm 0.6	109.3 \pm 2.5	117.1 \pm 1.6	108.3 \pm 1.5	100.0 \pm 0.0	88.2 \pm 0.3	4.80 \pm 0.22	4.53 \pm 0.32	4.01 \pm 0.17	4.61 \pm 0.33	11.8 \pm 0.3
Hetero-DNNs	92.4 \pm 4.0	109.8 \pm 1.4	113.6 \pm 1.1	107.9 \pm 3.0	80.7 \pm 0.4	69.7 \pm 0.4	4.88 \pm 0.23	3.72 \pm 0.08	3.47 \pm 0.05	3.84 \pm 0.18	11.0 \pm 0.6

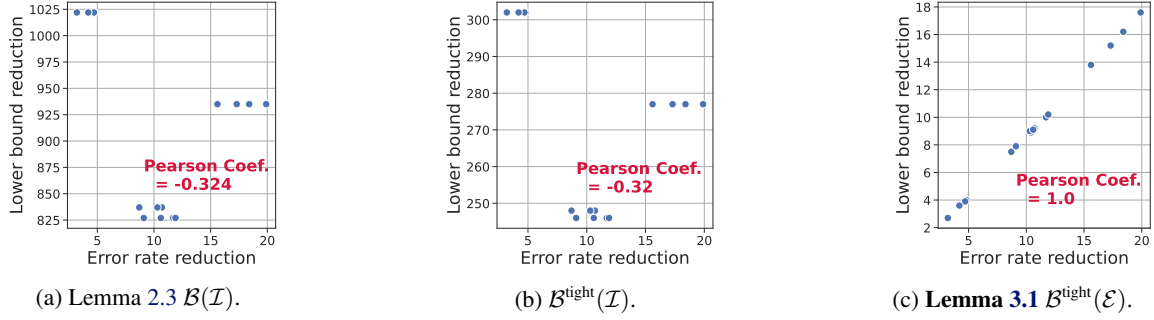


Figure K.11: **CosmosQA task.** Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.17 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 28.2 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 6.2 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 6.2 %. $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: 2.0 %.

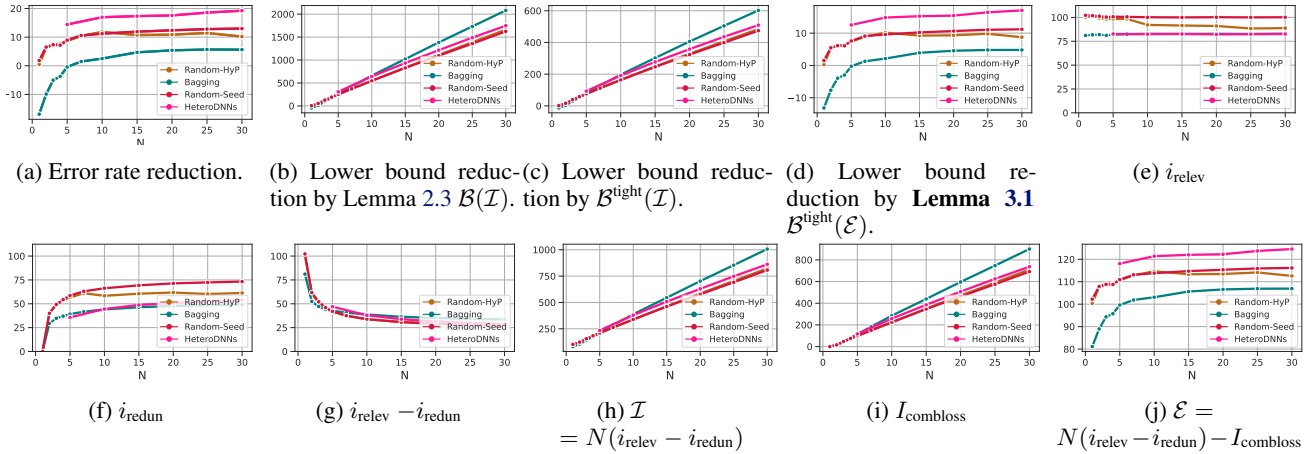


Figure K.12: **CosmosQA task.** The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as: $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$.

Table K.17: **CosmosQA task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 28.2 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 6.2 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 6.2 %. $LB(s_0)$ by $\mathcal{B}(\mathcal{I})$: 2.0 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	8.7 \pm 1.0	10.4 \pm 1.0	10.7 \pm 1.4	10.3 \pm 0.6	7.5 \pm 0.9	8.9 \pm 0.8	9.2 \pm 1.1	9.0 \pm 0.5	248 \pm 7	837 \pm 34
Bagging	3.2 \pm 0.8	4.8 \pm 0.2	4.7 \pm 0.3	4.2 \pm 0.7	2.7 \pm 0.7	4.0 \pm 0.2	3.9 \pm 0.2	3.6 \pm 0.6	302 \pm 3	1022 \pm 20
Random-Seed	10.6 \pm 1.1	11.7 \pm 0.3	11.9 \pm 0.6	9.1 \pm 2.8	9.1 \pm 1.0	10.0 \pm 0.3	10.2 \pm 0.6	7.9 \pm 2.2	246 \pm 1	827 \pm 7
Hetero-DNNs	15.6 \pm 1.3	18.4 \pm 0.7	17.3 \pm 0.7	19.9 \pm 1.2	13.8 \pm 1.1	16.2 \pm 0.6	15.2 \pm 0.6	17.6 \pm 1.1	277 \pm 5	935 \pm 9

