An Overview of Uncertainty Calibration for Text Classification and the Role of Distillation

Han Guo **Ramakanth Pasunuru** **Mohit Bansal**
UNC Chapel Hill
hanguo@cs.cmu.edu {ram, mbansal}@cs.unc.edu

Abstract

Recent advances in NLP systems, notably the pretraining-and-finetuning paradigm, have achieved great success in predictive accuracy. However, these systems are usually not well calibrated for uncertainty out-of-the-box. Many recalibration methods have been proposed in the literature for quantifying predictive uncertainty and calibrating model outputs, with varying degrees of complexity. In this work, we present a systematic study of a few of these methods. Focusing on the text classification task and finetuned large pretrained language models, we first show that many of the finetuned models are not well calibrated out-of-the-box, especially when the data come from out-of-domain settings. Next, we compare the effectiveness of a few widely-used recalibration methods (such as ensembles, temperature scaling). Then, we empirically illustrate a connection between distillation and calibration. We view distillation as a regularization term encouraging the student model to output uncertainties that match those of a teacher model. With this insight, we develop simple recalibration methods based on distillation with no additional inference-time cost. We show on the GLUE benchmark that our simple methods can achieve competitive out-of-domain (OOD) calibration performance w.r.t. more expensive approaches. Finally, we include ablations to understand the usefulness of components of our proposed method and examine the transferability of calibration via distillation.

1 Introduction

The recent success of NLP systems, notably the pretraining-and-finetuning paradigm has led to widespread applications (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019). However, these systems are not always well-calibrated; in many high-stake decision-making scenarios such as medical diagnosis, even small errors would have large damage. Suppose an ML system predicts a 20% probability a patient has cancer whereas the reality is 40%, diagnosis relying on inaccurate estimates could lead to devastating consequences (Kumar et al., 2019). Further, interpreting and communicating these uncertainties facilitates better trust between humans and ML systems (Bansal et al., 2020; Wilder et al., 2020; Ribeiro et al., 2016, 2018).

Hence, it is increasingly important for users to understand not only when the systems would succeed, but also when they could fail. One seemingly straightforward approach is to have the systems output predictions and some measure of their confidence/uncertainty. Users could then use both the predictions and associated uncertainties to decide how much they would trust the prediction. For example, one might decide to take an umbrella to work only if the confidence of the rain prediction is more than 50%. For many statistical methods, confidence/uncertainty is either part of the system by design (e.g., Bayesian methods) or could be efficiently estimated (e.g., linear regressions). Unfortunately, for large-scale DNNs, estimating uncertainty becomes a challenge (Gal, 2016): e.g., nominal probabilities from the softmax function are shown to be uncalibrated estimates of model uncertainty (Platt, 1999; Niculescu-Mizil and Caruana, 2005; Guo et al., 2017; Ovadia et al., 2019).

In this work, we present a systematic study on recalibrating current NLP systems, particularly those that fall in the recent popular pretraining-and-finetuning paradigm (Hendrycks et al., 2020; Desai and Durrett, 2020), as they are widely deployed in recent state-of-the-art systems and hence it is important that they are well calibrated for safety and transparency. However, the methods discussed in this work could generalize to a broader range of systems. We focus on the calibration not only of the task itself, but also under dataset distributional

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 289–306
Bangkok, Thailand (Online), August 6, 2021. ©2021 Association for Computational Linguistics

shift (Ovadia et al., 2019).

We start by introducing uncertainty and calibration, and cover related advances in the deep learning literature. In addition to widely-used maximum calibration error and expected calibration error, we follow previous works (Ovadia et al., 2019; Kumar et al., 2019) and include additional calibration evaluation metrics for better comparisons (e.g., Brier scores and ℓ_p calibration error).

We conduct experiments on GLUE classification tasks (Wang et al., 2019) and show that finetuned language models are usually not calibrated out-of-the-box, especially when the data comes from *a distribution different from the training data*. We use the term "out-of-domain" (or "out-of-distribution", OOD) to refer to the setting where the train and evaluation data come from different "distributions". Related works in NLP have considered data from similar tasks but from different datasets as OOD (Ovadia et al., 2019; Hendrycks and Gimpel, 2017). Next, in order to make models more calibrated, we study some of the widely-used recalibration methods, with various degrees of effectiveness and computational cost. For example, ensembling models has been shown to be very effective in out-of-domain settings (Ovadia et al., 2019), but the cost of computation scales with the size of ensembles. On the other hand, distillation (Hinton et al., 2015) is a widely-known method for improving the system's performance by learning from a stronger teacher model. In this work, we empirically examine the connection between distillation and calibration. Notably, we view the objective function of distillation as a regularization term that encourages the student model to match the predictive uncertainty of a stronger, more calibrated teacher model.

We conduct analysis experiments to show that the teacher's calibration performance could be distilled into the student model, even when the teacher model's accuracy remains similar. With this insight, we show that simple methods based on distillation could achieve competitive performance in out-of-domain calibration, without introducing extra computation at inference time. Finally, we also conduct ablation experiments to understand the usefulness of components of the method. In summary, our contributions are listed as follows:

- We present a systematic study on the performance of various recalibration methods on finetuned language models for both in-domain and out-of-domain settings.
- We empirically examine the connection between distillation and calibration, and conduct experiments showing that distillation can distill calibration performance.
- We describe two simple recalibration methods, and experimental results demonstrate their competitiveness in the out-of-domain settings; finally, we also ablate method's components and measure the extent to which distillation transfers teachers' calibration improvement.

2 Background and Related Works

Due to space constraints, we present some of the most relevant materials in the main paper. Please see the appendix (Sec. A) for extended background and related works.

The quality of the uncertainty measurement is usually measured via calibration (Kendall and Gal, 2017). In the context of calibration, the uncertainties often refer to predictive probabilities. The model is calibrated if the predictive probabilities match the empirical frequency of the data (Gal, 2016). Let $\hat{Y}$ and $\hat{P}$ be the predicted class and its associated confidence of a neural network. We would like the confidence estimates $\hat{P}$ to be calibrated, which intuitively means that we want $\hat{P}$ to represent true probabilities (Guo et al., 2017):

$$\mathbb{P}(\hat{Y} = Y | \hat{P} = p) = p, \forall p \in [0, 1]. \qquad (1)$$

Suppose a classification model is given N input examples, and made predictions $\hat{y}_1, ..., \hat{y}_N$, each with $\hat{p} = 0.35$. We would expect 35% of the predictions would be correct. The problem of uncertainty/confidence calibration and confidence scores have been studied and applied in various settings such as structured prediction problems (Kuleshov and Liang, 2015), online recalibration (with potentially adversarial/OOD input) (Kuleshov and Ermon, 2017), model regularization (Pereyra et al., 2017), and misclassified/OOD examples detection (Hendrycks and Gimpel, 2017). In practice, however, perfect calibration is almost impossible (Guo et al., 2017), and estimating the first term in Eq. 1 is not straightforward using finite samples, because in most cases $\hat{P}$ is a continuous random variable (Guo et al., 2017; Kumar et al., 2019). In Sec. 3, we describe ways to estimate the calibration performance.

It has been widely observed that modern neural networks are usually not calibrated out of the box (Platt, 1999; Zadrozny and Elkan, 2001; Guo et al., 2017; Ovadia et al., 2019). Recalibration methods improve calibration by transforming un-calibrated outputs into calibrated outputs/probabilities, and they include scaling-based methods (Platt, 1999; Guo et al., 2017), histogram-binning-based methods (Guo et al., 2017; Zadrozny and Elkan, 2001), and ensembles (Lakshminarayanan et al., 2017). Recently, Kumar et al. (2019) proposed the scaling-binning calibrator and a more sample-efficient estimator of calibration error. In our work, we describe simple approaches that combine the strength of ensembles and temperature scaling without introducing computation at inference time; we further apply the scaling-binning calibrator to ensure calibration.

Ensemble-based methods work by aggregating multiple networks trained independently on the entire dataset, and has been shown to achieve strong performance in out-of-domain calibration (Ovadia et al., 2019; Lakshminarayanan et al., 2017). More generally, there are randomization-based ensembles and boosting-based ensembles. Within the randomization-based ensembles, we use the entire training dataset to train each model instead of different bootstrap samples of the original training set (Lakshminarayanan et al., 2017).

Temperature scaling is an extension of Platt scaling (Guo et al., 2017). It uses a single scalar parameter $T > 0$ for all classes. Given output z_i, the confidence prediction is:

$$\hat{p}_i = \max_k \sigma(z_{i,k}/T). \tag{2}$$

An extension, called heteroscedastic regression, is used in our work, which replaces the constant scalar with learned values (Kendall and Gal, 2017; Kendall et al., 2018).

Knowledge distillation (Hinton et al., 2015) is a compression technique in which a compact model (usually referred to as the student model) is trained to mimic the behavior of a more powerful teacher model. In the context of classification, knowledge distillation works by augmenting the loss function with an additional term $D_{KL}(p_i \| p_j)$ where $p_i = \mathrm{softmax}(z_i/T)$ and $p_j = \mathrm{softmax}(z_j/T)$ with z_i and z_j the logits from two models, and T controls the smoothness of the output distribution. In this work, we show that distillation can also be used to distill calibration performance, and use it to build simple yet competitive recalibration methods.

Concurrently, Desai and Durrett (2020) studied the calibration of pretrained transformers when finetuned to downstream tasks, and Hendrycks et al. (2020) studied the out-of-distribution robustness of pretrained transformers. We are different from them in that first we present a systematic study on the out-of-distribution calibration; second we draw insights from the connection between distillation and temperature scaling to design simple yet competitive recalibration methods; third, we conduct experiments to understand the connection between them empirically; finally, we also include a more comprehensive set of calibration evaluations following Ovadia et al. (2019) and Kumar et al. (2019).

3 Measuring Calibration Errors

3.1 Calibration Error Metrics

Let $\mathcal{X}$ be the input space, and $\mathcal{Y} = \{1, ..., K\}$ be the label space, and $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables denoting the input and the label, respectively. Further, let $f : \mathcal{X} \to [0,1]^K$ be a neural network that outputs the model's confidence for each class. For simplicity of notation, we define $\hat{Y} = \arg\max_j f(X)_j$, and $\hat{P} = \max_j f(X)_j$.

Expected Calibration Error. One notion of miscalibration is the expected difference between confidence and accuracy,

$$\mathrm{ECE}(f) = \mathbb{E}\left[\left|\mathbb{P}\big(Y{=}\hat{Y}|P{=}\hat{P}\big) - \hat{P}\right|\right]. \tag{3}$$

As mentioned in Sec. 2, this cannot be estimated using finitely many samples if $\hat{P}$ is a continuous random variable. Expected Calibration Error (Naeini et al., 2015; Guo et al., 2017), or ECE, approximates this via partitioning predictions into multiple bins and computing the weighted average.

