Deep Reference Priors: What is the best way to pretrain a model?

Yansong Gao[1,*], Rahul Ramesh[2,*], and Pratik Chaudhari[2,3]

[1]Applied Mathematics and Computational Sciences, University of Pennsylvania
[2]Computer and Information Science, University of Pennsylvania
[3]Electrical and Systems Engineering, University of Pennsylvania
Email: gaoyans@sas.upenn.edu, rahulram@seas.upenn.edu, pratikac@seas.upenn.edu
[*]Equal contribution

Abstract

What is the best way to exploit extra data—be it unlabeled data from the same task, or labeled data from a related task—to learn a given task? This paper formalizes the question using the theory of reference priors. Reference priors are objective, uninformative Bayesian priors that maximize the mutual information between the task and the weights of the model. Such priors enable the task to maximally affect the Bayesian posterior, e.g., reference priors depend upon the number of samples available for learning the task and for very small sample sizes, the prior puts more probability mass on low-complexity models in the hypothesis space. This paper presents the first demonstration of reference priors for medium-scale deep networks and image-based data. We develop generalizations of reference priors and demonstrate applications to two problems. First, by using unlabeled data to compute the reference prior, we develop new Bayesian semi-supervised learning methods that remain effective even with very few samples per class. Second, by using labeled data from the source task to compute the reference prior, we develop a new pretraining method for transfer learning that allows data from the target task to maximally affect the Bayesian posterior. Empirical validation of these methods is conducted on image classification datasets.

1 Introduction

Exploiting extra data, e.g., labeled data from a related task, or unlabeled data from the same task, is a powerful way of reducing the number of training data required to learn a given task. This idea lies at the heart of burgeoning fields like transfer, meta-, semi- and self-supervised learning, and these fields have developed a wide variety of methods to incorporate such extra information. To give a few examples, methods for transfer learning fine-tune a representation that was pretrained on labeled data from another—ideally related—task. Methods for semi-/self-supervised learning pretrain the representation using unlabeled data, which may come from the same task or from other related tasks, before using the labeled data. In this paper, we ask the question: what is the *best* way to exploit extra data for learning a task? In other words, if we have *some* pool of data—be it labeled or unlabeled, from the same task, or from another task—what is the *optimal* way to pretrain a representation?

As posed, the answer to the above question depends upon the downstream task that we seek to solve and therefore it seems difficult to obtain a general solution. We can formulate a more reasonable question by recognizing that a pretrained representation can be thought of as a Bayesian prior (or a sample from it). Fundamentally, a prior restricts the set of models that can be fitted upon the task. So we could instead ask: how to best use the extra data to restrict the set of models that we could fit on the desired task. We formalize this question in this paper using the theory of reference priors in Bayesian statistics. Our paper has the following contributions.

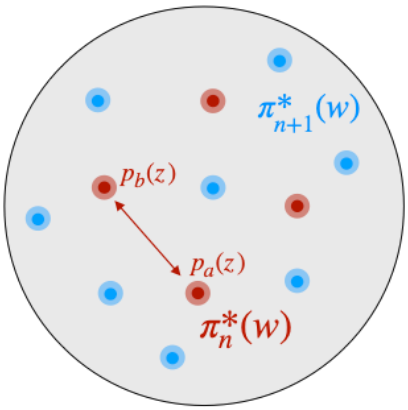


Figure 1: A schematic to understand the concept of a reference prior. The circle denotes the space of probability distributions $p_w(z)$ parameterized by weights w. A reference prior is supported on a set of diverse models that are spread out in model space. Given n samples z^n, this prior is a distribution supported on a discrete set (red) such that each of its atoms predicts confidently on z^n but very differently from other atoms of π^*_n. A reference prior of order n, denoted by $\pi^*_n(w)$, maximizes the mutual information $I(w; z^n) = H(z^n) - H(z^n \mid w)$ where H denotes entropy and $p(z^n) = \mathbb{E}_{w\sim\pi^*_n}[p(z^n \mid w)]$ is the average prediction of models from the prior. Given more data, say a dataset with one more sample z^{n+1}, the new reference prior π^*_{n+1} consists of more atoms because the models now have more ways of distinguishing their outputs from those of each other. As $n \to \infty$, the reference prior converges to the Jeffreys prior, modulo some technical conditions. This paper uses the concept of reference priors to develop theory and methods for Bayesian semi-supervised, self-supervised and transfer learning that work even when the sample size n is extremely small, e.g., our implementation of semi-supervised learning obtains an accuracy of 81.08% on CIFAR-10 with 5 labeled samples/class.

(1) We formalize the problem of "how to best pretrain a model" using the theory of reference priors, which are objective, uninformative Bayesian priors computed by maximizing the mutual information between the task and the weights. We show how these priors maximize the KL-divergence between the posterior computed from the task and the prior, on average over the distribution of the unknown future data. This allows the samples from the task to maximally influence the posterior. We discuss how reference priors are supported on a discrete set of atoms in the weight space. We develop a scalable, particle-based algorithm to compute reference priors for deep networks. To our knowledge, this is the first demonstration of reference priors for deep networks that maintains their characteristic nature, namely that they are supported in a discrete set of atoms.

(2) We formalize semi-supervised and self-supervised learning as computing a reference prior where the learner is given access to a pool of unlabeled data and seeks to compute a prior using this data. This formulation sheds light upon the theoretical underpinnings of existing state of the art methods such as FixMatch. In particular, we show that techniques such as consistency regularization and entropy minimization which are commonly used in practice can be directly understood using the reference prior formulation. This work therefore provides an information-theoretic formulation of semi-supervised and self-supervised learning.

(3) We develop a novel two-stage reference prior where the learner gets access to data in two stages and seeks to compute a prior that is optimal for data from the second stage. By solving for the optimal prior in this case, we show how it has the flavor of ignoring certain parts of the weight space depending upon whether data from the first stage was similar to that from the second stage, or not. This objective is closely related to the predictive Information Bottleneck principle.

(4) We formalize transfer learning using the two-stage reference prior where the goal is to build a prior using data from a source task that maximally leverages data from the target task. This formulation is useful because it is an information-theoretically optimal way to pretrain using a source task for the goal of transferring to the target task.

We show an empirical study of our formulations on the CIFAR-10 and CIFAR-100 datasets. We show that

our methods to compute reference priors are scalable, provide results that are competitive with state of the art methods for semi-supervised learning, and obtain significantly better accuracy than fine-tuning for transfer learning, even for very small sample sizes.

2 Background

This section introduces notation, develops the formulation for objective Bayesian priors and gives a few examples that explain how these priors work.

2.1 Setup

Consider a dataset $\hat{P}_n = \{(x_i, y_i)\}_{i=1}^n$ with n samples that consists of inputs $x_i \in \mathbb{R}^d$ and labels $y_i \in \{1, \dots, C\}$. Each sample of this dataset is drawn from a joint distribution $P(x, y)$ which we define to be the "task". It will be useful to use the shorthand $x^n = (x_1, \dots, x_n)$ and $y^n = (y_1, \dots, y_n)$ to denote all inputs and labels. Let $w \in \mathbb{R}^p$ be the weights of a probabilistic model which evaluates $p_w(y \mid x)$. We will use a random variable z with a probabilistic model denoted by $p_w(z)$ when we do not wish to distinguish between inputs and labels. Given a prior on weights $\pi(w)$, Bayes law gives the posterior

$$p(w \mid x^n, y^n) \propto p(y^n \mid x^n, w)\pi(w).$$

The Fisher Information Matrix (FIM) $g \in \mathbb{R}^{p \times p}$ has entries,

$$g(w)_{kl} = \frac{1}{n} \sum_{i=1}^{n} \sum_{y=1}^{C} p_w(y \mid x_i) \partial_{w_k} \log p_w(y \mid x_i) \partial_{w_l} \log p_w(y \mid x_i),$$

and it can be used to understand the differences between two probabilistic models. For a model with weights w, the Kullback-Leibler (KL) divergence $\mathrm{KL}(p_w, p_{w+\mathrm{d}w}) = \int \mathrm{d}P(x)\, \mathrm{d}y\, p_w(y \mid x) \log(p_w(y \mid x)/p_{w+\mathrm{d}w}(y \mid x))$ can be written as

$$\mathrm{KL}(p_w, p_{w+\mathrm{d}w}) = \frac{1}{2} \sum_{k,l=1}^{p} g(w)_{kl}\, \mathrm{d}w_k\, \mathrm{d}w_l\,.$$

The FIM is preserved under diffeomorphisms of the weights w (Amari, 2016). In Bayesian statistics, it is often used to build the Jeffreys prior $\pi_J(w) \propto \sqrt{\det g(w)}$. Jeffreys prior is reparameterization invariant, i.e., it assigns the same probability to a set of models irrespective of our choice of parameterization of those models. It is an example of an "uninformative prior" which means that it allows us to impose some generic structure upon the problem, e.g., that the prior be reparameterization invariant. Informative priors are different from such uninformative priors. For instance, if we were to choose π to be a Gaussian distribution with mean at a pretrained model then such a prior would express a very definite information and bias the ensuing posterior towards the pretrained model.

2.2 Reference Priors

The choice of a prior is typically subjective and left to the user. In order to make this choice more objective, Bernardo (1979) suggested that uninformative priors should maximize some divergence, say the Kullback-Leibler (KL) divergence $\mathrm{KL}(p(w \mid z), \pi(w)) = \int \mathrm{d}w\, p(w \mid z) \log\left(p(w \mid z)/\pi(w)\right)$, between the prior and the posterior for data z. The rationale for doing so is to allow the data to dominate the posterior rather than our choice of the prior. Since we do not know the data *a priori* while picking the prior, we should maximize the *average* KL-divergence over the data distribution $p(z)$. This amounts to maximizing the mutual information

$$\begin{aligned} \pi^* = \operatorname*{argmax}_{\pi} I_\pi(w; z) &:= \int \mathrm{d}z\, \mathrm{d}w\, p(z) p(w \mid z) \log \frac{p(w \mid z)}{\pi(w)} = H(w) - H(w \mid z) \\ &= \int \mathrm{d}z\, \mathrm{d}w\, p(z) p(w \mid z) \log \frac{p(z \mid w)}{p(z)} = H(z) - H(z \mid w), \end{aligned} \tag{1}$$

where $p(z) = \int \mathrm{d}w\, \pi(w) p(z \mid w)$ and $H(w) = \int \mathrm{d}w\, \pi(w) \log \pi(w)$ is the Shannon entropy; the conditional entropy $H(w \mid z)$ is defined analogously. Mutual information is a natural quantity for measuring the amount missing information about w provided by data z when the initial belief was π. The prior $\pi^*(w)$ is known as a reference prior. It is invariant to a reparameterization of the weight space because mutual information is invariant to reparameterization. The reference prior does not depend upon the samples $\hat{P}_n$ but only depends on their distribution P.

