Bayesian Adaptation for Covariate Shift

Anonymous Author(s)
Affiliation
Address
email

Abstract

When faced with distribution shift at test time, deep neural networks often make inaccurate predictions with unreliable uncertainty estimates. While improving the robustness of neural networks is one promising approach to mitigate this issue, an appealing alternative to robustifying networks against all possible test-time shifts is to instead adapt them to unlabeled test data. In this paper, we propose a Bayesian model relating unlabeled data from the shifted test inputs to the classifier parameters being inferred and show how we can instantiate approximate inference in this model with a simple test-time adaptation procedure. We evaluate our method on a variety of distribution shifts for image classification, including image corruptions, natural distribution shifts, and domain adaptation settings, and show that our method improves both accuracy and uncertainty estimation.

1 Introduction

Modern deep learning methods can provide high accuracy in settings where the model is evaluated on data from the same distribution as the training set, but accuracy often degrades severely when there is a mismatch between the training and test distributions [Hendrycks and Dietterich, 2019, Taori et al., 2020]. In safety-critical settings, effectively deploying machine learning models requires not only high accuracy, but also requires the model to reliably quantify uncertainty in its predictions in order to assess risk and potentially abstain from making dangerous, unreliable predictions. Reliably estimating uncertainty is especially important in settings with distribution shift where inaccurate predictions are more prevalent, but the reliability of uncertainty estimates often also degrades along with the accuracy as the shifts become more severe [Ovadia et al., 2019]. In real-world applications, distribution mismatch at test time is often inevitable, thus necessitating methods that can robustly handle distribution shifts, both in terms of retaining high accuracy but also in providing meaningful uncertainty estimation.

As it can be difficult to train a single model to be robust to all potential distribution shifts we might encounter at test time, we can instead robustify models by allowing for *adaptation* at test time, where we can finetune the network given unlabeled inputs from the shifted target distribution, thus allowing the model to specialize in the particular shift it encounters. Since test-time distribution shifts often cannot be anticipated during training time, we restrict the adaptation procedure to operate without any further access to the original training data. Prior work on test-time adaptation [Wang et al., 2020a, Sun et al., 2019] focused on improving accuracy, and found that simple objectives like entropy minimization capable of providing substantial improvements under distribution shift [Wang et al., 2020a]. However, these prior works do not consider uncertainty estimation, and an objective like entropy minmization can quickly make predictions overly confident and less calibrated, leaving us without reliable uncertainty estimates and thus unable to quantify risks when making predictions. Our goal in this paper is to design a test-time adaptation procedure that can not only improve predictive accuracy under distribution shift, but also provide reliable uncertainty estimates.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

One common way to approach uncertainty estimation is from the standpoint of Bayesian inference, which treats the learned parameters as a random variable, and leads to a probabilistic model relating the parameters and data. We can then perform inference in this model to obtain a posterior distribution over parameters that can be used to reason about epistemic uncertainty on the test inputs. However, as discussed by Seeger [2000], the typical Bayesian supervised learning model (Figure 1) assumes independence between the classifier parameters being inferred and the input distribution, so observing unlabeled inputs provides no additional information about the parameters, and thus does not help for predicting the labels.

In this work, we formulate test-time adaptation within a Bayesian inference framework in order to improve not only the accuracy of predictions, but also to provide meaningful uncertainty estimates under distribution shift. We propose a probabilistic model for covariate shift in which observing unlabeled inputs from the test distribution *does* provide information about the classifier parameters being inferred and allows for a principled way to incorporate adaptation via entropy minimization, while still having reliable uncertainty estimates by performing Bayesian inference. We demonstrate the benefits of Bayesian marginalization in this probabilistic model, finding that marginalizing predictions over an ensemble of adapted models can outperforms either ensembling or adaptation alone in both accuracy and calibration across a range of distribution shifts. Our test-time adaptation strategy is simple to implement, requires minimal changes to standard training procedures, and outperforms prior test-time adaptation techniques on a variety of benchmarks for robustness to distribution shifts.

2 Bayesian Adaptation for Covariate Shift

We first review the standard probabilistic model for Bayesian inference in supervised learning, a probabilistic model proposed by Seeger [2000] for incorporating unalabeled data in a semisupervised learning (SSL) setting, then propose an extension of the SSL model to account for covariate shift. We then discuss challenges with performing inference in our proposed model, and describe our approximate inference procedure that allows for an efficient test-time adaptation procedure that improves both accuracy and uncertainty estimation under distribution shift.

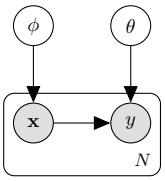

Figure 1: Probabilistic model for standard supervised learning, observing N labeled data points.

2.1 Probabilistic Model for Covariate Shift

In the standard Bayesian model for supervised learning (Figure 1), we assume that the inputs $\mathbf{X}$ are sampled i.i.d. from a generative model with parameters ϕ, while the corresponding labels are sampled from a conditional distribution $p(Y|\mathbf{X}, \theta)$ parameterized by θ. The parameters θ and ϕ are themselves random variables, sampled independently from prior distributions $p(\phi)$ and $p(\theta)$. We observe only a dataset $\mathcal{D} = \{\mathbf{X}_i, Y_i\}_{i=1}^n$, and then perform inference over the parameters θ using Bayes rule:

$$p(\theta|\mathcal{D}) \propto p(\theta) \prod_{i=1}^{n} p(Y_i|\mathbf{X}_i, \theta). \tag{1}$$

To make predictions on a new input $\mathbf{x}_{n+1}$, we marginalize over the posterior distribution of the classifier parameters θ to obtain the predictive distribution

$$p(Y_{n+1}|\mathbf{X}_{n+1}, \mathcal{D}) = \int p(Y_{n+1}|\mathbf{X}_{n+1}, \theta) p(\theta|\mathcal{D})\, d\theta.$$

Note that observing the (unlabeled) test input $\mathbf{x}_{n+1}$ (or more generally, any number of test inputs) does not affect the posterior distribution of parameters θ, and so within this probabilistic model, there is no benefit to observing multiple unlabeled datapoints before making predictions.

