Learning Priors for Invariance

Eric Nalisnick **Padhraic Smyth**

Department of Computer Science
University of California, Irvine
{enalisni, p.smyth}@uci.edu

Abstract

Informative priors are often difficult, if not impossible, to elicit for modern large-scale Bayesian models. Yet, often, some prior knowledge is known, and this information is incorporated via engineering tricks or methods less principled than a Bayesian prior. However, employing these tricks is difficult to reconcile with principled probabilistic inference. For instance, in the case of data set augmentation, the posterior is conditioned on artificial data and not on what is actually observed. In this paper, we address the problem of how to specify an informative prior when the problem of interest is known to exhibit invariance properties. The proposed method is akin to posterior variational inference: we choose a parametric family and optimize to find the member of the family that makes the model robust to a given transformation. We demonstrate the method's utility for dropout and rotation transformations, showing that the use of these priors results in performance competitive to that of non-Bayesian methods. Furthermore, our approach does not depend on the data being labeled and thus can be used in semi-supervised settings.

1 INTRODUCTION

Bayesian inference is characterized by its ability to naturally incorporate existing information. This information is encoded by the prior distribution, and its specification is widely regarded as "the most important step" in the Bayesian approach given that it can "drastically alter the subsequent inference" [30]. Unfortunately, for modern large-scale machine learning models, setting the prior based on existing knowledge is often hard, if not impossible, due to limitations in human intuition. Modelers can have difficulty reasoning about how parameters behave in high dimensions and translating abstract concepts into formal probability distributions. As an alternative, the modeler must resort to specifying a diffuse, noninformative prior with the hope that the data can overwhelm any pathology the arbitrary prior may introduce.

Nonetheless, the modeler often knows some prior information that is essential for obtaining good performance, and it is common to incorporate this knowledge via 'engineering tricks' or methods less principled than Bayesian inference. For example, achieving state-of-the-art performance on image classification frequently requires data set augmentation [15]: creating new training instances by flipping, scaling, rotating, etc. the original images [2]. Another example is using feature dropout on bag-of-words representations to simulate the effect of varying a document's length [35]. Training a model under these stochastic augmentation or perturbation strategies is in effect inducing a prior, one that encourages robustness with respect to these known invariances.

While methods for including prior information via means other than the likelihood or prior are undoubtedly highly effective in practice, they are difficult to reconcile with principled probabilistic inference. One problem is that the resulting Bayesian posteriors are conditioned on an artificial training set, not on what is truly observed. Another issue is that, as is the case when training with dropout, it is unclear how to interpret the regularization mechanism: are the masking variables model parameters and if so should we be computing their posterior? These and related questions motivate recent work on formulating dropout as (approximate) Bayesian inference [10, 11, 20].

In this paper, we propose a method for transferring

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

a modeler's knowledge about invariances into a corresponding Bayesian prior. Doing so allows data set augmentation, dropout, and other effective regularization strategies to be incorporated into the model as a proper Bayesian prior. Once this is done, Bayesian inference can proceed as usual without complication of or the need to re-interpret the inference strategy (whatever it may be: Markov Chain Monte Carlo, variational inference, maximum a posteriori estimation, etc.).

Our proposed approach is to formulate a variational problem [5]: given a parameteric family, find the member of the family that, when used as a prior, makes the model as near to invariant as possible. To do this, we first derive a lower bound that quantifies the model's invariance under some specific perturbation process. We then maximize this bound with respect to the parameters of the parameteric family. An important detail to note is that we are *not* performing empirical Bayesian inference. Rather, we learn the prior from the data *model*, similarly to how objective priors are specified [17, 4, 26]. For supervised models, this means that only the features are needed, making our method well suited for semi-supervised settings, as the experiments demonstrate.

2 PRELIMINARIES

Before describing the proposed methods, we begin by defining perturbation processes and invariant statistical models. We use the following notation throughout the paper. As our primary focus is on supervised learning, we denote input features as $\mathbf{x}_i \in \mathbb{R}^d$ and labels (indicating class membership or a real-valued response) as y_i, where i indexes the observed data. Define the data model (likelihood function) to be $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ are the model parameters. Thus, in the Bayesian setting, $p(\boldsymbol{\theta})$ denotes the prior and $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X})$ the posterior. We write all expectations, entropies, and divergences in their continuous form (i.e. with integrals), but sums should be used when the support is discrete.

2.1 Perturbation Processes

Many of the recent successes in supervised machine learning have come from data augmentation and corruption processes that perturb observations and parameters. These processes have the effect of regularizing the classifier to which they are applied by implicitly encoding user knowledge. We define them formally and generally as follows. Call a generative process that takes in a random variable $\mathbf{z} \in \mathcal{Z}$ and samples a random transformation $\tilde{\boldsymbol{z}} \in \tilde{\mathcal{Z}}(\boldsymbol{z})$ a *perturbation process* (PP):

$$\tilde{\mathbf{z}} \sim q(\tilde{\mathbf{z}}; \mathbf{z}, \boldsymbol{\zeta}) \tag{1}$$

where $\mathbf{z}$ denotes the random variable pre-transformation, $\tilde{\mathbf{z}}$ denotes the same variable post-transformation, and $\boldsymbol{\zeta}$ are the parameters of q. Below we describe dropout and rotation as a PP acting on the features (i.e. $\mathbf{z} = \mathbf{x}$).