The objective to calculate reference prior π^* above may not be analytically tractable. Bernardo therefore also suggested that we compute n-reference priors. In this paper, we call n the "order" and have deliberately overloaded the notation for the number of samples n; the reason for doing so will be clear soon.

$$\pi_n^* = \operatorname*{argmax}_{\pi} I_\pi(w; z^n) = H(w) - H(w \mid z^n), \tag{2}$$

using n samples and then setting $\pi^* := \lim_{n\to\infty} \pi_n^*$ under appropriate technical conditions (Berger et al., 1988). Reference priors are asymptotically equivalent to Jeffreys prior for one-dimensional problems. They differ in general for multi-dimensional problems but it can be shown that Jeffreys prior is the continuous prior that maximizes the mutual information (Clarke and Barron, 1994).

2.3 Blahut-Arimoto algorithm

The Blahut-Arimoto algorithm (Arimoto, 1972; Blahut, 1972) is a general method for maximizng functionals like (1) and leads to iterations of the form $\pi^{t+1}(w) \propto \exp\left(\mathrm{KL}(p(z \mid w), p(z))\right) \pi^t(w)$. It exploits the identity

$$\max_{\pi} I_\pi(w; z) = \max_{q} \max_{\pi} \mathop{\mathbb{E}}_{w\sim\pi} \left[p(z \mid w) \log \frac{q(w \mid z)}{\pi(w)} \right]$$

where the optimum on the right-hand-side is achieved when $q(w \mid z) \equiv p(w \mid z)$. The BA algorithm alternates the maximization over π and q in this identity to result in Maximizing mutual information is a convex problem and therefore the BA algorithm is guaranteed to converge. It is typically implemented for discrete variables, e.g., Information Bottleneck (Tishby et al., 1999). Such a discretization of the weight space is prohibitive for high-dimensional models, we therefore implement this algorithm using particles in Remark 4.

Example 1 (Estimating the bias of a coin). Consider the estimation of the bias of a coin $w \in [0, 1]$ using n trials. If z denotes the number of heads (which is a sufficient statistic), we have $p(z \mid w) = w^z (1-w)^{n-z} n!/(z!(n-z)!)$. For $n = 1$, since we know that $I(w; z^1) \leq \log 2$ with this one bit of information, we can see that $\pi_1^*(z) = (\delta(w) + \delta(1-w))/2$ is the reference prior that achieves the upper bound (Berger et al., 1988). This result is intuitive: if we *know* that we have only one observation and the outcome is either a heads or a tails, the optimal uninformative prior puts equal probability mass on $w = 0$ and $w = 1$. We can numerically find π_n^* for different values of n using the BA algorithm (Fig. 2).

Example 2 (Visualizing the reference priors in data space). We next construct a way to visualize the reference prior for deep learning classification problem. To compute the reference prior, we implement the algorithm in Remark 4 using particles(atoms). Note that w are weights of a deep network, it is not reasonable to visualize $\pi_n^*(w)$ in such a high dimension weight space. We use the following strategy to visualize the prior.

Given a classification problem with C classes and n samples in the training set, we think of the predictions of the network as a vector in $n \times C$ dimensions given by

$$\mathbb{R}^{nC} \ni f(w) = \frac{1}{\sqrt{2n}} \left(\sqrt{p_w(y = 1 \mid x_1)}, \sqrt{p_w(y = 2 \mid x_1)}, \ldots, \sqrt{p_w(y = C-1 \mid x_n)}, \sqrt{p_w(y = C \mid x_n)} \right).$$

This vector $f(w)$ is parameterized by the weights w of the network. The rationale behind this choice is to observe that for two weights w, w', the Euclidean distance between the prediction vectors is the Hellinger

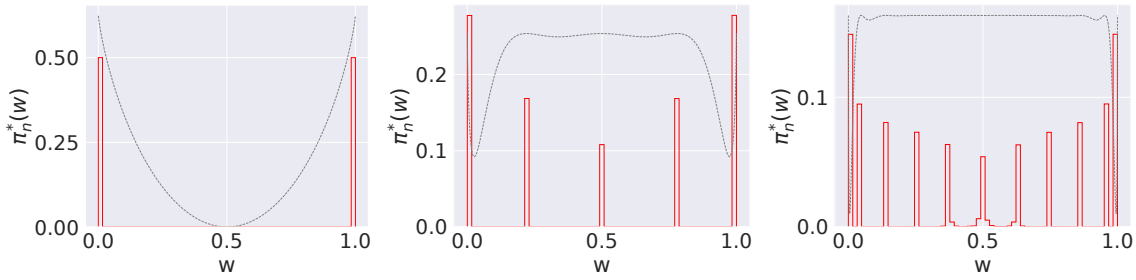


Figure 2: Reference prior for the coin-tossing model for $n = 1, 10, 50$ (from left to right) computed using the Blahut-Arimoto algorithm. Atoms are critical points of the gray line which is $\mathrm{KL}(p(z^n), p(z^n \mid w))$. The prior is discrete for finite order $n < \infty$, also see Mattingly et al. (2018). Atoms of the prior are maximally different from each other, e.g., for $n = 1$, they are on opposite corners of the parameter space. As the number of data increases, the separation between atoms of the prior reduces. The prior converges to Jeffreys prior $\pi_J(w) \propto (w(1-w))^{-1}$ as $n \to \infty$ for this one-dimensional setting.

divergence between the prediction probability distributions,

$$\|f(w) - f(w')\|^2 = \frac{1}{2n} \sum_{i=1}^{n} \sum_{k=1}^{C} \left(\sqrt{p_w(y = k \mid x_i)} - \sqrt{p_{w'}(y = k \mid x_i)} \right)^2$$
$$= \frac{1}{n} \sum_{i=1}^{n} d_H^2 \left(p_w(\cdot \mid x_i), p_{w'}(\cdot \mid x_i) \right).$$

where

$$d_H^2(P, Q) = \frac{1}{2} \int \left(\sqrt{\mathrm{d}P} - \sqrt{\mathrm{d}Q} \right)^2$$

is the Hellinger distance. In other words, the prediction vector $f(w)$ maps the weights w into a $(n \times C)-$dimensional Euclidean space, which makes it possible to visualize the reference prior, as shown in Fig. 3.

At the beginning stage of the optimization, all atoms w^k make similar predictions (blue points in Fig. 3 concentrate near each other). This is not surprising because the atoms are initialized randomly and w^k has a uniform distribution at its output. As the prior is optimized to maximize the mutual information, its atoms increasingly make more diverse predictions (orange) and towards the end stage (green), the prediction of the atoms forms a low-dimension manifold. We hypothesize that these green particles form a "boundary" of the possible prediction vectors made by the model and consist of simple, low-dimensional models (Mattingly et al., 2018).

3 Methods

This section first discusses a key property of these priors that enables us to calculate them numerically, namely the fact that they are supported on a discrete set in the weight space for finite order (§3.1). It then constructs priors that are designed to make maximal use of pre-training data for semi-supervised (§3.2) and transfer learning (§3.4).

3.1 Existence and discreteness of reference priors

Existence and boundedness. As (Berger et al., 1988) say, reference prior does not exist if $I_\pi(w; z^n)$ is infinite. To address the existence of the reference prior, we need to assume that (i) π is supported on a compact set $\Omega \subset \mathbb{R}^p$, and (ii) if $p_\pi(z^n) = \int_\Omega \mathrm{d}w\, \pi(w) p(z^n \mid w)$ is the marginal, $\mathrm{KL}(p_w, p_\pi)$ is a continuous function of w for any π. Then the n-reference prior π_n^* exists and $I_{\pi_n}(w; z^n)$ is finite; see Lemma 2.14 in (Zhang, 1994).
Discreteness. Suppose that z^n is a discrete random variable with number of atoms denoted by C. Suppose

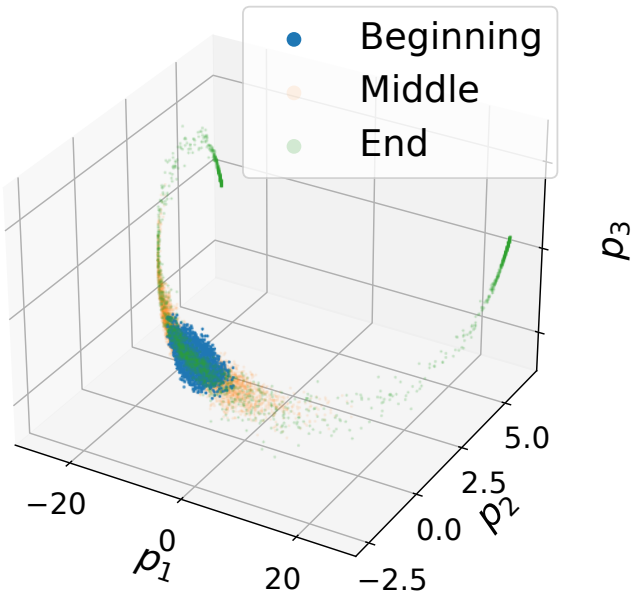


Figure 3: We compute and visualize the reference prior for a binary classification problem on MNIST (digits 3 vs. 5). We randomly initialize $K = 3000$ particles (blue) and then maximize the mutual information in (3). For each atom w^k of our prior, $k = 1, 2, \ldots, 3000$, we evaluate its prediction vector $f(w^k)$. By using PCA, we embed all K prediction vectors into a 3-dimensional Euclidean space spanned by the top three principle components $\vec{p_1}$, $\vec{p_2}$ and $\vec{p_3}$. The different colors show the prior at initialization (blue), after 5,000 iterations (orange) and towards the end of the computation after 50,000 iterations (green).

that the $n-$reference prior π_n^* exists and is unique(up to measure zero), Let $\Omega_n = \{w \in \Omega : \pi_n^*(w) > 0\}$ be the support of π_n^*. Suppose $\{p(z^n \mid w) : w \in \Omega_n\}$ is compact. Then π_n^* is discrete with number of atoms no greater than C (see Lemma 2.18 in (Zhang, 1994)). Rigorous development of reference priors has been done by these earlier works. Our paper focuses on some innovative applications to problems in machine learning.

