
Semi-supervised Deep Kernel Learning: Regression with Unlabeled Data by Minimizing Predictive Variance

Neal Jean*, Sang Michael Xie*, Stefano Ermon
Department of Computer Science
Stanford University
Stanford, CA 94305
{nealjean, xie, ermon}@cs.stanford.edu

Abstract

Large amounts of labeled data are typically required to train deep learning models. For many real-world problems, however, acquiring additional data can be expensive or even impossible. We present semi-supervised deep kernel learning (SSDKL), a semi-supervised regression model based on minimizing predictive variance in the posterior regularization framework. SSDKL combines the hierarchical representation learning of neural networks with the probabilistic modeling capabilities of Gaussian processes. By leveraging unlabeled data, we show improvements on a diverse set of real-world regression tasks over supervised deep kernel learning and semi-supervised methods such as VAT and mean teacher adapted for regression.

1 Introduction

The prevailing trend in machine learning is to automatically discover good feature representations through end-to-end optimization of neural networks. However, most success stories have been enabled by vast quantities of labeled data [1]. This need for supervision poses a major challenge when we encounter critical scientific and societal problems where fine-grained labels are difficult to obtain. Accurately measuring the outcomes that we care about—e.g., childhood mortality, environmental damage, or extreme poverty—can be prohibitively expensive. Although these problems have limited data, they often contain underlying structure that can be used for learning; for example, poverty is strongly correlated over both space and time.

Semi-supervised learning approaches offer promise when few labels are available by allowing models to supplement their training with unlabeled data [2]. Mostly focusing on classification tasks, these methods often rely on strong assumptions about the structure of the data (e.g., cluster assumptions, low data density at decision boundaries) that generally do not apply to regression [3, 4, 5].

In this paper, we present semi-supervised deep kernel learning, which addresses the challenge of semi-supervised regression by building on previous work combining the feature learning capabilities of deep neural networks with the ability of Gaussian processes to capture uncertainty [6]. SSDKL incorporates unlabeled training data by minimizing predictive variance in the posterior regularization framework, a flexible way of encoding prior knowledge in Bayesian models [7, 8].

Our main contributions are the following:

- We introduce semi-supervised deep kernel learning (SSDKL), a regression model that combines the strengths of heavily parameterized deep neural networks and nonparametric

*denotes equal contribution

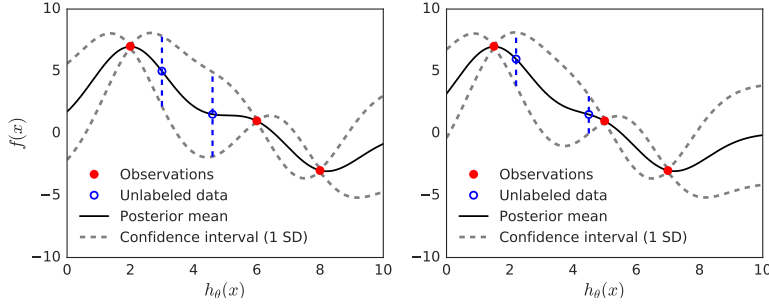


Figure 1: Depiction of the variance minimization approach behind semi-supervised deep kernel learning (SSDKL). The x -axis represents one dimension of a neural network embedding and the y -axis represents the corresponding output. **Left:** Without unlabeled data, the model learns an embedding by maximizing the likelihood of labeled data. The black and gray dotted lines show the posterior distribution after conditioning. **Right:** Embedding learned by SSDKL tries to minimize the predictive variance of unlabeled data, encouraging unlabeled embeddings to be near labeled embeddings. Observe that the representations of both labeled and unlabeled data are free to change.

Gaussian processes. While the deep Gaussian process kernel induces structure in an embedding space, the model also allows *a priori* knowledge of structure (i.e., spatial or temporal) in the input features to be naturally incorporated through kernel composition.