Dropout. Dropout corruption—where elements of the data or model parameters are set to zero at random—has been observed to consistently improve the held-out performance of logistic regression [36, 25] and deep neural networks [33]. In this paper we focus on feature dropout, which can be written as a PP as follows:

$$\tilde{\mathbf{x}} = \boldsymbol{b} \odot \mathbf{x} \quad \text{where} \quad \boldsymbol{b} \sim \text{Bernoulli}(1 - \zeta) \tag{2}$$

where $\odot$ denotes an element-wise product and $\zeta \in (0, 1)$ the dropout probability. The random variable $\boldsymbol{b}$ acts, simply, as a element-wise mask on the feature vector $\boldsymbol{x}$.

Rotation. As mentioned in the Introduction, many image data sets exhibit rotations, and classifier performance can be improved by augmenting the data set with rotated version of the true observations. As a PP, 2D rotation can be written as

$$\tilde{\mathbf{x}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3}$$

where $\phi \sim \text{Uniform}(\zeta \in [0, 2\pi])$. Usually padding or some other standardization is used to keep the image size consistent.

In this paper, we focus on dropout and rotation transformations, illustrating the proposed technique for pointwise and affine transformations, respectively. Applying the techniques to other operations in these classes would proceed in a similar manner. The techniques we propose can also be applied to just about any functional transform as long as the parameterized prior is sufficiently expressive.

2.2 Invariant Models

Invariant statistical models have been well studied, both in theory [9] and in practice [8, 31]. The classic formulation is group theoretic, as in Eaton (1989) [9]. We use a similar definition except that we require invariance with respect to all members of a PP's support, which may not form a proper algebraic group.

Definition 2.1. *Let $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}(\mathbf{x})$ be a realization from a perturbation process $q(\cdot; \boldsymbol{\zeta})$ acting on $\mathbf{x} \in \mathcal{X}$, and let $\mathcal{P}_{y|x}$ be a family of models indexed by their parameters. A statistical model $p \in \mathcal{P}_{y|x}$ is $q_{\boldsymbol{\zeta}}$-invariant if*

$$p(y|\mathbf{x}) \quad = \quad p(y|\tilde{\mathbf{x}}) \quad \forall\ \tilde{\mathbf{x}} \in \tilde{\mathcal{X}}(\mathbf{x}).$$

Intuitively, this invariance property can be thought of as robustness: a dropout-invariant classifier, for instance, should produce the same output distribution no matter how the input features are corrupted. In the case of the usual Bernoulli(0.5) noise, however, it is unlikely a classifier could be meaningfully dropout-invariant since the probability that all features will be masked is non-zero.

3 LEARNING INVARIANT PRIORS

Having introduced PPs and defined model invariance, we next detail the proposed methodology. We begin by proposing a quantity representing a 'nearness' to invariance and then discuss how to minimize such a quantity with respect to the model's prior.

3.1 Quantifying Approximate Invariance

Recall that our goal is to learn a prior that prefers invariance, and thus we need some continuous quantity that represents how near to invariant a model is. Definition 2.1 is not appropriate as is, because it would require the equality be checked for all $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}(\mathbf{x})$. Instead, we consider the expectation of the model under q, which is also invariant:

Corollary 3.0.1. *If $p \in \mathcal{P}$ is $q_{\boldsymbol{\varsigma}}$-invariant, then $\mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]$ is $q_{\boldsymbol{\varsigma}}$-invariant:*

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})] &= \int_{\tilde{\mathcal{X}}} p(y|\tilde{\mathbf{x}})\ q(\tilde{\mathbf{x}};\mathbf{x})\ d\tilde{\mathbf{x}} \\ &= p(y|\mathbf{x}) \int_{\tilde{\mathcal{X}}} q(\tilde{\mathbf{x}};\mathbf{x})\ d\tilde{\mathbf{x}} = p(y|\mathbf{x}).\end{aligned}$$

This fact is useful for quantifying nearness to invariance because it weights $p(y|\tilde{\mathbf{x}})$ over $\tilde{\mathcal{X}}$, meaning that a lack of invariance for a particular $\tilde{\mathbf{x}}$ can be excused or neglected if $q(\tilde{\mathbf{x}};\mathbf{x})$ is near zero. Thus, quantifying the degree of invariance of a model reduces to computing some divergence between $p(y|\mathbf{x})$ and $\mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]$. We use the Kullback-Leibler divergence—$\mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ \mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]\ ]$—which is zero if and only if $p(y|\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]$ almost everywhere and is positive otherwise. Since $\mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]$ will be intractable for most models of interest, we use the following upper bound on the divergence so that we can obtain an unbiased Monte Carlo approximation of the expectation, obtaining an upper bound via Jensen's inequality:

$$\begin{aligned}&\mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ \mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]\ ] \\ &= \mathbb{E}_{p(y|\mathbf{x})}[\log p(y|\mathbf{x})] - \mathbb{E}_{p(y|\mathbf{x})}[\log \mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]] \\ &\leq \mathbb{E}_{p(y|\mathbf{x})}[\log p(y|\mathbf{x})] - \mathbb{E}_{p(y|\mathbf{x})}\mathbb{E}_{q_{\boldsymbol{\varsigma}}}[\log p(y|\tilde{\mathbf{x}})] \\ &= \mathbb{E}_{q_{\boldsymbol{\varsigma}}}\mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ p(y|\tilde{\mathbf{x}})\ ].\end{aligned} \tag{4}$$