Inference for semi-supervised learning. By treating the parameters θ, ϕ as a-priori independent, the standard model assumes there is no relationship between the input distribution and the process that generates labels, leading to the inability to utilize unlabeled data in inferring θ. However, we often do have prior beliefs over this relationship, which we can incorporate using a model of the form in Figure 2. One reasonable belief in many classification tasks is that the inputs we encounter

typically have unambiguous labels, meaning that the true parameter θ should have very little aleatoric uncertainty.

Similar to Grandvalet and Bengio [2004], we can encode this belief about the relationship between the input distribution ϕ and the classifier θ with an additional factor corresponding to the labels having low entropy conditioned on the inputs

$$p(\theta|\phi) \propto \mu(\theta) \exp(-\alpha H_{\theta,\phi}(Y|\mathbf{X})) \tag{2}$$

$$= \mu(\theta) \exp\left(\alpha \mathbb{E}_{\mathbf{X}\sim p(X|\phi), Y\sim p(Y|X,\theta)}[\log p(Y|\mathbf{X},\theta)]\right). \tag{3}$$

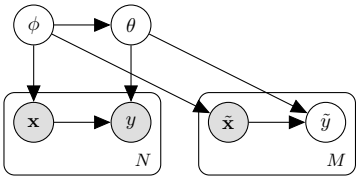

Figure 2: Model for SSL [Seeger, 2000], observing N labeled pairs and M unlabeled inputs, both generated from the same distribution. Observing unlabeled inputs $\tilde{\mathbf{x}}$ provides information about the input distribution ϕ, and thus about θ.

where $\mu(\theta)$ is a prior belief over the parameters θ that is agnostic of the input distribution parameter ϕ.

Now, observing unlabeled test inputs $\mathcal{U} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m\}$ provides information about the parameter ϕ governing the input distribution, which then allows us to update our belief over the learned parameters θ through Equation 2 and thus allows inference to utilize unlabeled data within a Bayesian framework.

Extension to covariate shift. The previous probabilistic model for SSL assumes all inputs were drawn from the same distribution (given by ϕ). However, our goal is use unlabeled data from a *different* test distribution to adapt our model, so we extend the model to incorporate *covariate shift*.

We now assume we have two input-generating distributions; ϕ specifies the input distribution for our labeled training set, and $\tilde{\phi}$ specifies the shifted distribution of unlabeled test inputs which we aim to adapt to. Under the assumption of covariate shift, the same classifier θ is used to generate labels in both the train and test domains, leading to the model in Figure 3.

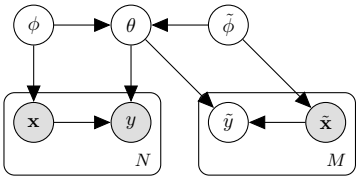

Figure 3: Our proposed probabilistic model for adaptation for covariate shift, observing a training set with N labeled pairs and M unlabeled inputs from a shifted test distribution.

Now, we argue that our prior belief of low aleatoric uncertainty is reasonable for *both* the distribution induced by ϕ and the one induced by $\tilde{\phi}$; even if there is some distribution shift from the training set, we can still often expect the new distribution to have unambiguous labels. We can then incorporate both ϕ and $\tilde{\phi}$ into our belief over θ with the factor

$$p(\theta|\phi, \tilde{\phi}) \propto \mu(\theta) \exp(-\alpha H_{\theta,\phi}(Y|\mathbf{X})) \exp(-\tilde{\alpha} H_{\theta,\tilde{\phi}}(Y|\mathbf{X})), \tag{4}$$

with the two scalar hyperparameters $\alpha, \tilde{\alpha}$ controlling how to weight the entropies from each distribution with the likelihoods of the labeled training set. While we can extend this model to include more labeled or unlabeled input distributions, we focus on simply having one training distribution and one test distribution as it is the most relevant for our problem setting of adapting to a particular distribution shift at test time.

2.2 Approximations for Tractable Inference

Performing inference over θ in the above model can be challenging in our test-time adaptation setting for several reasons. First of all, inference over θ would also require performing inference over the parameters of the generative models ϕ and $\tilde{\phi}$ to evaluate likelihoods in Equation 4. This is difficult in practice for two reasons: First, the inputs x might be high-dimensional and difficult to model, as in the case of images. Second, the amount of unlabeled data might be fairly small, and insufficient to accurately estimate the parameters $\tilde{\phi}$ if we employ a highly expressive generative model. As such, we would much prefer to avoid explict generative modeling and to only perform inference over the discriminative model parameters θ instead.

