Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift

Yaniv Ovadia[*]
Google Research
yovadia@google.com

Emily Fertig[*†]
Google Research
emilyaf@google.com

Jie Ren[†]
Google Research
jjren@google.com

Zachary Nado
Google Research
znado@google.com

D Sculley
Google Research
dsculley@google.com

Sebastian Nowozin
Google Research
nowozin@google.com

Joshua V. Dillon
Google Research
jvdillon@google.com

Balaji Lakshminarayanan[‡]
DeepMind
balajiln@google.com

Jasper Snoek[‡]
Google Research
jsnoek@google.com

Abstract

Modern machine learning methods including deep learning have achieved great success in predictive accuracy for supervised learning tasks, but may still fall short in giving useful estimates of their predictive *uncertainty*. Quantifying uncertainty is especially critical in real-world settings, which often involve input distributions that are shifted from the training distribution due to a variety of factors including sample bias and non-stationarity. In such settings, well calibrated uncertainty estimates convey information about when a model's output should (or should not) be trusted. Many probabilistic deep learning methods, including Bayesian-and non-Bayesian methods, have been proposed in the literature for quantifying predictive uncertainty, but to our knowledge there has not previously been a rigorous large-scale empirical comparison of these methods under dataset shift. We present a large-scale benchmark of existing state-of-the-art methods on classification problems and investigate the effect of dataset shift on accuracy and calibration. We find that traditional post-hoc calibration does indeed fall short, as do several other previous methods. However, some methods that marginalize over models give surprisingly strong results across a broad spectrum of tasks.

1 Introduction

Recent successes across a variety of domains have led to the widespread deployment of deep neural networks (DNNs) in practice. Consequently, the predictive distributions of these models are increasingly being used to make decisions in important applications ranging from machine-learning aided medical diagnoses from imaging (Esteva et al., 2017) to self-driving cars (Bojarski et al., 2016). Such high-stakes applications require not only point predictions but also accurate quantification of predictive uncertainty, i.e. meaningful confidence values in addition to class predictions. With sufficient independent labeled samples from a target data distribution, one can estimate how well

[*]Equal contribution
[†]AI Resident
[‡]Corresponding authors

Preprint. Under review.

a model's confidence aligns with its accuracy and adjust the predictions accordingly. However, in practice, once a model is deployed the distribution over observed data may shift and eventually be very different from the original training data distribution. Consider, e.g., online services for which the data distribution may change with the time of day, seasonality or popular trends. Indeed, robustness under conditions of distributional shift and out-of-distribution (OOD) inputs is necessary for the safe deployment of machine learning (Amodei et al., 2016). For such settings, calibrated predictive uncertainty is important because it enables accurate assessment of risk, allows practitioners to know how accuracy may degrade, and allows a system to abstain from decisions due to low confidence.

A wide variety of approaches have been developed for quantifying predictive uncertainty in DNNs. Probabilistic neural networks such as mixture density networks (MacKay & Gibbs, 1999) capture the inherent ambiguity in outputs for a given input, also referred to as *aleatoric uncertainty* (Kendall & Gal, 2017). Bayesian neural networks learn a posterior distribution over parameters that quantifies parameter uncertainty, a type of *epistemic uncertainty* that can be reduced through the collection of additional data. Popular approximate Bayesian approaches include Laplace approximation (MacKay, 1992), variational inference (Graves, 2011; Blundell et al., 2015), dropout-based variational inference (Gal & Ghahramani, 2016; Kingma et al., 2015), expectation propagation and stochastic gradient MCMC (Welling & Teh, 2011). Non-Bayesian methods include training multiple probabilistic neural networks and ensembling the predictions of individual models (Osband et al., 2016; Lakshminarayanan et al., 2017). Another popular non-Bayesian approach involves re-calibration of probabilities on a held-out validation set through temperature scaling (Platt, 1999), which was shown by Guo et al. (2017) to lead to well-calibrated predictions on the i.i.d. test set.

Using Distributional Shift to Evaluate Predictive Uncertainty While previous work has evaluated the quality of predictive uncertainty on OOD inputs (Lakshminarayanan et al., 2017), there has not to our knowledge been a comprehensive evaluation of uncertainty estimates from different methods under dataset shift. Indeed, we suggest that effective evaluation of predictive uncertainty is most meaningful under conditions of distributional shift. One reason for this is that post-hoc calibration gives good results in independent and identically distributed (i.i.d.) regimes, but can fail under even a mild shift in the input data. And in real world applications, as described above, distributional shift is widely prevalent. Understanding questions of risk, uncertainty, and trust in a model's output becomes increasingly critical as shift from the original training data grows larger.

Contributions In the spirit of calls for more rigorous understanding of existing methods (Lipton & Steinhardt, 2018; Sculley et al., 2018; Rahimi & Recht, 2017), this paper provides a benchmark for evaluating uncertainty that focuses not only on calibration in the i.i.d. setting but also *calibration under distributional shift.* We present a large-scale evaluation of popular approaches in probabilistic deep learning, focusing on methods that operate well in large-scale settings, and evaluate them on a diverse range of classification benchmarks across image, text, and categorical modalities. We use these experiments to evaluate the following questions:

- How trustworthy are the uncertainty estimates of different methods under dataset shift?
- Does calibration in the i.i.d. setting translate to calibration under dataset shift?
- How do uncertainty and accuracy of different methods co-vary under dataset shift? Are there methods that consistently do well in this regime?

In addition to answering the questions above, we will also release open-source code and our model predictions such that researchers can easily evaluate their approaches on these benchmarks.

2 Background

Notation and Problem Setup Let $\boldsymbol{x} \in \mathbb{R}^d$ represent a set of d-dimensional features and $y \in \{1, \ldots, k\}$ denote corresponding labels (targets) for k-class classification. We assume that a training dataset $\mathcal{D}$ consists of N i.i.d.samples $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$.

Let $p^*(\boldsymbol{x}, y)$ denote the true distribution (unknown, observed only through the samples $\mathcal{D}$), also referred to as the *data generating process*. We focus on classification problems, in which the true distribution is assumed to be a discrete distribution over k classes, and the observed $y \in \{1, \ldots, k\}$ is a sample from the conditional distribution $p^*(y|\boldsymbol{x})$. We use a neural network to model $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ and estimate the parameters $\boldsymbol{\theta}$ using the training dataset. At test time, we evaluate the model predictions

against a test set, sampled from the same distribution as the training dataset. However, here we would also like to evaluate the model against OOD inputs sampled from $q(\boldsymbol{x}, y) \neq p^*(\boldsymbol{x}, y)$. In particular, we consider two kinds of shifts:

- *shifted versions* of the test inputs where the ground truth label belongs to one of the k classes. We use shifts such as corruptions and perturbations proposed by Hendrycks & Dietterich (2019), and ideally would like the model predictions to become more uncertain with increased shift, assuming shift degrades accuracy.
- *a completely different OOD dataset*, where the ground truth label is not one of the k classes. Here we check if the model exhibits higher predictive uncertainty for those new instances and to this end report diagnostics that rely only on predictions and not ground truth labels.

High-level overview of existing methods A large variety of methods have been developed to either provide higher quality uncertainty estimates or perform OOD detection to inform model confidence. These can roughly be divided into:

1. Methods which deal with $p(y|\boldsymbol{x})$ only, we discuss these in more detail in Section 3.
2. Methods which model the joint distribution $p(y, \boldsymbol{x})$, e.g. deep hybrid models (Kingma et al., 2014; Alemi et al., 2018; Nalisnick et al., 2019; Behrmann et al., 2018).
3. Methods with an OOD-detection component in addition to $p(y|\boldsymbol{x})$ (Bishop, 1994; Lee et al., 2018; Liang et al., 2018), and related work on selective classification (Geifman & El-Yaniv, 2017).

We refer to Shafaei et al. (2018) for a recent summary of these methods. Due to the differences in modeling assumptions, a fair comparison between these different classes of methods is challenging; for instance, some OOD detection methods rely on knowledge of a known OOD set, or train using a none-of-the-above class, and it may not always be meaningful to compare predictions from these methods with those obtained from a Bayesian DNN. We focus on methods described by (1) above, as this allows us to focus on methods which make the same modeling assumptions about data and differ only in how they quantify predictive uncertainty.