3.2 Exposing the Prior

We now discuss how to introduce Bayesian thinking into our formulations of invariance. Consider the aforementioned models as marginal likelihoods: $p(y|\mathbf{x}) = \int_{\boldsymbol{\Theta}} p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{E}_{p(\boldsymbol{\theta})}[p(y|\mathbf{x},\boldsymbol{\theta})]$. Looking ahead, our ultimate goal is to optimize Equation 4 with respect to $p(\boldsymbol{\theta})$. Ideally we would do this in its current marginalized form, but computing the marginal likelihood is notoriously difficult, even for relatively simple models. Hence, we again upper bound the divergence, which in turn makes the quantity amenable to an unbiased Monte Carlo approximation:

$$\begin{aligned}&\mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ p(y|\tilde{\mathbf{x}})\ ] \\ &\quad= \mathrm{KLD}[\ \mathbb{E}_{p(\boldsymbol{\theta})}[p(y|\mathbf{x},\boldsymbol{\theta})]\ ||\ \mathbb{E}_{p(\boldsymbol{\theta})}[p(y|\tilde{\mathbf{x}},\boldsymbol{\theta})]\ ] \\ &\quad\leq \mathbb{E}_{p(\boldsymbol{\theta})}\mathrm{KLD}[\ p(y|\mathbf{x},\boldsymbol{\theta})\ ||\ p(y|\tilde{\mathbf{x}},\boldsymbol{\theta})\ ].\end{aligned} \tag{5}$$

The bound follows directly from the fact that KLD is a convex function over the domain of probability distributions. With this upper bound, we expose $p(\boldsymbol{\theta})$ and make it accessible for optimization.

3.3 Optimization Objective

Let $\boldsymbol{\lambda}$ denote the parameters of the prior $p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$. We propose optimizing $\boldsymbol{\lambda}$ by the following objective, which is formed by combining Equations 4 and 5 with an entropy term:

$$\begin{aligned}&\mathcal{L}^*(\boldsymbol{\lambda};\mathbf{x}) \\ &\quad= \mathbb{H}_{\boldsymbol{\lambda}}[\boldsymbol{\theta}] - \mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ \mathbb{E}_{q_{\boldsymbol{\varsigma}}}[p(y|\tilde{\mathbf{x}})]\ ] \\ &\quad\geq \mathbb{H}_{\boldsymbol{\lambda}}[\boldsymbol{\theta}] - \mathbb{E}_{q_{\boldsymbol{\varsigma}}}\mathrm{KLD}[\ p(y|\mathbf{x})\ ||\ p(y|\tilde{\mathbf{x}})\ ] \\ &\quad\geq \mathbb{H}_{\boldsymbol{\lambda}}[\boldsymbol{\theta}] - \mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\varsigma}}}\mathrm{KLD}[\ p(y|\mathbf{x},\boldsymbol{\theta})\ ||\ p(y|\tilde{\mathbf{x}},\boldsymbol{\theta})\ ] \\ &\quad= \mathcal{J}(\boldsymbol{\lambda};\mathbf{x})\end{aligned} \tag{6}$$

where $\mathbb{H}_{\boldsymbol{\lambda}}[\boldsymbol{\theta}] = -\int_{\boldsymbol{\theta}} p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})\ \log p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})\ d\boldsymbol{\theta}$. We assume the objective is optimized under the empirical distribution of feature observations, i.e. $\mathbb{E}_{p(\hat{\mathbf{x}})}[\mathcal{J}(\boldsymbol{\lambda};\mathbf{x})] = \frac{1}{N}\sum_i \mathcal{J}(\boldsymbol{\lambda};\mathbf{x}_i)$. Maximizing $\mathcal{J}$ w.r.t. $\boldsymbol{\lambda}$ means that we are finding the distribution that minimizes the expected divergence between the unperturbed and perturbed model—or in other words, the invariance—under the prior. We emphasize that this objective does not depend on the observed y's, only the model output distribution over y. Because of this fact we can use unlabeled feature observations during learning of the prior.

The inclusion of the entropy term in Equation 6 is motivated by the *principle of maximum entropy*, i.e., that the appropriate distribution for representing prior beliefs is one that obeys known constraints and has maximum entropy otherwise [16, 34]. This behavior is precisely what Equation 6 encourages; the first term encourages maximum entropy and the second imposes

the invariance constraints. In practice, the entropy term encourages the prior to avoid spurious solutions. For example, a neural network could become dropout-invariant by learning as the prior a delta function at zero. We will show this phenomenon analytically for linear regression in Section 4.

Equation 6 is amenable to a wide range of parametric forms for the prior. For example, it supports mixture densities $p_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_{\boldsymbol{\lambda}_k}(\boldsymbol{\theta})$ where $p_{\boldsymbol{\lambda}_k}$ is the kth component with parameters $\boldsymbol{\lambda}_k$ and π_k is the corresponding mixture weight. When using a mixture for the prior, the divergence component of the objective can be written as $\mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\zeta}}(\boldsymbol{\zeta})}\mathrm{KLD}[p_{\boldsymbol{\theta}}||p_{\boldsymbol{\theta}}(\boldsymbol{\zeta})] = \sum_k \pi_k \mathbb{E}_{p_{\boldsymbol{\lambda}_k}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\zeta}}(\tilde{\mathbf{x}};\boldsymbol{\zeta})}\mathrm{KLD}[p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i,\boldsymbol{\theta})||p_{\boldsymbol{\theta}}(y_i|\tilde{\mathbf{x}}_i,\boldsymbol{\theta})]$, where the objective is evaluated under each component distribution and a weighted average taken according to the mixture weights.