Remark 3 (The reference prior depends upon the number of samples in the task and finds a small set of diverse models). The likelihood of each atom in the optimal prior $p(z^n \mid w)$ is maximally different from that of other atoms, and thereby the marginal likelihood $p(z^n)$. It is intuitive why the prior should have finite atoms, this is because it is not possible to distinguish between a large set of models using few samples n. For models like deep networks, this is extremely promising in the low-sample regime because it automatically selects a small set of models from a large model space to fit to the few data. This is analogous to covering numbers in learning theory (Bousquet et al., 2003) where we endow the hypothesis space with a metric that measures disagreement between two hypotheses over n samples. Smaller the n, smaller the covering number, and smaller the set of models considered by the learner.

Remark 4 (Using multiple particles in place of the Blahut-Arimoto algorithm). Since the optimal prior is discrete under certain assumptions, we can maximize the mutual information directly by identifying the best set of atoms. We set $\pi_n^* = \sum_{i=1}^K \lambda_i \delta(w - w^i)$ where $\{w^1, \ldots, w^K\}$ are the K atoms of our prior π_n^* and λ_i are coefficients with $\sum_i \lambda_i = 1$. We call these atoms "particles". Abbott and Machta (2019) also suggest a scaling law for K in terms of n. For example, they suggest that $K \sim n^{4/3}$ for a problem with two biased coins. Each particle for us will be a deep network, so we will instead treat K as a hyper-parameter.

Remark 5 (Variational approximation of reference priors). Nalisnick and Smyth (2017) maximize a lower bound on $I_\pi(w; z)$ and replace the term $p(z) = \int \mathrm{d}w\, \pi(w) p(z \mid w)$ in (1) by the so-called VR-max estimator $\max_w \log p(z \mid w)$ where the maximum is evaluated across a set of samples from $\pi(w)$ (Li and Turner, 2016). They use a continuous variational family parameterized by neural networks. Our arguments show that the prior is supported on a discrete set and we need not use a continuous-valued distribution to model it. We therefore maintain $\pi(w)$ as a set of particles.

3.2 Reference priors for semi-supervised learning

Consider the situation where we are given inputs x^n and labels y^n, and an additional pool of

$$\text{unlabeled inputs } x^u$$

from the same task which we can use to compute $\pi^*(w)$. The key idea behind using reference priors for semi-supervised learning is that since the KL-divergence between the posterior on (x^n, y^n) and the prior $\pi(w)$ is maximized *on average* over the inputs, we can use the unlabeled inputs x^u to perform this averaging. More precisely, we set

$$\begin{aligned} \pi_n^* = \operatorname*{argmax}_\pi I_\pi(y^n, x^n; w) &:= \mathop{\mathbb{E}}_{x^n, (y^n \mid x^n, w), w \sim \pi} \left[\log \frac{p(y^n \mid x^n, w)}{p_\pi(y^n \mid x^n)} \right] \\ &= \mathop{\mathbb{E}}_{x^u} \left[H(y^u \mid x^u) \right] - \mathop{\mathbb{E}}_{x^u, w \sim \pi} \left[H(y^u \mid x^u, w) \right]. \end{aligned} \tag{3}$$

where $p_\pi(y^n \mid x^n) = \int \mathrm{d}w\, \pi(w) \prod_{i=1}^n p(y_i \mid x_i, w)$ and likewise for $p_\pi(y^u \mid x^u)$. We have switched the expectation in the final step to over x^u instead of labeled inputs x^n because they have the same distribution. For the same reason, we also switched the expectation on all possible labels y^n of the labeled data, to y^u, the unknown labels of unlabeled data.

Assume for now that we know the number of classes C and can compute expectations over $p(y^n \mid x^n, w)$. If our prior has K particles, then the second term is the average of the per-particle entropy of the predictions. The objective encourages each particle w_i to predict confidently, i.e., to have a small entropy in its output distribution $p_{w_i}(y \mid x)$. The first term is the entropy of the average predictions: $p_\pi(y^n \mid x^n)$, and it is large if

particles predict different outputs y^n for the same inputs x^n, i.e., they disagree with each other. Effectively, the objective encourages particles to be very dissimilar but confident (not necessarily correct) models.

We use the empirical average over unlabeled inputs to maximize (3). Predictions on new samples x are made using the Bayesian posterior predictive distribution

$$p(y \mid x, x^n, y^n) \propto \int \mathrm{d}w \, \pi^*_n(w) p(y \mid x, w) p(y^n \mid x^n, w). \tag{4}$$

Remark 6 (The number of classes need not be known a priori). Consider a model that has a feature generator with weights w in conjunction with a fixed (or a non-parametric) classifier that makes predictions over C' classes. Since (3) involves an *expectation* over the predictions $p(y^u \mid x^u, w)$, C' need not be the same as the number of classes C for the task. We can compute the prior over weights w of the feature generator and use it to learn a model for labeled data. This property mirrors the prevalent practice in self-supervised learning, e.g., (Chen et al., 2020b), where the feature generator is trained on pretext tasks that are different from the eventual goal of classification.

Remark 7 (Reference prior spreads out on the statistical manifold). Given the weight space, the prior spreads its probability mass such that while each particle that is likely under the prior can confidently (not necessarily correctly) make predictions on inputs (first term in (3)), predictions of these particles are maximally entropic when combined together (second term). If we consider the statistical manifold from information geometry (Amari, 2016) which consists of all models $p_w(y \mid x)$, then the reference prior has a finite number of atoms and is maximally spread out on this manifold to ensure that the particles are distinct. Asymptotically, as the number of atoms grows to infinity, it approaches the flat prior on the manifold, which corresponds to Jeffreys prior on the weight space (Clarke and Barron, 1994).

3.3 Reference priors for a two-stage experiment

In this section, we do not distinguish between inputs and labels and first develop the idea in terms of a generic random variable z^n. Consider a situation when we obtain data in two stages, first z^m, and then z^n. How should we select a prior, and thereby the posterior of the first stage, such that the posterior of the second stage makes maximal use of the new n samples? We can extend the idea in §2.2 in a natural way to address this question. We can maximize the divergence between the posterior of the second stage and the posterior after the first stage, on average, over the new samples z^n.

Since we have access to samples z^m, we need not average over them, we can compute the posterior $p(w \mid z^m)$ from these samples given the prior $\pi(w)$. First, notice that by chain rule $p(w, z^n \mid z^m) = p(w \mid z^{m+n}) p(z^n \mid z^m) = p(z^n \mid w) p(w \mid z^m)$. We can now write

$$\begin{aligned} \pi^*_{n \mid m} = \operatorname*{argmax}_{\pi} I_{p(w \mid z^m)}(w; z^n) &:= \int \mathrm{d}z^n \, p(z^n \mid z^m) \, \mathrm{KL}(p(w \mid z^{m+n}), p(w \mid z^m)) \\ &= \int \mathrm{d}w \, p(w \mid z^m) \int \mathrm{d}z^n \, p(z^n \mid w) \log \frac{p(z^n \mid w)}{p(z^n \mid z^m)}, \end{aligned} \tag{5}$$

where $p(w \mid z^m) \propto p(z^m \mid w)\pi(w)$ and $p(z^n \mid z^m) = \int \mathrm{d}w \, p(z^n \mid w) p(w \mid z^m)$. If (2) has a unique solution, we should have the optimal $p(w \mid z^m) \equiv \pi^*_n(w)$. This leads to

$$\pi^*_{n \mid m}(w) \propto \pi^*_n(w) \, p(z^m \mid w)^{-1}. \tag{6}$$

Notice that the reference prior puts *less* probability mass on regions which have high likelihood on old data z^m. This is consistent with intuition because the prior is designed to be such that the posterior is maximally informed by the new samples z^n. Given knowledge of old data, the prior *downweighs regions* in the weight space that could bias the posterior of the new data. We will exploit this peculiar property of reference priors in this section. We also have $\pi^*_{n \mid m} = \pi^*_n$ for $m = 0$ which is consistent with (2). As $m \to \infty$, this prior ignores the part of the weight space that was ideal for z^m. Appendix C shows this two-stage prior for the biased coin experiment.

Remark 8 (Averaging over z^m in the two-stage experiment). If we do not know the outcomes z^m yet, the prior should be calculated by averaging over both z^m, z^n

$$\begin{aligned}\pi^* = \operatorname*{argmax}_{\pi} \int \mathrm{d}z^m \, p(z^m) I_{p(w \mid z^m)}(w; z^n) &:= I_\pi(w; z^{m+n}) - I_\pi(w; z^m) \\ &= H(w \mid z^m) - H(w \mid z^{m+n}).\end{aligned} \tag{7}$$

The reference prior encourages multiple explanations of initial data z^m, i.e., high $H(w \mid z^m)$, so as to let the future samples x^n select the best one among these explanations, i.e., reduce the entropy $H(w \mid z^{m+n})$. It is interesting to note that neither is this two-stage prior equivalent to maximizing $I_\pi(w; z^{m+n})$, nor is it simply the optimal prior corresponding to objectives $I_\pi(w; z^m)$ or $I_\pi(w; z^n)$. Both (6) and (7) suggest that such priors may be useful when we have some data *a priori*, e.g., either unlabeled samples from the same task, or labeled samples from some other task.

Remark 9 (Connection to the Predictive Information Bottleneck). The objective in (7) resembles that of Bialek et al. (2001), or its variational version in Alemi (2020), which seek to learn a representation, say w, which maximally forgets past data while remaining predictive of future data

$$\max_{p(w \mid z^m)} I(w; z^n) - \beta I(w; z^m). \tag{8}$$

The parameter β in (8) gives this objective control over how much information from the past is retained in the representation w. This objective is therefore called the predictive information bottleneck (IB).