3 Methods and Metrics

We select a subset of methods from the probabilistic deep learning literature for their prevalence, scalability and practical applicability[4]. These include (see also references within):

- (*Vanilla*) Maximum softmax probability (Hendrycks & Gimpel, 2017)
- (*Temp Scaling*) Post-hoc calibration by temperature scaling using a validation set (Guo et al., 2017)
- (*Dropout*) Monte-Carlo Dropout (Gal & Ghahramani, 2016; Srivastava et al., 2015) with rate p
- (*Ensembles*) Ensembles of M networks trained independently on the entire dataset using random initialization (Lakshminarayanan et al., 2017) (we set $M = 10$ in experiments below)
- (*SVI*) Stochastic Variational Bayesian Inference for deep learning (Blundell et al., 2015; Graves, 2011; Louizos & Welling, 2017, 2016; Wen et al., 2018). We refer to Appendix A.6 for details of our SVI implementation.
- (LL) Approx. Bayesian inference for the parameters of the last layer only (Riquelme et al., 2018)
  - (*LL SVI*) Mean field stochastic variational inference on the last layer only
  - (*LL Dropout*) Dropout only on the activations before the last layer

In addition to metrics (we use arrows to indicate which direction is better) that do not depend on predictive uncertainty, such as classification accuracy ↑, the following metrics are commonly used:

Negative Log-Likelihood (NLL) ↓ Commonly used to evaluate the quality of model uncertainty on some held out set. *Drawbacks:* Although a proper scoring rule (Gneiting & Raftery, 2007), it can over-emphasize tail probabilities (Quinonero-Candela et al., 2006).

[4]The methods used scale well for training and prediction (see **in Appendix A.8.)**. We also explored methods such as scalable extensions of Gaussian Processes (Hensman et al., 2015), but they were challenging to train on the 37M example Criteo dataset or the 1000 classes of ImageNet.

Brier Score ↓ (Brier, 1950) Proper scoring rule for measuring the accuracy of predicted probabilities. It is computed as the squared error of a predicted probability *vector*, $p(y|x_n, \boldsymbol{\theta})$, and the one-hot encoded true response, y_n. That is,

$$\text{BS} = |\mathcal{Y}|^{-1} \sum_{y \in \mathcal{Y}} (p(y|\boldsymbol{x}_n, \boldsymbol{\theta}) - \delta(y - y_n))^2 = |\mathcal{Y}|^{-1} \Big(1 - 2p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) + \sum_{y \in \mathcal{Y}} p(y|\boldsymbol{x}_n, \boldsymbol{\theta})^2\Big). \quad (1)$$

The Brier score has a convenient interpretation as $BS = \text{uncertainty} - \text{resolution} + \text{reliability}$, where uncertainty is the marginal uncertainty over labels, resolution measures the deviation of individual predictions against the marginal, and reliability measures calibration as the average violation of long-term true label frequencies. We refer to (DeGroot & Fienberg, 1983) for the decomposition of Brier score into calibration and refinement for classification and to (Bröcker, 2009) for the general decomposition for any proper scoring rule. *Drawbacks:* Brier score is insensitive to predicted probabilities associated with in/frequent events.

Both the Brier score and the negative log-likelihood are proper scoring rules and therefore the optimum score corresponds to a perfect prediction. In addition to these two metrics, we also evaluate two metrics—*expected calibration error* and *entropy*—which focus on a particular aspect of the predicted probabilities. Neither of these metrics is a proper scoring rule, and thus there exist trivial solutions which make these metrics perfect; for example, returning the marginal probability $p(y)$ for every instance will yield perfectly calibrated but uninformative predictions. However, both metrics measure important properties that are not directly measured by proper scoring rules.

Expected Calibration Error (ECE) ↓ Measures predicted probability accuracy (Naeini et al., 2015). It is computed as the average gap between within bucket accuracy and within bucket predicted probability for S buckets $B_s = \{n \in 1 \dots N : p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) \in (\rho_s, \rho_{s+1}]\}$. That is, $\text{ECE} = \sum_{s=1}^{S} \frac{|B_s|}{N} |\operatorname{acc}(B_s) - \operatorname{conf}(B_s)|$, where $\operatorname{acc}(B_s) = |B_s|^{-1} \sum_{n \in B_s} [y_n = \hat{y}_n]$, $\operatorname{conf}(B_s) = |B_s|^{-1} \sum_{n \in B_s} p(\hat{y}_n|\boldsymbol{x}_n, \boldsymbol{\theta})$, and $\hat{y}_n = \arg\max_y p(y|\boldsymbol{x}_n, \boldsymbol{\theta})$ is the n-th prediction. When bins $\{\rho_s : s \in 1 \dots S\}$ are quantiles of the held-out predicted probabilities, $|B_s| \approx |B_k|$ and the estimation error is approximately constant. *Drawbacks:* Due to binning, ECE does not always monotonically increase as predictions approach ground truth. If $|B_s| \neq |B_k|$, the estimation error varies across bins.

There is no ground truth label for fully OOD inputs. Thus we report histograms of **confidence** and predictive **entropy** on known and OOD inputs and **accuracy versus confidence plots** (Lakshminarayanan et al., 2017): Given the prediction $p(y = k|\boldsymbol{x}_n, \boldsymbol{\theta})$, we define the predicted label as $\hat{y}_n = \arg\max_y p(y|\boldsymbol{x}_n, \boldsymbol{\theta})$, and the confidence as $p(y = \hat{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \max_k p(y = k|\boldsymbol{x}_n, \boldsymbol{\theta})$. We filter out test examples corresponding to a particular confidence threshold $\tau \in [0, 1]$ and compute the accuracy on this set.

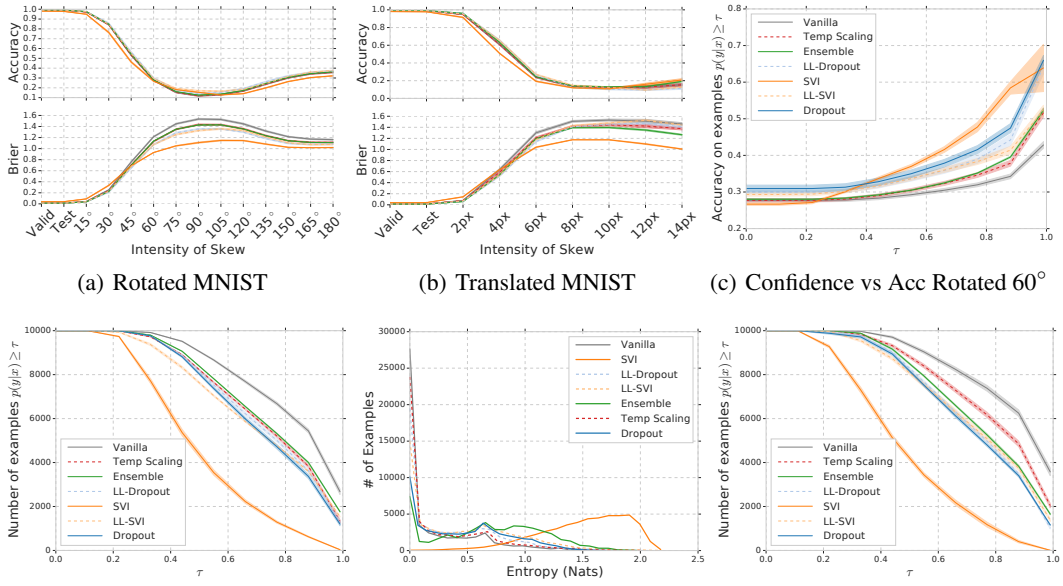
4 Experiments and Results

We evaluate the behavior of the predictive uncertainty of deep learning models on a variety of datasets across three different modalities: images, text and categorical (online ad) data. For each we follow standard training, validation and testing protocols, but we additionally evaluate results on increasingly shifted data and an OOD dataset. We detail the models and implementations used in Appendix A. Hyperparameters were tuned for all methods (except on ImageNet) as detailed in Appendix A.7.

4.1 An illustrative example - MNIST

We first illustrate the problem setup and experiments using the MNIST dataset. We used the LeNet (LeCun et al., 1998) architecture, and, as with all our experiments, we follow standard training, validation, testing and hyperparameter tuning protocols. However, we also compute predictions on increasingly shifted data (in this case increasingly rotated or horizontally translated images) and study the behavior of the predictive distributions of the models. In addition, we predict on a completely OOD dataset, Not-MNIST (Bulatov, 2011), and observe the entropy of the model's predictions. We summarize some of our findings in Figure 1 and discuss below.

What we would like to see: Naturally, we expect the accuracy of a model to degrade as it predicts on increasingly shifted data, and ideally this reduction in accuracy would coincide with increased forecaster entropy. A model that was well-calibrated on the training and validation distributions would



(a) Rotated MNIST (b) Translated MNIST (c) Confidence vs Acc Rotated 60°

(d) Count vs Confidence Rotated 60° (e) Entropy on OOD (f) Confidence on OOD

Figure 1: Results on MNIST: 1(a) and 1(b) show accuracy and Brier score as the data is increasingly shifted. Shaded regions represent standard error over 10 runs. SVI has lower accuracy on the validation and test splits, but it is significantly more robust to dataset shift as evidenced by a lower Brier score and higher predictive entropy under shift (1(c)) and OOD data (1(e),1(f)).