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as: $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	LogR	$\hat{i}_{\text{combloss}}$ SVM	RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.683)				100	0	0	0	0	0	100
Random-HyP	110.8 \pm 1.2	112.9 \pm 1.3	113.3 \pm 1.8	112.9 \pm 0.9	91.5 \pm 1.9	60.6 \pm 1.6	23.57 \pm 0.47	23.43 \pm 0.54	23.40 \pm 0.56	23.43 \pm 0.51	31.0 \pm 2.5
Bagging	103.9 \pm 1.0	105.8 \pm 0.3	105.6 \pm 0.4	105.1 \pm 0.9	82.7 \pm 0.4	46.4 \pm 0.3	29.41 \pm 0.27	29.28 \pm 0.33	29.29 \pm 0.30	29.33 \pm 0.36	36.3 \pm 0.5
Random-Seed	113.1 \pm 1.3	114.5 \pm 0.3	114.7 \pm 0.7	111.4 \pm 3.3	100.0 \pm 0.0	69.3 \pm 0.4	23.16 \pm 0.46	23.07 \pm 0.39	23.06 \pm 0.42	23.28 \pm 0.25	30.7 \pm 0.4
Hetero-DNNs	119.9 \pm 1.8	123.3 \pm 1.1	122.0 \pm 1.0	125.5 \pm 1.8	82.6 \pm 0.7	48.8 \pm 0.0	25.84 \pm 0.66	25.61 \pm 0.76	25.70 \pm 0.77	25.47 \pm 0.70	33.8 \pm 0.8

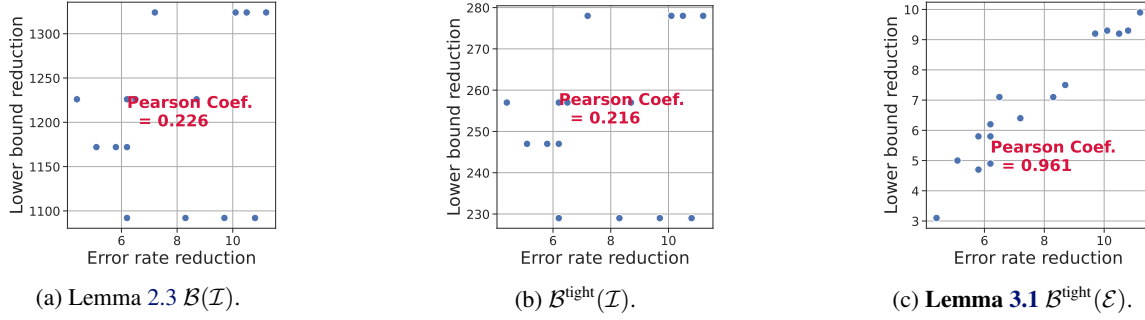


Figure K.13: **MNLI task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.18 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 18.6 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 3.7 %. $\text{LB}(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 3.7 %. $\text{LB}(s_0)$ by $\mathcal{B}(\mathcal{I})$: -1.1 %.

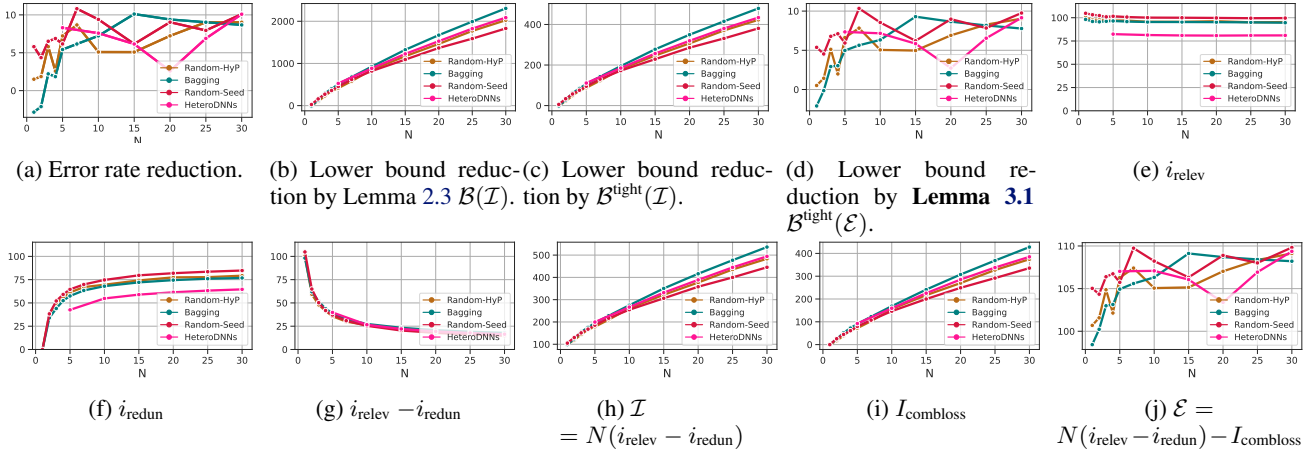


Figure K.14: **MNLI task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as $i_{\{\text{relev, redun, combloss}\}} = I_{\{\text{relev, redun, combloss}\}}/N$.