Another issue is that performing inference would require access to both the labeled training data and the unlabeled test data at the same time, while our test-time adaptation setting assumes that we can no longer look at the training set when it is time to adapt to the test distribution. We will discuss how to address these issues and describe our method, Bayesian Adaptation for Covariate Shift (BACS),

which provides a practical instantiation of inference in this Bayesian model in a computationally tractable test-time adaptation setting.

Plug-in approximation with empirical Bayes. We first propose to avoid explicit generative modeling by using with a plug-in empirical Bayes approach. Empirical Bayes is a common approximation for hierarchical Bayesian models that simply uses a point estimate of ϕ rather than marginalizing over ϕ's (similarly for $\tilde{\phi}$), which would reduce the computation to estimating only a single generative model for each input distribution given by parameters ϕ^* and $\tilde{\phi}^*$. To eliminate the need to train the parameters ϕ^* of a generative model of the inputs altogether, we note that $p(\theta|\phi, \tilde{\phi})$ only depends on ϕ and $\tilde{\phi}$ through the input distributions $p(x|\phi), p(\tilde{x}|\tilde{\phi})$. We can then approximate Equation 4 by plugging in the empirical distributions of x and $\tilde{x}$ in place of $p(x|\phi^*), p(\tilde{x}|\tilde{\phi}^*)$, resulting in

$$p(\theta|\hat{\phi}, \tilde{\phi}) \propto \mu(\theta) \exp\left(-\frac{\alpha}{n}\sum_{i=1}^{n} H(Y_i|\mathbf{x}_i, \theta)\right) \exp\left(-\frac{\tilde{\alpha}}{m}\sum_{i=1}^{m} H(Y_i|\tilde{\mathbf{x}}_i, \theta)\right).$$

Now, given a labeled training set $\mathcal{D}$ and unlabeled test points $\mathcal{U} = \{\tilde{\mathbf{x}}_i\}_{i=1}^m$, the new posterior distribution over parameters now has log probabilities (up to an additive normalizing constant)

$$\log p(\theta|\mathcal{D}, \mathcal{U}) = \log \mu(\theta) + \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \theta) - \frac{\alpha}{n}\sum_{i=1}^{n} H(Y|\mathbf{x}_i, \theta) - \frac{\tilde{\alpha}}{m}\sum_{j=1}^{m} H(Y|\tilde{\mathbf{x}}_j, \theta). \quad (5)$$

For convenience, we simply set $\alpha = 0$, as additionally minimizing entropy on the labeled training set is unnecessary when we are already maximizing the likelihood of the given labels with highly expressive models. Now, to infer θ given the observed datasets $\mathcal{D}$ and $\mathcal{U}$, the simplified log-density is

$$\log p(\theta|\mathcal{D}, \mathcal{U}) = \log \mu(\theta) + \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \theta) - \frac{\tilde{\alpha}}{m}\sum_{j=1}^{m} H(Y|\tilde{\mathbf{x}}_j, \theta). \quad (6)$$

We note that computing the maximum-a-posteriori (MAP) solution of this model corresponds to simply optimizing model parameters θ that on the supervised objective on the labeled training set in addition a minimum entropy regularizer on the unlabeled input. However, that we should not necessarily expect the MAP solution to provide reasonable uncertainty estimates, as a the model explicitly places more weight on parameters θ that assign confident predictions on the test inputs, and so any single model will likely provide overconfident predictions. Marginalizing the predictions over the posterior distribution over parameters is thus essential to recovering meaningful uncertainty estimates, as the different models we marginalize over, though each being individually confident in predictions, can still express uncertainty when the different models disagree.

Approximate inference for test-time adaptation. We now discuss how to perform inference in the above model in a way suitable for *test-time adaptation*, where we must adapt to the test data without any further access to the original training set. To allow for test-time adaptation, we propose to learn an approximate posterior

$$q(\theta) \approx p(\theta|\mathcal{D}) = \log \mu(\theta) + \sum_{i=1}^{n} \log p(y_i|\mathbf{x}_i, \theta) \quad (7)$$

during training time, and then use this approximate training set posterior $q(\theta)$ in place of the training set when performing inference on the unlabeled test data. This gives us an approximate posterior with log density

$$\log p(\theta|\mathcal{D}, \mathcal{U}) = \log q(\theta) - \frac{\tilde{\alpha}}{m}\sum_{j=1}^{m} H(Y|\tilde{\mathbf{x}}_j, \theta) \quad (8)$$

Unlike the full training set, the learned approximate posterior density $q(\theta)$, which can be as simple as a diagonal Gaussian distribution, can be much easier to store and optimize over for test-time adaptation, and the training time procedure would be identical to any approximate Bayesian method that learns a posterior density. In principle, we can now instantiate test-time adaptation by running any approximate Bayesian inference algorithm, such as variational inference or MCMC, to sample θ's from the density in Equation 8, and average predictions from these samples to compute the marginal probabilities for each desired test label.

Algorithm 1 Bayesian Adaptation for Covariate Shift (BACS)

Input: Ensemble size k, Entropy weight $\tilde{\alpha}$, Training Data: $\mathbf{x}_{1:n}, y_{1:n}$, Test Data: $\tilde{\mathbf{x}}_{1:m}$
Output: Predictive distributions $p(y|\tilde{\mathbf{x}}_j)$ for each test input $\tilde{\mathbf{x}}_j$
Training: For each $i \in (1, \ldots, k)$ compute an approximate density $q_i(\theta) \approx p(\theta|(\mathbf{x}_{1:n}, y_{1:n}))$
Adaptation:
for all ensemble members $i \in (1, \ldots, k)$ **do**
  Compute adapted parameters $\hat{\theta}_i = \arg\max_\theta \frac{\tilde{\alpha}}{m} \sum_{j=1}^{m} -H(Y|\tilde{\mathbf{x}}_j, \theta) + \log q_i(\theta)$
end for
For each test input $\tilde{\mathbf{x}}_i$, marginalize over ensemble $p(y|\tilde{\mathbf{x}}_j) = \frac{1}{k} \sum_{j=1}^{k} p(y|\tilde{\mathbf{x}}_j, \hat{\theta}_i)$.