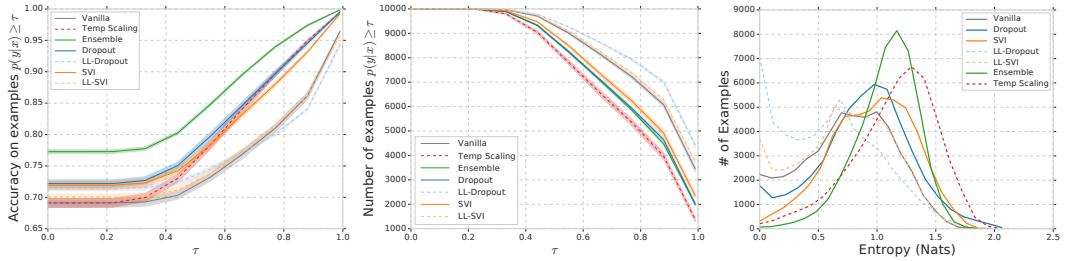
ideally remain so on shifted data. If calibration (ECE or Brier reliability) remained as consistent as possible, practitioners and downstream tasks could take into account that a model is becoming increasingly uncertain. On the completely OOD data, one would expect the predictive distributions to be of high entropy. Essentially, we would like the predictions to indicate that a model "knows what it does not know" due to the inputs straying away from the training data distribution.

What we observe: We see in Figures 1(a) and 1(b) that accuracy certainly degrades as a function of shift for all methods tested, and they are difficult to disambiguate on that metric. However, the Brier score paints a clearer picture and we see a significant difference between methods, i.e. prediction quality degrades more significantly for some methods than others. An important observation is that *while calibrating on the validation set leads to well-calibrated predictions on the test set, it does not guarantee calibration on shifted data*. In fact, nearly all other methods (except vanilla) perform better than the state-of-the-art post-hoc calibration (Temperature scaling) in terms of Brier score under shift. While SVI achieves the worst accuracy on the test set, it actually outperforms all other methods by a much larger margin when exposed to significant shift. We see in Figure 1(c) that SVI gives the highest accuracy at high confidence (or conversely is much less frequently confidently wrong) which can be important for high-stakes applications. Most methods demonstrate very low entropy (Figure 1(e)) and give high confidence predictions (Figure 1(f)) on data that is entirely OOD, i.e. they are confidently wrong about completely OOD data.

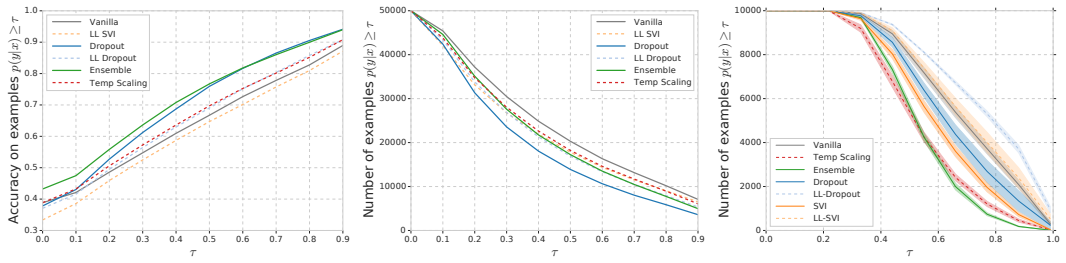
4.2 Image Models: CIFAR-10 and ImageNet

We now study the predictive distributions of residual networks (He et al., 2016) trained on two benchmark image datasets, CIFAR-10 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009), under distributional shift. We use 20-layer and 50-layer ResNets for CIFAR-10 and ImageNet respectively. For shifted data we use 80 different distortions (16 different types with 5 levels of intensity each, see Appendix B for illustrations) introduced by Hendrycks & Dietterich (2019). To evaluate predictions of CIFAR-10 models on entirely OOD data, we use the SVHN dataset (Netzer et al., 2011).

We summarize the results in Figures 2 and 3. Figure 2 inspects the predictive distributions of the models on CIFAR-10 (top) and ImageNet (bottom) for skewed (Gaussian blur) and OOD data. Figure 3 summarizes the accuracy and ECE for CIFAR-10 (top) and ImageNet (bottom) across all 80



(a) CIFAR: Confidence vs Accuracy (b) CIFAR: Count vs Confidence (c) CIFAR: Entropy on OOD



(d) ImageNet: Confidence vs Acc (e) ImageNet: Count vs Confidence (f) CIFAR: Confidence on OOD

Figure 2: Results on CIFAR-10 and ImageNet. Left column: 2(a) and 2(d) show accuracy as a function of confidence. Middle column: 2(b) and 2(e) show the number of examples greater than given confidence values for Gaussian blur of intensity 3. Right column: 2(c) and 2(f) show histogram of entropy and confidences from CIFAR-trained models on a completely different dataset (SVHN).

combinations of corruptions and intensities from (Hendrycks & Dietterich, 2019). Classifiers on both datasets show poorer accuracy and calibration with increasing degrees of skew. Comparing accuracy for different methods, we see that ensembles achieve highest accuracy under distributional skew. Comparing the ECE for different methods, we observe that while the methods achieve comparable low values of ECE for small values of skew, ensembles outperform the other methods for larger values of skew. Interestingly, *while temperature scaling achieves low ECE for low values of skew, the ECE increases significantly as the skew increases, which indicates that calibration on the i.i.d. validation dataset does not guarantee calibration under distributional skew*. (Note that for ImageNet, we found similar trends considering just the top-5 predicted classes, See Figure S6.) Furthermore, the results show that while temperature scaling helps significantly over the vanilla method, ensembles and dropout tend to be better. We refer to Appendix C for additional results; Figures S5 and S6 report additional metrics on CIFAR-10 and ImageNet, such as Brier score (and its component terms), as well as top-5 error for increasing values of skew.

Overall, ensembles consistently perform best across metrics and dropout consistently performed better than temperature scaling and last layer methods. *While the relative ordering of methods is consistent on both CIFAR-10 and ImageNet (ensembles perform best), the ordering is quite different from that on MNIST where SVI performs best.* Interestingly, LL-SVI and LL-Dropout perform worse than the vanilla method on skewed datasets as well as SVHN.

4.3 Text Models

Following Hendrycks & Gimpel (2017), we train an LSTM (Hochreiter & Schmidhuber, 1997) on the 20newsgroups dataset (Lang, 1995) and assess the model's robustness under distributional skew and OOD text. We use the even-numbered classes (10 classes out of 20) as in-distribution and the 10 odd-numbered classes as skewed data. We provide additional details in Appendix A.4.

We look at confidence vs accuracy when the test data consists of a mix of in-distribution and either skewed or completely OOD data, in this case the One Billion Word Benchmark (LM1B) (Chelba et al., 2013). Figure 4 (bottom row) shows the results. Ensembles significantly outperform all other methods, and achieve better trade-off between accuracy versus confidence. Surprisingly, LL-Dropout and LL-SVI perform worse than the vanilla method, especially when tested on fully OOD data.