A softer version of reference prior for the two-stage experiment. We take inspiration from the predictive IB and construct a variant of the prior in (6)

$$\begin{aligned}\pi^{\beta}_{n \mid m}(w) \propto \pi^*_n(w) p(z^m \mid w)^{-\beta} \quad \text{for } \beta \in (0, 1). \\ \Rightarrow p(w \mid z^{m+n}) \propto p(z^n \mid w) p(z^m \mid w)^{1-\beta} \pi^*_n(w).\end{aligned} \tag{9}$$

In this prior, we should use a value of $\beta = 0$ when we expect that data from the first stage z^m is very similar to data z^n from the second stage; this allows the posterior to *benefit from the past samples*. If we expect that the data are different, then we should set $\beta = 1$ which ignores regions in the weight space that predict well for z^m while selecting the prior for z^n. This is similar to the predictive IB where a small β encourages remembering the past data and $\beta = 1$ encourages forgetting. The difference between our reference prior approach and the predictive IB is that the prior in the latter need not be a reference prior.

3.4 Reference priors for transfer learning

Now consider the two-stage experiment where at each stage we obtain data from a different task. The first stage consists of m samples from a "source" task P^s and the second stage consists of n samples from the "target" task P^t. Our goal is to calculate a prior $\pi(w)$ that best utilizes the target task data. Bayesian inference for this problem involves first computing the posterior $p(w \mid x^m_s, y^m_s) \propto p(y^m_s \mid w, x^m_s)\pi(w)$ from the source task and then using it as a prior to compute the posterior for the target task $p(w \mid x^n_t, y^n_t, x^m_s, y^m_s)$. Just like §2.2, we would like to maximize the average KL-divergence between the two posteriors $\mathrm{KL}\left[p(w \mid x^n_t, y^n_t, x^m_s, y^m_s), p(w \mid x^m_s, y^m_s)\right]$ which measures the amount of the information learnt from samples from the target task. However, note that this KL-divergence is a random variable: its randomness comes from samples x^m_s and x^n_t. Therefore, instead of maximizing the divergence given a specific x^m_s and x^n_t, we maximize the average divergence. We consider the following cases.

Case 1: Access to a pool of unlabeled data from the source x^m_s and the target task x^n_t. In this situation, we should average the KL-divergence over the predictions of the model y^m_s on the source data x^m_s; this is in addition to the unknown labels y^n_t of the target task. This involves maximizing

$$\mathop{\mathbb{E}}_{x^m_s, x^n_t, y^m_s \mid x^m_s, y^n_t \mid x^n_t} \mathrm{KL}\left[p(w \mid x^n_t, y^n_t, x^m_s, y^m_s), p(w \mid x^m_s, y^m_s)\right] \tag{10}$$

over the prior π. Here $p_\pi(y_s^m \mid x_s^m) = \mathbb{E}_{w\sim\pi}\, p(y_s^m \mid x_s^m, w)$ and $p_\pi(y_t^n \mid x_t^n) = \mathbb{E}_{w\sim\pi}\, p(y_t^n \mid x_t^n, w)$, respectively.

Case 2: x_s^m, y_s^m are fixed and known, and we have a pool of unlabeled data for x_t^n. Since we already know the data and labels for the source task, we will only average over x_t^n and y_t^n and maximize the objective

$$\mathop{\mathbb{E}}_{x_t^n, y_t^n \mid x_t^n} \mathrm{KL}\left[p(w \mid x_t^n, y_t^n, x_s^m, y_s^m), p(w \mid x_s^m, y_s^m)\right], \tag{11}$$

over π; here $p_\pi(y_t^n \mid x_t^n) = \int \mathrm{d}w\, \pi(w) p(y_t^n \mid x_t^n, w)$. Note that we need not have a separate pool of unlabeled data from the target, we can simply use the samples x_t^n and average over the labels; this also applies to (3).

Remark 10 (Connecting (10) and (11) to practice). Both objectives can be written down as

$$\pi^* = \mathop{\mathrm{argmax}}_{\pi} I_\pi(w; y_t^n, x_t^n \mid x_s^m, y_s^m) - I_\pi(w; x_s^m, y_s^m) \tag{12}$$

with the distinction that while in Case 1, we average over all quantities, namely $p(x_s^m), p(y_s^m), p(x_t^n), p(y_t^n)$ while in Case 2, we fix the distributions of $p(x_s^m)$ and $p(y_s^m)$ to the empirical distribution of the source task data. Case 2 is commonly considered transfer learning. Case 1, where one has access to *only unlabeled data* from a source task *that is different from the target task* is not typically studied in practice. If the source and target tasks are the same, then the objective in (12) is the same as that of (3). If they are not, our theory indicates that the prior in (12) should *minimize* the mutual information with respect to the source task; this is similar to the argument in §3.3. Like (9), we can again introduce a coefficient β on the second term in (12) to control handle the relatedness between source and target tasks.

Remark 11 (Reference priors for semi-supervised and transfer learning are discrete). ?? extends directly to the semi-supervised learning case in (3) and the two-stage experiment in (6) and (9). Therefore we can show that the prior should be discrete if there are a finite amount of labeled samples in semi-supervised learning, or if the *target* task in transfer learning provides finitely many samples.

3.5 Reference priors for self-supervised learning

Reference priors shed light on understanding self-supervised learning techniques. Contrastive approaches of self-supervised learning (SelfSL) learn representations by minimizing the distance between two augmented views of the same data point (positive pairs) and maximizing views from different data points (negative pairs). In this section, w are weights of a stochastic feature extractor. Once an input data x was feed into the model, a stochastic representation $h \in \Gamma \subset \mathbb{R}^l$ is generated, where Γ is a compact subset in $l-$dimension real space. Let G be a set of data augmentations, we write the joint distribution, for $g \in G$

$$p(x, h, g \mid w) = \frac{1}{|G|} p(x) p(h \mid g(x), w). \tag{13}$$

then $h \mid x, w$ distributes as the ensemble averaging law

$$p(h \mid x, w) = \frac{1}{|G|} \sum_{g\in G} p(h \mid g(x), w) \tag{14}$$

where G is a set of data augmentations. We extend the spirit of reference prior in a natural way and write down the mutual information bwtween h^n and x^n given w.

$$\begin{aligned} I_w(h^n : x^n) &= \mathop{\mathbb{E}}_{x^n} \int \mathrm{d}h^n\, p_w^{\text{ave}}(h^n \mid x^n) \log \frac{p_w^{\text{ave}}(h^n \mid x^n)}{p_w^{\text{ave}}(h^n)} \\ &= H(h^n \mid w) - \mathop{\mathbb{E}}_{x^n} H(h^n \mid x^n, w), \end{aligned}$$

where $p_w^{\text{ave}}(h^n \mid x^n) = \prod_{i=1}^n p_w^{\text{ave}}(h_i \mid x_i)$ and $p_w^{\text{ave}}(h^n) = \mathbb{E}_{x^n}\, p_w^{\text{ave}}(h^n \mid x^n)$.

Remark 12. The first term in (15) matches with the negative pair penalty in contrastive learning, while the second tern in (15) coincides with the positive pairs. Therefore contrastive learning is equivalent to maximizing mutual information between representations and data. That explains why contrastive approaches works since they are selecting models that do their best to learn information from data:

$$w^* = \arg\max_w I_w(h^n : x^n) = \arg\max_w H(h^n \mid w) - \mathop{\mathbb{E}}_{x^n} H(h^n \mid x^n, w). \tag{15}$$

Remark 13 (connecting with free energy principle for representation learning). (Gao and Chaudhari, 2020b) pick reconstruction of the original data—as a way to measure the discarded information in the representation when it is fitted on a specific task, this idea leads to the study of the following Lagrangian which is similar to the Information Bottlenck of (Tishby et al., 2000)

$$\min_w R + \lambda D + \gamma C, \tag{16}$$

where the rate R is an variational upper bound on the mutual information $I_w(x \mid h)$, distortion D measures the quality of reconstruction of the decoder, and C measures the classification loss. The canonical reconstruction task forces the representation learning redundant information about the data, (Gao and Chaudhari, 2020b) empirically show that these extra information are potentially helpful for transfer learning. Instead of attaching a canonical task, we directly setting the constrains

$$\begin{aligned} &\min_w C \\ &\text{such that } I_w(x^n \mid h^n) \geq MI, \end{aligned}$$

which forces the representation learning redundant information.

3.6 Practical tricks for implementing reference priors

The reference prior objective is conceptually simple but it is difficult to implement it directly using deep networks and modern datasets. We next discuss some practical tricks to compute reference priors.

Order of the prior n versus the number of samples. Theory sets the order of the prior n to be the same as the number of samples (Bernardo, 1979). We make a distinction between the two and restrict our experiments to order $n = 2, 3$. Mathematically, this amounts to computing averages in (2) or (3) over only sets of n samples at a time rather than all of them. This significantly reduces the class of models considered in the reference prior by pretending that there is an extremely small number of samples available for training the task; this is useful for large models like deep networks. Note that we are *not* restricting to small order n for computational reasons, i.e., computing the expectation over all classes y^n in (3) can be done in a single forward pass.

Using cross-entropy loss to bias particles towards good parts of the weight space. Consider Case 2 in §3.4. The posterior (4) suggests that we should first compute the prior, and then weight each sample by the likelihood of the labeled data. In practice, we combine these two steps into a single objective

$$\max_\pi \gamma I_\pi(w; y^u, x^u) + \mathop{\mathbb{E}}_{w\sim\pi} \left[\log p(y^n \mid x^n, w)\right], \tag{17}$$

where γ is a hyper parameter, x^n, y^n are labeled samples. (17) makes sense since it allows us to directly obtain particles that both have high probability under the prior and that have a high likelihood. Using such particles for inference is different from using the correct Bayesian posterior, in particular due to the coefficient γ, which we set to be $\gamma = 1/2$. We initialize the prior $\pi(w) = \sum_{i=1}^K \lambda_i \delta(w - w^i)$ at randomly chosen weights $\{w^i\}$ and perform gradient ascent steps on the above objective to directly compute the likelihood weighted particles. This strategy further restricts the search space for the particles in $\pi(w)$. A warm posterior, i.e., $\gamma < 1$, as opposed to cold posteriors (Wenzel et al., 2020) used in supervised learning, is a reasonable thing to do here because our prior $\pi(w)$ is highly entropic.