- By formalizing the semi-supervised variance minimization objective in the posterior regularization framework, we unify previous semi-supervised approaches such as minimum entropy and minimum variance regularization under a common framework. To our knowledge, this is the first paper connecting semi-supervised methods to posterior regularization.
- We demonstrate that SSDKL can use unlabeled data to learn more generalizable features and improve performance on a range of regression tasks, outperforming the supervised deep kernel learning method and semi-supervised methods such as virtual adversarial training (VAT) and mean teacher [9, 10]. In a challenging real-world task of predicting poverty from satellite images, SSDKL outperforms the state-of-the-art by 14.9% on average—by incorporating prior knowledge of spatial structure, the improvement increases to 18.4%.

2 Preliminaries

We assume a training set of n labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and m unlabeled examples $\{\mathbf{x}_j\}_{j=n+1}^{n+m}$ with instances $\mathbf{x} \in \mathbb{R}^d$ and labels $y \in \mathbb{R}$. Let X_L, \mathbf{y}_L, X_U refer to the aggregated features and targets, where $X_L \in \mathbb{R}^{n \times d}$, $\mathbf{y}_L \in \mathbb{R}^n$, and $X_U \in \mathbb{R}^{m \times d}$. At test time, we are given examples $X_T \in \mathbb{R}^{t \times d}$ that we would like to predict.

We will consider both inductive and transductive semi-supervised learning. In *inductive* semi-supervised learning, the labeled data (X_L, \mathbf{y}_L) and unlabeled data X_U are used to learn a function $f : \mathcal{X} \mapsto \mathcal{Y}$ that generalizes well and is a good predictor on unseen test examples X_T [2]. In *transductive* semi-supervised learning, the unlabeled examples are exactly the test data that we would like to predict, i.e., $X_T = X_U$ [11]. A transductive learning approach tries to find a function $f : \mathcal{X}^{n+m} \mapsto \mathcal{Y}^{n+m}$, with no requirement of generalizing to additional test examples.

Gaussian processes A Gaussian process (GP) is a collection of random variables, any finite number of which form a multivariate Gaussian distribution. Following the notation of [12], a Gaussian process defines a distribution over functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$ from inputs to target values. If

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k_\phi(\mathbf{x}_i, \mathbf{x}_j))$$

with mean function $\mu(\mathbf{x})$ and covariance kernel function $k_\phi(\mathbf{x}_i, \mathbf{x}_j)$ parameterized by ϕ , then any collection of function values is jointly Gaussian,

$$f(X) = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T \sim \mathcal{N}(\boldsymbol{\mu}, K_{X,X}),$$

with mean vector and covariance matrix defined by the GP, s.t. $\boldsymbol{\mu}_i = \mu(\mathbf{x}_i)$ and $(K_{X,X})_{ij} = k_\phi(\mathbf{x}_i, \mathbf{x}_j)$. In practice, we often assume that observations include i.i.d. Gaussian noise, i.e., $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x})$ where $\epsilon \sim \mathcal{N}(0, \phi_n^2)$, and the covariance function becomes

$$\text{Cov}(y(\mathbf{x}_i), y(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j) + \phi_n^2 \delta_{ij}$$

where $\delta_{ij} = \mathbb{I}[i = j]$. To make predictions at unlabeled points X_U , we can compute a Gaussian posterior distribution in closed form by conditioning on the observed data (X_L, \mathbf{y}_L) . For a more thorough introduction, we refer readers to [13].

Deep kernel learning Deep kernel learning (DKL) combines neural networks with GPs by using a neural network embedding as input to a deep kernel [6]. Given input data $\mathbf{x} \in \mathcal{X}$, a neural network parameterized by w is used to extract features $h_w(\mathbf{x}) \in \mathbb{R}^p$. The outputs are modeled as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(h_w(\mathbf{x})), k_\phi(h_w(\mathbf{x}_i), h_w(\mathbf{x}_j)))$$

for some mean function $\mu(\cdot)$ and base kernel function $k_\phi(\cdot, \cdot)$ with parameters ϕ . Parameters $\theta = (w, \phi)$ of the deep kernel are learned jointly by minimizing the negative log likelihood of the labeled data [12]:

$$L_{\text{likelihood}}(\theta) = -\log p(\mathbf{y}_L | X_L, \theta) \quad (1)$$

For Gaussian distributions, the marginal likelihood is a closed-form, differentiable expression, allowing DKL models to be trained via backpropagation.