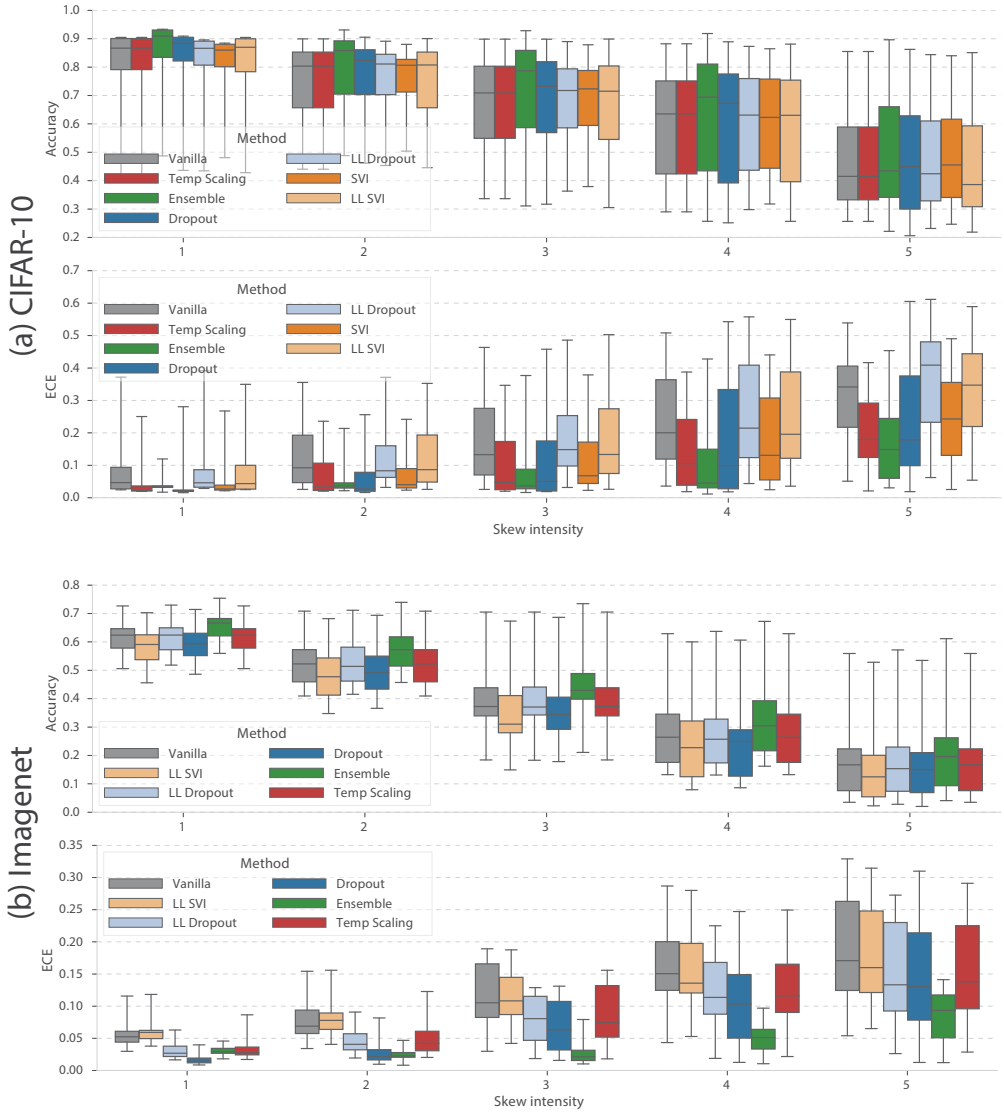
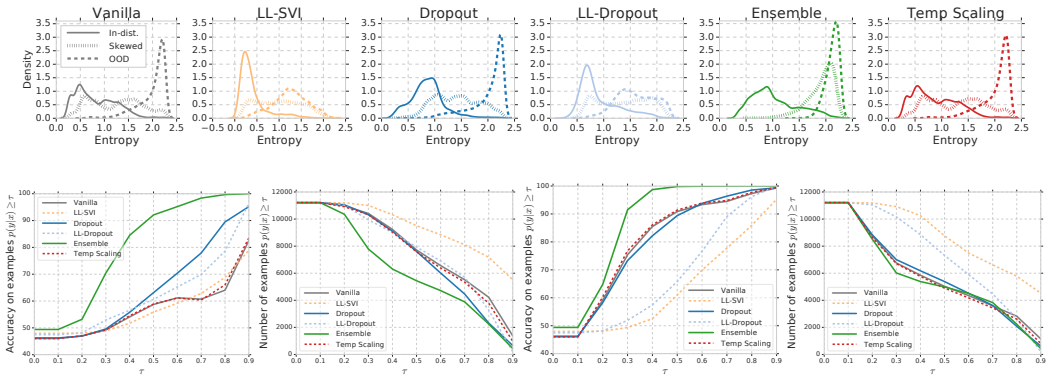


Figure 3: Calibration under distributional shift: boxplots showing a detailed comparison of Brier score and ECE under all types of corruptions on (a) CIFAR-10 and (b) ImageNet. Each box shows the quartiles summarizing the results across all types of skew while the error bars indicate the min and max across different skew types. Figures showing additional metrics are provided in Figures S5 (CIFAR-10) and S6 (ImageNet). Tables for numerical comparisons are provided in Appendix E.

Figure 4 reports histograms of predictive entropy on in-distribution data and compares them to those for the skewed and OOD datasets. As expected, most methods achieve the highest predictive entropy on the completely OOD dataset, followed by the skewed dataset and then the in-distribution test dataset. Only ensembles have consistently higher entropy on the skewed data, which explains why they perform best on the confidence vs accuracy curves in the second row of Figure 4. Compared with the vanilla model, Dropout and LL-SVI have more a distinct separation between in-distribution and skewed or OOD data. While Dropout and LL-Dropout perform similarly on in-distribution, LL-Dropout exhibits less uncertainty than Dropout on skewed and OOD data. Temperature scaling does not appear to increase uncertainty significantly on the skewed data.



(a) Confidence vs Acc. (b) Confidence vs Count (c) Confidence vs Accuracy (d) Confidence vs Count

Figure 4: Top row: Histograms of the entropy of the predictive distributions for in-distribution (solid lines), skewed (dotted lines), and completely different OOD (dashed lines) text examples. Bottom row: Confidence score vs accuracy and count respectively when evaluated for in-distribution and in-distribution shift text examples (a,b), and in-distribution and OOD text examples (c,d).

4.4 Ad-Click Model with Categorical Features

Finally, we evaluate the performance of different methods on the *Criteo Display Advertising Challenge*[5] dataset, a binary classification task consisting of 37M examples with 13 numerical and 26 categorical features per example. We introduce skew by reassigning each categorical feature to a random new token with some fixed probability that controls the intensity of skew. This coarsely simulates a type of skew observed in non-stationary categorical features as category tokens appear and disappear over time. The model consists of a 3-hidden-layer multi-layer-perceptron (MLP) with hashed and embedded categorical features and achieves a negative log-likelihood of approximately 0.5 (contest winners achieved 0.44). Due to class imbalance ($\sim 25\%$ of examples are positive), we report AUC instead of classification accuracy.

Results from these experiments are depicted in Figure 5. (Figure S7 in Appendix C shows additional results including ECE and Brier score decomposition.) We observe that ensembles are superior in terms of both AUC and Brier score for most of the values of skew, with the performance gap between ensembles and other methods generally increasing as the skew increases. Both Dropout model variants yielded improved AUC on skewed data, and Dropout surpassed ensembles in Brier score at skew-randomization values above 60%. SVI proved challenging to train, and the resulting model uniformly performed poorly; LL-SVI fared better but generally did not improve upon the vanilla model. *Strikingly, temperature scaling has a worse Brier score than Vanilla indicating that post-hoc calibration on the validation set actually harms calibration under dataset shift.*

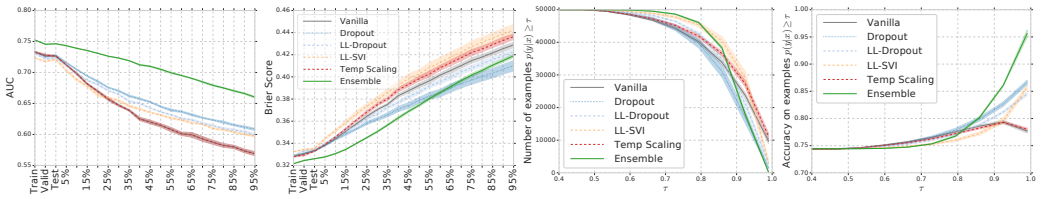


Figure 5: Results on Criteo: The first two plots show degrading AUCs and Brier scores with increasing skew while the latter two depict the distribution of prediction confidences and their corresponding accuracies at 75% randomization of categorical features. SVI is excluded as it performed too poorly.

[5] https://www.kaggle.com/c/criteo-display-ad-challenge

5 Takeaways and Recommendations

We presented a large-scale evaluation of different methods for quantifying predictive uncertainty under dataset shift, across different data modalities and architectures. Our take-home messages are the following:

- Quality of uncertainty consistently degrades with increasing dataset shift regardless of method.
- Better calibration and accuracy on i.i.d. test dataset does not usually translate to better calibration under dataset shift (skewed versions as well as completely different OOD data).
- Post-hoc calibration (on i.i.d validation) with temperature scaling leads to well-calibrated uncertainty on i.i.d. test and small values of skew, but is significantly outperformed by methods that take epistemic uncertainty into account as the skew increases.
- Last layer Dropout exhibits less uncertainty on skewed and OOD datasets than Dropout.
- SVI is very promising on MNIST/CIFAR but it is difficult to get to work on larger datasets such as ImageNet and other architectures such as LSTMs.
- The relative ordering of methods is mostly consistent (except for MNIST) across our experiments. The relative ordering of methods on MNIST is not reflective of their ordering on other datasets.
- Deep ensembles seem to perform the best across most metrics and be more robust to dataset shift. We found that relatively small ensemble size (e.g. $M = 5$) may be sufficient (Appendix D).

We hope that this benchmark is useful to the community and inspires more research on uncertainty under dataset shift, which seems challenging for existing methods. While we focused only on the quality of predictive uncertainty, applications may also need to consider computational and memory costs of the methods; Table S1 in Appendix A.8 discusses these costs, and the best performing methods tend to be more expensive. Reducing the computational and memory costs, while retaining the same performance under dataset shift, would also be a key research challenge.