Data augmentations. Data augmentation is a key component of state-of-the-art approaches for semi-supervised learning. Methods like Fix-match employ heavy data augmentations. We also use data augmentations for experiments on CIFAR-10 and CIFAR-100. For implementing (17), we define the averaging prediction

$$p_w^{\text{ave}}(y \mid x) = \frac{1}{2} p(y \mid G_{\text{weak}}(x), w) + \frac{1}{2} p_w(y \mid G_{\text{strong}}(x), w)$$

as a linear combination of the predictions on the same input augmented by different augmentations G_{weak} and G_{strong}, where weak augmentations are flips/shifts, and strong ones are from AutoAugment (Sohn et al., 2020). Notice that by Jensen's inequality,

$$-\log p_w^{\text{ave}}(y \mid x) \leq -\frac{1}{2} \log p(y \mid G_{\text{weak}}(x), w) - \frac{1}{2} \log p(y \mid G_{\text{strong}}(x), w).$$

The reference prior for semi-supervised learning with weak and strong data augmentations is computed using

$$\max_\pi \gamma \, \mathbb{E}_{w \sim \pi, x^u} \int \mathrm{d}y^u p_w^{\text{ave}}(y^u \mid x^u) \frac{\log p(y^u \mid G_{\text{weak}}(x^u), w) + \log p(y^u \mid G_{\text{strong}}(x^u), w)}{2} + \gamma \, \mathbb{E}_{x^u}[H(y^u \mid x^u)] + \mathbb{E}_{w \sim \pi}[\log p(y^n \mid x^n, w)], \tag{18}$$

where x^u is sampled from unlabelled data, (x^n, y^n) are labeled samples. If G_{weak} and G_{strong} are identical augmentations, then $p_w^{\text{ave}}(y \mid x) = p(y \mid x, w)$ and the first term converges to $-\gamma H(y^u \mid x^u, w)$. We can therefore recover (17) from (18).

3.7 Relationship of the reference prior objective to existing methods for semi-supervised learning

The reference prior objective in (3) is closely connected to existing methods in semi-supervised learning. It is a difference of two entropies: $I_\pi(w; y^n, x^n) = \mathbb{E}_{x^n}[S(y^n \mid x^n)] - \mathbb{E}_{x^n, w \sim \pi}[S(y^n \mid x^n, w)]$. A number of methods, e.g., FixMatch (Sohn et al., 2020) or MixMatch (Berthelot et al., 2019), force the model to identify whether the underlying input image is the same across strong augmentations. Let G be a set of such augmentations. These methods use a quadratic penalty for $p(y^n \mid x^n)$ with respect to deviations from $p_G(y^n \mid x^n, w) = |G|^{-1} \sum_{g \in G} p(y^n \mid g(x^n), w)$ where $g(x^n)$ denotes the result of applying the augmentation g to inputs x^n. For our objective, minimizing the second term, $\mathbb{E}_{x^n, w \sim \pi}[H(y^n \mid x^n, w)]$ with this augmentation-averaged distribution $p_G(y^n \mid x^n, w)$ achieves the same purpose.

Disagreement-based methods (Zhou and Li, 2010) employ multiple models and use the confident models to soft-annotate unlabeled samples for others. Our reference prior is discrete and therefore also consists of multiple models. Disagreements among them are measured by $H(y^n \mid x^n)$. If $p(y^n \mid x^n)$ is uniform, which is encouraged by the reference prior objective, we obtain strong disagreements.

The first term $\mathbb{E}_{w \sim \pi, x^u} \int \mathrm{d}y^u \, p_w^{\text{ave}}(y^u \mid x^u) \left[\frac{1}{2} \log p(y^u \mid G_{\text{weak}}(x^u), w) + \frac{1}{2} \log p(y^u \mid G_{\text{strong}}(x^u), w)\right]$ in (18) closely relates to FixMatch which uses a cross-entropy loss that encourages consistent predictions across weak and strong augmentations. As we saw in (18), the key difference between FixMatch and our objective is the second term $\mathbb{E}_{x^u}[H(y^n \mid x^n)]$ that encourages disagreements between particles.

4 Empirical Study

4.1 Setup

We evaluate on the CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) datasets. For semi-supervised learning, we show results using 50–1000 labeled samples, i.e., 5–100 samples/class and use the rest of the samples in the training set as unlabeled samples. For transfer learning, we construct 20 5-way classification tasks using the super classes of the CIFAR-100 dataset. For all experiments on CIFAR-10/CIFAR-100, we use the WRN 28-2 architecture (Zagoruyko and Komodakis, 2016) which is identical to that used in Berthelot et al. (2019). We use the Adam (Kingma and Ba, 2015) to train with a constant learning rate of 0.002. For all our experiments,

the reference prior is of order $n = 2$ and has $K = 4$ particles. We run our methods for 512 epochs, with the coefficient γ scaled linearly from 0 to 0.25 over the entire training period and α fixed at 0.1. During evaluation, we use the Exponential moving average (EMA) of the model parameters which is commonly used in semi-supervised learning (Tarvainen and Valpola, 2017). Appendix A provides more details.

4.2 Baseline methods for semi-supervised and transfer learning

We compare Deep reference priors to existing methods in semi-supervised learning such as FixMatch (Sohn et al., 2020), MixMatch (Berthelot et al., 2019), Mean Teacher (Tarvainen and Valpola, 2017), Π-Model (Sajjadi et al., 2016b), Pseudo-Label (Lee et al., 2013), Virtual Adversarial Training (Miyato et al., 2018), and Mixup between labeled and unlabeled data (Berthelot et al., 2019). For transfer learning, we use fine-tuning to adapt a model trained on source to the target data. We additionally compare to using just the labeled target data (standard supervised learning), using labeled and unlabeled target data (self-supervised learning). We consider 5 different tasks from CIFAR-100, with 1000 labeled source samples and 100 labeled target samples for all experiments.

4.3 Semi-supervised learning

We evaluate Deep reference priors in the semi-supervised learning setting using the formulation described in §3.6. Table 1 compares Deep reference priors with competing semi-supervised learning methods on the CIFAR-10 dataset. Our methods outperform 5 other baseline methods (by 5-10%), except MixMatch and FixMatch. This indicates that reference priors can effectively leverage unlabeled data to pretrain representations and are competitive, if not better, than the state of the art. Deep reference priors are applicable to semi-supervised learning without being explicitly formulated to enforce properties like confident predictions on unlabeled samples or label consistency with respect to augmentations.

Fig. 4 compares Deep reference priors to MixMatch in settings where the learner is given access to different number of labeled data. Both methods have similar trends as the number of labeled samples is changed. The implementation of MixMatch by the original authors implements a number of techniques such as a moving averaging of the weights, an "interleaved" version of Mixup across labeled and unlabeled data, intricate schedules for various coefficients in their method. Our theoretical argument in §3.7 shows that the objective used in reference prior is very close to that of MixMatch, so we expect to match their accuracy on this problem. We have investigated the gap in accuracy at depth and we currently believe that the gap could be mitigated by using a larger number of particles in the reference prior. The utility of Deep reference priors is further validated in Table 2 (left) which evaluates on 5 semi-supervised learning tasks from CIFAR-100. Deep reference priors make use of unlabeled samples in addition to the 100 labeled samples for each task using, which results in accuracy gains as large as 18.6% when compared to using just labeled samples (vanilla supervised learning).

Method	Samples				
	50	100	250	500	1000
PiModel	-	-	46.58	58.18	68.47
PseudoLab	-	-	50.02	59.45	69.09
Mixup	-	-	52.57	63.86	74.28
VAT	-	-	63.97	73.89	81.32
Mean Teacher	-	-	52.68	57.99	82.68
MixMatch	64.21	80.29	88.91	90.35	92.25
FixMatch	86.19	-	94.53	-	-
Flex-UDA	94.67 (40)	-	94.95	-	
FlexMatch	95.01 (40)	-	95.2	-	
Deep Reference Prior	81.08	85.93	90.13	91.23	92.15

Table 1: Comparison to existing work in semi-supervised learning.

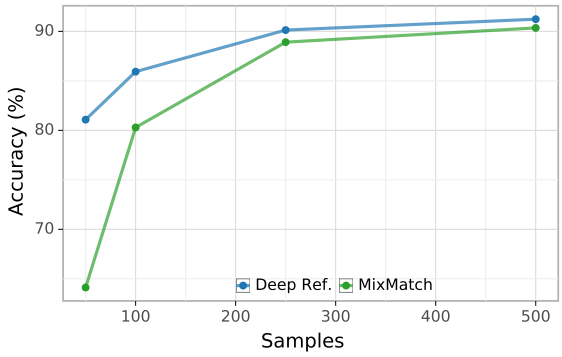


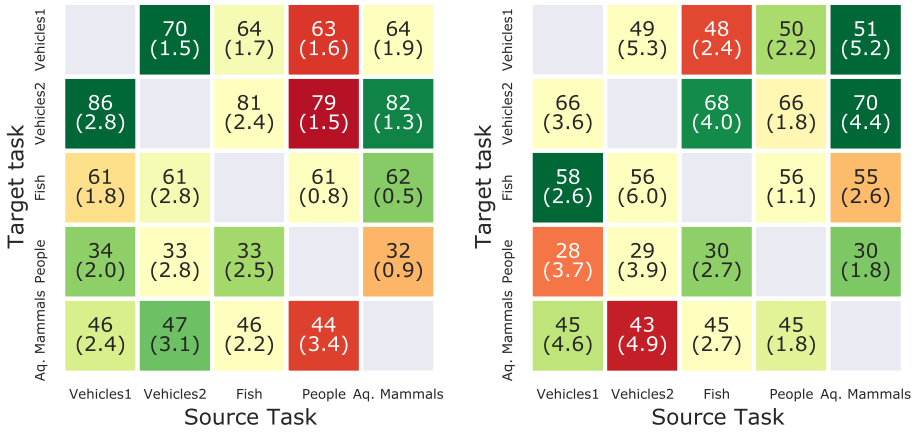
Figure 4: We compare Deep reference prior to a variant with no EMA during evaluation, and to MixMatch, over a varying range of the number of labeled samples. EMA gives a marginal yet consistent improvement in accuracy; this has also been observed in other work in semi-supervised learning. Trends in the accuracy are broadly similar for Deep reference priors and MixMatch and the gap between them reduces as the number of labeled samples increases. See §4.3 for more details.