Posterior regularization In probabilistic models, domain knowledge is generally imposed through the specification of priors. These priors, along with the observed data, determine the posterior distribution through the application of Bayes' rule. However, it can be difficult to encode our knowledge in a Bayesian prior. *Posterior regularization* offers a more direct and flexible mechanism for controlling the posterior distribution.

Let $\mathcal{D} = (X_L, \mathbf{y}_L)$ be a collection of observed data. [8] present a regularized optimization formulation called *regularized Bayesian inference*, or RegBayes. In this framework, the regularized posterior is the solution of the following optimization problem:

$$\text{RegBayes: } \inf_{q(M|\mathcal{D}) \in \mathcal{P}_{\text{prob}}} \mathcal{L}(q(M|\mathcal{D})) + \Omega(q(M|\mathcal{D})) \quad (2)$$

where $\mathcal{L}(q(M|\mathcal{D}))$ is defined as the KL-divergence between the desired post-data posterior $q(M|\mathcal{D})$ over models M and the standard Bayesian posterior $p(M|\mathcal{D})$, $D_{KL}(q||p)$. The goal is to learn a posterior distribution that is not too far from the standard Bayesian posterior while also fulfilling some requirements imposed by the regularization.

3 Semi-supervised deep kernel learning

We introduce *semi-supervised deep kernel learning* (SSDKL) for problems where labeled data is limited but unlabeled data is plentiful. To learn from unlabeled data, we observe that a Bayesian approach provides us with a predictive posterior *distribution*—i.e., we are able to quantify predictive uncertainty. Thus, we regularize the posterior by adding an unsupervised loss term that minimizes the predictive variance at unlabeled data points:

$$L_{\text{semisup}}(\theta) = \frac{1}{n} L_{\text{likelihood}}(\theta) + \frac{\alpha}{m} L_{\text{variance}}(\theta) \quad (3)$$

$$L_{\text{variance}}(\theta) = \sum_{x \in X_U} \text{Var}(f(x)) \quad (4)$$

where n and m are the numbers of labeled and unlabeled training examples and α is a hyperparameter controlling the trade-off between supervised and unsupervised components.

3.1 Variance minimization as posterior regularization

Let $X = (X_L, X_U)$ be the observed input data and $\mathcal{D} = (X, \mathbf{y}_L)$ be the input data with labels for the labeled data X_L . Let \mathcal{F} denote a space of functions where for $f \in \mathcal{F}$, $f: \mathbb{R}^d \rightarrow \mathbb{R}$ maps from the

inputs to the target values, and let $q(f|\mathcal{D}, \theta) \in \mathcal{P}_{prob}$ be a posterior distribution over the functions, where $\theta \in \Theta$ are parameters of q from some parameter space Θ and $\mathcal{P}_{prob} = \{q : q(f, \theta|\mathcal{D}) = q(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})\}$ is a family of distributions where $q(\theta|\mathcal{D})$ is restricted to be a Dirac delta centered on $\bar{\theta} \in \Theta$. We assume that a likelihood density $p(\cdot|X)$ exists and let $q^*(f, \theta|\mathcal{D}) = p(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})$, where p is the Bayesian posterior.

Instead of maximizing the marginal likelihood of the labeled training data in a purely supervised approach, we train our model in a semi-supervised fashion by minimizing the compound objective

$$L_{semisup}(\bar{\theta}) = -\frac{1}{n} \log p(\mathbf{y}_L | X_L, \bar{\theta}) + \frac{\alpha}{m} \sum_{x \in X_U} \text{Var}_{f \sim q^*}(f(x)) \quad (5)$$

where α controls the trade-off between supervised and unsupervised components.