4 ANALYTICAL SOLUTION FOR LINEAR REGRESSION

To build intuition and to further examine the proposed objective (Equation 6), we next show an analytical solution for linear regression under dropout noise and its connection to the popular *g-prior* [12]. We use the unbiased form of dropout, meaning $\mathbb{E}[\tilde{\mathbf{x}}] = \mathbf{x}$ and $\mathrm{Var}[\tilde{\mathbf{x}}] = \frac{1}{1-\zeta}\mathbf{x}^2$ [25], and we set the prior to be a multivariate normal $p_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\lambda}}, \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}))$ with diagonal covariance matrix. Define the data model to be a standard linear model with Gaussian error: $y = \mathbf{x}^T\boldsymbol{\theta} + \epsilon_0$, $\epsilon_0 \sim \mathrm{N}(0, \sigma_0^2)$. The divergence portion of the objective simplifies to:

$$\begin{aligned}
&\mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\zeta}}}\mathrm{KLD}[\ p(y|\mathbf{x},\boldsymbol{\theta})\ ||\ p(y|\tilde{\mathbf{x}},\boldsymbol{\theta})\ ] \\
&\quad= \mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\zeta}}}\left[\frac{(\mathbf{x}^T\boldsymbol{\theta} - \tilde{\mathbf{x}}^T\boldsymbol{\theta})^2}{2\sigma_0^2}\right] \\
&\quad= \mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\left[\frac{(\mathbf{x}^T\boldsymbol{\theta})^2}{2\sigma_0^2(1-\zeta)}\right] \\
&\quad= \frac{(\mathbf{x}^T\boldsymbol{\mu}_{\boldsymbol{\lambda}})^2 + \mathbf{x}^T\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}\mathbf{x}}{2\sigma_0^2(1-\zeta)}
\end{aligned} \tag{7}$$

If the proposed objective consisted of only the divergence (invariance) term, minimizing the equation above would clearly lead to both $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$ being set to zero. In other words, the optimal prior would be $p_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) = \boldsymbol{\delta}_0$, the delta function placed at zero.

The solution becomes much more interesting when the entropy term is included. The full objective can be written as:

$$\mathcal{J}_{\mathrm{LR}}(\boldsymbol{\lambda};\mathbf{x}) = \log\det(\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}) - \frac{(\mathbf{x}^T\boldsymbol{\mu}_{\boldsymbol{\lambda}})^2 + \mathbf{x}^T\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}\mathbf{x}}{2\sigma_0^2(1-\zeta)}. \tag{8}$$

Since the entropy term does not include the prior's mean, the optimal solution for this parameter is still $\boldsymbol{\mu}_{\boldsymbol{\lambda}} = \mathbf{0}$. Differentiating $\mathcal{J}_{\mathrm{LR}}$ with respect to $\sigma_{\boldsymbol{\lambda}}$, the optimal covariance matrix is $\sigma_0^2(1-\zeta)\mathrm{diag}(\mathbf{x}^T\mathbf{x})^{-1}$. Putting these together we obtain the final solution for the prior: $p^*_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) = \mathrm{N}(\mathbf{0}, \sigma_0^2(1-\zeta)\mathrm{diag}(\mathbf{x}^T\mathbf{x})^{-1})$.

Interestingly, the solution is equivalent to a diagonalized version of the well-known *g-prior* [12]—$\mathrm{N}(\mathbf{0}, g(\mathbf{x}^T\mathbf{x})^{-1})$—with g set by the dropout level. The g-prior has the nice property that the posterior mean is a linear combination of the prior mean and maximum likelihood estimator: $\boldsymbol{\theta}_{\mathrm{post}} = \frac{g}{1+g}\boldsymbol{\theta}_{\mathrm{MLE}} + (1 - \frac{g}{1+g})\boldsymbol{\mu}_{\lambda}$. Thus, in the case of the prior learned by our proposed method, we see the dropout rate plays the role of multiplicative shrinkage of the ML solution.

5 BLACK-BOX LEARNING FOR INTRACTABLE MODELS

For most problems of interest we will not be able to analytically solve the objective's required integrals, as in the previous section. Hence, in this section we describe how to make learning derivation-free and 'black-box' using recently developed techniques from posterior variational inference [28, 19]. Specifically, we use Monte Carlo approximations combined with differentiable non-centered parameterizations [19] to make learning fully gradient-based no matter how complicated the likelihood function is. We also discuss how to use what we call an 'implicit prior'—a highly expressive functional sampler.

Monte Carlo Expectations. For most modern, large-scale models, computing the expectations w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ will not be feasible analytically. Thus we turn to a nested Monte Carlo (MC) approximation:

$$\begin{aligned}
&\mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_{\boldsymbol{\zeta}}}\mathrm{KLD}[p(y|\mathbf{x},\boldsymbol{\theta})||p(y|\tilde{\mathbf{x}},\boldsymbol{\theta})] \\
&\approx \frac{1}{SM}\sum_{s=1}^{S}\sum_{m=1}^{M}\mathrm{KLD}[p(y|\mathbf{x},\hat{\boldsymbol{\theta}}_s)||p(y|\hat{\tilde{\mathbf{x}}}_m,\hat{\boldsymbol{\theta}}_s)]
\end{aligned} \tag{9}$$

such that M samples are drawn from the perturbation process $\hat{\tilde{\mathbf{x}}}_m \sim q(\tilde{\mathbf{x}};\boldsymbol{\zeta})$ and S samples are drawn from the prior we wish to learn $\hat{\boldsymbol{\theta}}_s \sim p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})$.