Maximum Calibration Error. In high-risk scenarios, we might be interested in measuring the worst-case performance. Maximum Calibration Error (Naeini et al., 2015; Guo et al., 2017), or MCE, estimates the following quantity via binning,

$$\mathrm{MCE}(f) = \max \left|\mathbb{P}\big(\hat{Y}{=}Y|P{=}\hat{P}\big) - \hat{P}\right|. \tag{4}$$

Brier Score. Calibration alone is not sufficient. We could construct cases in which the outputs of the model are calibrated but not useful. An example includes always outputting 50% in a binary classification task containing 50% of both labels (Kumar et al., 2019). An alternative measure is the Brier score (Brier, 1950), $\mathbb{E}[(f(X)-Y)^2]$. Note that the Brier Score is a proper scoring rule, thus the optimum score corresponds to a system with perfect calibration. We refer a more detailed discussion on proper scoring rule to Lakshminarayanan et al. (2017) (Sec 2.2). An extension of Brier Score is Brier Skill Scores (BSS). BSS is favored when the classes are imbalanced. In our early experiments, we did not observe significant ranking changes between these two measures, so we report Brier Score for simplicity.[1]

ℓ_p Calibration Error. A generalized notion of the calibration error is described in Kumar et al. (2019),

$$\mathrm{CE}(f) = \left(\mathbb{E}\big[|\mathbb{P}(Y{=}\hat{Y}|P{=}\hat{P}) - \hat{P}|^{p}\big]\right)^{1/p}. \quad (5)$$

This recovers the MCE when $p = \infty$ and ECE when $p = 1$ (Kumar et al., 2019). When $p = 2$, we refer to it as Squared Calibration Error (SCE).[2] This is estimated via binning the outputs and labels in practice similar to ECE and MCE. The plugin estimate for each term in the calibration error has been shown to be a biased estimate in Kumar et al. (2019), and the authors encouraged the use of a debiased estimator for the calibration error. We refer to this as the debiased Squared Calibration Error.

3.2 Underestimation of Calibration Errors for Model with Continuous Outputs

As noted in Sec. 2, the key to estimating the calibration error is estimating the conditional expectation $\mathbb{E}[Y|f(X)]$. However, if $f(X)$ is continuous, without smoothness assumptions on $\mathbb{E}[Y|f(X)]$, this is impossible (Kumar et al., 2019). An approximation could be made via binning the outputs into B intervals, as is done in most of the metrics aforementioned. However, Kumar et al. (2019) showed that the binned version always has a lower calibration error. The authors introduced the scaling-binning calibrator, which first fits a parametric function and then bins the function values to ensure calibration. Thus, in addition to reporting results using the metrics described in Sec. 3.1, we report results by running the scaling-binning calibrator on top of each method that we considered.[3] We further include ECE results with multiple bin-values in order to reduce the gap.

4 Methods

4.1 Baseline Model

Our baseline model follows the general finetuning of large pretrained language models on downstream tasks: we finetune RoBERTa-base (Liu et al., 2019) on downstream tasks.

4.2 Distillation and Uncertainty

Despite the strong empirical performance of many calibration methods (e.g., ensembles), their usefulness in practice is limited due to increased computation and/or memory costs at inference time (Ovadia et al., 2019). In Sec. 4.3, we describe a simple baseline: recalibrate, ensemble, and distill.

Distillation has been shown to mostly "preserve" performance in terms of accuracy – stronger teacher models tend to translate to stronger students (Hinton et al., 2015). However, whether distillation could also "preserve" calibration performance is less studied. A model with better performance does not necessarily translate to better calibration (Guo et al., 2017). Here, we briefly look at the distillation's objective from an angle of uncertainty matching, and show that they are related intuitively. Sec. 6.1 provides empirical evidence showing that the teacher model's calibration performance could be distilled into the student model.

There are two ways to see the connection. First, note that distillation tries to minimize the KL-divergence between the teacher output distribution and the student output distribution. This intuitively regularizes the student model to output confidence values that would be close to the confidence values from the teacher model. Later in Sec. 6.1, experimental results show that the confidences from two models indeed correlate positively. Another perspective, which we elaborate below, considers distillation as encouraging the students to output uncertainty close to that of teacher models.

[1] One can further include negative log-likelihood score. However, we want to avoid overcrowding the results table with too many numbers (which is already large, please see the supplementary materials Table 3-6). Since both Brier Score and NLL are proper-scoring rules (see Sec.3 in Ovadia et al. (2019)), we believe the results would be qualitatively similar.

[2] Technically, this is 2-norm Calibration Error. But we refer to this as the Squared Calibration Error for notation simplicity.

[3] The top-label variant of scaling-binning calibrator we use outputs calibrated probabilities of the top predictions, whereas Brier Scores require full probability vectors. Thus we exclude Brier Scores when using the scaling-binning calibrator.

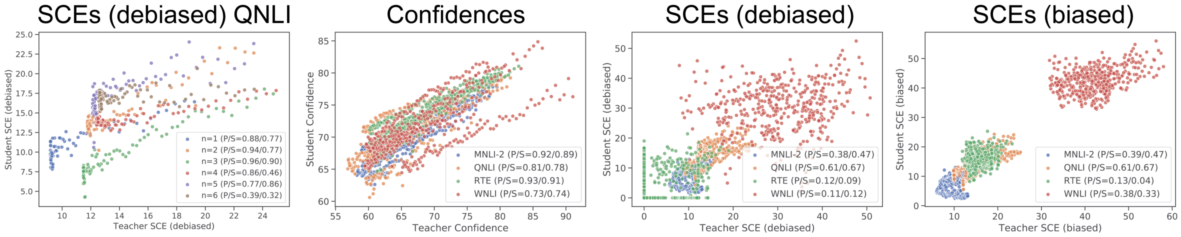


Figure 1: **Left-most Figure:** Visualization of calibration performance, measured by SCEs (debiased), between teacher and student models, trained on RTE and evaluated on QNLI. The n in the legend refers to the size of ensemble(s). *One metric/task, emphasizing different ensemble sizes.* **The Other Three Figures:** These are zoomed-out versions of the left-most figure, along with other tasks. Instead of using color to imply the ensemble size, here the color refers to the task in which the models are evaluated, and points of different ensemble sizes but the same evaluation task are aggregated and represented by the same color. Each sub-figure represents the evaluation metric. *More tasks/metrics, less emphasis on ensemble sizes.* **All Figures:** The X-axis refers to the teacher model performance, and the Y-axis refers to the student model performance. Each dot represents a different configuration used in the teacher model. The P/S in the legends refer to the Pearson/Spearman correlations.

We start by defining a loss function as a weighted combination of the regular cross entropy loss function and a regularization term that measures the difference in the uncertainty between the student model, θ, and the teacher model, $\theta^\star$,

$$L(\theta) = (1-\alpha)L_{XE}(\theta) + \alpha|H(\theta) - H(\theta^\star)|, \quad (6)$$

where H refers to predictive entropy (Gal, 2016), and is defined as (θ is ignored for simplicity),

$$H(y|x,\mathcal{D}) = -\sum_c p(y{=}c|x,\mathcal{D}) \log p(y{=}c|x,\mathcal{D}). \quad (7)$$

Gal (2016) showed that $H(y|x,\mathcal{D})$ could be approximated using samples from the (approximate) posterior distribution of the parameters. In practice, this could be satisfied, for example, if the student model is trained using dropout, and the teacher model uses either MC-dropout or ensembles.[4] Next, suppose we approximate one of the predictive entropy terms using cross entropy. This turns the second term in Eq. 6 into KL-divergence, and hence recovers the distillation objective.[5]

4.3 Recalibrate, Ensemble, and Distill

This simple algebraic manipulation shows that distillation has the effect of encouraging the student model to match the teacher model's uncertainty, and motivates us to build a simple recalibration method **"recalibrate, ensemble, and distill"** by first building an expensive yet calibrated teacher model (an ensemble of models each of which is recalibrated using temperature scaling),[6] and then distilling the expensive teacher model into a cheaper student model.

The training cost is roughly $(N+1)C_0 + C_1$, where N is the ensemble size, C_0 the cost of training the baseline, $+1$ comes from distillation, and C_1 comes from training the temperature scaling model (which is relatively cheap). However, the inference cost is almost the same as a single model (i.e., small overhead), which is very useful when inference is the primary concern (e.g., deployment).

4.4 Choosing the Distillation Temperature

The distillation term is often written as:

$$D_{\mathrm{KL}}\Big(P(x;\theta^\star,T) \;||\; P(x;\theta,T)\Big), \quad (8)$$

where $P(x;\theta,T) = \mathrm{softmax}(f(x;\theta)/T)$ and T is usually a hyperparameter to be tuned. One might notice that this is similar to the equation of temperature scaling (Eq. 2). This, together with the uncertainty matching viewpoint, motivates a small change to the distillation: we can remove the T from the student, and choose the constant $\hat{T}$ for the teacher that minimizes the calibration error,

$$D_{\mathrm{KL}}\Big(P(x;\theta^\star, \underset{\hat{T}}{\arg\min}\,\mathrm{CE}(\theta^\star,\hat{T})) \;||\; P(x;\theta)\Big), \quad (9)$$

[4] Note that the samples from a model using dropout (MC-dropout) or ensemble could be used to approximate the posterior distribution (Gal, 2016; Lakshminarayanan et al., 2017).

[5] Note that the approximation error equals the KL divergence, the term that the objective function seeks to minimize. As KL-divergence decreases, the approximation error also decreases.

[6] There are many ways to construct a powerful/expensive teacher model, and we choose the popular ensemble method for simplicity. Alternatives includes MC-dropout (with multiple forward passes) and SWA (Izmailov et al., 2018).

which is similar to performing another temperature scaling. The motivation is that we want the student model to produce calibrated probabilities rather than the scaled version of the student. If we simultaneously scale the student by T, then $f(x;\theta)/T$ would be calibrated, but the student model itself would not. We want to emphasize here that we are not the first ones to describe the connection between distillation and calibration, related findings have been presented in previous works (Tang et al., 2020; Müller et al., 2019). However, we believe our view from the angle of predictive entropy is novel. More importantly, we conduct extensive experiments and analyses in the context of finetuned language models for several text classification tasks, to empirically verify that calibration performance between student and teacher model is correlated.

5 Setup

We include additional details in the supplementary materials. Also included are expanded experiment results, such as figures evaluated on more tasks using more evaluation metrics (Sec. 6.1), and detailed/expanded results tables as well as accuracy and ECEs with multiple bin-sizes (Sec. 6.2).

Model. Our codebase is largely based on HuggingFace `Transformers` (Wolf et al., 2019). When applicable, we use an ensemble size 2, and choose $\hat{T}$ (Eq. 9) based on the Brier Scores on the validation dataset. The baseline model has 125.2M parameters, the temperature-scaling model (heteroscedastic variant) has 125.8M, and our method has 125.2M (same as the baseline model).