This semi-supervised variance minimization objective is a specific form of posterior regularization in the RegBayes framework. As in [8], we assume that \mathcal{F} is a complete separable metric space and Π is an absolutely continuous probability measure (with respect to background measure η) on $(\mathcal{F}, \mathcal{B}(\mathcal{F}))$, where $\mathcal{B}(\mathcal{F})$ is the the Borel σ -algebra, such that a density π exists where $d\Pi = \pi d\eta$.

Theorem 1. *Given \mathcal{D} , a suitable space of functions \mathcal{F} , and parameter space Θ , the semi-supervised variance minimization problem (5)*

$$\inf_{\bar{\theta}} L_{semisup}(\bar{\theta})$$

is equivalent to the RegBayes optimization problem (2)

$$\inf_{q(f, \theta|\mathcal{D}) \in \mathcal{P}_{prob}} \mathcal{L}(q(f, \theta|\mathcal{D})) + \Omega(q(f, \theta|\mathcal{D}))$$

$$\Omega(q(f, \theta|\mathcal{D})) = \alpha' \sum_{i=1}^m \left(\int_{\mathcal{F}} (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 \cdot \int_{\Theta} q(\theta|\mathcal{D}) p(f|\theta, \mathcal{D}) d\eta(f, \theta) \right),$$

where $\alpha' = \frac{\alpha n}{m}$, and $\mathcal{P}_{prob} = \{q : q(f, \theta|\mathcal{D}) = q(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})\}$ is a family of distributions where $q(\theta|\mathcal{D})$ is restricted to be a Dirac delta centered on $\bar{\theta} \in \Theta$.

We include a formal derivation in Appendix A.1 and give a brief outline here. It can be shown that solving the variational optimization objective

$$\inf_{q(f, \theta|\mathcal{D})} D_{KL}(q(f, \theta|\mathcal{D}) || p(f, \theta|X)) - \int_{f, \theta} q(f, \theta|\mathcal{D}) \log p(\mathbf{y}_L | f, \theta, X) d\eta(f, \theta)$$

is equivalent to minimizing the first term of the RegBayes objective in Theorem 1, and the minimizer is precisely the Bayesian posterior $p(f, \theta|\mathcal{D})$. When we restrict $q \in \mathcal{P}_{prob}$, given any $\bar{\theta}$, the optimizing value is in the form $q^*(f, \theta|\mathcal{D}) = p(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})$ where p is the Bayesian posterior.

In general, the optimal post-data posterior q (after regularization) may be in a different form than the Bayesian posterior. However, note that the variance regularizer only depends on $q(\theta|\mathcal{D}) = \delta_{\bar{\theta}}(\theta|\mathcal{D})$. In this case, the optimal post-data posterior q in the regularized objective is still of the form $q^*(f, \theta|\mathcal{D}) = p(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})$, and q is modified by the regularization function only through changing $\bar{\theta}$. From here, we can recover the variance minimization objective.

Intuition for variance minimization By minimizing $L_{semisup}$, we trade off maximizing the likelihood of our observations with minimizing the posterior variance on unlabeled data that we wish to predict. Since the deep kernel parameters are jointly learned, the neural net is encouraged to learn a feature representation in which the unlabeled examples are similar to the labeled examples, thereby reducing the variance on our predictions. If we imagine the labeled data as ‘‘supports’’ for the surface representing the posterior mean, we are optimizing for embeddings where unlabeled data tend to cluster around these labeled supports.

Another interpretation is that the semi-supervised objective is a regularizer that reduces overfitting to labeled data. The model is discouraged from learning features from labeled data that are not also useful for making low-variance predictions at unlabeled data points. In settings where unlabeled data provide additional variation beyond labeled examples, this can improve model generalization.