Differentiable Sampling. Using MC approximations makes computing derivatives w.r.t. the prior's parameters $\boldsymbol{\lambda}$ difficult, as they need to be computed through the samples $\hat{\boldsymbol{\theta}}_s$:

$$\begin{aligned}
&\frac{\partial}{\partial\boldsymbol{\lambda}}\sum_{s=1}^{S}\sum_{m=1}^{M}\mathrm{KLD}[p(y|\mathbf{x},\hat{\boldsymbol{\theta}}_s)||p(y|\hat{\tilde{\mathbf{x}}}_m,\hat{\boldsymbol{\theta}}_s)] = \\
&\sum_{s=1}^{S}\sum_{m=1}^{M}\frac{\partial}{\partial\hat{\boldsymbol{\theta}}_s}\mathrm{KLD}[p(y|\mathbf{x},\hat{\boldsymbol{\theta}}_s)||p(y|\hat{\tilde{\mathbf{x}}}_m,\hat{\boldsymbol{\theta}}_s)]\frac{\partial\hat{\boldsymbol{\theta}}_s}{\partial\boldsymbol{\lambda}}.
\end{aligned} \tag{10}$$

One way we can ensure $\frac{\partial \hat{\boldsymbol{\theta}}_s}{\partial \boldsymbol{\lambda}}$ is computable is by sampling $\boldsymbol{\theta}$ by way of a *differentiable non-centered parameterization* [19] (DNCP), which has the general form $\hat{\boldsymbol{\theta}} = g(\boldsymbol{\lambda}, \hat{\boldsymbol{\epsilon}})$ where $\hat{\boldsymbol{\epsilon}} \sim p(\boldsymbol{\epsilon})$. $\boldsymbol{\epsilon}$ is an auxiliary variable drawn from some fixed distribution and g is a differentiable function. A well-known example of a DNCP is the Gaussian's location-scale form $\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \hat{\boldsymbol{\epsilon}}$ where $\boldsymbol{\epsilon}$ is drawn from a standard Normal distribution.

Implicit Priors. Notice that when using MC approximations of the integrals (Equation 9), the only term in Equation 6 that requires the prior's density be evaluated is the entropy term. Thus, using a nonparametric estimate for $\mathbb{H}[\boldsymbol{\theta}]$ [3] can completely remove the need to evaluate the prior. Doing so allows us to use what we call an *implicit* prior: a prior from which we can draw samples but which we cannot evaluate as a density function, i.e. $\hat{\boldsymbol{\theta}} = f(\boldsymbol{\lambda}, \hat{\boldsymbol{\epsilon}})$ where $\hat{\boldsymbol{\epsilon}}$ is a sample drawn from some fixed distributions and $f(\cdot)$ is some differentiable, sufficiently flexible function such as a neural network. Treating the prior as a simulator in this way is similar to the ideas behind *Generative Adversarial Networks* [13] and *Variational Programs* [29]. The benefit of using an implicit prior is that we can have an extremely flexible distribution over $\boldsymbol{\theta}$; the downside is that we will eventually need to evaluate the implicit prior's density—possibly having to turn to nonparametric density estimation.

6 RELATED WORK

The closest work to what we propose in this paper is prior work on the definition and specification of objective priors [17, 4, 26]. We say that not because our method learns noninformative priors—quite the opposite—but because the method we propose here learns a prior based on the data model, just as objective priors do. For conditional models, the feature variables must be included to define the model, and thus, the prior is dependent on the observed data. This fact links our method (and objective priors) to empirical Bayesian inference [6]. However, a significant difference between our method and empirical Bayesian methods is that the variable being modeled (the classification label or regression response) is not considered in our prior's specification, as it typically is for most empirical Bayesian methods [6].

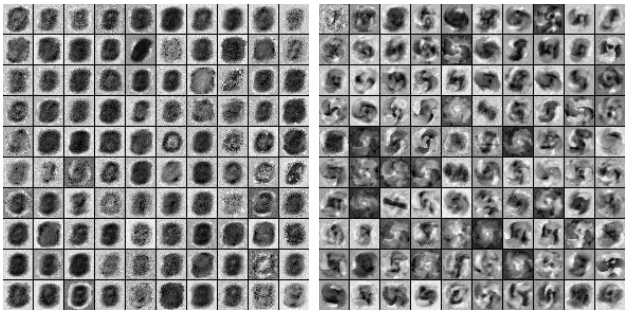
As for the specifics of the proposed optimization objective, its form is motivated by the principle of maximum entropy, which has a long history dating back to statistical physics [16, 14]. There has been some work on learning invariant maximum entropy distributions [27] and approximations to such distributions [23], but this previous work is tailored to specific settings (soil analysis and pairwise moment mean parameters, respectively). In contrast, our approach requires only samples from the perturbation and the distribution to be estimated (the prior).