Practical instantiation with ensembles. As previously mentioned, marginalizing over different models that provide diverse labelings of the test set is crucial to providing uncertainty estimates after adaptation. We thus propose to use an ensembling approach [Lakshminarayanan et al., 2016]. Deep ensembles simply train multiple models from different random initializations, each independently optimizing the target likelihoods, and averages together the models' predicted probabilities at test time. They are able to provide effective approximations to Bayesian marginalization due to their ability to aggregate models across highly distinct modes of the loss landscape [Fort et al., 2019, Wilson and Izmailov, 2020].

Our method, BACS, summarized in Algorithm 1, trains an ensemble of k different models on the training set, each with their approximate posterior $q_i(\theta)$ that captures the local loss landscape around each mode in the ensemble. Then at test time, we independently optimize each of the k models by minimizing Equation 8 (using the corresponding $q_i(\theta)$ for each ensemble member), and then aggregate the predictions across all adapted ensemble members.

3 Related Work

Entropy minimization. Entropy minimization has been used as a self-supervised objective in many settings, including domain adaptation [Saito et al., 2019, Carlucci et al., 2017], semisupervised learning [Grandvalet and Bengio, 2004, Berthelot et al., 2019, Lee and Lee, 2013], and few-shot learning [Dhillon et al., 2015]. Grandvalet and Bengio [2004] proposed a probabilistic model incorporating entropy minimization for semisupervised learning (without distribution shift), but only use the probabilistic model to motivate entropy minimization as a regularizer for a MAP solution in order improve accuracy, which does not capture any epistemic uncertainty. In contrast, we are concerned with test-time adaptation under distribution shift, which requires introducing a model of separate training-time and test-time input distributions, and reliable epistemic uncertainty estimates, which we obtain via Bayesian marginalization. We also devise an approximate inference scheme to allow for adaptation without access to the training data. Test time entropy minimization (TENT) [Wang et al., 2020a] uses entropy minimization as the sole objective to adapt to the test distribution (though without an explicit Bayesian interpretation) and adapts without any further access to the training data, but they only aim to improve accuracy and not uncertainty estimation. Similarly to Grandvalet and Bengio [2004], TENT only learns a single model using entropy minimization, whereas we show that explicitly performing Bayesian inference and marginalizing over multiple models is crucial for effective uncertainty estimation under distribution shift.

Uncertainty estimation under distribution shift (no adaptation). Various methods have been proposed for uncertainty estimation with distribution shift that do not incorporate test time adaptation. Ovadia et al. [2019] measured the calibration of various models under various distribution shift without any adaptation, finding that deep ensembles [Lakshminarayanan et al., 2016] and some other Bayesian methods that marginalize over multiple models perform well compared to methods that try to recalibrate predictions using only the source data. Beyond ensembles, other Bayesian techniques [Dusenberry et al., 2020, Maddox et al., 2019] have also demonstrated improved uncertainty estimation under distribution shift compared to standard models. Various other techniques have been found to improving calibration under distribution shift through significant changes to the training procedure, for example utilizing different loss functions [Padhy et al., 2020, Tomani and Buettner, 2020], extensive data-augmentations [Hendrycks et al., 2019], or extra pre-training [Xie et al., 2019b].

Uncertainty estimation with both train and test distributions. We now discuss prior work that considers uncertainty estimation assuming access to both the train and test data simultaneously. Recent work studying calibration for domain adaptation algorithms [Pampari and Ermon, 2020, Park et al., 2020, Wang et al., 2020b] found that predictions are poorly calibrated in the target domain even if the models were well-calibrated on the labeled source domain. These works all propose methods based on importance weighting between the target domain and the labeled source domain in order to recalibrate target domain predictions using only labels in the source domain. They are not directly applicable in our test-time adaptation setting, since they require estimates of density ratios between target and source distributions, which we cannot compute without access to either a generative model of training distribution, or access to the training data during adaptation. Our method also differs in how we approach uncertainty estimation. Instead of relying on extra labeled data and post-hoc recalibration techniques for classifiers, our method relies on Bayesian inference to provide meaningful uncertainty estimates.

For uncertainty estimation in regression problems, Chan et al. [2020] also propose to adapt Bayesian posteriors to unlabeled data by optimizing the predictive variance of Bayesian neural network at an input to serve as a binary classifier of whether the point is in-distribution or not. thus encouraging higher variance for out-of-distribution points. Their method is again not applicable in our test-time setting because they require access to both the train and test data at once.

Uncertainty estimation with test time adaptation. Nado et al. [2020] evaluate various techniques for uncertainty estimation in conjunction with adapting batch-norm statistics to the shifted test domain, and again find that deep ensembles provide well-calibrated prediction in addition to improved accuracy. While our method similarly utilizes ensembles and adapts batch norm statistics, and we show that additionally adapting via entropy minimization at test time further improves predictive accuracy without sacrificing calibration.