Training Semi-supervised deep kernel learning scales well with large amounts of unlabeled data since the unsupervised objective L_{variance} naturally decomposes into a sum over conditionally independent terms. This allows for mini-batch training on unlabeled data with stochastic gradient descent. Since all of the labeled examples are interdependent, computing exact gradients for labeled examples requires full batch gradient descent on the labeled data. However, previous work in approximate gradients in GPs using structured or sparse matrices allows for stochastic batched training and can directly be applied in our model, allowing SSDKL to scale with respect to both labeled and unlabeled data [14, 15, 16].

4 Experiments and results

We apply SSDKL to a variety of real-world regression tasks, beginning with eight datasets from the UCI repository [17]. We also explore the novel and challenging task of predicting local poverty measures from high-resolution satellite imagery [18]. In our reported results, we use the squared exponential or radial basis function kernel. We also experimented with polynomial kernels, but it had generally worse performance. Additional training details are provided in Appendix A.3.

4.1 Baselines

We first compare SSDKL to the purely supervised DKL, showing the contribution of unlabeled data. Following previous work, the DKL model is initialized from a NN+GP model that holds the weights of a pre-trained neural network fixed while optimizing the parameters of the GP.

In addition to the supervised DKL method, we compare against semi-supervised methods including co-training, consistency regularization, generative models, and label propagation. Since many methods are originally for semi-supervised classification, we adapt them for regression.

COREG, or CO-training REGressors, uses two k -nearest neighbor (k NN) regressors, each of which generates labels for the other during the learning process [19]. Unlike traditional co-training, which requires splitting features into sufficient and redundant views, COREG achieves regressor diversity by using different distance metrics for its two regressors [20].

Consistency regularization methods aim to make model outputs invariant to local input perturbations [9, 21, 10]. For semi-supervised classification, [22] found that VAT and mean teacher were the best methods using fair evaluation guidelines. Virtual adversarial training (VAT) via local distributional smoothing (LDS) enforces consistency by training models to be robust to adversarial local input perturbations [9, 23]. Unlike adversarial training [24], the *virtual* adversarial perturbation is found without labels, making semi-supervised learning possible. We adapt VAT for regression by choosing the output distribution $\mathcal{N}(h_{\theta}(\mathbf{x}), \sigma^2)$ for input \mathbf{x} , where $h_{\theta} : \mathbb{R}^d \rightarrow \mathbb{R}$ is a parameterized mapping and σ is fixed. Optimizing the likelihood term is then equivalent to minimizing squared error; the LDS term is the KL-divergence between the model distribution and a perturbed Gaussian. (see Appendix A.2). Mean teacher enforces consistency by penalizing deviation from the outputs of a model with the exponential weighted average of the parameters over SGD iterations [10].

Label propagation defines a graph structure over the data with edges that define the probability for a categorical label to propagate from one data point to another [25]. If we encode this graph in a transition matrix T and let the current class probabilities be y , then the algorithm iteratively propagates $y \leftarrow Ty$, row-normalizes y , clamps the labeled data to their known values, and repeats until convergence. We make the extension to regression by letting y be real-valued labels. As in [25], we use a fully-connected graph and the radial-basis kernel for edge weights. The kernel scale hyperparameter is chosen using a validation set.

Generative models such as the variational autoencoder (VAE) have shown promise in semi-supervised classification especially for visual and sequential tasks [26, 27, 28, 29]. We compare against a semi-supervised VAE by first learning an unsupervised embedding of the data and then using the embeddings as input to a supervised multilayer perceptron.