References

Alemi, A. A., Fischer, I., and Dillon, J. V. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Behrmann, J., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

Bishop, C. M. Novelty Detection and Neural Network Validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *ICML*, 2015.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, abs/1604.07316, 2016.

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

Bröcker, J. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009.

Bulatov, Y. NotMNIST dataset, 2011. URL http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

DeGroot, M. H. and Fienberg, S. E. The comparison and evaluation of forecasters. *The statistician*, 1983.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition*, 2009.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542, 1 2017.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Hendrycks, D. and Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *ICLR*, 2017.

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational gaussian process classification. In *International Conference on Artificial Intelligence and Statistics*. JMLR, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

Kendall, A. and Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. In *ICLR*, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, 2015.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Lang, K. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pp. 331–339. Elsevier, 1995.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, November 1998.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

Liang, S., Li, Y., and Srikant, R. Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks. *ICLR*, 2018.

Lipton, Z. C. and Steinhardt, J. Troubling trends in machine learning scholarship. *arXiv preprint arXiv:1807.03341*, 2018.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix Gaussian posteriors. *arXiv preprint arXiv:1603.04733*, 2016.

Louizos, C. and Welling, M. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. In *ICML*, 2017.

MacKay, D. J. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

MacKay, D. J. and Gibbs, M. N. Density Networks. *Statistics and Neural Networks: Advances at the Interface*, 1999.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *AAAI*, pp. 2901–2907, 2015.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Hybrid models with deep and invertible features. *arXiv preprint arXiv:1902.02767*, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, 2016.

Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, 1999.

Quinonero-Candela, J., Rasmussen, C. E., Sinz, F., Bousquet, O., and Schölkopf, B. Evaluating predictive uncertainty challenge. In *Machine Learning Challenges*. Springer, 2006.

Rahimi, A. and Recht, B. An addendum to alchemy, 2017.

Riquelme, C., Tucker, G., and Snoek, J. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *ICLR*, 2018.

Sculley, D., Snoek, J., Wiltschko, A., and Rahimi, A. Winner's curse? on pace, progress, and empirical rigor. 2018.

Shafaei, A., Schmidt, M., and Little, J. J. Does Your Model Know the Digit 6 Is Not a Cat? A Less Biased Evaluation of "Outlier" Detectors. *ArXiv e-Print arXiv:1809.04729*, 2018.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Training Very Deep Networks. In *Advances in Neural Information Processing Systems*, 2015.

Welling, M. and Teh, Y. W. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *ICML*, 2011.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernandez-Lobato, J. M., and Gaunt, A. L. Deterministic Variational Inference for Robust Bayesian Neural Networks. In *ICLR*, 2019.

Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift: Appendix

A Model Details

A.1 MNIST

We evaluated both LeNet and a fully-connected neural network (MLP) under shift on MNIST. We observed similar trends across metrics for both models, so we report results only for LeNet in Section 4.1. LeNet and MLP were trained for 20 epochs using the Adam optimizer (Kingma & Ba, 2014) and used ReLU activation functions. For stochastic methods, we averaged 300 sample predictions to yield a predictive distribution, and the ensemble model used 10 instances trained from independent random initializations. The MLP architecture consists of two hidden layers of 200 units each with dropout applied before every dense layer. The LeNet architecture (LeCun et al., 1998) applies two convolutional layers 3x3 kernels of 32 and 64 filters respectively) followed by two fully-connected layers with one hidden layer of 128 activations; dropout was applied before each fully-connected layer. We employed hyperparameter tuning (See Section A.7) to select the training batch size, learning rate, and dropout rate.

A.2 CIFAR-10

Our CIFAR model used the ResNet-20 V1 architecture with ReLU activations. Model parameters were trained for 200 epochs using the Adam optimizer and employed a learning rate schedule that multiplied an initial learning rate by 0.1, 0.01, 0.001, and 0.0005 at steps 80, 120, 160, and 180 respectively. Training inputs were randomly distorted using horizontal flips and random crops preceded by 4-pixel padding as described in (He et al., 2016). For relevant methods, dropout was applied before each convolutional and dense layer (excluding the raw inputs), and stochastic methods sampled 128 predictions per sample. Hyperparameter tuning was used to select the initial learning rate, training batch size, and the dropout rate.

A.3 ImageNet 2012

Our ImageNet model used the ResNet-50 V1 architecture with ReLU activations and was trained for 90 epochs using SGD with Nesterov momentum. The learning rate schedule linearly ramps up to a base rate in 5 epochs and scales down by a factor of 10 at each of epochs 30, 60, and 80. As with the CIFAR-10 model, stochastic methods used a sample-size of 128. Training images were distorted with random horizontal flips and random crops.

A.4 20 Newsgroups

We use a pre-processing strategy similar to the one proposed by Hendrycks & Gimpel (2017) for 20 Newsgroups. We build a vocabulary of size 30,000 words and words are indexed based on the word frequencies. The rare words are encoded as unknown words. We fix the length of each text input by setting a limit of 250 words, and those longer than 250 words are truncated, and those shorter than 250 words are padded with zeros. Text in even-numbered classes are used as in-distribution inputs, and text from the odd-numbered of classes are used skewed OOD inputs. A dataset with the same number of randomly selected text inputs from the LM1B dataset (Chelba et al., 2013) is used as completely different OOD dataset. The classifier is trained and evaluated only using the text from the even-numbered in-distribution classes in the training dataset. The final test results are evaluated based on in-distribution test dataset, skew OOD test dataset, and LM1B dataset.

The vanilla model uses a one-layer LSTM model of size 32 and a dense layer to predict the 10 class probabilities based on word embedding of size 128. A dropout rate of 0.1 is applied to both the LSTM layer and the dense layer for the Dropout model. The LL-SVI model replaces the last dense layer with a Bayesian layer, the ensemble model aggregates 10 vanilla models, and stochastic methods sample 5 predictions per example. The vanilla model accuracy for in-distribution test data is 0.955.

A.5 Criteo

Each categorical feature x_k from the Criteo dataset was encoded by hashing the string token into a fixed number of buckets N_k and either encoding the hash-bin as a one-hot vector if $N_k < 110$ or embedding each bucket as a d_k dimensional vector otherwise. This dense feature vector, concatenated with 13 numerical features, feeds into a batch-norm layer followed by a 3-hidden-layer MLP. Each model was trained for one epoch using the Adam optimizer with a non-decaying learning rate.

Values of N_k and d_k were tuned to maximize log-likelihood for a vanilla model, and the resulting architectural parameters were applied to all methods. This tuning yielded hidden-layers of size 2572, 1454, and 1596, and hash-bucket counts and embedding dimensions of sizes listed below:

$$\begin{aligned} N_k &= [1373, 2148, 4847, 9781, 396, 28, 3591, 2798, 14, 7403, 2511, 5598, 9501, \\ &\quad 46, 4753, 4056, 23, 3828, 5856, 12, 4226, 23, 61, 3098, 494, 5087] \\ d_k &= [3, 9, 29, 11, 17, 0, 14, 4, 0, 12, 19, 24, 29, 0, 13, 25, 0, 8, 29, 0, 22, 0, 0, 31, 0, 29] \end{aligned}$$

Learning rate, batch size, and dropout rate were further tuned for each method. Stochastic methods used 128 prediction samples per example.

A.6 Stochastic Variational Inference Details

For MNIST we used Flipout (Wen et al., 2018), where we replaced each dense layer and convolutional layer with mean-field variational dense and convolutional Flipout layers respectively. Variational inference for deep ResNets (He et al., 2016) is non-trivial, so for CIFAR we replaced a single linear layer per residual branch with a Flipout layer, removed batch normalization, added Selu non-linearities (Klambauer et al., 2017), empirical Bayes for the prior standard deviations as in Wu et al. (2019) and careful tuning of the initialization via Bayesian optimization.

A.7 Hyperparameter Tuning

Hyperparameters were optimized using Bayesian optimization via an automated tuning system to maximize the log-likelihood on a validation set that was held out from training (10K examples for MNIST and CIFAR-10, 125K examples for ImageNet). We optimized log-likelihood rather than accuracy since the former is a proper scoring rule.

A.8 Computational and Memory Complexity of Different methods

In addition to performance, applications may also need to consider computational and memory costs; Table S1 discusses them for each method.