4.4 Transfer learning

We next evaluate Deep reference priors in the transfer learning setting. The posterior (9) looks to find a model that is influenced by the target data by varying degrees controlled by β, with $\beta = 1$ corresponding to the scenario where the posterior is maxmimally influenced by target data after being pre-trained on the source data. We instantiate (9), by combining prior selection, pre-training on the source task and likelihood on the target task, into one objective,

$$\max_\pi \gamma I_\pi(w; y^u, x^u) + \mathop{\mathbb{E}}_{w\sim\pi} \log p(w; y^n \,|\, x^n) + (1-\beta) \mathop{\mathbb{E}}_{w\sim\pi} \log p(w; y^s \,|\, x^s), \tag{19}$$

where γ and β are hyper-parameters.



(a) **(b)**

Figure 5: Each cell shows the accuracy of using Deep reference priors (left) and Fine-tuning (right). Cells are colored red/green relative to the median accuracy of each row, with darker shades of green indicating that the source task is more suitable for transfer. For example, Vehicles-1 is particularly well suited source task for Deep Reference priors. The accuracy of most cells in Fig. 5a is better than the corresponding cells in Fig. 5b indicating that reference priors are capable of leveraging the labeled source data and unlabeled target data. The difference in accuracy between using Deep reference priors for transfer and fine-tuning is quite large in some cases, the gap is 34.8% for Vehicles2-Vehicles1.

Method	Task				
	Vehicles-1	Vehicles-2	Fish	People	Aq. Mammals
Supervised Learning	42.2	63.2	56.8	31.0	42.6
Deep Reference Prior (SSL)	63.6	75.2	54.6	34.0	47.4

Table 2: Accuracy (%) of 5 tasks from the CIFAR-100 dataset. Additional unlabeled data from the same task is useful in building a prior across all tasks except one task (Fish); improvement in accuracy is as large as 18.6%. This table should be read in conjunction to Fig. 5. We can see that transfer using source and target data using Deep reference prior results in an even larger improvement in performance, indicating that our method is effectively making use of both the labeled source data and unlabeled target data.

4.5 Semi-Supervised Learning Ablation Experiments

The following ablation experiments are conducted for semi-supervised learning on CIFAR-10 with 1000 labeled samples.

Order of prior. The order of the prior is an independent hyper-parameter as discussed in §3.6. Changing the order leads to a marginal (about 1%) improvement in accuracy. This justifies our choice of using $n = 2$ as the order in all experiments.

Method	Order			
	2	3	4	5
Deep Reference Prior ($K = 2$)	84.49	86.5	86.05	86.44

Table 3: The order of the reference prior has a minimal impact on accuracy.

Number of particles in the prior. We vary the number of particles in the prior and observe an improvement in accuracy as we increase the number of particles. The downside of this is however that a large number of particles implies increasing training time for the prior, which grows linearly with the number of particles.

Method	Particles			
	2	4	8	16
Deep Reference Prior ($n = 2$)	84.49	86.77	87.98	87.35

Table 4: Increased number of particles has a minimal impact on accuracy, the accuracy improves by 2.84% with 16 particles compared to 2.

Particles Weights in the Prior. As discussed in Remark 4, the prior is represented by

$$\pi_n^* = \sum_{i=1}^{K} \lambda_i \delta(w - w^i)$$

where $\{w^1, \ldots, w^K\}$ are the K atoms of the prior and λ_i are coefficients with $\sum_i \lambda_i = 1$. In our implementation, we set λ_i to be trainable parameters. We observe that λ_i assumes a uniform distribution over the particles in our experiments (Fig. 6). We expect this observation to change for a very large number of particles. For a small number of particles, the individual particles are extremely confident in their predictions. Hence, the term $H(y^u \mid x^u)$ is maximized when the predictions averaged across all particles is close to uniform distribution which is achieved when λ_i is uniform over all particles.

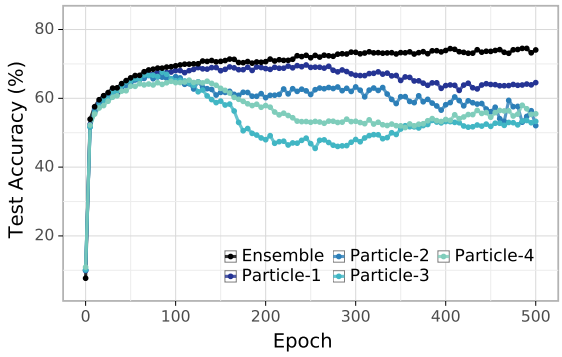


Figure 7: The evolution of the accuracies of individual particles in the reference prior when trained on 250 labeled samples. While the individual particles become diverse due to the term $H(y^n \mid x^n)$, the accuracy of the ensemble on the other hand consistently increases.

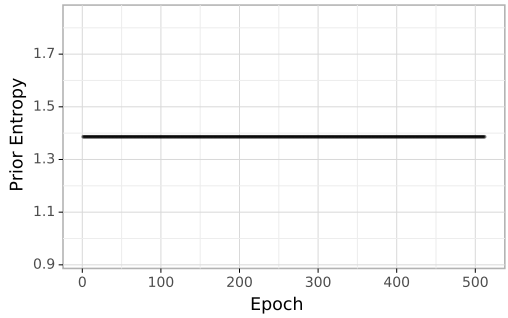


Figure 6: Entropy of the weights of the particles (λ_i) in the prior. The plot corresponds to the entropy of 4 particles which is essentially constant throughout training. In fact, the value is close the theoretical maximum of the entropy which is $\ln 4 \sim 1.38$ indicating that the weights λ_i are all equal.

4.6 Analysis and additional ablation studies

Diversity of the models in the prior. Fig. 7 shows how the entropy $H(y^u \mid x^u)$ enforces diversity between the different particles. We see that the accuracy of particles diverge during training. While one particle clearly dominates the posterior in terms of predicting correctly on most of the samples, the Bayesian posterior predictive distribution has a slightly higher accuracy than the best particle.

Design Choices. We next seek to dissect different components in our model, like EMA, Mixup, and the inclusion of the cross-entropy term (17) (referred to as cross-entropy warmup). We evaluate on a 2-way classification problem from CIFAR-10 (Cat vs. Dog) on 200 labeled samples run for 70 epochs. This is a challenging binary classification problem (relatively speaking, it is the hardest pair of classes in CIFAR-10). Fig. 9a indicates the benefit of including Mixup and the cross-entropy term. This benefit is more pronounced in the 10-way classification problem. Fig. 4 shows why using EMA in the evaluation step in beneficial.

Choice of α. We again evaluate on the Cats vs. Dogs 2-way classification task to understand the impact of the hyper-parameter choice for α. Fig. 9a indicates that using $\alpha = 1.0$ in (18) has lower accuracy compared to using $\alpha = 0.1$ or 0.5. The freedom to control this hyper-parameter is hence beneficial.

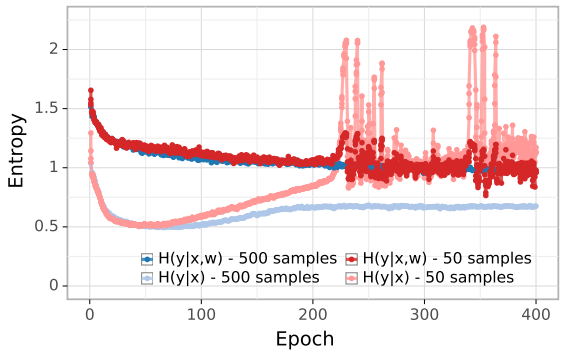


Figure 8: The evolution of the entropy terms $H(y^n \mid x^n, w)$ and $H(y^n \mid x^n)$, when trained on 500 and 50 samples. Note that while $H(y^n \mid x^n)$ is expected to be larger than $H(y^n \mid x^n, w)$ (since the KL-divergence is always positive), this is not the case since we approximate the term $H(y^n \mid x^n, w)$ by an upper-bound obtained from Jensen's inequality. Note that when there are only 50 labeled samples, the training dynamics become unstable, which is indicated by the oscillations in the entropy terms indicated in the plot

α	Accuracy (%)
0.1	75.35
0.5	74.45
1.0	71.4

(a)

Methods	Accuracy (%)
Ref Prior	75.35
No Mixup	73.4
No CE Warmup	69.4

(b)

Figure 9: **Fig. 9a**: This table computes the accuracies of 2-class classification (Cat vs Dog.) on 200 labeled samples run for 70 epochs. Using $\alpha = 1.0$ is clearly deterimental to performance and it is hence it is beneficial to introduce this tunable hyper-parameter. **Fig. 9b**: We ablate over some additional components in our model which are cross-entropy warmup, and mixup. We consider a simple 2-class classification problem (Cat vs Dog on CIFAR10) with 200 labeled samples and run our model for 70 epochs as opposed to the 1000 epochs. While this results, in less pronounced differences in accuracies, the trends are stil indicative of the importance of using cross-entropy warmup and mixup to train the model.

5 Related Work and Discussion

Reference priors in Bayesian statistics. We build upon the theory of reference priors which was developed in the objective Bayesian statistics literature Bernardo (1979); Berger et al. (1988, 2009). The main idea used in our work is that non-asymptotic reference priors allow us to exploit the finite samples from the task in a fundamentally different way than classical Bayesian inference. If the number of samples from the task available to the learner is finite, then the prior should also select only a finite number of models. Reference priors are not common in the machine learning literature. A notable exception is Nalisnick and Smyth (2017) who optimize a variational lower bound and demonstrate results on small-scale problems. The main technical distinction of our work is that we explicitly use the discrete prior instead of a variational approximation. To our knowledge, our work is the first instantiation of reference priors for medium-scale deep networks and image-based tasks.