4.2 UCI regression experiments

We evaluate SSDKL on eight regression datasets from the UCI repository. For each dataset, we train on $n = \{100, 300\}$ labeled examples, retain 1000 examples as the hold out test set, and treat the

Dataset	N	d	Percent reduction in RMSE compared to DKL											
			$n = 100$						$n = 300$					
			SSDKL	COREG	Label Prop	VAT	Mean Teacher	VAE	SSDKL	COREG	Label Prop	VAT	Mean Teacher	VAE
Skilcraft	3,325	18	7.7	7.5	8.5	1.4	-15.4	-7.4	1.7	-6.4	-3.6	-7.0	-29.9	-23.4
Parkinsons	5,875	20	24.9	15.4	-5.6	-38.5	-41.0	-79.6	19.6	7.1	-22.9	-51.6	-88.9	-149.6
Elevators	16,599	18	1.9	-6.7	26.8	-3.5	4.7	6.9	1.8	-8.5	27.5	-10.7	-8.1	9.0
Protein	45,730	9	3.4	0.6	1.87	-2.5	-3.9	-6.4	1.4	-0.7	-0.2	-3.3	-12.1	-13.0
Blog	52,397	280	3.7	8.6	6.1	8.4	4.1	10.0	-0.3	6.8	7.9	7.3	3.1	5.6
CTslice	53,500	384	15.5	36.7	20.5	18.6	-9.0	-28.5	32.9	42.2	26.8	27.5	-13.4	-42.9
Buzz	583,250	77	13.7	24.8	10.3	-2.4	-60.9	-24.8	15.7	17.9	4.6	-21.3	-91.3	-29.3
Electric	2,049,280	6	43.7	-17.5	-78.6	-120.6	-196.5	-272.4	17.8	-76.8	-218.7	-241.6	-475.3	-646.1
Average			14.3	8.7	-1.3	-17.4	-39.8	-50.3	11.3	-2.3	-22.3	-37.6	-89.5	-111.2

Table 1: Percent reduction in RMSE compared to baseline supervised deep kernel learning (DKL) model for semi-supervised deep kernel learning (SSDKL), COREG, label propagation, virtual adversarial training (VAT), mean teacher, and variational auto-encoder (VAE) models. Results are averaged across 10 trials for each UCI regression dataset. Here N is the total number of examples, d is the input feature dimension, and n is the number of labeled training examples. Final row shows average percent reduction in RMSE achieved by using unlabeled data.

remaining data as unlabeled examples. Following [22], the labeled data is randomly split 90-10 into training and validation samples, giving a realistically small validation set. We use the validation set for hyperparameter search and early stopping. We report test RMSE averaged over 10 trials of random splits to combat the small data sizes. Following [12], we choose a neural network with a similar $[d-100-50-50-2]$ architecture and two-dimensional embedding. We reduced the size of the lower layers since our labeled training sets are much smaller, but similarly to [12], results were not sensitive to these choices. Following [22], we use this same base model for all deep models, including SSDKL, DKL, VAT, mean teacher, and the VAE encoder, in order to make results comparable across methods. Since label propagation creates a kernel matrix of all data points, we limit the number of unlabeled examples for label propagation to a maximum of 20000 due to memory constraints. We initialize labels in label propagation with a kNN regressor with $k = 5$ to speed up convergence.

SSDKL performs at least as well as DKL across all datasets, and a Wilcoxon signed-rank test shows significance at the $p = 0.05$ level for at least one labeled training set size for 6 of the 8 datasets. SSDKL gives a 14.3% and 11.3% average RMSE improvement over the supervised DKL in the $n = 100, 300$ cases respectively, superior to other semi-supervised methods adapted for regression.

The same hyperparameters and initializations are used across *all* UCI datasets for SSDKL. We use learning rates of 1×10^{-3} and 0.1 for the neural network and GP parameters respectively and initialize all GP parameters to 1. In Fig. 2 (**right**), we study the effect of varying α to trade off between maximizing the likelihood of labeled data and minimizing the variance of unlabeled data. A large α emphasizes minimization of the predictive variance while a small α focuses on fitting labeled