Table K.18: **MNLI task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 18.6 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 3.7 %. $LB(s_0)$ by $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 3.7 %. $LB(s_0)$ by $\mathcal{B}(\mathcal{I})$: -1.1 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	6.2 \pm 3.0	5.8 \pm 3.4	5.1 \pm 1.8	5.8 \pm 1.8	4.9 \pm 2.9	4.7 \pm 3.4	5.0 \pm 2.3	5.8 \pm 2.0	247 \pm 20	1172 \pm 52
Bagging	11.2 \pm 3.8	10.5 \pm 3.9	10.1 \pm 3.8	7.2 \pm 3.0	9.9 \pm 3.9	9.2 \pm 3.7	9.3 \pm 3.4	6.4 \pm 2.6	278 \pm 21	1324 \pm 103
Random-Seed	8.3 \pm 1.4	10.8 \pm 0.6	6.2 \pm 2.2	9.7 \pm 4.2	7.1 \pm 1.3	9.3 \pm 0.3	6.2 \pm 1.0	9.2 \pm 3.9	229 \pm 22	1092 \pm 137
Hetero-DNNs	4.4 \pm 2.0	8.7 \pm 2.8	6.2 \pm 1.4	6.5 \pm 3.3	3.1 \pm 2.1	7.5 \pm 2.6	5.8 \pm 1.1	7.1 \pm 2.4	257 \pm 13	1226 \pm 116

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	i_{relev}	i_{redun}	Voting	i_{combloss} LogR	SVM	RForest	$i_{\text{relev}} - i_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.681)				100	0	0	0	0	0	100
Random-HyP	104.3 \pm 2.6	104.2 \pm 3.0	104.4 \pm 2.1	105.1 \pm 1.8	95.6 \pm 0.5	74.2 \pm 1.2	14.47 \pm 1.26	14.48 \pm 1.28	14.46 \pm 1.21	14.41 \pm 1.13	21.4 \pm 1.3
Bagging	108.8 \pm 3.4	108.2 \pm 3.3	108.2 \pm 3.1	105.7 \pm 2.3	95.4 \pm 0.6	72.1 \pm 1.7	16.05 \pm 0.97	16.09 \pm 1.05	16.09 \pm 1.01	16.26 \pm 1.24	23.3 \pm 1.8
Random-Seed	106.3 \pm 1.1	108.3 \pm 0.2	105.5 \pm 0.9	108.1 \pm 3.5	100.0 \pm 0.0	79.6 \pm 1.4	13.27 \pm 1.38	13.14 \pm 1.39	13.32 \pm 1.44	13.15 \pm 1.61	20.4 \pm 1.4
Hetero-DNNs	102.7 \pm 1.9	106.7 \pm 2.3	105.2 \pm 1.0	106.3 \pm 2.1	81.0 \pm 0.8	58.9 \pm 1.0	15.19 \pm 0.83	14.93 \pm 0.83	15.03 \pm 0.89	14.96 \pm 0.93	22.0 \pm 1.3

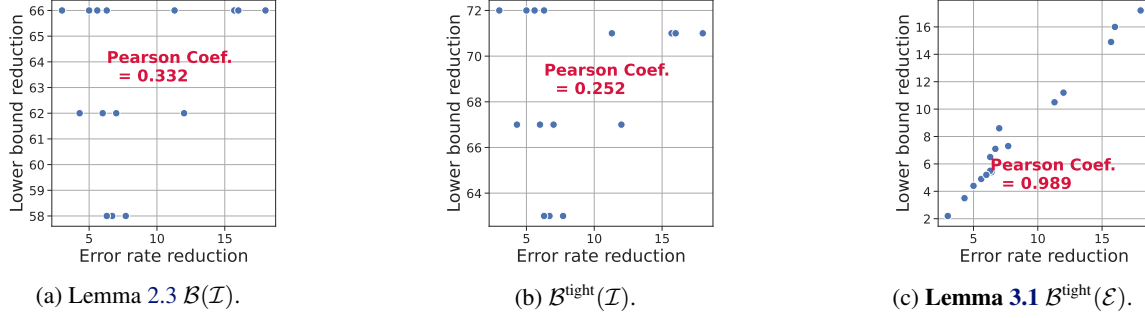


Figure K.15: **MRPC task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.19 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 13.6 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.6 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.6 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -4.0 %.