Data. We perform experiments on the classification tasks from the GLUE Benchmark (Wang et al., 2019), and we refer readers to Wang et al. (2019) regarding dataset statistics. Because the calculation of calibration errors requires access to the ground truth data, which is not available for GLUE data, we split the validation dataset into two halves, one for validation and the other for test, following Desai and Durrett (2020). For MultiNLI, we merge the results for both MultiNLI matched and mismatched sections. When computing the out-of-domain performance between the 3-label MultiNLI and other 2-label NLI tasks, we follow `jiant` (Pruksachatkun et al., 2020) and merge the predictions/labels that correspond to “neutral” and “contradiction” into a single category.

Evaluation. Our evaluation follows Guo et al. (2017), Ovadia et al. (2019), and Kumar et al. (2019). The train and evaluation data come from the same task for in-domain evaluations, but they come from different tasks of the same type for out-of-domain evaluations. We group MRPC and QQP (paraphrase tasks), and group MNLI (2-label version), QNLI, RTE, and WNLI (NLI tasks). We leave SST-2 (sentiment), CoLA (acceptability), and MNLI (3-label version, NLI) as separate groups. We use the in-domain validation data to train the scaling-binning calibrator.[7]

Analysis Experiments Details. We conduct experiments on RTE, in which we distill teacher models with different ensemble-sizes (from 1 to 6) and the temperature scaling constant (from 0.50 to 2.00 with a step size of 0.02) to student models. Each model is then evaluated on both in-domain task (RTE) and out-of-domain tasks (MNLI-2, QNLI, WNLI) using confidence, ECE, MCE, Brier Scores, SCE (debiased) and SCE (biased). The numbers represent performances on the validation dataset.

6 Experiments

6.1 Analysis Experiments

Sec. 4.2 shows the connection between distillation and uncertainty regularization. In this section, we perform analysis experiments examining the correlation between the calibration performance of the teacher models and student models. We conduct experiments on RTE, in which we distill teacher models with different ensemble-sizes and the temperature scaling constant to student models. Each model is then evaluated on both in-domain and out-of-domain tasks. Numbers here represent performances on the validation dataset.

We start by examining the calibration performances of teacher and student models, where we vary the calibration performance of the teacher model while holding the accuracy almost the same.[8] Fig. 1 (left) shows the debiased Squared Calibration Error of models trained on RTE and

[7] We only use the 2-label version of MNLI for evaluation. We use accuracy for CoLA evaluation so that calibration error computations would be more consistent across tasks.

[8] Note the accuracy of teacher models with the same ensemble size but different temperature scaling constants would be almost the same, as for each model, temperature scaling constant sharpens/flattens the probabilities but usually does not change their relative ranking. The motivation here is to reduce external influences, as comparing calibration performance might not be very meaningful if the predictions/accuracies change significantly.

	Without Scaling-Binning Calibrator					With Scaling-Binning Calibrator			
	MCE	ECE	Brier Score	SCE (d)	SCE (b)	MCE	ECE	SCE (d)	SCE (b)
In Domain									
Baseline	24.51	5.80	12.20	**6.28**	12.18	9.11	3.78	1.71	5.95
Ensemble	23.96	6.10	**11.83**	7.81	12.03	11.81	**2.94**	4.36	7.46
TempScale	23.49	**4.39**	11.87	7.31	**10.75**	**8.81**	3.91	**0.93**	**5.41**
Ours	17.19	5.66	12.19	8.18	12.28	12.94	3.24	4.39	7.51
Ours ($\hat{T}$)	**16.21**	4.93	12.09	8.58	11.91	10.78	3.43	4.66	8.11
Out of Domain									
Baseline	29.66	19.30	29.00	20.06	23.92	30.16	17.83	19.44	21.40
Ensemble	30.71	16.61	27.60	18.95	22.95	**23.77**	**14.39**	**13.45**	**17.17**
TempScale	**26.45**	16.35	27.53	18.71	22.35	33.60	17.55	18.61	20.65
Ours	28.26	17.17	28.08	17.63	22.15	25.11	14.50	15.55	17.92
Ours ($\hat{T}$)	29.79	**15.52**	**27.21**	**17.20**	**21.28**	28.95	14.62	15.97	18.82

Table 1: In-domain and out-of-domain experiment results averaged across tasks. **SCE(d)/SCE(b)**: Squared Calibration Errors (debiased/biased). Lower scores indicate better calibration. Bold/underscored numbers are the best/second-best among comparisons, respectively.

evaluated on QNLI. We can observe that, by varying the teacher model's calibration performance, the calibration performance of the student model also changes in similar directions.

Next, Fig. 1(right) depicts the calibration performances of each teacher-student pair across multiple calibration metrics. Similarly, these figures indicate that correlation of calibration performance between teacher/student models are in general positive. This confirms the intuition described in Sec. 4.2 that calibration performance of the teacher model could be distilled into the student model.

6.2 Main Experiments

Next, we show our experimental results comparing the following four models: Baseline (**Baseline**, Sec. 4.1), Ensemble (Lakshminarayanan et al., 2017) (**Ensemble**, Sec. 2), Temperature Scaling (Guo et al., 2017) (**TempScale**, Sec. 2), our method (**Ours**, Sec. 4.2), and its variant with automatic distillation temperature selection (**Ours ($\hat{T}$)**, Sec. 4.4). For each table, we report results with and without running the scaling-binning calibrator following the description in Sec. 3.2. Due to space constraints, we discuss and display the average performances in here (please see Sec. 5).

Baseline Performances. Results are shown in Table 1; here, we can see that the baseline has relatively high calibration errors. Notably, the out-of-domain ECE values are around $18-19$, interpreted as over/under-estimating the probability by about $18-19\%$ in expectation.

Ensemble and Temperature Scaling. Next, we add ensembles/temperature scaling to the baseline. Results in Table 1 show that performances improve in general, especially in the out-of-domain settings: $3/9$ in-domain metrics improve ($2/9$ metrics similar) and $8/9$ out-of-domain metrics improve for ensembles, $6/9$ in-domain metrics improve ($2/9$ metrics similar) and $7/9$ out-of-domain metrics improve ($1/9$ metrics similar) for temperature-scaling. The results are largely consistent with previous observations that temperature-scaling performed better when the data come from in-domain (it outperforms ensembles among $7/9$ metrics and $1/9$ similar in in-domain settings), whereas ensembles are more competitive in out-of-domain settings at the cost of extra computation (it out-performs temperature scaling in $4/9$ metrics in out-of-domain settings while being similar in $3/9$).

Our Methods. Then, we apply our method, which has the same computation at inference time as the baseline. Table 1 showed that performances improve as well despite having no extra inference-time computation cost: $2/9$ metrics improve ($3/9$ metrics similar) in-domain and $9/9$ metrics improve out-of-domain. Applying the automatic temperature selection on top of our method further improves out-of-domain performance in 4 metrics. However, using automatic temperature does not further improve the performance when we additionally apply the scaling-binning calibrator. We hypothesize that this is because temperature values are chosen based on evaluation metrics before applying the scaling-binning calibrator, thus fail to take it into account. Also, comparing our method to ensembles and temperature scaling, our method improves upon temperature scaling in $5/9$ metrics in out-of-domain settings ($1/9$ similar), but outperforms the more expensive ensembles in just $3/9$

	Without Scaling-Binning Calibrator					With Scaling-Binning Calibrator			
	MCE	ECE	Brier Score	SCE (d)	SCE (b)	MCE	ECE	SCE (d)	SCE (b)
				In Domain					
Ours	17.19	5.66	12.19	8.18	12.28	12.94	3.24	4.39	7.51
−**Ensemble**	18.10	5.90	12.20	10.11	13.19	17.09	5.48	6.40	9.33
−**TempScale**	21.70	6.13	12.28	8.33	12.58	10.49	3.79	4.45	8.26
−**Distillation**	13.04	4.51	11.58	4.40	10.09	6.14	2.61	4.30	7.38
				Out of Domain					
Ours	28.26	17.17	28.08	17.63	22.15	25.11	14.50	15.55	17.92
−**Ensemble**	27.25	17.40	27.96	18.75	22.89	32.20	16.70	19.86	21.81
−**TempScale**	29.18	19.89	29.50	21.28	24.89	30.68	17.40	20.76	22.63
−**Distillation**	21.74	15.39	26.71	15.85	20.58	19.89	14.82	13.05	16.85

Table 2: In-domain/out-of-domain ablation results averaged across tasks. **SCE(d)/SCE(b)**: Squared Calibration Errors (debiased/biased). Lower scores indicate better calibration.

metrics ($1/9$ similar). Comparing our method with automatic temperature selection, we can see $8/9$ metrics in out-of-domain settings improves compared to temperature scaling, and $5/9$ compared to ensembles ($1/9$ similar). This shows that our methods are competitive in out-of-domain settings with little extra computation.

6.3 Ablation Experiments

In this section, we (1) ablate our method by removing components to gain insights into how each of the components contribute to the final performance,[9] and (2) measure how well distillation transfers calibration performance.

First, we remove ensembles (or temperature scaling), and include only temperature scaling (or ensembles) and distillation (−**Ensembles** and −**TempScale**, respectively). We can see from the results in Table 2 that removing either of them leads to worse performances in general: $7/9$ in-domain ($2/9$ being similar) and $6/9$ ($2/9$ being similar) out-of-domain for removing ensembles, $4/9$ in-domain ($4/9$ similar) and $9/9$ out-of-domain for removing temperature scaling. This shows that the additional calibration gains from the teacher model can be effectively distilled into the student models.

Next, we compare the models before/after distillation (−**Distillation**).[10] As expected, the teacher model (before distillation) achieved strong performance at the expense of extra inference-time computation. We then study to what extent distillation transfers calibration performance. Let A_t and B_t be two different teacher models (before distillation) with difference in only one of the components (e.g., ensemble or temperature-scaling), and let A_s and B_s be the corresponding student models (after distillation). Then, we compute the *relative* percentage of improvement because of a component from teacher to student model (assuming A is more powerful than B), denoted as ρ_{AB}:

$$\rho_{AB} = \frac{\varepsilon(A_s) - \varepsilon(B_s)}{\varepsilon(A_t) - \varepsilon(B_t)} \times 100, \qquad (10)$$

where $\varepsilon(\cdot)$ denotes the out-of-domain calibration performance. We compute ρ_{AB} for each metric, and use the median of percentages as the summary statistic. We found 40.8% (111.2%) of the improvements from adding ensembles (temperature scaling) as extra components in teacher models are transferred to students models via distillation.[11]

7 Conclusion and Discussion

We presented a study of calibration of finetuned language models in the context of text classification, where models are evaluated on in-domain and out-of-domain data. We showed the effectiveness of a few widely-used calibration methods. We illustrated the intuitive connection between distillation and calibration, and described simple yet competitive calibration methods. We conducted experiments to empirically understand whether distillation can be used to distill calibration performance, and showed that the simple methods we described achieved competitive out-of-domain calibration performances. We further presented ablation studies on the usefulness of components of

[9] For ease of comparison, we only ablate the system without the automatic temperature selection.