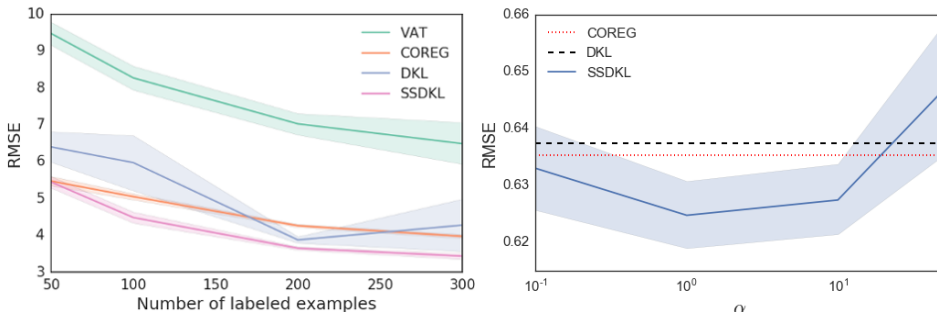


Figure 2: **Left:** Average test RMSE vs. number of labeled examples, $n = \{50, 100, 200, 300\}$, for UCI Parkinsons dataset. SSDKL outperforms supervised DKL, co-training regressors (COREG), and virtual adversarial training (VAT). **Right:** SSDKL performance on poverty prediction (section 4.3) as a function of α , which controls the trade-off between labeled and unlabeled objectives, for $n = 300$. The dotted lines plot the performance of DKL and COREG. VAT performed worse and is excluded for clarity. All results averaged over 10 trials. All shading represents one standard deviation.

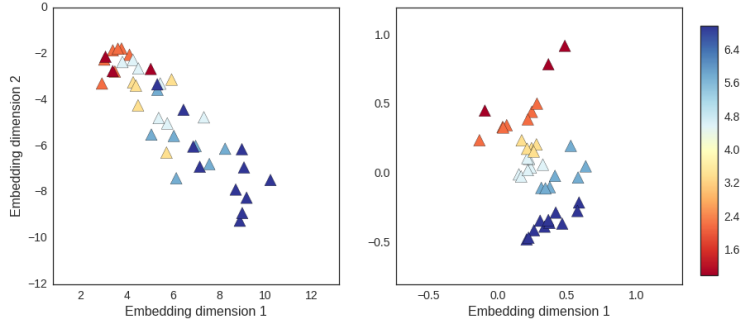


Figure 3: **Left:** Two-dimensional embeddings learned by supervised deep kernel learning (DKL) model on the Skillcraft dataset using $n = 50$ labeled training examples. The colorbar shows the magnitude of the outputs. **Right:** Embeddings learned by semi-supervised deep kernel learning (SSDKL) model using the same 50 labeled training examples plus an additional $m = 1000$ unlabeled examples. By using unlabeled data for regularization, SSDKL learns a better representation.

data. SSDKL improves on DKL for values of α between 0.1 and 10.0, indicating that performance is not overly reliant on the choice of this hyperparameter. Fig. 2 (**left**) compares SSDKL to purely supervised DKL, COREG, and VAT as we vary the labeled training set size.

Surprisingly, COREG outperformed SSDKL on the Blog, CTslice, and Buzz datasets. We found that these datasets happen to be better-suited for nearest neighbors-based methods such as COREG. A kNN regressor using only the labeled data outperformed DKL and SSDKL on all three datasets for $n = 100$, beat DKL on all three for $n = 300$, and beat SSDKL on two of three for $n = 300$. Thus, the kNN regressor is already outperforming SSDKL with only labeled data—it is unsurprising that SSDKL is unable to close the gap on a semi-supervised nearest neighbors method like COREG.

Representation learning To gain some intuition about how the unlabeled data helps in the learning process, we visualize the neural network embeddings learned by the DKL and SSDKL models on the Skillcraft dataset. In Fig. 3 (**left**), we first train DKL on $n = 50$ labeled training examples and plot the two-dimensional neural network embedding that is learned. In Fig. 3 (**right**), we train SSDKL on the same $n = 50$ labeled training examples along with $m = 1000$ additional unlabeled data points and plot the resulting embedding. In the left panel, DKL learns a poor embedding—different colors representing different output magnitudes are intermingled. In the right panel, SSDKL is able to use the unlabeled data for regularization, and learns a better representation of the dataset.