More closely related is the work of Bachman et al. on *pseudo-ensemble agreement* regularization [1]. They propose a regularization penalty of the form: $\mathcal{R}(\boldsymbol{\theta}) = \mathbb{E}_{x \sim p_x} \mathbb{E}_{\xi \sim p_\xi} \mathcal{V}[f_{\boldsymbol{\theta}}(x), f_{\boldsymbol{\theta}}(x; \xi)]$, where the first expectation is with respect to the empirical distribution of the features, the second expectation is with respect to a noise process (such as dropout corruption [33]), and $\mathcal{V}[\cdot, \cdot]$ is some way to measure the discrepancy between the unperturbed and pertrubed model f_θ. The divergence term we propose in Equation 6 is a special case of Bachman et al.'s penalty: Equation 6 can be obtained by setting $\mathcal{V}[\cdot, \cdot]$ to be KLD (as Bachman et al. do in some experiments) and adding an expectation over the model parameters. The key difference between Bachman et al.'s work and what we propose here is that they use their regularization term within a penalized likelihood framework. There is no concept of learning a Bayesian prior nor one of transferring the stochastic regularization into a probability distribution.

Lastly, we note that this work has been inspired by recent efforts to analyze dropout both from the perspectives of penalized likelihood [1, 36, 37] and approximate Bayesian inference [20, 10, 11]. In the former category, Bachman et al. [1], Wager et al. [36], and Wang & Manning [37] carry out analyses of linear regression that are similar to that in Section 4. However, their analyses are motivated by seeking a closed-form regularization penalty that mimics the effect of dropout. There are no notions of Bayesian priors, and our development of the connection to the g-prior is new. In the latter category, Kingma et al. [20] and Gal & Ghahramani [10, 11] show that dropout can be interpreted as approximate Bayesian inference under certain variational posteriors. Our work has similar motivations—that is, to link dropout and Bayesian methodology—but we do so via the Bayesian prior. We formulate the prior that corresponds to dropout thus allowing inference to proceed with no constraints and by way of either MCMC or variational methods.

7 EXPERIMENTS

In this section we carry out empirical analyses of the proposed methods, focusing on dropout and rotation transformations. First, we discuss some qualitative properties of the learned priors by visualizing them as weight filters. Second, we quantitatively analyze the degree of invariance of several distributions with respect to dropout and rotation perturbations. And lastly, we perform classification tasks to demonstrate



(a) Dropout Invariant (b) Rotation Invariant

Figure 1: *Weight Visualization.* Above we show filter visualizations for 100 weight matrices sampled from two learned implicit priors, one invariant to dropout and one invariant to rotation. Both were trained on MNIST. The dropout invariant prior can be seen to down-weight features found around the center of the image, which is where the active features usually are found. The rotation invariant prior learns spiral feature transformations roughly similar to some of the features learned by Toroidal Subgroup Analysis (see Figure 3 in [7]).

that using the proposed invariant priors results in performance on par with non-Bayesian methods. The multi-class classification experiments use neural network likelihoods of the form $y_i \sim \text{Multinoulli}(\mathbf{p} = \gamma_{\boldsymbol{\theta}}(\mathbf{x}_i))$ where $\gamma_{\boldsymbol{\theta}}(\mathbf{x}_i) = \text{softmax}(\mathbf{h}_L \boldsymbol{\theta}_{L+1})$, the softmax output of one or more neural network layers with the form $\mathbf{h}_{l+1} = \text{ReLU}(\mathbf{h}_l \boldsymbol{\theta}_{l+1})$, where $\mathbf{h}_0 = \mathbf{x}_i$. The binary classification (sentiment analysis) experiment uses a logistic regression likelihood of the form $y_i \sim \text{Bernoulli}(p = \text{logistic}(\mathbf{x}_i \boldsymbol{\theta}))$.

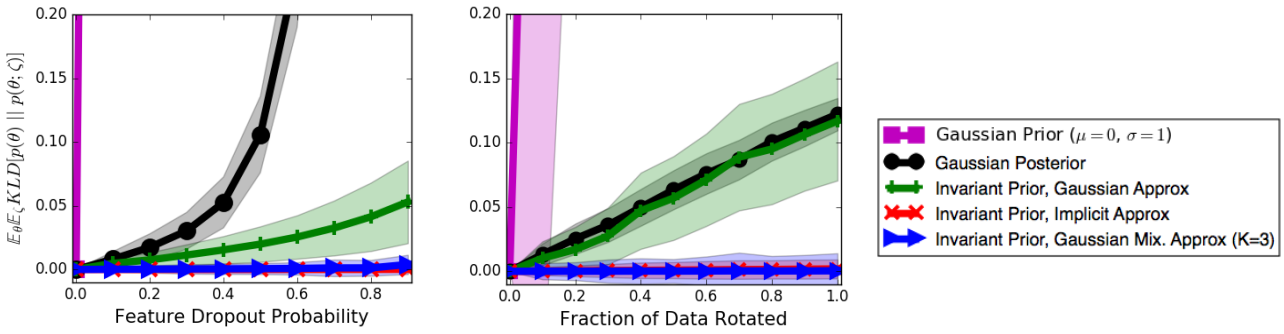
Regarding hyperparameter selection, Adam [18] was used for all experiments with a learning rate chosen from $\{.001, .0005, .0001, .00005\}$ via a validation set (other parameters kept at Tensorflow defaults). For the Monte Carlo approximations used to learn the invariant priors, 50 samples were used for both the parameters and perturbation process. The best priors were selected based on those which obtained the highest value of Equation 6 upon convergence. All posteriors were obtained via *Stochastic Gradient Variational Bayes* [21], and the posterior mean was used to calculate test performance in all cases.