Information theory. Discreteness is seen in many problems with an information-theoretic formulation, e.g., capacity of a Gaussian channel under an amplitude constraint (Smith, 1971), representations of neurons in the brain Laughlin (1981); Tkačik et al. (2008), and biological systems (Mayer et al., 2015). (Mattingly et al., 2018; Abbott and Machta, 2019) have developed a rich theory of scaling laws that give the number of atoms in the prior as a function of the number of data. They use examples to study how reference priors with finite data select "simple models" which lie on the edges of the parameter space (see Fig. 2). Although we cannot yet characterize a similar simplicity bias of reference priors for deep networks, we believe that the methods developed in our paper are effective because of this phenomenon; our choice of using a small order n for the prior is directly motivated by their examples. On the other hand, information based methods are widely applied in miscellaneous aspects of deep learning. One of the common motivation behind unsupervised contrastive representation learning **??**Oord et al. (2018); Henaff (2020); Tian et al. (2020); Bachman et al. (2019); Hjelm et al. (2018); He et al. (2020); Chen et al. (2020a); Wang and Isola (2020) is the InfoMax principle (Linsker, 1988), which instantiated by maximizing the mutual information (MI) between multi-views of the data. While in regularized information maximization for clustering, (Gomes et al., 2010; Bridle et al., 1992; Hu et al., 2017) directly maximize the mutual information between inputs and their representations. Beyond all these works on various information criterion, our work heavily explores the statistical dependency between the model weights and the data, which shed lights on understanding what the optimal way to pre-train a representation is.

Semi-Supervised Learning. Semi-supervised learning is a mature field with a wide diversity of approaches (Chapelle et al., 2009; Xiaojin, 2008; Zhu and Goldberg, 2009; Ouali et al., 2020). Ideas such as consistency regularization, which enforce consistent predictions across different augmentations of the inputs, form an important component of state of the art methods. Prior work (Bachman et al., 2014; Laine and Aila, 2016; Sajjadi et al., 2016b; Tarvainen and Valpola, 2017; Berthelot et al., 2019) explicitly enforces such losses while we show in §3.7 how the reference prior can automatically enforce consistency regularization. Minimizing the entropy of predictions on unlabeled data, either explicitly (Grandvalet et al., 2005; Miyato et al., 2018) or using pseudo-labeling methods (Lee et al., 2013; Sajjadi et al., 2016a), is another popular technique. We show in §3.7 how entropy minimization is automatically achieved by the reference prior objective.

Transfer learning. is a key component of a large number of applications today, e.g, (Devlin et al., 2019; Kolesnikov et al., 2020; Dhillon et al., 2020). A central question that remains unanswered today is how one should pretrain a model on the source task given that the eventual purpose is to transfer to a target task, although there have been some partial attempts at addressing it via the Information Bottleneck, e.g.,Gao and Chaudhari (2020a). This question becomes particularly challenging when transferring across domains, or for small sample sizes (Davatzikos, 2019). Reference priors are uniquely suited to tackle this question: the two-stage experiment that we have defined is the "optimal" way to leverage data from the source task for the target task, and learning such a prior is particularly useful in low-sample regimes (see §4.4).

References

Michael C. Abbott and Benjamin B. Machta. A Scaling Law From Discrete to Continuous Solutions of Channel Capacity Problems in the Low-Noise Limit. *Journal of Statistical Physics*, 176(1):214–227, July 2019. ISSN 1572-9613. doi: 10.1007/s10955-019-02296-2.

Alexander A Alemi. Variational predictive information bottleneck. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–6. PMLR, 2020.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer Japan, Tokyo, 2016.

Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

James O Berger, Jos M Bernardo, and Manuel Mendoza. *On Priors That Maximize Expected Information*. Purdue University. Department of Statistics, 1988.

James O Berger, José M Bernardo, and Dongchu Sun. The formal definition of reference priors. *The Annals of Statistics*, 37(2):905–938, 2009.

Jose M Bernardo. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):113–128, 1979.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural computation*, 13(11): 2409–2463, 2001.

Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473, 1972.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer, 2003.

John S Bridle, Anthony JR Heading, and David JC MacKay. Unsupervised classifiers, mutual information and'phantom targets'. 1992.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020b.

Bertrand S Clarke and Andrew R Barron. Jeffreys' prior is asymptotically least favorable under entropy risk. *Journal of Statistical planning and Inference*, 41(1):37–60, 1994.

Christos Davatzikos. Machine learning in neuroimaging: Progress and challenges. *NeuroImage*, 197:652, 2019.

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

Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *Proc. of International Conference of Learning and Representations (ICLR)*, 2020.

Yansong Gao and Pratik Chaudhari. A free-energy principle for representation learning. In *Proc. of International Conference of Machine Learning (ICML)*, 2020a.

Yansong Gao and Pratik Chaudhari. A free-energy principle for representation learning, 2020b.

Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. 2010.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. *arXiv:1912.11370 [cs]*, May 2020.

A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, Computer Science, University of Toronto, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv:1610.02242*, 2016.

Simon Laughlin. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c*, 36(9-10):910–912, 1981.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

Yingzhen Li and Richard E. Turner. R\'enyi Divergence Variational Inference. *arXiv:1602.02311 [cs, stat]*, October 2016.

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Henry H Mattingly, Mark K Transtrum, Michael C Abbott, and Benjamin B Machta. Maximizing the information learned from finite data selects a simple model. *Proceedings of the National Academy of Sciences*, 115(8):1760–1765, 2018.

Andreas Mayer, Vijay Balasubramanian, Thierry Mora, and Aleksandra M Walczak. How a well-adapted immune system is organized. *Proceedings of the National Academy of Sciences*, 112(19):5950–5955, 2015.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin 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, 2018.

Eric Nalisnick and Padhraic Smyth. Variational reference priors. 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

Rahul Ramesh and Pratik Chaudhari. Boosting a model zoo for multi-task and continual learning. *arXiv preprint arXiv:2106.03027*, 2021.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1908–1912. IEEE, 2016a.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016b.

Joel G Smith. The information capacity of amplitude-and variance-constrained sclar Gaussian channels. *Information and control*, 18(3):203–219, 1971.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-Th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

Gašper Tkačik, Curtis G Callan, and William Bialek. Information flow and optimization in transcriptional regulation. *Proceedings of the National Academy of Sciences*, 105(34):12265–12270, 2008.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020.

Zhu Xiaojin. Semi-supervised learning literature survey. *Computer Sciences TR*, 1530, 2008.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. *arXiv:1710.09412 [cs, stat]*, April 2018.

Zhongxin Zhang. *Discrete noninformative priors*. PhD thesis, Yale University, 1994.

Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3): 415–439, 2010.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

References

Michael C. Abbott and Benjamin B. Machta. A Scaling Law From Discrete to Continuous Solutions of Channel Capacity Problems in the Low-Noise Limit. *Journal of Statistical Physics*, 176(1):214–227, July 2019. ISSN 1572-9613. doi: 10.1007/s10955-019-02296-2.

Alexander A Alemi. Variational predictive information bottleneck. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–6. PMLR, 2020.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer Japan, Tokyo, 2016.

Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

James O Berger, Jos M Bernardo, and Manuel Mendoza. *On Priors That Maximize Expected Information*. Purdue University. Department of Statistics, 1988.

James O Berger, José M Bernardo, and Dongchu Sun. The formal definition of reference priors. *The Annals of Statistics*, 37(2):905–938, 2009.

Jose M Bernardo. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):113–128, 1979.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural computation*, 13(11): 2409–2463, 2001.

Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473, 1972.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer, 2003.

John S Bridle, Anthony JR Heading, and David JC MacKay. Unsupervised classifiers, mutual information and'phantom targets'. 1992.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020b.

Bertrand S Clarke and Andrew R Barron. Jeffreys' prior is asymptotically least favorable under entropy risk. *Journal of Statistical planning and Inference*, 41(1):37–60, 1994.

Christos Davatzikos. Machine learning in neuroimaging: Progress and challenges. *NeuroImage*, 197:652, 2019.

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

Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *Proc. of International Conference of Learning and Representations (ICLR)*, 2020.

Yansong Gao and Pratik Chaudhari. A free-energy principle for representation learning. In *Proc. of International Conference of Machine Learning (ICML)*, 2020a.

Yansong Gao and Pratik Chaudhari. A free-energy principle for representation learning, 2020b.

Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. 2010.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. *arXiv:1912.11370 [cs]*, May 2020.

A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, Computer Science, University of Toronto, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv:1610.02242*, 2016.

Simon Laughlin. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c*, 36(9-10):910–912, 1981.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

Yingzhen Li and Richard E. Turner. R\'enyi Divergence Variational Inference. *arXiv:1602.02311 [cs, stat]*, October 2016.

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Henry H Mattingly, Mark K Transtrum, Michael C Abbott, and Benjamin B Machta. Maximizing the information learned from finite data selects a simple model. *Proceedings of the National Academy of Sciences*, 115(8):1760–1765, 2018.

Andreas Mayer, Vijay Balasubramanian, Thierry Mora, and Aleksandra M Walczak. How a well-adapted immune system is organized. *Proceedings of the National Academy of Sciences*, 112(19):5950–5955, 2015.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin 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, 2018.

Eric Nalisnick and Padhraic Smyth. Variational reference priors. 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

Rahul Ramesh and Pratik Chaudhari. Boosting a model zoo for multi-task and continual learning. *arXiv preprint arXiv:2106.03027*, 2021.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Mutual exclusivity loss for semi-supervised deep learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1908–1912. IEEE, 2016a.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016b.

Joel G Smith. The information capacity of amplitude-and variance-constrained sclar Gaussian channels. *Information and control*, 18(3):203–219, 1971.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-Th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

Gašper Tkačik, Curtis G Callan, and William Bialek. Information flow and optimization in transcriptional regulation. *Proceedings of the National Academy of Sciences*, 105(34):12265–12270, 2008.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on

the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020.

Zhu Xiaojin. Semi-supervised learning literature survey. *Computer Sciences TR*, 1530, 2008.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. *arXiv:1710.09412 [cs, stat]*, April 2018.

Zhongxin Zhang. *Discrete noninformative priors*. PhD thesis, Yale University, 1994.

Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3): 415–439, 2010.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

A Details of the experimental setup

Architecture. For experiments on CIFAR-10 and CIFAR-100 (§4), we consider a modified version of the Wide-Resnet 28-2 architecture (Zagoruyko and Komodakis, 2016), which is identical to the one used in Berthelot et al. (2019). This architecture differs from the standard Wide-Resnet architecture in a few important aspects. The modified architecture has Leaky-ReLU with slope 0.1 (as opposed to ReLU), no activations or batch normalization before any layer with a residual connection, and a momentum of 0.001 for batch-normalization running mean and standard-deviation (as opposed to 0.1, in other words these statistics are made to change very slowly). We observed that the change to batch-normalization momentum has a very large effect on the accuracy of semi-supervised learning.