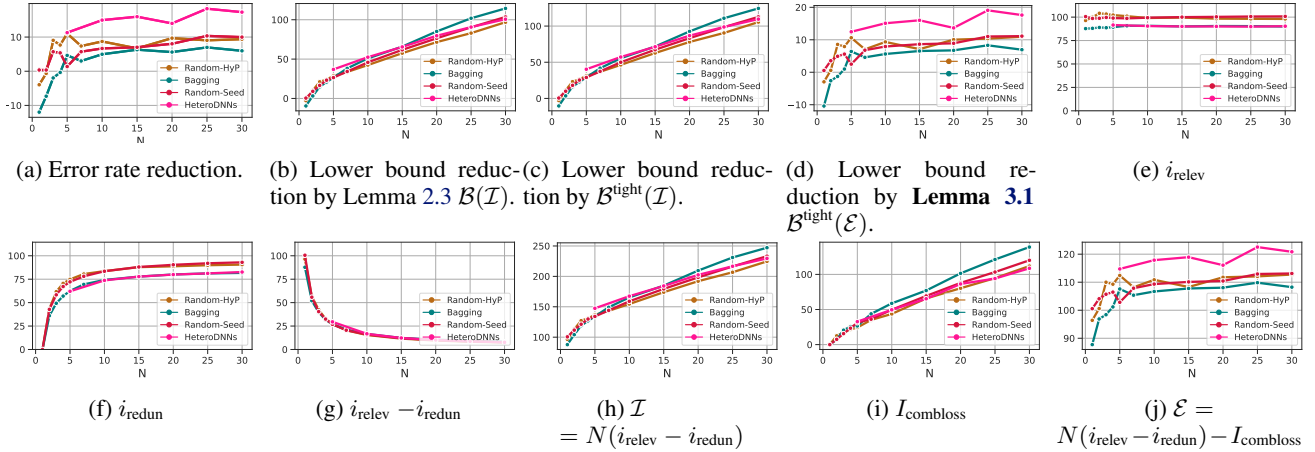


Figure K.16: **MRPC task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as: $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$.

Table K.19: **MRPC task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 13.6 %. $LB(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.6 %. $LB(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.6 %. $LB(s_0)$ of $\mathcal{B}(\mathcal{I})$: -4.0 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	7.7 \pm 3.7	6.4 \pm 2.3	6.7 \pm 3.8	6.3 \pm 2.5	7.3 \pm 3.7	5.4 \pm 2.3	7.1 \pm 3.7	5.5 \pm 2.4	63 \pm 4	58 \pm 4
Bagging	5.0 \pm 4.5	5.6 \pm 3.0	6.3 \pm 3.2	3.0 \pm 2.4	4.4 \pm 4.2	4.9 \pm 3.0	6.5 \pm 2.6	2.2 \pm 2.2	72 \pm 4	66 \pm 5
Random-Seed	12.0 \pm 3.2	4.3 \pm 2.5	7.0 \pm 2.0	6.0 \pm 2.1	11.2 \pm 3.0	3.5 \pm 2.2	8.6 \pm 2.5	5.2 \pm 1.9	67 \pm 6	62 \pm 6
Hetero-DNNs	15.7 \pm 2.1	18.0 \pm 2.2	16.0 \pm 4.3	11.3 \pm 5.6	14.9 \pm 2.1	17.2 \pm 2.3	16.0 \pm 4.7	10.5 \pm 6.1	71 \pm 2	66 \pm 4

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as: $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	LogR	$\hat{i}_{\text{combloss}}$ SVM	RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.336)				100	0	0	0	0	0	100
Random-HyP	108.5 \pm 4.1	106.3 \pm 2.6	108.3 \pm 4.1	106.5 \pm 3.0	99.5 \pm 1.7	87.9 \pm 1.5	4.37 \pm 0.50	4.51 \pm 0.52	4.38 \pm 0.49	4.50 \pm 0.41	11.6 \pm 2.3
Bagging	105.3 \pm 5.1	105.8 \pm 3.6	107.8 \pm 3.2	102.6 \pm 2.6	90.1 \pm 1.9	77.8 \pm 1.5	5.30 \pm 0.20	5.26 \pm 0.23	5.13 \pm 0.27	5.48 \pm 0.55	12.3 \pm 2.4
Random-Seed	113.1 \pm 3.2	104.1 \pm 2.6	110.1 \pm 2.6	106.1 \pm 2.2	100.0 \pm 0.0	88.0 \pm 0.5	4.42 \pm 0.67	5.02 \pm 0.72	4.62 \pm 0.70	4.89 \pm 0.69	12.0 \pm 0.5
Hetero-DNNs	117.5 \pm 2.7	120.3 \pm 3.1	118.9 \pm 5.9	112.5 \pm 7.4	90.1 \pm 1.4	77.8 \pm 1.2	4.45 \pm 0.24	4.26 \pm 0.14	4.35 \pm 0.12	4.78 \pm 0.19	12.3 \pm 1.9

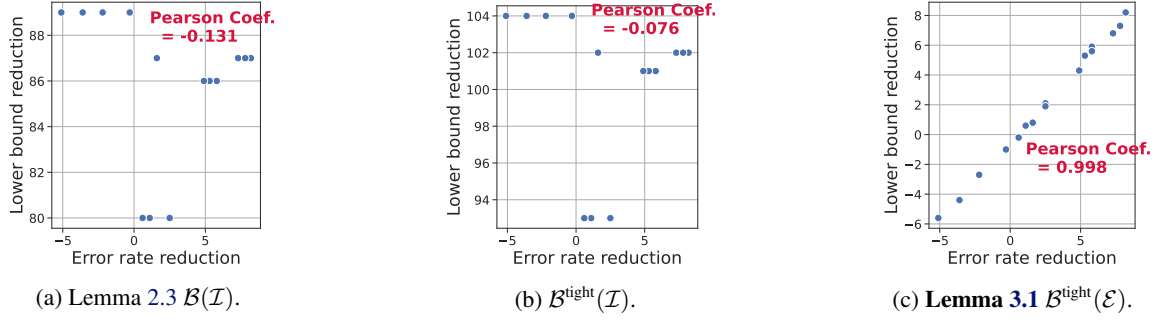


Figure K.17: **QQP task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.20 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 14.0 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.1 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.1 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -2.9 %.