Bayesian semi-supervised methods: Our proposed probabilistic model for covariate shift extends the model Seeger [2000] proposed for incorporating unlabeled data in semi-supervised learning. They use the probabilistic model to motivate regularization for the classifier that depend on the inputs in MAP inference, and do not tackle uncertainty estimation or distribution shift. Gordon and Hernández-Lobato [2020] perform Bayesian semi-supervised learning combining both generative and discriminative models. In contrast, our method does not need to learn a generative model of the data, and explicitly tackles the problem of distribution shift instead of assuming the labeled and unlabeled data come from the same distribution. Another line of work [Ng et al., 2018, Ma et al., 2019, Walker and Glocker, 2019, Liu et al., 2020] propose Bayesian methods for semisupervised learning specialized graph-structured data.

4 Experiments

In our experiments, we aim to analyze how our test-time adaptation procedure in BACS performs when adapting to various types of distribution shift, in comparison to prior methods, in terms of *both* the accuracy of the adapted model, and its ability to estimate uncertainty and avoid over-confident but incorrect predictions. We evaluate our method and prior techniques across a range of distribution shifts, including corrupted datasets, natural distribution shifts, and domain adaptation settings.

Architectures and implementation. For ImageNet [Deng et al., 2009] experiments, we use the ResNet50 [He et al., 2016] model, while for other datasets, we use ResNet26 [He et al., 2016]. For all methods utilizing ensembles, we use ensembles of 10 models, and report results averaged over the same 10 seeds for the non-ensembled methods. While adapting our networks using the entropy loss, we also allow the batch normalization statistics to adapt to the target distribution. To obtain approximate posteriors for each ensemble member, we use SWAG-D [Maddox et al., 2019], which estimates a Gaussian posterior with diagonal covariance from a trajectory of SGD and requires minimal changes to standard training procedures. During adaptation, we initialize each model from the corresponding posterior mean, corresponding to the solution obtained by Stochastic Weight Averaging [Izmailov et al., 2018]. For methods that optimize on the test distribution, we report results after one epoch of adaptation unless otherwise stated.

Comparisons. We compare our method against two state-of-the-art prior methods for test-time adaptation: TENT [Wang et al., 2020a], which minimizes entropy on the test data with a single model, without the additional posterior term accounting for the training set that we use in BACS, and

	CIFAR10-C				CIFAR100-C			
Method	Acc	NLL	Brier	ECE	Acc	NLL	Brier	ECE
Vanilla	60.84	1.775	0.5975	0.2348	35.24	4.264	0.9934	0.3968
Vanilla ensemble	62.37	1.507	0.5341	0.1655	38.84	3.324	0.8440	0.2490
BN Adapt	82.74	0.8420	0.2844	0.1176	58.27	2.246	0.6301	0.2340
Ensemble BN Adapt	86.44	0.4582	0.1985	0.03353	64.83	1.433	0.4717	**0.04956**
TENT (1 epoch)	86.57	0.7248	0.2289	0.09914	64.37	3.265	0.6111	0.2759
TENT (5 epochs)	87.73	0.8804	0.2233	0.1066	63.67	3.475	0.7135	0.3554
BACS (ours) (1 epoch)	88.11	0.4063	0.1754	0.02660	68.84	**1.296**	0.4292	0.06500
BACS (ours) (5 epochs)	**88.87**	**0.3842**	**0.1642**	**0.02541**	**69.22**	1.308	**0.4258**	0.06799

Table 1: **CIFAR-10/100 Corrupted** results at the highest level of corruption, averaged over all corruption types at severity level 5 (the most severe level). With one epoch of adaptation, BACS consistently outperforms all baselines in terms of accuracy, NLL and Brier score, and can further improve with more training. In terms of ECE, BACS performs similarly to ensembles with batchnorm adaptation, and both methods substantially outperform other baselines in uncertainty estimation.

ensembles adapted using the batch norm statistics of the shifted test set, as discussed by Nado et al. [2020]. We also compare to deep ensembles [Lakshminarayanan et al., 2016] without any adaptation as a baseline for uncertainty estimation under distribution shift.

Metrics. For each method, we measure the accuracy of the predictions, and also evaluate uncertainty estimation using the negative log likelihood (NLL), Brier score [Brier, 1950] and expected calibration error (ECE) [Naeini et al., 2015] metrics. NLL and Brier score are both proper scoring rules [Gneiting and Raftery, 2007], and are minimized if and only if the predicted distribution is identical to the true distribution. ECE measures calibration by binning predictions according to the predicted confidence and measuring the difference between the average confidence and empirical accuracy within each bin. ECE intuitively captures how reliably the model's predicted confidence actually reflects the actual accuracy, which may be more meaningful under distributional shift, for example if we would like to measure whether the model's uncertainty actually anticipates potential mistakes (e.g., for applications such as selective classification).

Corrupted Images. We evaluate our method on CIFAR-10-C, CIFAR-100-C, and ImageNet-C [Hendrycks and Dietterich, 2019], where distribution-shifted datasets are generated by applying different image corruptions at different intensities to the test sets of CIFAR10, CIFAR100 [Krizhevsky, 2012], and ImageNet [Deng et al., 2009] respectively. In Table 1, we show comparisons at the most severe level of corruption for CIFAR10-C and CIFAR100-C. For CIFAR10-C, we see that our method is able to outperform all baselines in accuracy, NLL, and Brier score with just one epoch of adaptation, and can further improves with more optimization.