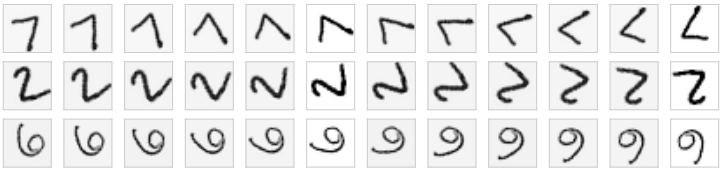
Table S1: Computational and memory costs for evaluated methods. Notation: m represents flops or storage for the full model, d represents flops or storage for the last layer, k denotes replications, n denotes number of evaluated points, and v denotes the validation set size. Serving/training compute is identical except that $v = 0$ for serving. Implicit in this table is a memory/compute tradeoff for sampling. Sampled weights/masks need not be stored explicitly via PRNG seed reuse; we assume the computational cost of sampling is zero.

Method	Compute/n	Storage
Vanilla	m	m
Temp Scaling	$m + vm/n$	m
LL-Dropout	$m + d(k-1)$	m
LL-SVI	$m + d(k-1)$	$m + d$
SVI	mk	$2m$
Dropout	mk	m
Ensemble	mk	mk

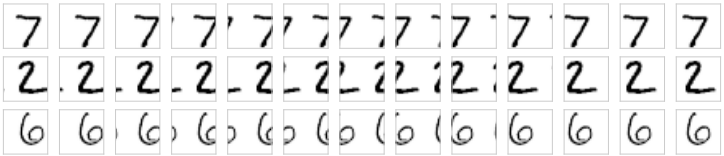
B Skewed Images

We distorted MNIST images using rotations with spline filter interpolation and cyclic translations as depicted in Figure S1.

For the corrupted ImageNet dataset, we used ImageNet-C (Hendrycks & Dietterich, 2019). Figure S2 shows examples of ImageNet-C images at varying corruption intensities. Figure S3 shows ImageNet-C images with the 16 corruptions analyzed in this paper, at intensity 3 (on a scale of 1 to 5).



(a) Rotations



(b) Cyclic translations

Figure S1: Examples of rotated and cyclically translated MNIST digits. Results for accuracy and calibration on rotated/translated MNIST are shown in Figure 1.

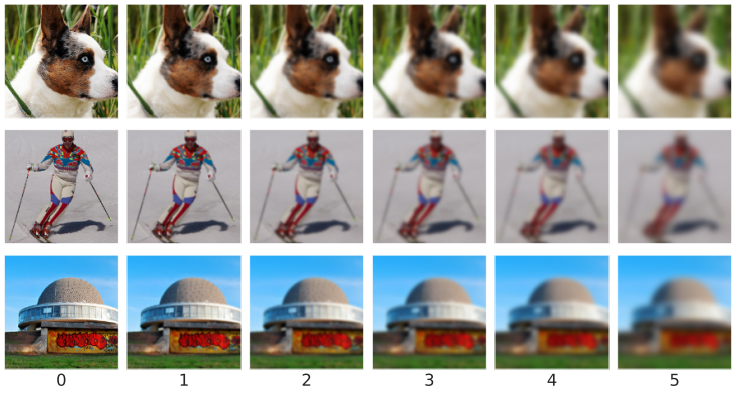


Figure S2: Examples of ImageNet images corrupted by Gaussian blur, at intensities of 0 (uncorrupted image) through 5 (maximum corruption included in ImageNet-C).

Figure S3: Examples of 16 corruption types in ImageNet-C images, at corruption intensity 3 (on a scale from 1–5). The same corruptions were applied to CIFAR-10. Figure 3 and Section C show boxplots for each uncertainty method and corruption intensity, spanning all corruption types.

C Calibration under distributional skew: Additional Results

Figures S4, S5, S6 and S7 show comprehensive results on MNIST, CIFAR-10, ImageNet and Criteo respectively across various metrics including Brier score, along with the components of the Brier score : reliability (lower means better calibration) and resolution (higher values indicate better predictive quality). Ensembles and dropout outperform all other methods across corruptions, while LL SVI shows no improvement over the baseline model.

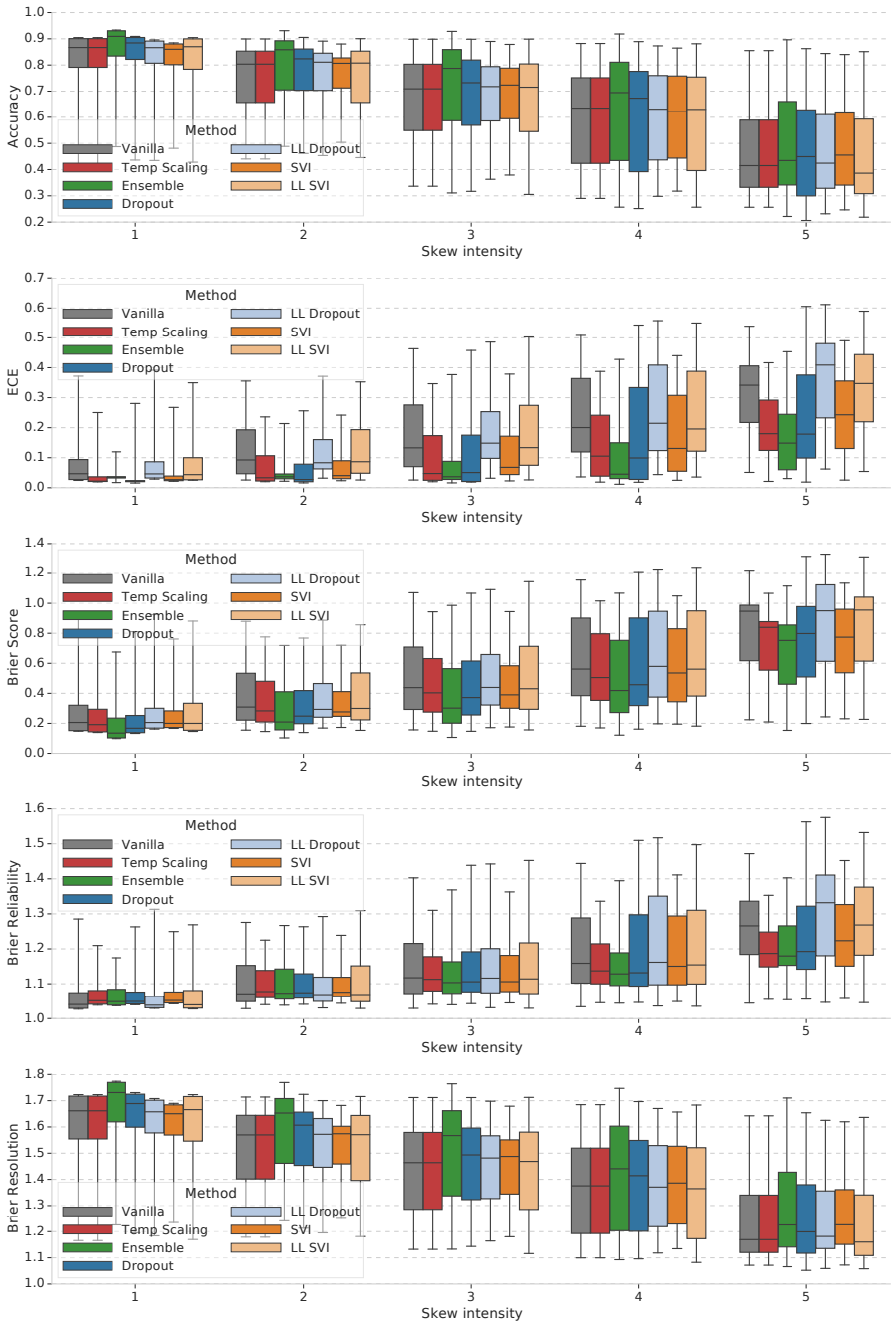


Figure S4: Boxplots facilitating comparison of methods for each skew level showing detailed comparisons of various metrics under all types of corruptions on MNIST. Each box shows the quartiles summarizing the results across all types of skew while the error bars indicate the min and max across different skew types.