[10] The −**Distillation** in Table 2 is the result of combining ensembles and temperature-scaling. In Table 1, we showed that distillation (especially when combined with automatic temperature) could be helpful compared to either ensembles or temperature-scaling alone.

[11] We chose median as it is simple and less affected by outliers. Please see the supplementary materials Sec. C for more details.

the proposed method and examined the transferability of calibration via distillation. However, our method is limited in that it requires an overhead cost involved in training the student model, which could be expensive in some settings. We leave it to future works to investigate more efficient inference-time recalibration techniques.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by DARPA YFA17-D17AP00022, ONR Grant N00014-18-1-2871, and Microsoft PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

Eneko Agirre, Llu'is M'arquez, and Richard Wicentowski, editors. 2007. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Association for Computational Linguistics, Prague, Czech Republic.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *NeurIPS*.

Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. 2020. Optimizing ai for teamwork. *arXiv:2004.13102*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *RTE*.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*.

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc Le. 2019. Bam! born-again multi-task networks for natural language understanding. In *ACL*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP*.

Pedro Domingos. 1997. Knowledge acquisition from examples via multiple models. In *ICML*.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. In *EMNLP*.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *ICML*.

Yarin Gal. 2016. *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *RTE*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *ICML*.

Corina Gurau, Alex Bewley, and Ingmar Posner. 2018. Dropout distillation for efficiently estimating model confidence. *arXiv:1809.10562*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *ACL*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *UAI*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Volodymyr Kuleshov and Stefano Ermon. 2017. Estimating uncertainty online against an adversary. In *AAAI*.

Volodymyr Kuleshov and Percy S Liang. 2015. Calibrated structured prediction. In *NeurIPS*.

Ananya Kumar, Percy S Liang, and Tengyu Ma. 2019. Verified uncertainty calibration. In *NeurIPS*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *NeurIPS*.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *ICML*.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshop*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*.

Yada Pruksachatkun, Phil Yeres, Haokun Liu, Jason Phang, Phu Mon Htut, Alex Wang, Ian Tenney, and Samuel R Bowman. 2020. jiant: A software toolkit for research on general-purpose text understanding models. In *Proceedings of ACL (demonstration track)*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Siddharth Reddy, Anca D Dragan, Sergey Levine, Shane Legg, and Jan Leike. 2019. Learning human objectives by evaluating hypothetical behavior. *arXiv:1912.05652*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *KDD*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *EMC2*.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *ICML*.

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. 2020. Understanding and improving knowledge distillation. *arXiv:2002.03532*.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. 2017. Distral: Robust multitask reinforcement learning. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv:1811.10959*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *TACL*.

Bryan Wilder, Eric Horvitz, and Ece Kamar. 2020. Learning to complement humans. *ACL*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisit knowledge distillation: a teacher-free framework. In *CVPR*.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *CVPR*.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*.

Xinchuan Zeng and Tony R. Martinez. 2000. Using a neural network to approximate an ensemble of classifiers. *Neural Processing Letters*.

A Background and Related Works

A.1 Epistemic and Aleatoric Uncertainty

Two types of uncertainty commonly appear in machine learning literature: epistemic uncertainty and aleatoric uncertainty (Gal, 2016; Kendall and Gal, 2017). **Epistemic uncertainty** accounts for uncertainty in model parameters, and tends to decrease as the amount of observed data increases. **Aleatoric uncertainty** conveys the noise inherent in the observations, and thus cannot be explained away with an increasing amount of data available. In the case of classification, examples of aleatoric uncertainty include the probability of the top class,[12] and the entropy of the probability distribution over classes (Kendall et al., 2018); examples of epistemic uncertainties include the mutual information.[13] In the literature of uncertainty calibration, we usually calibrate aleatoric uncertainty measured by the probability of the prediction. In Sec. 4.2, we also view distillation from the angle of matching another uncertainty between teacher model and student model, the predictive entropy (Gal, 2016).

A.2 Uncertainty Calibration

The quality of the uncertainty measurement is usually measured via calibration (Kendall and Gal, 2017). In the context of calibration, the uncertainties often refer to predictive probabilities. The model is calibrated if the predictive probabilities match the empirical frequency of the data (Gal, 2016). Let $\hat{Y}$ and $\hat{P}$ be the predicted class and its associated confidence (probability of correctness) of a neural network. We would like the confidence estimates $\hat{P}$ to be calibrated, which intuitively means that we want $\hat{P}$ to represent true probabilities (Guo et al., 2017):

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = p) = p, \forall p \in [0, 1]. \tag{11}$$

Suppose a classification model is given N input examples, and made predictions $\hat{y}_1, ..., \hat{y}_N$, each with $\hat{p} = 0.35$. We would expect 35% of the predictions would be correct. The problem of uncertainty/confidence calibration and confidence scores have been studied and applied in various settings (Kuleshov and Liang, 2015; Kuleshov and Ermon, 2017; Pereyra et al., 2017; Hendrycks and Gimpel, 2017; Elsahar and Gallé, 2019; Reddy et al., 2019). In practice, however, perfect calibration is almost impossible (Guo et al., 2017), and estimating the first term in Eq. 11 is not straightforward using finite samples, because in most cases $\hat{P}$ is a continuous random variable (Guo et al., 2017; Kumar et al., 2019). In Sec. 3, we describe ways to estimate the calibration performance.

It has been widely observed that modern neural networks are usually not calibrated out of the box (Platt, 1999; Zadrozny and Elkan, 2001; Guo et al., 2017; Ovadia et al., 2019). Recalibration methods improve calibration by transforming un-calibrated outputs into calibrated outputs/probabilities, and they include scaling-based methods (Platt, 1999; Guo et al., 2017), histogram-binning-based methods (Guo et al., 2017; Zadrozny and Elkan, 2001), and ensembles (Lakshminarayanan et al., 2017). Recently, Kumar et al. (2019) proposed the scaling-binning calibrator and a more sample-efficient estimator of calibration error. In our work, we describe simple approaches that combines the strength of ensembles and temperature scaling without introducing computation at inference time; we further apply the scaling-binning calibrator to ensure calibration.

Ensembles work by aggregating multiple networks trained independently on the entire dataset, and has been shown to achieve strong performance in out-of-domain calibration (Ovadia et al., 2019; Lakshminarayanan et al., 2017).[14] Temperature

[12] More specifically, it is one minus the probability/confidence of the top class.

[13] Please see page 54 in Gal (2016) for details.

[14] More generally, there are randomization-based ensembles and boosting-based ensembles. Within the randomization-based ensembles, in our work we use the entire training dataset to train each model instead of different bootstrap samples of the original training set (Lakshminarayanan et al., 2017).

scaling is an extension of Platt scaling (Guo et al., 2017). It uses a single scalar parameter $T > 0$ for all classes. Given output z_i (usually logits vectors), the confidence prediction is:

$$\hat{p}_i = \max_k \sigma(z_{i,k}/T). \tag{12}$$

An extension, called heteroscedastic regression, is used in our work, which replaces the constant scalar with learned values (Kendall and Gal, 2017; Kendall et al., 2018).

A.3 Distillation

Knowledge distillation (Hinton et al., 2015; Domingos, 1997; Blum and Mitchell, 1998; Zeng and Martinez, 2000; Ba and Caruana, 2014) is a compression technique in which a compact model (usually referred to as the student model) is trained to mimic the behavior of a more powerful teacher model. In the context of classification, knowledge distillation works by augmenting the loss function with an additional term $D_{KL}(p_i \| p_j)$ where $p_i = \text{softmax}(z_i/T)$ and $p_j = \text{softmax}(z_j/T)$ with z_i and z_j the logits from two models, and T controls the smoothness of the output distribution. Knowledge distillation has been used in a wide range of applications (Buciluă et al., 2006; Wang et al., 2018; Kim and Rush, 2016; Furlanello et al., 2018; Clark et al., 2019; Teh et al., 2017; Schwarz et al., 2018; Sanh et al., 2019). In this work, we show that distillation can also be used to distill calibration performance, and use it to build simple yet competitive recalibration methods.

A related area of research is label smoothing (Yuan et al., 2020). Label smoothing replaces the hard/one-hot targets y_k with modified targets $y_k(1-\alpha)+\alpha/K$, where K is the number of classes and α is a hyper-parameter. Pereyra et al. (2017) showed that label smoothing provides consistent gains across many tasks and proposed a new regularizer, termed confidence penalty. Müller et al. (2019) studied when label smoothing is helpful , and found that label smoothing can implicitly calibrate model's predictions. Instead, our use of a teacher model can be seen as adaptively deciding how much smoothing is needed (Tang et al., 2020).

A.4 Recent Related Works

Finally, there are also a few recent related works in the computer vision literature, e.g., Yun et al. (2020) proposed to distill the predictive distribution between different samples of the same label during training to improve calibration performance, Gurau et al. (2018) proposed Distilled Dropout Network which distills knowledge from multiple MC samples from the teacher to improve the reliability of its uncertainty scores. In our work, we mainly focus on language tasks. Concurrent to our work, Desai and Durrett (2020) studied the calibration of pretrained transformers when finetuned to downstream tasks, and Hendrycks et al. (2020) studied the out-of-distribution robustness of pretrained transformers. We are different from these two works in that first we present a systematic study on the out-of-distribution calibration; second we draw insights from the connection between distillation and temperature scaling to design simple yet competitive recalibration methods; third, we conduct experiments to understand the connection between these two concepts empirically; finally, we also include a more comprehensive set of calibration evaluations following Ovadia et al. (2019) and Kumar et al. (2019).

B Setup Details

Model Details and Hyperparameter Search. Our codebase is largely based on the `Transformers` library from HuggingFace (Wolf et al., 2019).[15] We used RoBERTa-base (Liu et al., 2019) for the language model backbone and used most of the default/recommended hyperparameters in the `Transformers` library. We tried two values of the learning rate in our initial experiments: $2e{-}5$ and $1e{-}5$; these numbers are chosen based on the hyperparameter search described in the library, and we stick to one of them ($1e{-}5$) based on accuracy. For experiments that involve ensembles, we use an ensemble size 2. For experiments that involve distillation, we set $T = 1.0$ (i.e., no scaling) for models without automatic temperature selection unless we explicitly mention otherwise. When automatic temperature selection is used, we chose $\hat{T}$ based on the Brier Scores on the validation dataset. All of our experiments ran on a single V100 GPU. The baseline model has 125.2M parameters, the temperature-scaling (heteroscedastic variant) has 125.8M parameters, and our method has 125.2M (same as the baseline model). We train multiple models using different random seeds before ensembling them, but otherwise run the training

[15] `https://github.com/huggingface/transformers`. We used `v2.4.1`.

and inference once. The runtime varies among tasks, but most of them could finish within a day.