Method	Acc	NLL	Brier	ECE
Vanilla	45.64	2.867	0.6750	**0.0614**
Vanilla ensemble	50.48	2.519	0.6211	0.07878
BN Adapt	55.97	2.363	0.6050	0.1606
Ensemble BN Adapt	62.18	2.137	0.5802	0.2438
TENT	58.05	2.076	0.5529	0.08137
BACS (ours)	**65.44**	**1.702**	**0.4542**	0.1479
BACS (MAP)	59.98	1.927	0.5293	0.06642

Table 2: **ImageNet-C** results averaged over all corruption types and levels. BACS again substantially outperforms all baselines in accuracy, NLL, and Brier scores. Ensemble BN is the strongest baseline in terms of accuracy, and BACS also outperforms it in ECE.

We note that TENT, despite improving accuracy on batchnorm adaptation with a single model, actually has *worse* NLL and ECE, indicating that the predictions become severely overconfident, especially with more test-time optimization as entropy is minimized further. In contrast, BACS is able to consistently improve upon or remain similar to ensemble BN in uncertainty estimation even with more optimization, indicating the importance of combining Bayesian marginalization with test-time adaptation for uncertainty estimation.

In Table 2, we show comparisons on ImageNet-C averaged over all corruption levels and corruption types. BACS is able to significantly outperform all prior methods in terms of accuracy, NLL, and Brier score. We note that for ImageNet-C, ensembling and batch norm adaptation actually *worsen calibration*, as measured by ECE, compared to the single model or non-adapted baselines respectively, despite each technique providing substantial improvements on all other metrics. The baseline models with no adaptation are actually the most calibrated (as measured by ECE), despite performing much worse in other metrics, with the non-ensembled model with no adaptation performing the best in ECE

(though the worst in all other metrics). We further see that an ablation of our method that only uses a single model (BACS (MAP)) outperforms the other non-ensemble methods in accuracy, NLL, and ECE, and achieves lower ECE than all methods but the vanilla model with no adaptation.

CIFAR10 to STL10. We further evaluate all methods on a more natural distribution shift by considering evaluating a model trained CIFAR-10 and evaluated on STL-10 [Coates et al., 2011]. We only use the 9 overlapping classes in each dataset. In this setting, BACS and BN ensembles outperform the other methods.

Method	Acc	NLL	Brier	ECE
Vanilla	82.38	0.7825	0.2886	0.1185
Vanilla Ensemble	84.03	0.5360	0.2333	0.06561
TENT	84.05	0.8831	0.2753	0.1216
BN adapt	83.72	0.7926	0.2709	0.1147
Ensemble BN Adapt	85.40	0.5301	0.2195	**0.05949**
BACS (ours)	85.47	**0.5284**	**0.2184**	0.06114

Table 3: **CIFAR10 to STL10**: Source model trained on CIFAR10 and evaluated on STL10 test set the (with nonoverlapping classes removed during both training and test). Here, the ensembled approaches perform the best overall and perform similarly to one another in uncertainty estimation. Compared to ensembles with BN adaptation, BACS performs similarly, with slightly improved accuracy, NLL, and Brier score and a small increase in ECE.

While TENT is able improves accuracy slightly over the non-ensembled baselines, we again see that uncertainty estimates degrade as entropy is minimized (as seen by the increases in NLL, Brier, and ECE scores). In contrast, BACS still remains well-calibrated, again emphasizing the importance of Bayesian marginalization for reliable uncertainty estimation when adapting models via entropy minimization.

SVHN to MNIST transfer. We also evaluate our method on a digit classification task, transferring a model trained on SVHN [Netzer et al., 2011] to MNIST [LeCun et al., 2010], which is commonly studied as a the domain adaptation setting [Ganin et al., 2015, Liang et al., 2020]. We find that BACS is able to significantly outperform the models with no adaptation as well as ensembles with batch norm adaptation in accuracy, NLL, and Brier score. Compared to TENT, our method improves in NLL and ECE despite having substantially lower accuracies.

We find that this severe distribution shift requires larger changes in parameters to effectively adapt, as seen by the substantial improvements between optimizing for one epoch and optimizing for 10. We also find that approximate posterior term used in our method overly constrains the network during adaptation, preventing BACS from reaching similar accuracy to TENT. We thus also experiment with removing the regularizer from the method (denoted BACS-posterior in the table), and find this method performs the best in all metrics. This again emphasizes the importance of marginalizing over multiple different models for uncertainty estimation.

Method	Acc	NLL	Brier	ECE
Vanilla	76.99	0.9967	0.3645	0.1290
Vanilla ensemble	79.52	0.7368	0.3024	0.03856
BN adaptation	73.43	1.4502	0.4453	0.1878
Ensemble BN adaptation	76.84	0.9444	0.3442	0.07855
TENT (1 epoch)	87.35	0.7708	0.2228	0.1010
TENT (10 epochs)	92.72	0.6554	0.1364	0.0653
BACS (ours) (1 epoch)	85.04	0.6024	0.2285	0.05595
BACS (ours) (10 epochs)	86.56	0.5553	0.2094	0.05133
BACS - posterior (1 epoch)	89.30	0.4214	0.1650	0.02603
BACS - posterior (10 epochs)	**94.07**	**0.2371**	**0.0965**	**0.02404**

Table 4: **SVHN to MNIST**: In this domain adaptation setting, we find adapting batch norm statistics alone hurts performance compared to the unadapated models, but methods utilizing entropy minimization are able to improve substantially in accuracy.