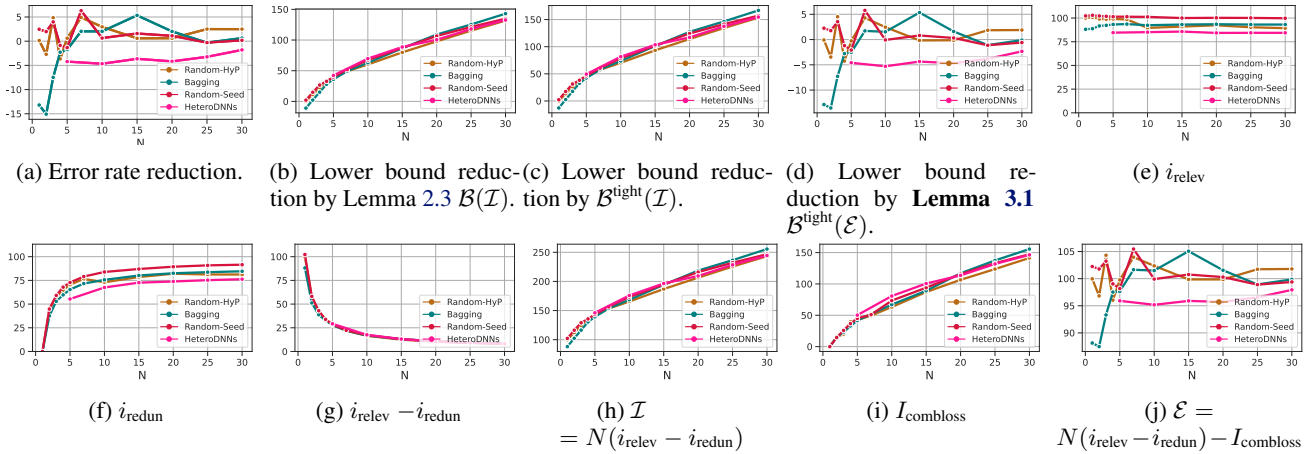


Figure K.18: **QQP task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as: $i_{\{\text{relev}, \text{redu}, \text{combloss}\}} = I_{\{\text{relev}, \text{redu}, \text{combloss}\}}/N$.

Table K.20: **QQP task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $ER(s_0)$: 14.0 %. $LB(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.1 %. $LB(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.1 %. $LB(s_0)$ of $\mathcal{B}(\mathcal{I})$: -2.9 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	2.5 \pm 2.7	1.1 \pm 4.3	0.6 \pm 3.7	2.5 \pm 1.5	2.1 \pm 2.9	0.6 \pm 4.6	-0.2 \pm 4.0	1.9 \pm 1.7	93 \pm 17	80 \pm 16
Bagging	5.8 \pm 6.0	5.8 \pm 7.1	5.3 \pm 5.4	4.9 \pm 6.0	5.9 \pm 6.7	5.6 \pm 7.7	5.3 \pm 5.9	4.3 \pm 6.5	101 \pm 3	86 \pm 2
Random-Seed	8.2 \pm 0.3	7.3 \pm 1.4	1.6 \pm 0.4	7.8 \pm 1.7	8.2 \pm 0.3	6.8 \pm 1.4	0.8 \pm 0.4	7.3 \pm 1.8	102 \pm 15	87 \pm 14
Hetero-DNNs	-5.1 \pm 1.8	-2.2 \pm 1.1	-3.6 \pm 4.0	-0.3 \pm 2.5	-5.6 \pm 1.9	-2.7 \pm 0.6	-4.4 \pm 4.0	-1.0 \pm 2.7	104 \pm 11	89 \pm 11

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as: $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	LogR	$\hat{i}_{\text{combloss}}$ SVM	RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.343)				100	0	0	0	0	0	100
Random-HyP	102.0 \pm 2.8	100.6 \pm 4.4	99.9 \pm 3.8	101.8 \pm 1.6	91.0 \pm 1.6	78.5 \pm 1.8	5.66 \pm 1.12	5.76 \pm 0.97	5.81 \pm 0.99	5.68 \pm 1.14	12.5 \pm 2.4
Bagging	105.6 \pm 6.4	105.4 \pm 7.3	105.1 \pm 5.7	104.1 \pm 6.2	93.1 \pm 2.5	80.2 \pm 2.4	5.91 \pm 0.42	5.92 \pm 0.49	5.94 \pm 0.38	6.01 \pm 0.32	12.9 \pm 3.5
Random-Seed	107.6 \pm 0.5	106.4 \pm 1.4	100.7 \pm 0.3	106.8 \pm 1.7	100.0 \pm 0.0	87.0 \pm 1.0	5.81 \pm 0.95	5.89 \pm 1.02	6.27 \pm 0.98	5.86 \pm 1.05	13.0 \pm 1.0
Hetero-DNNs	94.8 \pm 1.7	97.5 \pm 0.6	95.9 \pm 3.6	99.1 \pm 2.4	85.7 \pm 1.3	72.6 \pm 0.6	6.78 \pm 0.74	6.61 \pm 0.81	6.71 \pm 0.71	6.50 \pm 0.87	13.1 \pm 1.5