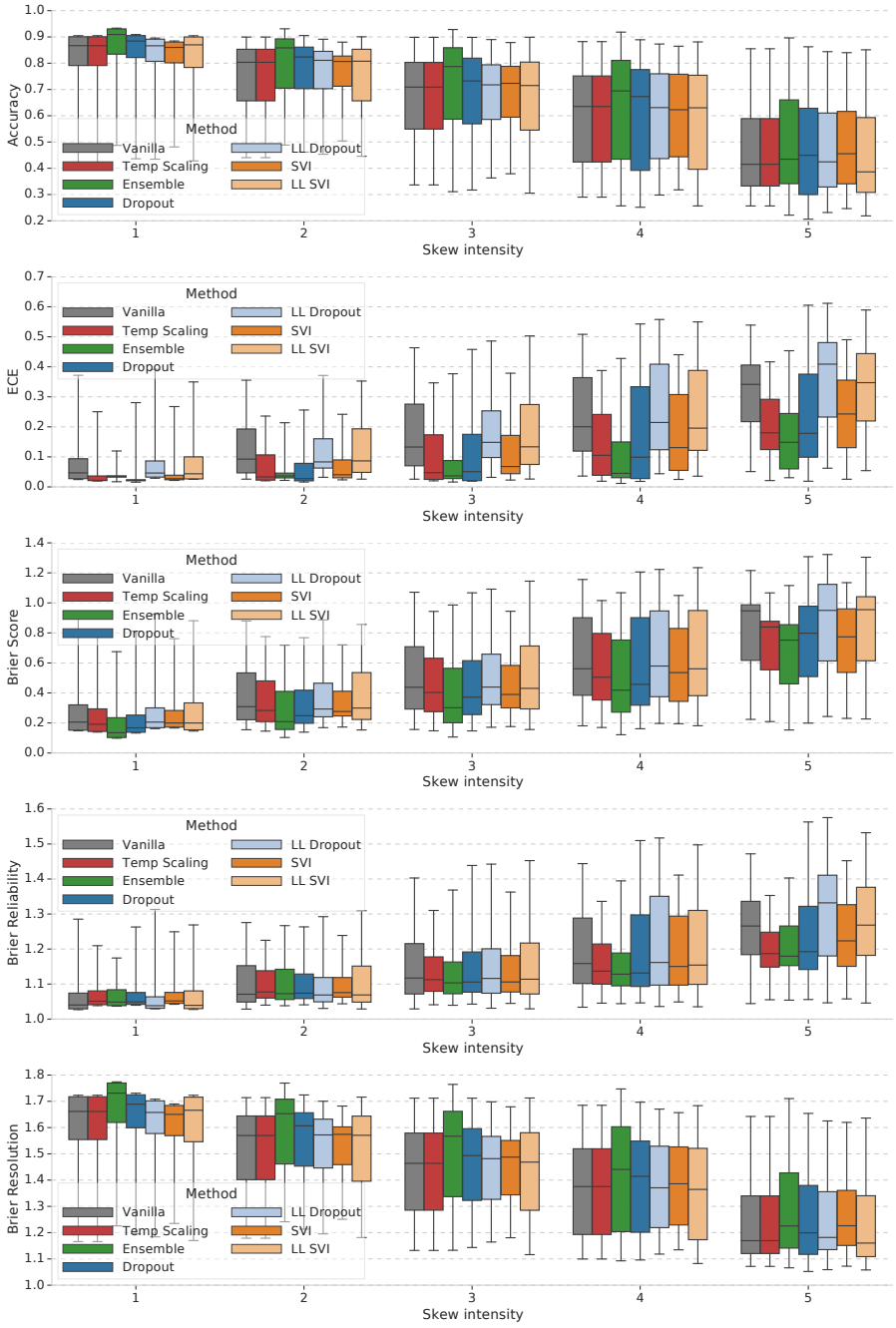


Figure S5: Boxplots facilitating comparison of methods for each skew level showing detailed comparisons of various metrics under all types of corruptions on CIFAR-10. Each box shows the quartiles summarizing the results across all types of skew while the error bars indicate the min and max across different skew types.

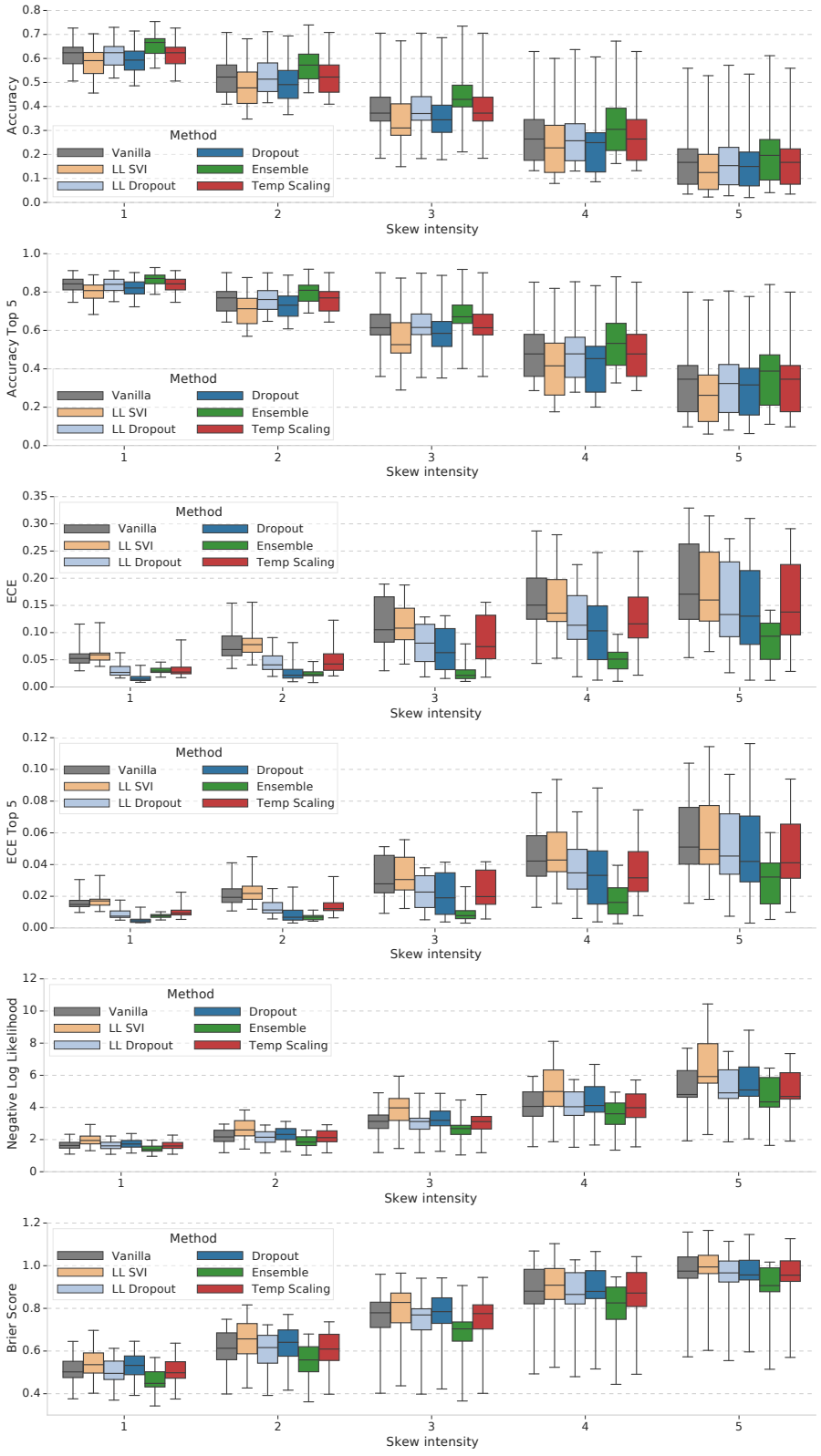


Figure S6: Boxplots facilitating comparison of methods for each skew level showing detailed comparisons of various metrics under all types of corruptions on ImageNet. Each box shows the quartiles summarizing the results across all types of skew while the error bars indicate the min and max across different skew types.

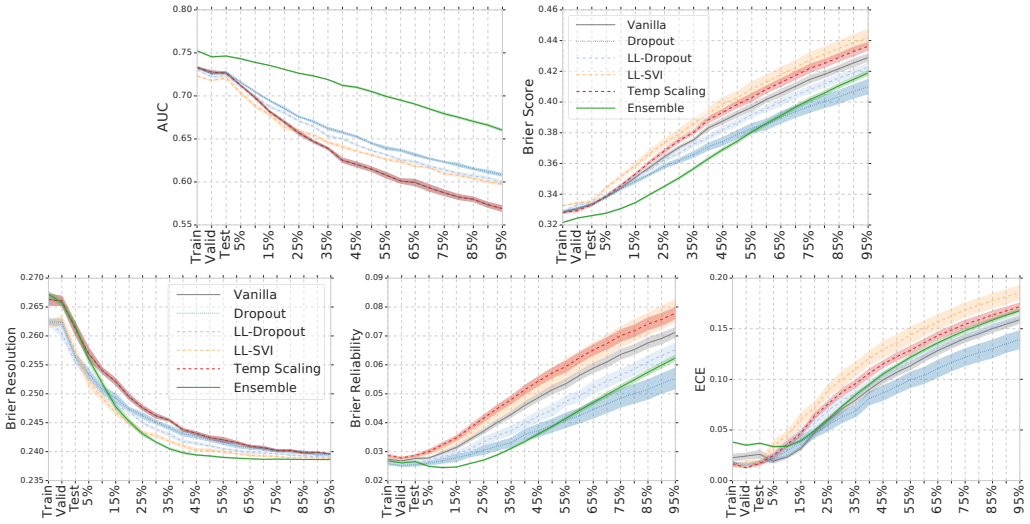
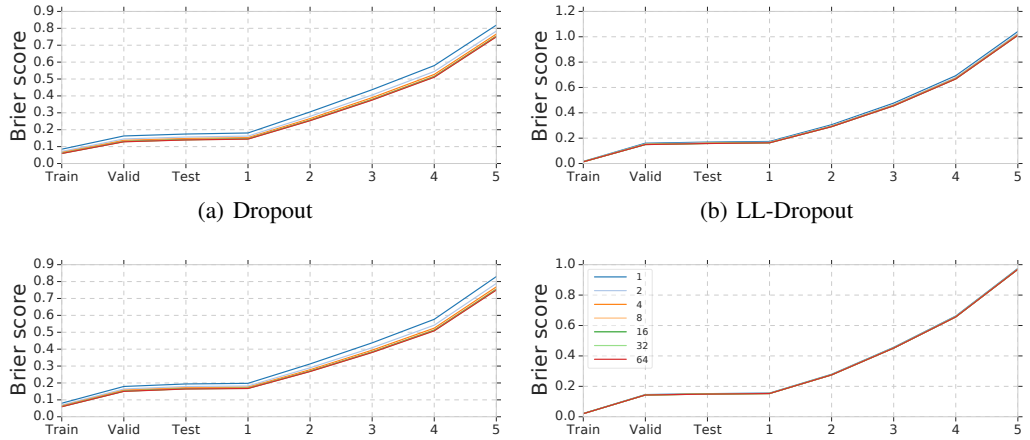