Data. We perform experiments on the classification tasks from the GLUE Benchmark (Wang et al., 2019; Warstadt et al., 2019; Dolan and Brockett, 2005; Agirre et al., 2007; Williams et al., 2018; Rajpurkar et al., 2016; Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009; Levesque et al., 2011).[16][17], and we refer readers to Wang et al. (2019) regarding dataset statistics. Because the calculation of calibration errors requires access to the ground truth data, which is not available for GLUE data, we split the validation dataset into two halves, one for validation and the other for test, following the approach of Desai and Durrett (2020). For MultiNLI, we merge the results for both MultiNLI matched and mismatched sections. When computing the out-of-domain performance between the 3-label MultiNLI and other 2-label NLI tasks, we follow the approach used in the `jiant` library (Pruksachatkun et al., 2020) and merge the predictions/labels that correspond to "neutral" and "contradiction" into a single category. We follow the `Transformers` library for the rest of the data preprocessing.

Evaluation Details. Our evaluation follows Guo et al. (2017), Ovadia et al. (2019), and Kumar et al. (2019). For MCE, ECE, and Brier Score, our implementation follows Ovadia et al. (2019),[18] and we use the default bin-size of 10 (in the tables below, we additionally include the performances when evaluating using bin-sizes of 15 and 50). For squared calibration errors (debiased or biased), we use the `uncertainty-calibration` library from Kumar et al. (2019) and follow default configurations whenever possible.[19]

For in-domain evaluations, the train data and evaluation data come from the same task. For out-of-domain evaluations, the train data and evaluation data come from different tasks of the same type. We group MRPC and QQP (paraphrase tasks), and group MNLI (2-label version),[20] QNLI, RTE, and WNLI (NLI tasks). We leave SST-2 (sentiment), CoLA[21] (acceptability), and MNLI (3-label version, NLI) as separate groups. We use the in-domain validation data to train the scaling-binning calibrator.

Analysis Experiments Details. We conduct experiments on RTE, in which we distill teacher models with different ensemble-sizes (from 1 to 6) and the temperature scaling constant (from 0.50 to 2.00 with a step size of 0.02) to student models. Each model is then evaluated on both in-domain task (RTE) and out-of-domain tasks (MNLI-2, QNLI, WNLI) using confidence, ECE, MCE, Brier Scores, SCE (debiased) and SCE (biased). The numbers represent performances on the validation dataset.

C Further Experiment Details

Distillation Transferability of Calibration. We compute ρ_{AB} based on both Table 1 and Table 2 of the main paper. The percentage of improvement presented in Sec. 6.3 of the main paper on ensembles is computed based on temperature scaling + ensembles (−**Distillation** in main paper Table 2) as A_t, ensembles only (**Ensemble** in main paper Table 1) as B_t, temperature scaling + ensembles + distillation (**Ours** in main paper Table 2) as A_s, and ensembles + distillation (−**TempScale** in main paper Table 2) as B_s. The percentage of improvement on temperature scaling is computed similarly with temperature scaling component as the main difference between the teacher/student models.

D Expanded Analysis Experiments

Please see Fig. 2 for the expanded visualization of the analysis experiments.

E Detailed Main Experiment Results

Please see Table 3 and Table 4 for detailed in-domain and out-of-domain experiment results.

F Detailed Ablation Experiment Results

Please see Table 5 and Table 6 for detailed in-domain and out-of-domain ablation experiment results.

[16] https://gluebenchmark.com/

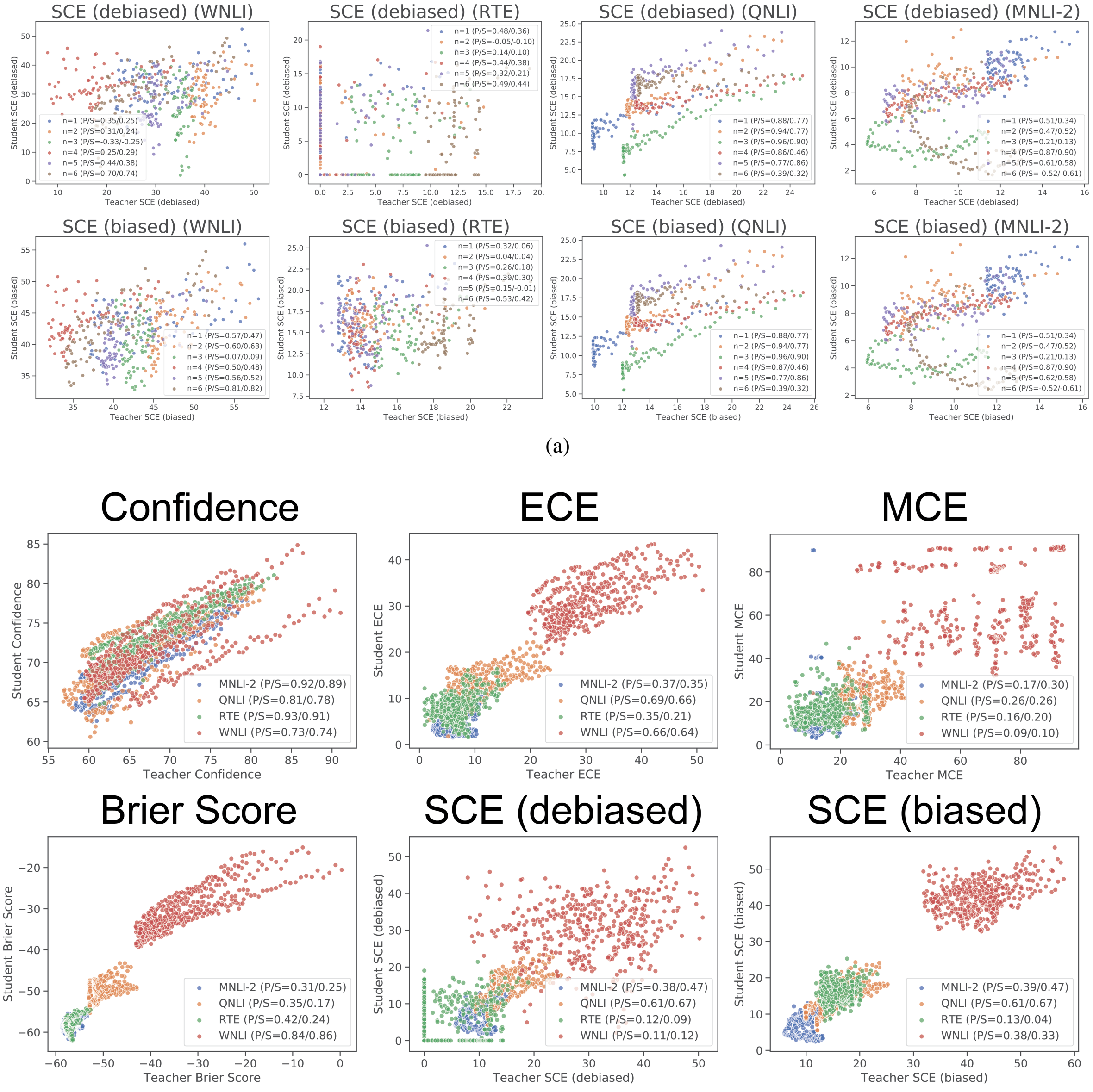
[17] QQP dataset: https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[18] https://github.com/google-research/google-research/tree/master/uq_benchmark_2019

[19] https://github.com/p-lambda/verified_calibration

[20] We only use the 2-label version of MNLI for evaluation.

[21] We use accuracy for CoLA evaluation so that calibration error computations would be more consistent across tasks.



(a)

(b)

Figure 2: **Figure (a):** Visualization of calibration performance, measured by SCEs (debiased and biased), between teacher models and student models, trained on RTE evaluated on RTE (in-domain), WNLI, QNLI, and 2-label version of MNLI (out-of-domain). The n in the legend refers to the size of ensemble(s). **Figure (b):** This is a zoomed-out version of Figure (a). Instead of using color to imply the ensemble size, here the color refers to the task in which the models are evaluated, and points of different ensemble sizes but the same evaluation task are aggregated and represented by the same color. Here each sub-figure represents the evaluation metric. **All Figures:** The X-axis refers to the performance of the teacher model, and the Y-axis refers to the performance of the student model. Within each sub-figure, each dot represents a different configuration used in the teacher model. The P/S in the legends refer to the Pearson/Spearman correlations.