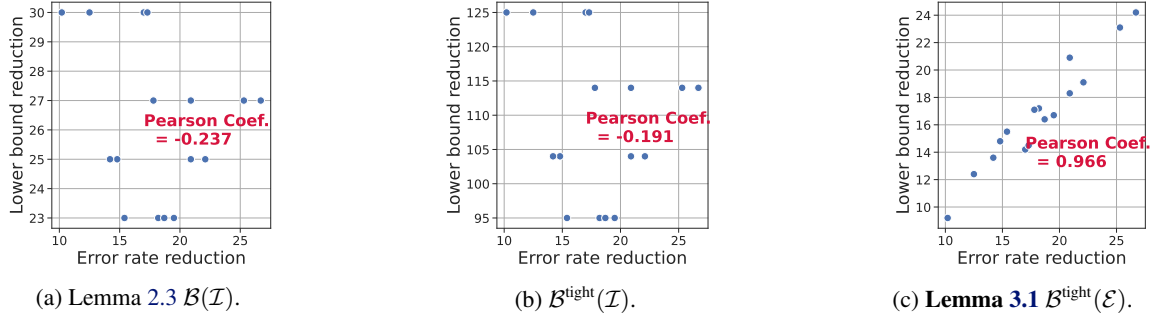


Figure K.19: **SciTail task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.21 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 5.7 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 1.2 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 1.2 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -5.2 %.

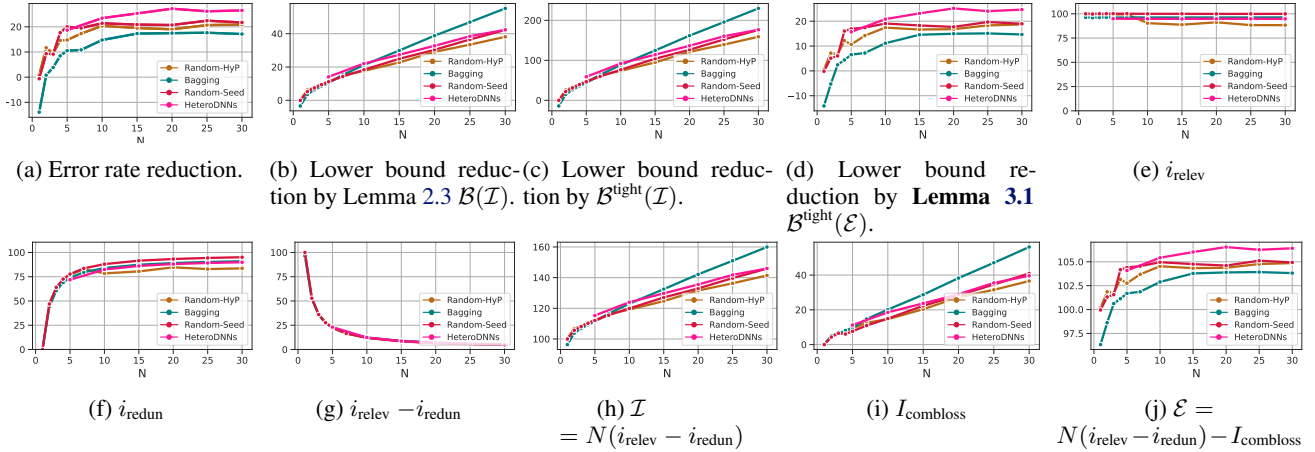


Figure K.20: **SciTail task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as: $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$.

Table K.21: **SciTail task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $\text{ER}(s_0)$: 5.7 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 1.2 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 1.2 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -5.2 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$				$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	18.2 \pm 1.1	18.7 \pm 1.1	19.5 \pm 2.3	15.4 \pm 1.5	17.2 \pm 1.3	16.4 \pm 1.8	16.7 \pm 2.6	15.5 \pm 1.5	95 \pm 2	23 \pm 0
Bagging	12.5 \pm 1.5	17.0 \pm 2.5	17.3 \pm 2.7	10.2 \pm 2.6	12.4 \pm 1.4	14.2 \pm 2.9	14.5 \pm 3.1	9.2 \pm 2.6	125 \pm 2	30 \pm 0
Random-Seed	14.8 \pm 1.4	22.1 \pm 0.5	20.9 \pm 0.9	14.2 \pm 0.7	14.8 \pm 1.6	19.1 \pm 0.5	18.3 \pm 0.7	13.6 \pm 0.4	104 \pm 4	25 \pm 1
Hetero-DNNs	20.9 \pm 2.0	26.7 \pm 1.0	25.3 \pm 2.2	17.8 \pm 4.5	20.9 \pm 2.0	24.2 \pm 1.1	23.1 \pm 2.2	17.1 \pm 4.3	114 \pm 2	27 \pm 1