Figure S7: Comprehensive comparison of metrics on Criteo models. The Brier decomposition reveals that the majority of its degradation is due to worsening reliability, and this component alone appears to largely explain the ranking of methods in total Brier score. Ensemble notably degrades most rapidly in resolution but persists with better reliability compared other methods for most of the data-corruption range; on ECE it remains roughly in the middle among explored methods. Dropout (and to a lesser extend LL-Dropout) perform best on ECE and experience slower degradation in both resolution and reliability leading it to surpass ensembles at the severe range of data corruption. Total Brier score and AUC results are discussed in detail in Section 4.4.

D Effect of the number of samples on the quality of uncertainty

Figure S8 shows the effect of the number of sample sizes used by Dropout, SVI (and last-layer variants) on the quality of predictive uncertainty, as measured by the Brier score. Increasing the number of samples has little effect on last-layer variants, whereas increasing the number of samples improves the performance for SVI and Dropout, with diminishing returns beyond size 5.



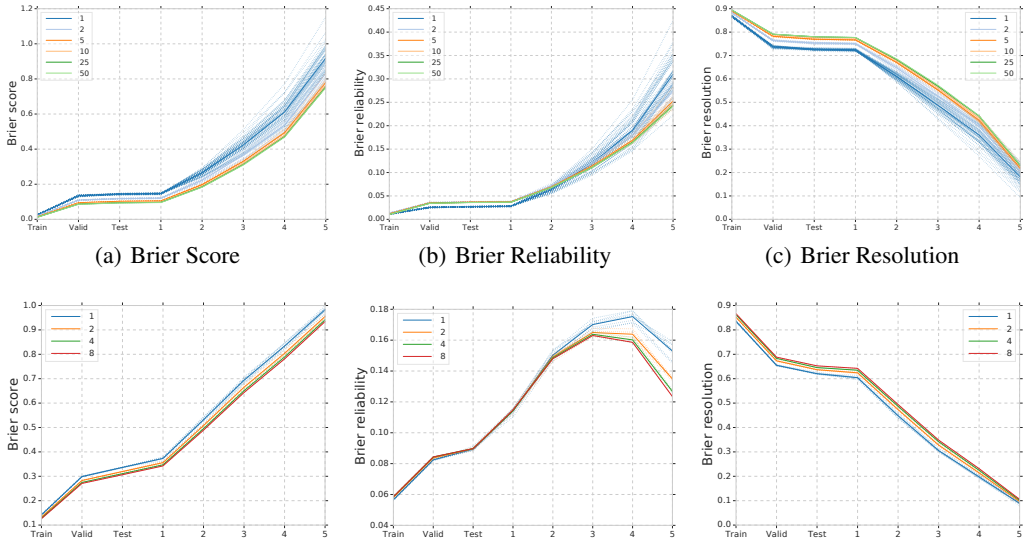
(a) Dropout (b) LL-Dropout

(c) SVI (d) LL-SVI

Figure S8: Effect of Dropout and SVI sample sizes on CIFAR-10 Brier scores under increasing Gaussian blur. See Section 4.2 for full results on CIFAR-10.

Figure S9 shows the effect of ensemble size on CIFAR-10 (top) and ImageNet (bottom). Similar to SVI and Dropout, we see that increasing the number of models in the ensemble improves performance with diminishing returns beyond size 5. As mentioned earlier, the Brier score can be further

decomposed into $\text{BS} = \text{calibration} + \text{refinement} = \text{reliability} + \text{uncertainty} - \text{resolution}$ where $\text{reliability} \downarrow$ measures calibration as the average violation of long-term true label frequencies, and $\text{refinement} = \text{uncertainty} - \text{resolution}$, where uncertainty is the marginal uncertainty over labels (independent of predictions) and $\text{resolution} \uparrow$ measures the deviation of individual predictions from the marginal.



(a) Brier Score (b) Brier Reliability (c) Brier Resolution

(d) Brier Score (e) Brier Reliability (f) Brier Resolution

Figure S9: Effect of the ensemble size on CIFAR-10 (top row) and ImageNet (bottom row) Brier scores under increasing Gaussian-blur skew. We additionally show the Brier score components: Reliability (lower means better calibration) and Resolution (higher values indicate better predictive quality). Note that the scales for Reliability are significantly smaller than the other plots.

E Tables of Metrics

The tables below report quartiles of Brier score, negative log-likelihood, and ECE for each model and dataset where quartiles are computed over all corrupted variants of the dataset.

E.1 CIFAR-10

Dataset	Vanilla	Temp. Scaling	Ensembles	Dropout	LL-Dropout	SVI	LL-SVI
Brier Score (25th)	0.243	0.227	0.165	0.215	0.259	0.250	0.246
Brier Score (50th)	0.425	0.392	0.299	0.349	0.416	0.363	0.431
Brier Score (75th)	0.747	0.670	0.572	0.633	0.728	0.604	0.732
NLL (25th)	2.356	1.685	1.543	1.684	2.275	1.628	2.352
NLL (50th)	1.120	0.871	0.653	0.771	1.086	0.823	1.158
NLL (75th)	0.578	0.473	0.342	0.446	0.626	0.533	0.591
ECE (25th)	0.057	0.022	0.031	0.021	0.069	0.029	0.058
ECE (50th)	0.127	0.049	0.037	0.034	0.136	0.064	0.135
ECE (75th)	0.288	0.180	0.110	0.174	0.292	0.187	0.275

E.2 ImageNet

Dataset	Vanilla	Temp. Scaling	Ensembles	Dropout	LL-Dropout	LL-SVI
Brier Score (25th)	0.553	0.551	0.503	0.577	0.550	0.590
Brier Score (50th)	0.733	0.726	0.667	0.754	0.723	0.766
Brier Score (75th)	0.914	0.899	0.835	0.922	0.896	0.938
NLL (25th)	1.859	1.848	1.621	1.957	1.830	2.218
NLL (50th)	2.912	2.837	2.446	3.046	2.858	3.504
NLL (75th)	4.305	4.186	3.661	4.567	4.208	5.199
ECE (25th)	0.057	0.031	0.022	0.017	0.034	0.065
ECE (50th)	0.102	0.072	0.032	0.043	0.071	0.106
ECE (75th)	0.164	0.129	0.053	0.109	0.123	0.148

E.3 Criteo

Dataset	Vanilla	Temp. Scaling	Ensembles	Dropout	LL-Dropout	SVI	LL-SVI
Brier Score (25th)	0.353	0.355	0.336	0.350	0.353	0.512	0.361
Brier Score (50th)	0.385	0.391	0.366	0.373	0.379	0.512	0.396
Brier Score (75th)	0.409	0.416	0.395	0.393	0.403	0.512	0.421
NLL (25th)	0.581	0.594	0.508	0.532	0.542	7.479	0.554
NLL (50th)	0.788	0.829	0.552	0.577	0.600	7.479	0.633
NLL (75th)	0.986	1.047	0.608	0.624	0.664	7.479	0.711
ECE (25th)	0.041	0.055	0.044	0.043	0.052	0.254	0.066
ECE (50th)	0.097	0.113	0.100	0.085	0.100	0.254	0.127
ECE (75th)	0.135	0.149	0.141	0.116	0.136	0.254	0.162