Train	SST-2	CoLA	MNLI	MRPC	QQP	QNLI	RTE	WNLI	Average	SST-2	CoLA	MNLI	MRPC	QQP	QNLI	RTE	WNLI	Average
	Accuracy									**Accuracy (+SBC)**								
Baseline	93.1	80.3	87.2	86.8	91.0	92.3	64.7	55.6	81.38	/	/	/	/	/	/	/	/	/
Ensemble	93.8	79.9	87.4	85.8	91.2	92.4	66.2	55.6	81.54	/	/	/	/	/	/	/	/	/
TempScale	93.1	80.3	87.2	86.8	91.0	92.3	64.7	55.6	81.38	/	/	/	/	/	/	/	/	/
Ours	93.3	79.7	87.4	84.8	91.0	92.2	65.5	55.6	81.19	/	/	/	/	/	/	/	/	/
Ours ($\hat{T}$)	93.6	79.7	87.3	86.3	91.0	92.4	64.0	55.6	81.24	/	/	/	/	/	/	/	/	/
	Maximum Calibration Error									**Maximum Calibration Error (+SBC)**								
Baseline	49.7	32.1	38.6	19.6	12.9	21.5	17.6	4.1	24.51	1.6	7.9	4.1	5.1	2.6	6.3	14.3	31.0	9.11
Ensemble	74.0	29.2	16.3	21.5	12.6	14.6	19.0	4.5	23.96	6.6	9.5	4.9	8.7	3.2	3.9	17.9	39.8	11.81
TempScale	33.0	12.3	24.2	16.7	3.3	13.2	80.5	4.7	23.49	1.0	11.8	3.3	5.0	2.5	4.2	12.8	29.9	8.81
Ours	34.0	20.8	14.3	16.6	9.3	14.0	24.0	4.5	17.19	2.8	13.5	3.8	22.8	2.9	3.8	17.2	36.7	12.94
Ours ($\hat{T}$)	44.3	20.4	11.9	11.7	7.3	8.9	21.1	4.1	16.21	1.2	10.5	3.8	15.4	1.9	3.9	12.7	36.8	10.78
	Expected Calibration Error (bin size = 10**)**									**Expected Calibration Error (bin size =** 10**) (+SBC)**								
Baseline	5.5	11.5	6.3	5.2	4.2	4.1	5.5	4.1	5.80	1.0	3.3	0.8	1.5	0.4	1.2	7.3	14.7	3.78
Ensemble	4.6	11.9	5.5	4.2	3.4	3.3	11.4	4.5	6.10	1.6	4.0	0.9	2.3	0.6	0.9	7.5	5.7	2.94
TempScale	3.4	8.2	3.9	4.2	1.7	1.9	7.1	4.7	4.39	0.2	4.0	0.8	2.0	0.5	1.2	8.2	14.4	3.91
Ours	4.6	11.6	5.2	4.1	3.0	3.5	8.8	4.5	5.66	0.6	4.0	1.1	3.3	0.5	0.8	7.7	7.9	3.24
Ours ($\hat{T}$)	3.6	9.3	4.1	4.0	2.4	2.2	9.7	4.1	4.93	0.2	5.5	0.8	4.5	0.4	0.7	9.6	5.7	3.43
	Expected Calibration Error (bin size = 15**)**									**Expected Calibration Error (bin size =** 15**) (+SBC)**								
Baseline	5.5	11.6	6.3	6.3	4.2	4.1	5.5	4.1	5.95	1.0	4.6	1.1	1.7	0.4	1.5	6.8	16.3	4.18
Ensemble	4.7	11.9	5.5	4.4	3.4	3.3	8.5	4.5	5.78	1.6	5.4	1.0	3.1	0.6	1.0	10.6	5.7	3.63
TempScale	3.6	8.6	3.9	5.4	1.7	1.9	9.0	4.7	4.85	0.2	4.6	0.8	2.4	0.7	1.5	7.4	16.0	4.20
Ours	4.8	12.1	5.2	5.2	3.1	3.5	9.4	4.5	5.98	0.6	4.0	1.4	4.4	0.5	1.3	9.4	7.9	3.69
Ours ($\hat{T}$)	3.7	9.4	4.1	5.1	2.4	2.2	9.7	4.1	5.09	0.3	4.9	1.1	4.5	0.4	1.4	8.5	5.7	3.35
	Expected Calibration Error (bin size = 50**)**									**Expected Calibration Error (bin size =** 50**) (+SBC)**								
Baseline	6.2	12.6	6.4	7.9	4.2	4.3	11.2	4.1	7.11	2.1	3.9	1.1	2.2	0.5	1.5	9.9	19.7	5.11
Ensemble	5.2	12.6	5.6	6.9	3.5	3.6	16.8	4.5	7.34	3.3	5.2	1.5	3.8	0.6	1.2	11.9	16.2	5.46
TempScale	5.0	9.7	4.1	9.5	1.8	2.6	11.3	4.7	6.09	0.7	5.0	1.2	3.3	0.7	1.5	8.3	19.3	5.00
Ours	5.7	12.9	5.2	11.6	3.1	3.7	15.5	4.5	7.78	1.8	5.6	1.6	5.2	0.6	1.4	11.9	13.0	5.14
Ours ($\hat{T}$)	4.4	10.7	4.4	10.0	2.4	2.6	16.6	4.1	6.90	1.7	6.0	1.5	5.5	0.4	1.5	11.1	8.2	4.49
	Brier Score									**Brier Score (+SBC)**								
Baseline	5.80	15.25	6.63	10.35	7.00	6.20	21.45	24.90	12.20	/	/	/	/	/	/	/	/	/
Ensemble	5.10	15.20	6.43	10.05	6.70	5.85	20.35	24.95	11.83	/	/	/	/	/	/	/	/	/
TempScale	5.15	14.35	6.37	10.35	6.65	5.85	21.35	24.90	11.87	/	/	/	/	/	/	/	/	/
Ours	5.50	15.85	6.43	10.45	6.75	6.15	21.45	24.90	12.19	/	/	/	/	/	/	/	/	/
Ours ($\hat{T}$)	5.20	15.35	6.37	10.60	6.70	5.95	21.65	24.90	12.09	/	/	/	/	/	/	/	/	/
	Squared Calibration Error (debiased)									**Squared Calibration Error (debiased, +SBC)**								
Baseline	9.4	12.0	8.2	3.6	6.0	5.5	5.5	0.0	6.28	1.3	1.5	1.3	0.0	0.7	1.8	7.1	0.0	1.71
Ensemble	5.7	13.7	7.2	0.0	5.0	4.7	9.7	16.5	7.81	2.6	4.1	1.7	0.0	1.0	0.7	8.9	15.9	4.36
TempScale	5.1	8.4	5.0	7.4	2.0	1.9	0.0	28.7	7.31	0.0	3.9	1.2	0.0	0.9	1.4	0.0	0.0	0.93
Ours	7.3	13.0	6.8	0.0	4.2	4.7	13.0	16.4	8.18	0.0	4.8	1.9	5.3	0.9	1.1	8.4	12.7	4.39
Ours ($\hat{T}$)	4.8	11.7	5.6	4.2	3.3	3.1	12.6	23.3	8.58	0.0	5.5	1.5	1.1	0.6	1.4	14.4	12.8	4.66
	Squared Calibration Error (biased)									**Squared Calibration Error (biased, +SBC)**								
Baseline	10.2	13.5	8.3	9.2	6.1	5.8	15.6	28.7	12.18	3.3	5.2	1.6	3.1	0.9	2.3	13.9	17.3	5.95
Ensemble	7.0	15.0	7.3	8.1	5.0	5.0	17.0	31.8	12.03	4.0	6.3	2.0	5.2	1.1	1.6	14.4	25.1	7.46
TempScale	6.5	10.5	5.2	11.2	2.2	2.6	10.6	37.2	10.75	1.6	6.3	1.5	4.2	1.0	2.0	11.6	15.1	5.41
Ours	8.3	14.5	6.9	8.3	4.2	5.0	19.2	31.8	12.28	2.2	6.9	2.1	8.5	1.1	1.8	14.6	22.9	7.51
Ours ($\hat{T}$)	6.3	13.3	5.7	9.6	3.4	3.5	19.0	34.5	11.91	2.1	7.4	1.8	7.2	0.8	2.0	18.8	24.8	8.11

Table 3: In-domain performances on SST-2 (S), CoLA (C), MultiNLI (MN), MRPC (MR), QQP (QQ), QNLI (QN), RTE (R), WNLI (W). Note that for the metrics we considered here, lower scores indicate better calibration.