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as: $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	LogR	$\hat{i}_{\text{combloss}}$ SVM	RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.641)				100	0	0	0	0	0	100
Random-HyP	104.5 \pm 0.4	104.3 \pm 0.5	104.3 \pm 0.7	104.0 \pm 0.4	88.8 \pm 1.8	80.5 \pm 1.8	1.35 \pm 0.03	1.36 \pm 0.01	1.36 \pm 0.02	1.38 \pm 0.03	8.3 \pm 2.5
Bagging	103.2 \pm 0.4	103.7 \pm 0.7	103.8 \pm 0.8	102.4 \pm 0.7	96.2 \pm 0.3	87.4 \pm 0.3	1.95 \pm 0.01	1.92 \pm 0.03	1.91 \pm 0.04	2.01 \pm 0.04	8.8 \pm 0.5
Random-Seed	103.8 \pm 0.4	105.0 \pm 0.2	104.8 \pm 0.2	103.6 \pm 0.1	100.0 \pm 0.0	91.5 \pm 0.1	1.55 \pm 0.07	1.47 \pm 0.08	1.49 \pm 0.08	1.57 \pm 0.08	8.5 \pm 0.1
Hetero-DNNs	105.4 \pm 0.5	106.3 \pm 0.3	106.0 \pm 0.6	104.5 \pm 1.1	94.8 \pm 0.5	86.1 \pm 0.5	1.62 \pm 0.08	1.56 \pm 0.07	1.58 \pm 0.09	1.68 \pm 0.12	8.6 \pm 0.7

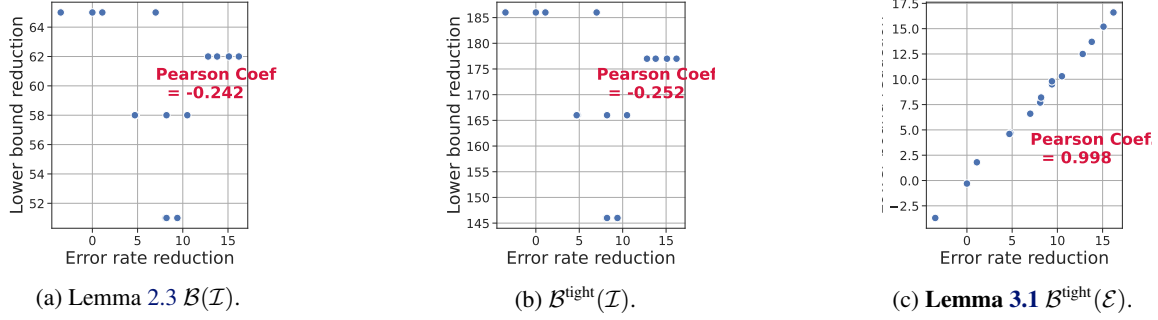


Figure K.21: **SST task**. Correlations between error rate reductions and lower bound reductions. Each figure uses different type of lower bound. Each point in the figures shows a quantity of a specific ensemble system s and the quantity is the average over the eight tasks. See Table K.22 for the real value of each point. We used the 16 ensemble systems described in Section 4.2. Each system s used $N = 15$ models. The baseline values in (8) and (9) were the followings: $\text{ER}(s_0)$: 15.7 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.3 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.3 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -3.0 %.

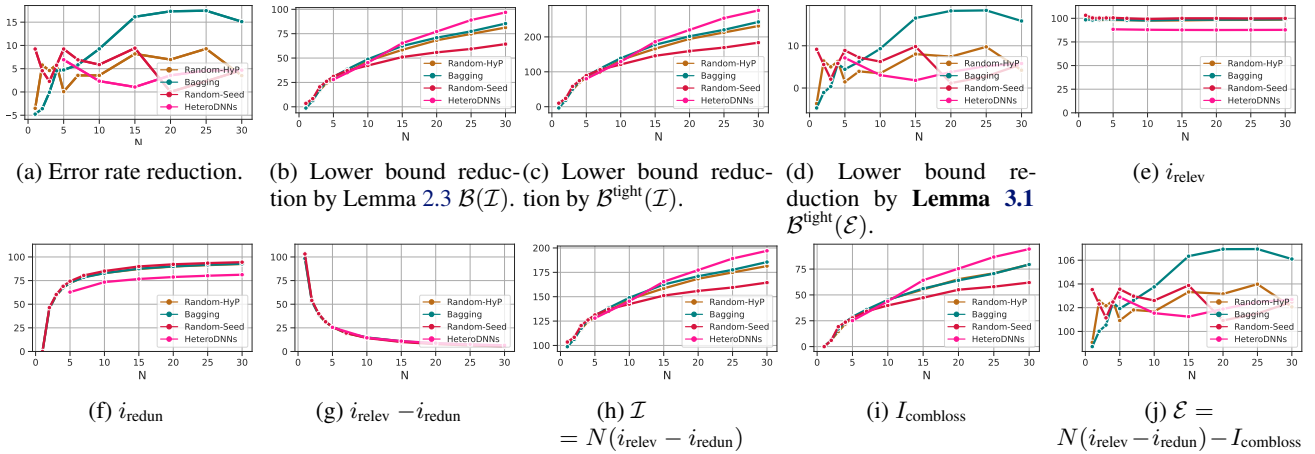


Figure K.22: **SST task**. The change in ensemble quantities when the number of models N is changed. Each figure shows a specific quantity. The ensemble systems used the SVM model combination. Each value is an averages of the eight tasks. i denotes per-model metric values defined as: $i_{\{\text{relev}, \text{redun}, \text{combloss}\}} = \mathcal{I}_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$.