Qualitative Analysis. We begin by performing visual inspection of the invariant priors. We do this by learning an implicit prior (one-hidden-layer neural network, 1000 hidden units) for the two weight matrices of a one-hidden-layer neural network with 500 hidden units. We trained the prior on the MNIST data set under dropout and rotation perturbations (separately). Samples from the prior on the first layer weights are shown in Figure 1. Subfigure (a) shows filter samples from the prior learned under dropout noise. The weights near the center of the image are conspicuously lower (i.e. darker) than those on the edges. This is expected, as placing low-weights on frequently active features reduces the effect of dropping out those features. Wager et al. come to a similar conclusion: dropout penalizes the weights of rare features less harshly than it does those of common features [36]. Subfigure (b) shows the filter samples learned under rotation perturbations. We see they exhibit spiral transformations, which is expected since being rotation invariant would require that features similar distances from the image center receive near equal weight.

Quantitative Analysis. Next we quantitatively analyze the invariance properties of the priors. We quantify invariance based on the the proposed objective's KLD term, i.e. $\mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})} \mathbb{E}_{q_{\boldsymbol{\zeta}}} \text{KLD}[\, p(y|\mathbf{x}, \boldsymbol{\theta}) \,||\, p(y|\tilde{\mathbf{x}}, \boldsymbol{\theta}) \,]$. We calculate this quantity by drawing a sample from the prior, drawing a sample perturbation, and computing the KLD between the unperturbed and perturbed models with the sampled parameters and perturbation. We repeat the process 500 times and average the runs to obtain the final result. Again, the model we used for the experiment was a one-hidden-layer neural network (500 hidden units) and the data set was MNIST undergoing dropout and rotation perturbations.

We trained three forms of invariant priors—a factorized Gaussian (green), a three-component mixture of factorized Gaussians (blue), and an implicit prior (red) parameterized by a one-hidden-layer neural network (1000 hidden units)—and compare them to a standard Normal prior (pink) and a factorized Gaussian posterior (black) in Figure 2. The Gaussian posterior was obtained by training the network on MNIST with stochastic perturbations sampled for each forward pass. We see that, in the case of dropout (Subfigure a), all learned priors are markedly more robust to dropout noise than the two Gaussian baselines. The Gaussian mixture and implicit priors remain invariant at even a high noise level (> 0.8), showing only a slight upward trend. In the case of rotation (Subfigure b), the factorized Gaussian posterior and invariant prior have nearly identical invariance, but again the implicit and mixture invariant priors are notably robust across all perturbation levels.

Fully-Supervised Classification. Next we report results on (fully) supervised classification experiments on the rotated MNIST data set [22], which consists of 12,000 training images and 50,000 test images. 2,000 images were used as a validation set and recombined into the training set to obtain the final test perfor-



(a) Feature Dropout (b) Rotation

Figure 2: *Invariance vs Perturbation Magnitude.* The plots above shows the robustness of several distributions (y-axis shows $\mathbb{E}_{p_{\boldsymbol{\lambda}}(\boldsymbol{\theta})}\mathbb{E}_{q_\zeta}\text{KLD}[\, p(y|\mathbf{x}, \boldsymbol{\theta}) \,||\, p(y|\tilde{\mathbf{x}}, \boldsymbol{\theta}) \,]$) to dropout and rotation perturbations of increasing magnitude (x-axis). We compare the proposed invariant priors—three approximations: implicit (red), factorized Gaussian (green), Gaussian mixture (blue)—to a standard Normal prior (pink) and the posterior (black) after training on perturbed data. We see the learned invariant priors exhibit invariance across all perturbation magnitudes, especially when using implicit or mixture approximations.

	Prior Distribution	Test Error (%)
SVM [22]		10.38
Bayesian Neural Net	N(0, .0025)	10.08
Neural Net w/ Dropout		8.85
CNN [8]		5.03
Bayesian Neural Net	**Invariant Prior** (Factorized Gaussian)	9.41
Bayesian Neural Net	**Invariant Prior** (Mixture of Three Gaussians)	8.29
Rotation-Invariant RBM [32]		4.20
Rotation-Aware ConvRBM [31]		3.98
Group Equivariant CNN [8]		2.28
Harmonic Networks [38]		**1.69**

Table 1: *Rotated MNIST.* Test classification error on a data set of rotated hand-written digits [22]. The first four models (from the top) have no notion of rotation, the next two have rotation invariant priors (ours), and the last two have rotations explicitly parameterized in the model and represent the current state-of-the-art.

	Prior Distribution	Test Error (%)
Logistic Regression w/ L2		14.22*
Bayesian Logistic Regression	N(0, .25)	14.19*
Transductive SVM		13.98
Logistic Regression w/ MC Dropout		12.83*
Logistic Regression w/ CF Dropout		11.90
Bayesian Logistic Regression	**Invariant Prior** (Factorized Gaussian)	11.93
Bayesian Logistic Regression	**Invariant Prior** (Mixture of Three Gaussians)	**11.81**

Table 2: *IMDB Sentiment Analysis.* Test classification error on the (large) IMDB sentiment analysis data set [24]. $*$ indicates a method was trained without the unlabeled examples. *MC*: Monte Carlo, *CF*: Closed-Form.

mance, following [8, 22]. We trained three Bayesian neural networks (NNs)—one with a standard Normal prior (variance chosen by validation set), one with a factorized Gaussian rotation invariant prior, and one with a mixture of Gaussians ($K = 3$) rotation invariant prior—and a NN with Bernoulli(.5) dropout. All networks had two hidden layers with 2,750 units each.