Train	MR	QQ	QN			R			W			Average	MR	QQ	QN			R			W			Average
Eval	QQ	MR	M2	R	W	M2	QN	W	M2	QN	R		QQ	MR	M2	R	W	M2	QN	W	M2	QN	R	
	Accuracy												**Accuracy (+SBC)**											
Baseline	68.3	68.1	55.2	56.8	38.9	67.6	55.4	44.4	35.1	50.1	48.2	53.46	/	/	/	/	/	/	/	/	/	/	/	/
Ensemble	67.9	65.2	58.5	54.7	47.2	67.7	57.7	44.4	35.1	50.1	48.2	54.25	/	/	/	/	/	/	/	/	/	/	/	/
TempScale	68.3	68.1	55.2	56.8	38.9	67.6	55.4	44.4	35.1	50.1	48.2	53.46	/	/	/	/	/	/	/	/	/	/	/	/
Ours	68.4	68.1	58.6	57.6	47.2	67.3	61.0	44.4	35.1	50.1	48.2	55.09	/	/	/	/	/	/	/	/	/	/	/	/
Ours ($\hat{T}$)	67.2	71.1	61.1	59.7	38.9	66.2	61.2	44.4	35.1	50.1	48.2	54.84	/	/	/	/	/	/	/	/	/	/	/	/
	Maximum Calibration Error												**Maximum Calibration Error (+SBC)**											
Baseline	35.8	24.3	37.3	34.6	68.4	18.5	19.7	67.2	16.2	1.2	3.1	29.66	33.3	19.7	33.2	32.3	58.7	7.4	19.8	31.4	37.8	26.5	31.7	30.16
Ensemble	36.9	46.7	35.6	34.7	94.7	9.6	16.9	42.7	16.0	1.1	2.9	30.71	34.7	17.3	33.9	38.8	33.7	6.5	20.5	34.4	21.6	11.2	8.9	23.77
TempScale	34.5	35.0	35.0	34.6	61.1	9.9	20.0	41.8	15.7	0.8	2.6	26.45	32.5	26.1	34.2	33.7	55.8	8.4	22.2	65.9	35.8	25.9	29.1	33.60
Ours	35.7	33.5	33.2	31.2	70.9	11.3	11.4	64.1	15.9	0.9	2.8	28.26	33.9	22.8	27.4	32.1	52.7	4.3	13.7	30.1	26.9	15.0	17.3	25.11
Ours ($\hat{T}$)	39.2	22.7	30.3	30.7	94.0	11.2	12.6	66.2	16.3	1.3	3.2	29.79	37.5	19.8	26.4	40.5	91.0	4.1	8.2	35.6	26.0	15.7	13.6	28.95
	Expected Calibration Error (bin size = 10**)**												**Expected Calibration Error (bin size =** 10**) (+SBC)**											
Baseline	19.5	22.9	26.4	29.9	40.0	9.5	8.4	35.2	16.2	1.2	3.1	19.30	18.2	13.7	14.7	17.1	25.4	5.8	9.1	30.6	27.2	16.3	18.0	17.83
Ensemble	20.6	22.9	21.7	28.4	22.9	3.5	10.6	32.1	16.0	1.1	2.9	16.61	21.3	15.2	12.2	18.2	12.9	5.3	9.3	28.5	21.3	5.4	8.7	14.39
TempScale	20.4	17.6	21.6	24.6	34.3	6.9	5.5	29.8	15.7	0.8	2.6	16.35	14.8	12.9	16.3	18.0	26.4	6.5	8.5	31.6	26.2	15.6	16.3	17.55
Ours	18.3	19.3	21.2	24.5	38.7	3.7	9.0	34.6	15.9	0.9	2.8	17.17	15.8	12.1	12.8	15.5	22.4	3.7	5.2	27.6	23.7	9.1	11.6	14.50
Ours ($\hat{T}$)	20.4	17.3	15.5	19.4	32.1	4.8	7.5	32.9	16.3	1.3	3.2	15.52	20.4	12.3	9.9	11.4	27.6	2.2	7.6	26.2	23.4	8.6	11.2	14.62
	Expected Calibration Error (bin size = 15**)**												**Expected Calibration Error (bin size =** 15**) (+SBC)**											
Baseline	19.5	22.9	26.4	31.5	40.0	10.6	8.5	35.2	16.2	1.2	3.1	19.55	18.2	13.7	14.7	17.1	25.4	8.6	9.0	30.6	27.2	16.7	18.0	18.11
Ensemble	21.0	24.3	21.7	28.4	22.9	4.9	10.8	32.1	16.0	1.1	2.9	16.92	21.3	15.2	12.2	18.2	12.9	3.4	8.6	28.5	21.3	5.4	8.7	14.15
TempScale	20.8	18.5	21.6	24.9	34.3	8.5	5.4	29.8	15.7	0.8	2.6	16.63	17.7	12.9	16.3	18.0	26.4	7.9	8.5	31.6	26.2	15.6	16.4	17.95
Ours	18.3	19.9	21.3	24.5	38.7	4.6	9.1	34.6	15.9	0.9	2.8	17.33	15.8	12.1	12.8	15.5	22.4	2.1	5.2	27.6	23.7	9.1	13.0	14.48
Ours ($\hat{T}$)	20.5	17.5	15.5	18.7	34.7	5.3	7.5	32.9	16.3	1.3	3.2	15.76	20.4	12.3	9.9	11.4	27.6	3.0	7.0	26.2	23.4	8.6	11.2	14.64
	Expected Calibration Error (bin size = 50**)**												**Expected Calibration Error (bin size =** 50**) (+SBC)**											
Baseline	19.8	25.7	26.4	32.4	45.4	10.7	8.5	35.2	16.2	1.2	3.1	20.42	18.2	13.7	14.7	17.1	25.4	9.0	9.1	32.4	27.2	16.7	18.4	18.35
Ensemble	21.4	24.9	21.7	31.9	41.3	5.4	11.0	32.1	16.0	1.1	2.9	19.06	21.3	15.2	12.2	18.2	12.9	5.4	9.5	28.5	21.3	5.4	8.7	14.42
TempScale	21.1	19.6	21.6	26.1	44.2	9.6	6.8	31.6	15.7	0.8	2.6	18.15	17.7	12.9	16.3	18.0	26.4	8.1	8.7	31.6	26.2	15.6	17.3	18.07
Ours	18.5	21.2	21.4	27.4	43.5	5.2	9.1	38.4	15.9	0.9	2.8	18.57	15.8	12.1	12.8	15.5	22.4	4.3	5.5	32.4	23.7	9.1	13.0	15.15
Ours ($\hat{T}$)	21.0	21.5	15.6	23.6	45.3	5.8	7.5	38.5	16.3	1.3	3.2	18.15	20.4	12.3	9.9	11.4	27.6	3.0	7.6	28.4	23.4	8.6	12.6	15.02
	Brier Score												**Brier Score (+SBC)**											
Baseline	26.20	26.50	33.00	33.00	40.65	22.40	24.70	37.05	25.45	25.00	25.05	29.00	/	/	/	/	/	/	/	/	/	/	/	/
Ensemble	26.85	26.40	30.55	33.35	29.80	21.60	24.45	35.25	25.35	25.00	25.05	27.60	/	/	/	/	/	/	/	/	/	/	/	/
TempScale	26.15	23.95	30.65	30.65	36.00	22.35	24.45	33.35	25.25	25.00	25.05	27.53	/	/	/	/	/	/	/	/	/	/	/	/
Ours	26.20	24.45	29.90	31.05	39.40	21.95	23.85	36.75	25.30	25.00	25.05	28.08	/	/	/	/	/	/	/	/	/	/	/	/
Ours ($\hat{T}$)	27.55	23.80	27.00	28.90	34.70	22.50	23.45	35.95	25.45	25.00	25.05	27.21	/	/	/	/	/	/	/	/	/	/	/	/
	Squared Calibration Error (debiased)												**Squared Calibration Error (debiased, +SBC)**											
Baseline	24.2	21.5	28.9	27.3	39.1	12.0	10.0	30.5	16.3	1.1	9.8	20.06	22.2	15.2	19.4	17.4	28.9	10.2	11.0	23.9	28.7	18.3	18.6	19.44
Ensemble	25.6	22.5	25.2	28.1	21.2	7.0	14.3	43.0	16.3	1.4	3.9	18.95	24.3	13.0	18.5	20.5	0.0	6.2	13.2	15.9	21.4	6.5	8.5	13.45
TempScale	24.3	17.7	24.5	23.4	33.6	10.7	8.9	37.3	15.8	0.0	9.6	18.71	21.5	14.3	19.8	18.7	28.6	9.2	10.3	22.6	27.5	17.6	14.6	18.61
Ours	23.0	18.5	24.1	26.4	33.8	6.9	9.4	34.0	15.9	1.9	0.0	17.63	21.1	11.6	17.0	16.5	25.5	4.9	6.8	29.1	23.9	10.1	4.5	15.55
Ours ($\hat{T}$)	26.9	16.6	18.6	21.3	39.0	7.7	7.9	32.8	16.3	2.1	0.0	17.20	26.1	12.6	14.5	14.2	22.4	5.9	10.3	27.5	23.6	9.6	9.0	15.97
	Squared Calibration Error (biased)												**Squared Calibration Error (biased, +SBC)**											
Baseline	24.2	25.0	29.0	31.6	48.6	12.2	10.7	43.1	16.4	3.9	18.4	23.92	22.3	17.2	19.5	20.5	31.0	10.3	11.3	33.4	28.7	18.6	22.6	21.40
Ensemble	25.7	25.7	25.2	32.4	35.8	7.2	14.8	49.0	16.4	3.9	16.3	22.95	24.3	15.6	18.6	23.1	13.8	6.4	13.4	29.9	21.4	7.1	15.3	17.17
TempScale	24.3	21.5	24.6	28.3	44.3	10.8	9.6	45.2	15.9	3.0	18.3	22.35	21.5	16.4	19.8	21.4	30.8	9.3	10.6	32.4	27.5	17.8	19.6	20.65
Ours	23.1	22.3	24.2	30.6	45.5	7.2	10.1	44.6	16.0	4.2	15.9	22.15	21.1	14.3	17.0	19.2	29.3	5.1	7.4	35.1	23.9	10.5	14.2	17.92
Ours ($\hat{T}$)	26.9	20.8	18.7	26.5	46.6	7.9	8.6	43.7	16.4	4.3	13.7	21.28	26.2	15.2	14.6	19.3	29.6	6.1	10.6	36.1	23.6	10.0	15.7	18.82

Table 4: Out-of-domain performances on MRPC (MR), QQP (QQ), QNLI (QN), RTE (R), WNLI (W). We use M2 to denote the 2-label version of MultiNLI task. Note that for the metrics we considered here, lower scores indicate better calibration.

Train	SST-2	CoLA	MNLI	MRPC	QQP	QNLI	RTE	WNLI	Average	SST-2	CoLA	MNLI	MRPC	QQP	QNLI	RTE	WNLI	Average
	Accuracy									**Accuracy (+SBC)**								
Ours	93.3	79.7	87.4	84.8	91.0	92.2	65.5	55.6	81.19	/	/	/	/	/	/	/	/	/
−**Ensemble**	93.3	79.5	87.4	85.3	91.0	92.5	66.2	55.6	81.35	/	/	/	/	/	/	/	/	/
−**TempScale**	93.1	79.1	87.3	86.3	91.1	92.3	65.5	55.6	81.29	/	/	/	/	/	/	/	/	/
−**Distillation**	94.0	79.9	87.4	86.3	91.2	92.3	66.9	55.6	81.70	/	/	/	/	/	/	/	/	/
	Maximum Calibration Error									**Maximum Calibration Error (+SBC)**								
Ours	34.0	20.8	14.3	16.6	9.3	14.0	24.0	4.5	17.19	2.8	13.5	3.8	22.8	2.9	3.8	17.2	36.7	12.94
−**Ensemble**	49.6	25.3	15.5	14.3	8.6	11.3	16.4	3.8	18.10	2.3	11.9	3.5	26.3	2.8	9.6	6.7	73.6	17.09
−**TempScale**	58.7	34.3	16.7	7.6	15.6	16.6	19.8	4.3	21.70	12.5	9.9	3.7	3.8	3.8	4.0	5.5	40.7	10.49
−**Distillation**	28.3	22.8	8.4	12.8	3.1	8.2	15.9	4.8	13.04	4.5	9.7	3.0	9.3	2.1	3.1	15.7	1.7	6.14
	Expected Calibration Error (bin size = 10)									**Expected Calibration Error (bin size = 10) (+SBC)**								
Ours	4.6	11.6	5.2	4.1	3.0	3.5	8.8	4.5	5.66	0.6	4.0	1.1	3.3	0.5	0.8	7.7	7.9	3.24
−**Ensemble**	4.5	11.8	5.3	3.8	3.0	3.3	11.7	3.8	5.90	0.7	4.5	1.0	6.0	0.5	1.2	6.4	23.5	5.48
−**TempScale**	5.4	13.3	6.2	3.6	4.0	3.6	8.6	4.3	6.13	1.8	3.5	0.9	1.5	0.5	0.7	5.5	15.9	3.79
−**Distillation**	3.2	9.3	3.1	2.4	1.3	1.3	10.7	4.8	4.51	1.2	4.3	0.8	1.7	0.6	0.8	9.8	1.7	2.61
	Expected Calibration Error (bin size = 15)									**Expected Calibration Error (bin size = 15) (+SBC)**								
Ours	4.8	12.1	5.2	5.2	3.1	3.5	9.4	4.5	5.98	0.6	4.0	1.4	4.4	0.5	1.3	9.4	7.9	3.69
−**Ensemble**	4.5	11.8	5.3	4.7	3.0	3.3	11.7	3.8	6.01	0.7	3.8	1.4	5.4	0.5	1.6	8.9	23.5	5.73
−**TempScale**	5.5	13.3	6.2	5.6	4.0	3.7	8.9	4.3	6.44	2.7	4.4	1.2	2.1	0.4	1.3	5.5	18.0	4.45
−**Distillation**	3.4	9.4	3.0	4.3	1.3	1.7	12.6	4.8	5.06	1.2	4.4	0.8	2.0	0.6	0.8	12.1	1.7	2.95
	Expected Calibration Error (bin size = 50)									**Expected Calibration Error (bin size = 50) (+SBC)**								
Ours	5.7	12.9	5.2	11.6	3.1	3.7	15.5	4.5	7.78	1.8	5.6	1.6	5.2	0.6	1.4	11.9	13.0	5.14
−**Ensemble**	5.5	12.7	5.5	10.7	3.1	3.5	15.5	3.8	7.54	1.7	5.5	1.7	6.7	0.5	1.6	10.2	27.5	6.93
−**TempScale**	6.3	13.7	6.3	9.7	4.1	4.0	11.9	4.3	7.54	4.3	4.1	1.4	2.6	0.5	1.4	5.5	18.9	4.84
−**Distillation**	4.7	11.6	3.2	7.3	1.4	2.0	17.2	4.8	6.53	2.1	5.2	1.0	2.7	0.6	1.0	12.5	4.8	3.74
	Brier Score									**Brier Score (+SBC)**								
Ours	5.50	15.85	6.43	10.45	6.75	6.15	21.45	24.90	12.19	/	/	/	/	/	/	/	/	/
−**Ensemble**	5.25	15.65	6.47	10.15	6.80	6.05	22.35	24.85	12.20	/	/	/	/	/	/	/	/	/
−**TempScale**	5.75	16.05	6.57	10.75	6.90	6.10	21.25	24.90	12.28	/	/	/	/	/	/	/	/	/
−**Distillation**	4.70	14.40	6.20	10.05	6.45	5.65	20.25	24.95	11.58	/	/	/	/	/	/	/	/	/
	Squared Calibration Error (debiased)									**Squared Calibration Error (debiased, +SBC)**								
Ours	7.3	13.0	6.8	0.0	4.2	4.7	13.0	16.4	8.18	0.0	4.8	1.9	5.3	0.9	1.1	8.4	12.7	4.39
−**Ensemble**	7.4	14.6	7.3	0.0	4.2	4.0	10.3	33.1	10.11	0.0	7.2	1.8	8.2	0.8	2.9	11.7	18.6	6.40
−**TempScale**	9.7	14.8	8.1	0.0	5.6	5.0	0.0	23.4	8.33	4.8	6.2	1.5	0.0	1.1	1.1	10.4	10.5	4.45
−**Distillation**	3.7	12.1	3.8	0.0	1.5	1.8	12.3	0.0	4.40	0.0	3.8	1.1	0.0	0.7	0.0	10.1	18.7	4.30
	Squared Calibration Error (biased)									**Squared Calibration Error (biased, +SBC)**								
Ours	8.3	14.5	6.9	8.3	4.2	5.0	19.2	31.8	12.28	2.2	6.9	2.1	8.5	1.1	1.8	14.6	22.9	7.51
−**Ensemble**	8.4	15.9	7.4	7.9	4.2	4.4	17.8	39.5	13.19	2.5	8.7	2.0	10.5	1.0	3.3	16.9	29.7	9.33
−**TempScale**	10.5	16.1	8.2	7.7	5.7	5.3	12.5	34.6	12.58	5.8	7.9	1.8	3.4	1.2	1.8	16.4	27.8	8.26
−**Distillation**	5.4	13.5	3.9	6.5	1.7	2.5	18.4	28.8	10.09	2.6	6.1	1.4	4.5	0.9	1.3	15.6	26.6	7.38