We show that our method, BACS, is able to achieve strong results in accuracy *and* uncertainty estimates across different distribution shifts. Compared to ensembles with batch-norm adaptation, adapting each ensemble member with our regularized entropy minimization objective often leads to substantially improved accuracy, NLL, and Brier scores, while ECE either stays similar or improves. We show that TENT often produces highly overconfident predictions and becomes poorly calibrated, while BACS is able to improve on TENT both in terms of accuracy and uncertainty estimation through the use of Bayesian marginalization. These observations support our hypothesis that framing entropy minimization in the context of a well-defined Bayesian model can lead to significantly more effective test-time adaptation techniques.

Impact of posterior term. In this section, we discuss the effects of using the training set posterior density in our objective (Equation 8) when adapting at test time. Intuitively, the training posterior density ensures that our adapted classifiers, while minimizing entropy on the target distribution, remain constrained to still perform well on the training set and stay near the initial solution found during training. We empirically find that the posterior term can be important for preventing entropy minimization from finding degenerate solutions: adapting the whole network on several ImageNet-C corruptions without the posterior term can lead to poor models that achieve close to 0 accuracy. While TENT, which adapts via entropy minimization without any term accounting for the training data, uses an ad-hoc solution that restricts adaptation to only the learnable scale and shift parameters in the batch norm layers, our use of the training set posterior density is motivated directly from our proposed probabilistic model and does not require any heuristic choices over which parameters should or should not be adapted. In Table 2, we see that an ablation of our method without ensembles, which corresponds simply to the MAP solution in our proposed model, outperforms TENT in all metrics. The difference between the BACS (MAP) and TENT is precisely that BACS (MAP) adapts the whole network, but with an additional regularization term, while TENT adapts only a small part of the network without any regularizer.

However, we find that the approximate posterior density can also be limiting in certain situations, which we can observe in the SVHN to MNIST experiments. Here, there is a large discrepancy between the two domains, and we find that adaptation with the posterior is too limiting and is unable to adjust the parameters enough to achieve as high accuracy on MNIST as TENT. Ablating our method to remove the posterior term however does achieve the best performance in both accuracy and uncertainty estimation, again illustrating the benefits of marginalizing over different models after adaptation.

5 Discussion

We presented Bayesian Adaptation for Covariate Shift (BACS), a Bayesian approach for utilizing test-time adaptation to obtain both improved accuracy and well-calibrated uncertainty estimates when faced with distribution shift. Our method is straightforward to implement, requires minimal changes to standard training procedures, and improves both accuracy and uncertainty estimation for a variety of distribution shifts in image classification.

Limitations and future work. Another limitation of our approach is that it requires effective techniques for estimating the parameter posterior from the training set. While the study of such methods, and Bayesian neural networks more broadly, is an active area of research, it remains a significant practical challenge. Indeed, it is likely that the simple Gaussian posteriors we employ provide a poor estimation of the true posterior and can overly constrain the network during adaptation. Therefore, a particularly relevant direction for future work is to integrate BENT with more sophisticated Bayesian neural network methods.

Another promising direction for future work is to explore other objectives objectives that had success in semi-supervised learning settings, such as consistency based losses [Sajjadi et al., 2016, Miyato et al., 2017, Xie et al., 2019a, Sohn et al., 2020] or information maximization [Gomes et al., 2010], which can be straightforwardly incorporated into our method as suitable priors on the relationship between the data distribution and classifier. More broadly, we hope that our work will spur further research into test-time adaptation techniques based on well-defined Bayesian models that describe how unlabeled test data should inform our posterior estimates of the model parameters, even in the presence of distributional shift.

References

D. Berthelot, G. Research, N. Carlini, I. Goodfellow, A. Oliver, N. Papernot, and C. Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. Technical report, 2019. URL `https://github.com/google-research/mixmatch`.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78 (1), 1 1950. ISSN 0027-0644. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò. AutoDIAL: Automatic DomaIn Alignment Layers. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:5077–5085, 4 2017. URL http://arxiv.org/abs/1704.08082.

A. J. Chan, A. M. Alaa, Z. Qian, and M. Van Der Schaar. Unlabelled Data Improves Bayesian Uncertainty Calibration under Covariate Shift. Technical report, 11 2020. URL http://proceedings.mlr.press/v119/chan20a.html.

A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto. A Baseline for Few-Shot Image Classification. In *International Conference on Learning Representations*. arXiv, 9 2015. URL http://arxiv.org/abs/1909.02729.

M. W. Dusenberry, G. Jerfel, Y. Wen, Y.-a. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, and D. Tran. Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors. *arXiv preprint arXiv:2005.07186*, 2020.

S. Fort, H. Hu, and B. Lakshminarayanan. Deep Ensembles: A Loss Landscape Perspective. *arXiv*, 12 2019. URL http://arxiv.org/abs/1912.02757.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-Adversarial Training of Neural Networks. *Advances in Computer Vision and Pattern Recognition*, 17(9783319583464):189–209, 5 2015. URL http://arxiv.org/abs/1505.07818.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 3 2007. ISSN 01621459. doi: 10.1198/016214506000001437. URL https://www.tandfonline.com/doi/abs/10.1198/016214506000001437.