Test set classification error is shown in Table 1. The table is divided into three sections: the first has no concept of rotation, the second has a rotation invariant prior, and the third has rotation-invariance built into the data model. We see that the invariant priors allow the Bayesian NNs to perform comparably to (factortized Gaussian) or better than (mixture of Gaussians) all of the models with no built-in concept of rotation except the Convolutional NN. However, the performance gap between the models with invariant priors and models with rotations explicitly parameterized (bottom four) is still considerable. We conjecture that the gap is due to the prior learning coarse rotational invariance. To elaborate, the filters preferred by the prior (Figure 1 (b)) do not exhibit the fine, digit-specific rotated edge detectors learned by the parameterized models, as seen in [31]. The ability to learn these refined rotations likely boosts performance considerably. Moreover, we note that these models have been extensively hand-crafted to be rotationally invariant while our method is general and requires no additional effort from the modeler.

Semi-Supervised Classification. Lastly we report results on semi-supervised classification experiments on the large IMDB data set [24], which consists of 50,000 unlabeled examples and 25,000 for training and testing each. 5,000 of the training examples were used as a validation set. We trained several Bayesian and non-Bayesian logistic regression models, including one with the closed-form (CF) dropout penalty proposed by Wager et al. [36][1]. We used the unlabeled data to train dropout invariant priors as well as the CF dropout penalty.

Test set classification error is shown in Table 2. The Bayesian logistic regression model with a Gaussian mixture invariant prior achieves the lowest error rate, even besting the closed-form dropout penalty, which has the ability to learn the regularization and data model jointly. We conjecture that invariant priors were able to achieve better comparative performance in this setting because dropout is a simpler perturbation.

[1]The error is higher than what is reported by Wager et al. [36] due to using unigrams and a smaller vocabulary (20,000 words).

8 CONCLUSIONS

We have proposed an optimization objective (Equation 6) for learning priors that represent known invariance constraints. When the objective has an analytical solution (Section 4), we see that the resulting distribution is sensible and that both the objective's components are necessary. Experimentally, we demonstrated use of the prior results in better performance than when the invariance is not accounted for. Only models extensively hand-crafted for the invariance setting outperformed use of our proposed prior. This work, we believe, represents an important first step in allowing subjective priors to be specified for modern, large-scale Bayesian models.

Acknowledgments

The work in this paper was supported in part by the National Science Foundation under awards IIS-1320527, NRT-1633631, CNS-1730158, by the National Institutes of Health under award number U01TR001801, by Adobe Research, and by a Google Faculty Award.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3365–3373, 2014.

[2] Henry S Baird. Document image defect models. In *Structured Document Image Analysis*, pages 546–556. Springer, 1992.

[3] Jan Beirlant, Edward J Dudewicz, László Györfi, and Edward C Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

[4] José M Bernardo. Reference analysis. *Handbook of statistics*, 25:17–90, 2005.

[5] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

[6] George Casella. An introduction to empirical bayes data analysis. *The American Statistician*, 39(2):83–87, 1985.

[7] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1755–1763, 2014.

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd*

International Conference on Machine Learning (ICML), pages 2990–2999, 2016.

[9] Morris L Eaton. Group invariance applications in statistics. In *Regional conference series in Probability and Statistics*, pages i–133. JSTOR, 1989.

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[11] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1019–1027, 2016.

[12] P. Goel and A. Zellner. On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques*, pages 233–243, 1986.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[14] Silviu Guiasu and Abe Shenitzer. The principle of maximum entropy. *The Mathematical Intelligencer*, 7(1):42–48, 1985.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

[17] Harold Jeffreys. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 186, pages 453–461. The Royal Society, 1946.

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[19] Diederik Kingma and Max Welling. Efficient gradient-based inference through transformations between bayes nets and neural nets. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1782–1790, 2014.

[20] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2575–2583, 2015.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[22] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine learning (ICML)*, pages 473–480, 2007.

[23] Yuanzhi Li and Andrej Risteski. Approximate maximum entropy principles via goemans-williamson with applications to provable variational methods. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4635–4643, 2016.

[24] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150, 2011.

[25] Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Q Weinberger. Learning with marginalized corrupted features. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 410–418, 2013.

[26] Eric Nalisnick and Padhraic Smyth. Learning Approximately Objective Priors. *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[27] Veronica Nieves, Jingfeng Wang, Rafael L Bras, and Elizabeth Wood. Maximum entropy distributions of scale-invariant processes. *Physical Review Letters*, 105(11):118701, 2010.

[28] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 814–822, 2014.

[29] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems (NIPS)*, pages 496–504, 2016.

[30] Christian Robert. *The Bayesian Choice*. Springer, 2001.

[31] Uwe Schmidt and Stefan Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2050–2057. IEEE, 2012.

[32] Kihyuk Sohn and Honglak Lee. Learning invariant representations with local transformations. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1311–1318, 2012.

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

[34] Jos Uffink. The constraint rule of the maximum entropy principle. *Studies in History and Philosophy of Modern Physics*, 27(1):47–79, 1996.

[35] Stefan Wager, William Fithian, and Percy Liang. Data augmentation via levy processes. *Perturbations, Optimization, and Statistics*, 2016.

[36] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 351–359, 2013.

[37] Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 118–126, 2013.

[38] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5028–5037, 2017.