Table 5: In-domain ablation performances on SST-2 (S), CoLA (C), MultiNLI (MN), MRPC (MR), QQP (QQ), QNLI (QN), RTE (R), WNLI (W). Note that for the metrics we considered here, lower scores indicate better calibration.

Train	MR	QQ	QN			R			W			Average	MR	QQ	QN			R			W			Average
Eval	QQ	MR	M2	R	W	M2	QN	W	M2	QN	R		QQ	MR	M2	R	W	M2	QN	W	M2	QN	R	
	Accuracy												**Accuracy (+SBC)**											
Ours	68.4	68.1	58.6	57.6	47.2	67.3	61.0	44.4	35.1	50.1	48.2	55.09	/	/	/	/	/	/	/	/	/	/	/	/
−Ensemble	66.3	66.2	59.7	59.0	44.4	67.5	55.7	44.4	35.1	50.1	48.2	54.24	/	/	/	/	/	/	/	/	/	/	/	/
−TempScale	66.6	67.2	56.0	56.1	47.2	68.0	54.6	44.4	35.1	50.1	48.2	53.95	/	/	/	/	/	/	/	/	/	/	/	/
−Distillation	67.8	65.2	58.7	54.7	50.0	67.8	57.9	44.4	35.1	50.1	48.2	54.54	/	/	/	/	/	/	/	/	/	/	/	/
	Maximum Calibration Error												**Maximum Calibration Error (+SBC)**											
Ours	35.7	33.5	33.2	31.2	70.9	11.3	11.4	64.1	15.9	0.9	2.8	28.26	33.9	22.8	27.4	32.1	52.7	4.3	13.7	30.1	26.9	15.0	17.3	25.11
−Ensemble	39.3	27.4	32.5	40.7	58.6	8.7	27.1	43.5	16.7	1.7	3.5	27.25	37.3	21.4	28.5	28.6	17.5	15.5	27.4	28.0	56.5	45.6	47.9	32.20
−TempScale	39.1	31.5	37.0	38.3	54.6	11.6	21.1	67.3	16.2	1.2	3.1	29.18	37.7	16.6	31.9	30.5	61.8	3.0	16.4	26.6	38.4	31.0	43.6	30.68
−Distillation	35.8	25.6	33.2	36.7	34.6	4.8	17.4	31.9	15.7	0.8	2.6	21.74	33.5	18.3	30.9	27.6	14.8	4.7	18.4	31.5	22.3	7.7	9.1	19.89
	Expected Calibration Error (bin size = 10**)**												**Expected Calibration Error (bin size =** 10**) (+SBC)**											
Ours	18.3	19.3	21.2	24.5	38.7	3.7	9.0	34.6	15.9	0.9	2.8	17.17	15.8	12.1	12.8	15.5	22.4	3.7	5.2	27.6	23.7	9.1	11.6	14.50
−Ensemble	18.9	20.6	20.9	24.7	31.9	3.3	15.2	34.0	16.7	1.7	3.5	17.40	18.4	13.2	10.6	12.6	17.2	3.6	15.5	28.0	24.9	17.9	21.8	16.70
−TempScale	23.9	22.4	27.5	31.1	40.6	3.4	14.6	34.8	16.2	1.2	3.1	19.89	21.2	12.7	18.9	22.5	26.7	3.0	16.4	26.6	23.3	9.5	10.6	17.40
−Distillation	20.1	18.4	18.0	25.4	26.7	3.7	8.8	29.1	15.7	0.8	2.6	15.39	20.6	15.2	13.6	21.0	11.8	3.8	9.3	28.6	22.3	7.7	9.1	14.82
	Expected Calibration Error (bin size = 15**)**												**Expected Calibration Error (bin size =** 15**) (+SBC)**											
Ours	18.3	19.9	21.3	24.5	38.7	4.6	9.1	34.6	15.9	0.9	2.8	17.33	15.8	12.1	12.8	15.5	22.4	2.1	5.2	27.6	23.7	9.1	13.0	14.48
−Ensemble	20.5	21.2	20.9	24.7	31.9	4.4	15.2	35.5	16.7	1.7	3.5	17.84	20.8	13.2	10.6	12.6	17.2	4.4	15.6	28.0	24.9	17.9	21.8	17.00
−TempScale	24.0	22.9	27.5	30.4	39.4	5.6	14.6	34.8	16.2	1.2	3.1	19.97	21.2	12.7	18.9	22.5	26.7	3.0	16.4	26.6	23.3	10.1	13.3	17.70
−Distillation	20.9	18.8	18.0	25.4	28.4	3.7	8.9	29.1	15.7	0.8	2.6	15.66	20.6	15.2	13.6	21.0	11.8	3.1	8.8	28.6	22.3	7.7	9.1	14.71
	Expected Calibration Error (bin size = 50**)**												**Expected Calibration Error (bin size =** 50**) (+SBC)**											
Ours	18.5	21.2	21.4	27.4	43.5	5.2	9.1	38.4	15.9	0.9	2.8	18.57	15.8	12.1	12.8	15.5	22.4	4.3	5.5	32.4	23.7	9.1	13.0	15.15
−Ensemble	21.2	23.3	21.0	27.5	35.2	4.8	15.2	34.0	16.7	1.7	3.5	18.55	20.8	13.8	10.6	12.6	17.2	3.6	16.7	28.0	25.1	17.9	21.8	17.10
−TempScale	24.5	24.2	27.7	34.1	46.2	5.9	14.6	41.0	16.2	1.2	3.1	21.70	21.2	12.7	18.9	22.5	26.8	3.0	16.4	26.6	23.3	10.1	16.5	18.00
−Distillation	21.8	20.0	18.1	29.8	35.6	4.5	9.1	31.6	15.7	0.8	2.6	17.24	20.6	15.2	13.6	21.0	11.8	4.0	9.3	28.6	22.3	7.7	9.1	14.84
	Brier Score												**Brier Score (+SBC)**											
Ours	26.20	24.45	29.90	31.05	39.40	21.95	23.85	36.75	25.30	25.00	25.05	28.08	/	/	/	/	/	/	/	/	/	/	/	/
−Ensemble	27.90	25.70	29.35	30.45	33.75	21.70	26.10	36.95	25.55	25.05	25.10	27.96	/	/	/	/	/	/	/	/	/	/	/	/
−TempScale	28.60	25.25	33.45	34.15	42.00	21.50	26.25	37.70	25.40	25.05	25.10	29.50	/	/	/	/	/	/	/	/	/	/	/	/
−Distillation	26.75	24.25	28.80	31.45	28.25	21.45	24.15	33.40	25.25	25.00	25.05	26.71	/	/	/	/	/	/	/	/	/	/	/	/
	Calibration Error (debiased)												**Calibration Error (debiased, +SBC)**											
Ours	23.0	18.5	24.1	26.4	33.8	6.9	9.4	34.0	15.9	1.9	0.0	17.63	21.1	11.6	17.0	16.5	25.5	4.9	6.8	29.1	23.9	10.1	4.5	15.55
−Ensemble	27.3	21.0	23.4	24.4	28.9	6.9	17.6	30.2	16.7	0.5	9.4	18.75	26.2	15.2	16.2	19.2	12.3	8.1	19.7	24.7	30.5	21.5	24.9	19.86
−TempScale	28.7	20.0	30.0	30.1	40.6	6.8	16.1	35.6	16.2	2.9	7.1	21.28	26.2	11.9	22.0	23.0	32.4	8.9	19.3	29.4	24.8	12.7	17.8	20.76
−Distillation	25.4	16.2	21.5	24.5	17.9	4.9	13.0	23.2	16.0	1.9	9.8	15.85	23.5	13.5	18.2	20.4	0.0	4.7	12.9	20.2	22.5	7.7	0.0	13.05
	Calibration Error (biased)												**Calibration Error (biased, +SBC)**											
Ours	23.1	22.3	24.2	30.6	45.5	7.2	10.1	44.6	16.0	4.2	15.9	22.15	21.1	14.3	17.0	19.2	29.3	5.1	7.4	35.1	23.9	10.5	14.2	17.92
−Ensemble	27.3	24.5	23.5	29.0	40.8	7.1	18.0	42.9	16.8	3.7	18.2	22.89	26.2	17.0	16.3	21.3	17.2	8.2	19.9	33.9	30.5	21.7	27.7	21.81
−TempScale	28.7	23.6	30.0	34.0	49.8	7.0	16.5	45.9	16.3	4.7	17.3	24.89	26.3	14.3	22.1	25.2	36.6	9.0	19.5	36.1	24.8	13.0	22.0	22.63
−Distillation	25.4	20.5	21.6	29.2	34.0	5.2	13.5	38.3	16.1	4.2	18.4	20.58	23.5	15.6	18.3	23.3	11.9	4.9	13.2	31.5	22.6	8.3	12.2	16.85

Table 6: Out-of-domain ablation performances on MRPC (MR), QQP (QQ), QNLI (QN), RTE (R), WNLI (W). We use M2 to denote the 2-label version of MultiNLI task. Note that for the metrics we considered here, lower scores indicate better calibration.