R. Gomes, A. Krause, and P. Perona. Discriminative Clustering by Regularized Information Maximization. Technical report, 2010. URL http://www.cs.ubc.ca/.

J. Gordon and J. M. Hernández-Lobato. Combining deep generative and discriminative models for Bayesian semi-supervised learning. *Pattern Recognition*, 100:107156, 4 2020. ISSN 00313203. doi: 10.1016/j.patcog.2019.107156.

Y. Grandvalet and Y. Bengio. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 770–778. IEEE Computer Society, 12 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90. URL http://image-net.org/challenges/LSVRC/2015/.

D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *7th International Conference on Learning Representations, ICLR 2019*, 3 2019. URL http://arxiv.org/abs/1903.12261.

D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *arXiv*, 12 2019. URL http://arxiv.org/abs/1912.02781.

P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. In *UAI*, 2018.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 6 2012.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 2017-December:6403–6414, 12 2016. URL http://arxiv.org/abs/1612.01474.

Y. LeCun, C. Cortes, and C. J. Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

D.-h. Lee and D.-h. Lee. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. *ICML Workshop on challenges in representation learning*, 2013. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.664.3543.

J. Liang, D. Hu, and J. Feng. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. *arXiv*, 2 2020. URL http://arxiv.org/abs/2002.08546.

Z.-Y. Liu, S.-Y. Li, S. Chen, Y. Hu, and S.-J. Huang. Uncertainty Aware Graph Gaussian Process for Semi-Supervised Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 4957–4964, 4 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i04.5934. URL www.aaai.org.

J. Ma, W. Tang, J. Zhu, and Q. Mei. A Flexible Generative Framework for Graph-based Semi-supervised Learning. *arXiv*, 5 2019. URL http://arxiv.org/abs/1905.10769.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13132–13143, 2019.

T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 4 2017. URL http://arxiv.org/abs/1704.03976.

Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek. Evaluating Prediction-Time Batch Normalization for Robustness under Covariate Shift. *arXiv*, 6 2020. URL http://arxiv.org/abs/2006.10963.

M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

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

Y. C. Ng, N. Colombo, and R. Silva. Bayesian Semi-supervised Learning with Graph Gaussian Processes. *Advances in Neural Information Processing Systems*, 2018-December:1683–1694, 9 2018. URL http://arxiv.org/abs/1809.04379.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *arXiv*, 6 2019. URL http://arxiv.org/abs/1906.02530.

S. Padhy, Z. Nado, J. Ren, J. Liu, J. Snoek, and B. Lakshminarayanan. Revisiting One-vs-All Classifiers for Predictive Uncertainty and Out-of-Distribution Detection in Neural Networks. *arXiv*, 7 2020. URL http://arxiv.org/abs/2007.05134.

A. Pampari and S. Ermon. Unsupervised Calibration under Covariate Shift. *arXiv*, 6 2020. URL http://arxiv.org/abs/2006.16405.

S. Park, O. Bastani, J. Weimer, and I. Lee. Calibrated Prediction with Covariate Shift via Unsupervised Domain Adaptation. *arXiv*, 2 2020. URL http://arxiv.org/abs/2003.00343.

K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised Domain Adaptation via Minimax Entropy. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:8049–8057, 4 2019. URL http://arxiv.org/abs/1904.06487.

M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. *Advances in Neural Information Processing Systems*, pages 1171–1179, 6 2016. URL http://arxiv.org/abs/1606.04586.

M. Seeger. Input-dependent Regularization of Conditional Density Models. Technical report, 2000. URL https://infoscience.epfl.ch/record/175482.

K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *arXiv*, 1 2020. URL http://arxiv.org/abs/2001.07685.

Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. 9 2019. URL http://arxiv.org/abs/1909.13231.

R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring Robustness to Natural Distribution Shifts in Image Classification. *arXiv*, 7 2020. URL http://arxiv.org/abs/2007.00644.

C. Tomani and F. Buettner. Towards Trustworthy Predictions from Deep Neural Networks with Fast Adversarial Calibration. 12 2020. URL http://arxiv.org/abs/2012.10923.

I. Walker and B. Glocker. Graph Convolutional Gaussian Processes. In *International Conference on Machine Learning*, 5 2019. URL http://arxiv.org/abs/1905.05739.

D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization. *arXiv*, 6 2020a. URL http://arxiv.org/abs/2006.10726.

X. Wang, M. Long, J. Wang, and M. I. Jordan. Transferable Calibration with Lower Bias and Variance in Domain Adaptation. *arXiv*, 7 2020b. URL http://arxiv.org/abs/2007.08259.

A. G. Wilson and P. Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *arXiv*, 2 2020. URL http://arxiv.org/abs/2002.08791.

Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. Unsupervised Data Augmentation for Consistency Training. *arXiv*, 4 2019a. URL http://arxiv.org/abs/1904.12848.

Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with Noisy Student improves ImageNet classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 11 2019b. URL http://arxiv.org/abs/1911.04252.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our contributions are in improving algorithms for robustness and adaptation, rather than any specific application. We expect the broader impacts of our work to be similar to those of prior works that are concerned with robustness and adaptation, and do not see any negative societal impacts specific to our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Included in supplemental materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See appendix A.1
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See expanded results in appendix B.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See appendix A.1
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All datasets are cited in the text of the experiments section.
   (b) Did you mention the license of the assets? [Yes] See appendix A.3.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use only standard, widely used datasets in our experiments.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]