Table K.22: **SST task**. Statistics of ensemble systems described in Section 4.2. The rows and columns list the model generation and combination methods of Table 2, respectively. Each cell shows a quantity of a specific system s . Each quantity is the average over the eight tasks. Each system contains $N = 15$ models. Color shows the rank within *each column* (brighter is better).

(a) Error rate reductions and lower bound reductions. The baseline values used in (8) and (9) were the followings. $\text{ER}(s_0)$: 15.7 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{E})$: 2.3 %. $\text{LB}(s_0)$ of $\mathcal{B}^{\text{tight}}(\mathcal{I})$: 2.3 %. $\text{LB}(s_0)$ of $\mathcal{B}(\mathcal{I})$: -3.0 %.

	Error rate reductions (8)				Lower bound reductions (9)					
	Voting	LogR	SVM	RForest	Voting	Lemma 3.1 $\mathcal{B}^{\text{tight}}(\mathcal{E})$			$\mathcal{B}^{\text{tight}}(\mathcal{I})$	Lemma 2.3 $\mathcal{B}(\mathcal{I})$
Random-HyP	4.8 \pm 7.4	10.5 \pm 3.4	8.2 \pm 2.9	4.7 \pm 7.6	4.7 \pm 7.3	10.3 \pm 3.5	8.0 \pm 2.8	4.6 \pm 7.5	166 \pm 28	58 \pm 11
Bagging	15.1 \pm 7.7	12.8 \pm 6.1	16.2 \pm 10.8	13.8 \pm 9.9	15.2 \pm 8.1	12.5 \pm 6.3	16.6 \pm 11.0	13.7 \pm 10.2	177 \pm 28	62 \pm 11
Random-Seed	9.4 \pm 5.8	8.1 \pm 2.8	9.4 \pm 5.8	8.2 \pm 2.8	9.5 \pm 6.1	7.7 \pm 2.8	9.8 \pm 5.6	8.2 \pm 2.9	146 \pm 16	51 \pm 6
Hetero-DNNs	-3.5 \pm 4.5	7.0 \pm 1.8	1.1 \pm 5.8	0	-3.7 \pm 4.5	6.6 \pm 1.8	1.8 \pm 6.1	-0.3 \pm 1.9	186 \pm 14	65 \pm 7

(b) Breakdown of ensemble strength defined in (7). We show per-model metric values defined as: $\hat{i}_{\{\text{relev}, \text{redun}, \text{combloss}\}} = I_{\{\text{relev}, \text{redun}, \text{combloss}\}}/N$. Thus, $\mathcal{E} = (\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}} - \hat{i}_{\text{combloss}}) \times N$ holds. For intuitive understanding, all the values are normalized by the ensemble strength of baseline \mathcal{E}_{s_0} , for example, $I_{\text{relev}} = \hat{I}_{\text{relev}}/\mathcal{E}_{s_0} \times 100$ where \hat{I}_{relev} is the raw value.

	$\mathcal{E}(\mathbf{O}, Y, \hat{Y})$				Per-model metric values						
	Voting	LogR	SVM	RForest	\hat{i}_{relev}	\hat{i}_{redun}	Voting	$\hat{i}_{\text{combloss}}$		RForest	$\hat{i}_{\text{relev}} - \hat{i}_{\text{redun}}$
Baseline (s_0)	100 (the raw value is 0.705)				100	0	0	0	0	0	100
Random-HyP	101.6 \pm 2.5	103.5 \pm 1.2	102.8 \pm 0.9	101.6 \pm 2.6	97.7 \pm 0.5	87.2 \pm 0.2	3.76 \pm 0.72	3.63 \pm 0.63	3.68 \pm 0.67	3.76 \pm 0.60	10.5 \pm 0.6
Bagging	105.3 \pm 2.8	104.3 \pm 2.2	105.8 \pm 3.9	104.8 \pm 3.6	98.4 \pm 1.2	87.6 \pm 0.6	3.78 \pm 0.53	3.84 \pm 0.58	3.74 \pm 0.45	3.81 \pm 0.50	10.8 \pm 1.4
Random-Seed	103.2 \pm 2.1	102.7 \pm 1.0	103.4 \pm 1.9	102.8 \pm 0.9	100.0 \pm 0.0	90.0 \pm 0.3	3.16 \pm 0.23	3.19 \pm 0.39	3.15 \pm 0.24	3.19 \pm 0.29	10.0 \pm 0.3
Hetero-DNNs	98.7 \pm 1.5	102.3 \pm 0.7	100.7 \pm 2.1	99.9 \pm 0.7	87.7 \pm 0.6	76.7 \pm 0.3	4.41 \pm 0.27	4.17 \pm 0.34	4.28 \pm 0.31	4.33 \pm 0.33	11.0 \pm 0.6