For experiments on MNIST (Appendix B.1), we use a fully-connected network with 1 hidden layer of size 32. We use the hardtanh activation in place of ReLU for this experiment; this is because maximizing the mutual information has the effect of increasing the magnitude of the activations for ReLU networks. One may use weight decay to control the scale of the weights and thereby that of the activations but in an effort to implement the reference prior exactly, we did not use weight decay in this model. Note that the nonlinearities for the CIFAR models are ReLUs.

Datasets. For semi-supervised learning, we consider the CIFAR-10 dataset with the number of labeled samples varying from 50–1000 (i.e., 5–100 labeled samples per class). We also consider a 2-class classification problem in some ablation experiments (§4.6) with the two classes being Cat/Dog, which is among the harder 2-class classification problems in CIFAR-10. Semi-supervised learning experiments use all samples that are not a part in the labeled set, as unlabeled samples.

For transfer learning, we construct two tasks from MNIST (task one is a 5-way classification task for digits 0–4, and task two is another 5-way classification task for digits 5–9). For this experiment, we use labeled source data but do not use any labeled target data. This makes our approach using a reference prior similar to a purely unsupervised method.

The CIFAR-100 dataset is also utilized in the transfer learning setup (§4.4). We consider five 5-way classification tasks from CIFAR-100 constructed using the super-classes. The five tasks considered are Vehicles-1, Vehicles-2, Fish, People and Aquatic Mammals. The selection of these tasks were motivated from the fact that some pairs of tasks are known to positively impact each other (Vehicles-1, Vehicles-2), while other pairs are known to be detrimental to each other (Vehicles-2, People); see the experiments in Ramesh and Chaudhari (2021).

Optimization. Adam with a constant learning rate of 0.002 was used in our experiments on CIFAR-10 and CIFAR-100. Mixed-precision (32-bit weights, 16-bit gradients) was used to expedite training. Training was

performed for 512 epochs unless specified otherwise. As is common practice in semi-supervised learning literature Berthelot et al. (2019), the coefficient of the objective of the unsupervised data (mutual information term in our case) was increased linearly during the course of training. Doing so uses the cross-entropy loss on the labeled data to effectively reduce the space of weight configurations that could be the atoms of the reference prior; this is beneficial while implementing the reference prior over a high-dimensional weight space.

We also conducted experiments with SGD for computing the reference prior on MNIST. SGD was used with a constant learning rate of 0.001 with Nesterov's acceleration, momentum coefficient of 0.9 and weight decay of 10^{-5}.

Definition of a single Epoch. Note that since we iterate over the unlabeled and labeled data (each with different number of samples), the notion of what is an epoch needs to be defined differently. In our work, one epoch refers to 1024 weight updates, where each weight update is calculated using a batch-size of 64 for the labeled data of batch size 64, and a batch-size of 60 for the unlabeled data.

Exponential Moving Average (EMA). In all CIFAR-10 and CIFAR-100 experiments, we also implement the Exponential Moving Average (EMA) (Tarvainen and Valpola, 2017). In each step, the EMA model is updated such that the new weights are the weighted average of the old EMA model weights, and the latest trained model weights. The weights for averaging used in our work (and most other methods) are 0.999 and 0.001 respectively. Note that EMA only affects the model that used for testing, it does not affect how weight updates are calculated during training.

Data Augmentations. We use random-horizontal flips and random-pad-crop (padding of 4 pixels on each side) as augmentations for the CIFAR-10 and CIFAR-100 datasets. Additionally, Mixup (Zhang et al., 2018) across both labeled and unlabeled samples was used. Pseudo-labels of the unlabeled samples where computed using the the average softmax predictions across multiple augmentations of the same image, as explained in §3.6. Note that this step is necessary in order to apply Mixup between labeled and unlabeled samples.

No data augmentations were used for MNIST.

B Additional Experiments

B.1 Unsupervised transfer learning on MNIST

For all our MNIST experiments, the reference prior is of order $n = 2$ and has $K = 50$ particles. We run our methods for 1024 epochs.

We first compare Deep reference priors with fine-tuning for transfer learning. The posterior (9) seeks to find a model that is influenced by the target data to varying degrees controlled by β, with $\beta = 1$ corresponding to the scenario where the posterior is maximally influenced by target data after being pre-trained on the source data. We instantiate (9), by combining prior selection, pre-training on the source task into one objective,

$$\max_{\pi} \gamma I_{\pi}(w; y^u, x^u) + (1 - \beta)\, \mathbb{E}_{w\sim\pi} \log p(w; y^s \,|\, x^s), \tag{S-20}$$

where γ and β are hyper-parameters. Solving (S-20) requires no knowledge from target data labels, therefore the setting here is pure unsupervised clustering for target task dataset. We compare the deep reference unsupervised transfer to fine-tuning method which adapts a model trained on labelled source to the labelled target data. All samples from the source task (about 30,000 images across 5 classes) were used for both the reference prior and fine-tuning.

C Two-stage experiment for coin tossing

In §3.3, we consider a situation when we obtain data in two stages, first z^m, and then z^n. We propose a prior π^* ((7)) such that the posterior of the second stage makes maximal use of the new n samples. In this section, we visualize π^* in the parameter space using a two-stage coin tossing experiment.

Method	Number of labeled target task data				
	0	50	100	250	500
Fine-Tuning	-	71.1	78.8	86.6	93.0
Deep Reference Prior Unsupervised Transfer	87.4	-	-	-	-

Table S-5: Accuracy (%) of transfer from source task (digits 0–4) to the target task (digits 5–9). We see that transfer using source and unlabelled target data using Deep reference prior performs as well as fine tuning with labelled source data and 250 labelled target data. This indicates that our method is making effective use of both the labeled source data and unlabeled target data.

Method	Number of labeled target task data				
	0	50	100	250	500
Fine-Tuning	-	90.2	92.4	94.7	96.2
Deep Reference Prior Unsupervised Transfer	95.2	-	-	-	-

Table S-6: Accuracy (%) of transfer from source task (digits 5–9) to the target task (digits 0–4). We see that transfer using source and unlabelled target data using Deep reference prior performs better than fine tuning with labelled source data and 250 labelled target data. This indicates that our method is making effective use of both the labeled source data and unlabeled target data.

Consider the estimation of the bias of a coin $w \in [0, 1]$ using two-stage $m + n$ trials. If z denotes the number of heads in total, we have $p(z \mid w) = w^z(1 - w)^{m+n-z}(m + n)!/(z!(m + n - z)!)$. There are m trials in first stage and n trails in second stage. We numerically find π^* for different values of m and n using the BA algorithm (Fig. S-10 and Fig. S-11).

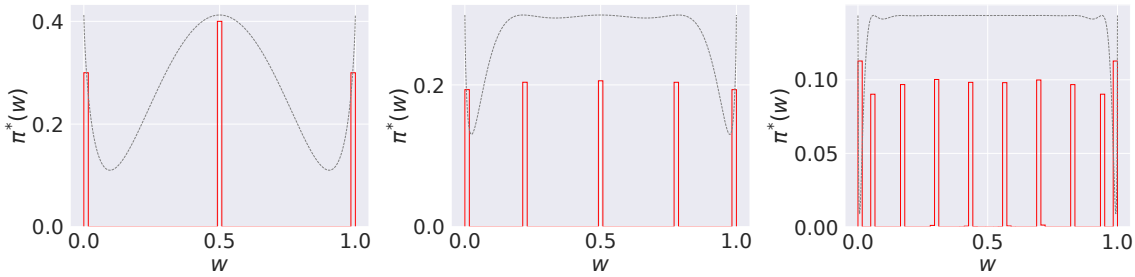


Figure S-10: Reference prior for the two stage coin-tossing model(see (7)), for $m = 1$ and $n = 1, 10, 40$ (from left to right) computed using the Blahut-Arimoto algorithm. Atoms are critical points of the gray line which is $\mathrm{KL}(p(z^{m+n}), p(z^{m+n} \mid w)) - \mathrm{KL}(p(z^m), p(z^m \mid w))$. The prior is discrete for finite order $n < \infty$, also see Mattingly et al. (2018). We now see how this reference prior behaves for different values of $\alpha = m/n$, e.g., for $\alpha \to 0$ this prior π^* is closer with π_n^* in (2), but still remains differences with π_n^*.

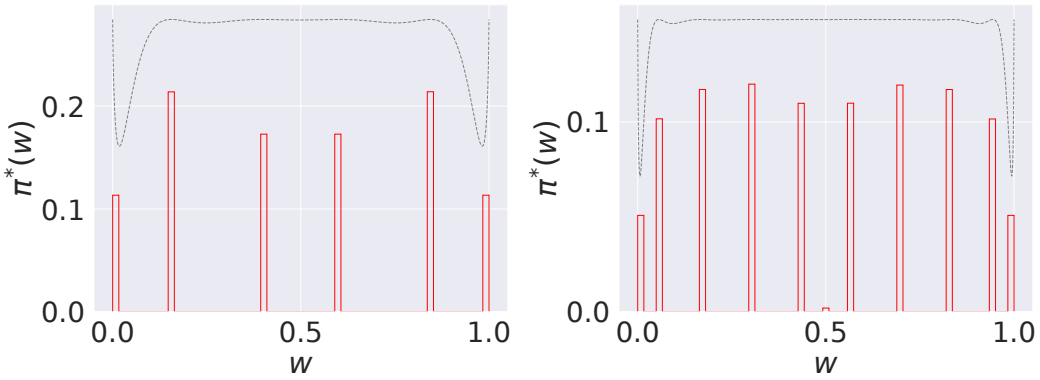


Figure S-11: Reference prior for the two stage coin-tossing model(see (7)), for $n = 1$ and $m = 10, 30$ (from left to right) computed using the Blahut-Arimoto algorithm. Atoms are critical points of the gray line which is $\mathrm{KL}(p(z^{m+n}), p(z^{m+n} \mid w)) - \mathrm{KL}(p(z^m), p(z^m \mid w))$. The prior is discrete for finite order $n < \infty$, also see Mattingly et al. (2018).