4.3 Poverty prediction

High-resolution satellite imagery offers the potential for cheap, scalable, and accurate tracking of changing socioeconomic indicators. In this task, we predict local poverty measures from satellite images using limited amounts of poverty labels. As described in [30], the dataset consists of 3,066

Percent reduction in RMSE ($n = 300$)			
Country	Spatial SSDKL	SSDKL	DKL
Malawi	13.7	16.4	15.7
Nigeria	17.9	4.6	1.7
Tanzania	10.0	15.5	9.2
Uganda	25.2	12.1	13.8
Rwanda	27.0	25.4	21.3
Average	18.4	14.9	12.5

Table 2: Percent RMSE reduction in a poverty measure prediction task compared to baseline ridge regression model used in [30]. SSDKL and DKL models use only satellite image data. Spatial SSDKL incorporates both location and image data through kernel composition. Final row shows average RMSE reduction of each model averaged over 10 trials.

villages across five Africa countries: Nigeria, Tanzania, Uganda, Malawi, and Rwanda. These include some of the poorest countries in the world (Malawi and Rwanda) as well as some that are relatively better off (Nigeria), making for a challenging and realistically diverse problem.

In this experiment, we use $n = 300$ labeled satellite images for training. With such a small dataset, we can not expect to train a deep convolutional neural network (CNN) from scratch. Instead we take a transfer learning approach as in [18], extracting 4096-dimensional visual features and using these as input. More details are provided in Appendix A.4.

Incorporating spatial information In order to highlight the usefulness of kernel composition, we explore extending SSDKL with a spatial kernel. Spatial SSDKL composes two kernels by summing an image feature kernel and a separate location kernel that operates on location coordinates (lat/lon). By treating them separately, it explicitly encodes the knowledge that location coordinates are spatially structured and distinct from image features.

As shown in Table 2, all models outperform the baseline state-of-the-art ridge regression model from [30]. Spatial SSDKL significantly outperforms all other models, while the concatenated kernel does not improve over the basic SSDKL model, which uses only image features. Spatial SSDKL outperforms the other models by directly modeling location coordinates as spatial features, showing that kernel composition can effectively incorporate prior knowledge of structure.

5 Related work

[31] introduced deep Gaussian processes, which stack GPs in a hierarchy by modeling the outputs of one layer with a Gaussian process in the next layer. Despite the suggestive name, these models do not integrate deep neural networks and Gaussian processes.

[6] proposed deep kernel learning, combining neural networks with the non-parametric flexibility of GPs and training end-to-end in a fully supervised setting. Extensions have explored approximate inference, stochastic gradient training, and recurrent deep kernels for sequential data [14, 15, 16].

Our method draws inspiration from transductive experimental design, which chooses the most informative points (experiments) to measure by seeking data points that are both hard to predict and informative for the unexplored test data [32]. Similar prediction uncertainty approaches have been explored in semi-supervised classification models, such as minimum entropy and minimum variance regularization, which can now also be understood in the posterior regularization framework [3, 33].

Recent work in generative adversarial networks (GANs) [26], variational autoencoders (VAEs) [27], and other generative models have achieved promising results on various semi-supervised classification tasks [28, 29]. However, we find that these models are not as well-suited for generic regression tasks such as in the UCI repository as for audio-visual tasks.

Consistency regularization posits that the model’s output should be invariant to reasonable perturbations of the input [9, 21, 10]. Combining adversarial training [24] with consistency regularization, virtual adversarial training uses a label-free regularization term that allows for semi-supervised training [9]. Mean teacher adds a regularization term that penalizes deviation from an exponential weighted average of the parameters over SGD iterations [10]. For semi-supervised classification, [22] found that VAT and mean teacher were the best methods across a series of fair evaluations.

Label propagation defines a graph structure over the data points and propagates labels from labeled data over the graph. The method must assume a graph structure and edge distances on the input feature space without the ability to adapt the space to the assumptions. Label propagation is also subject to memory constraints since it forms a kernel matrix over *all* data points, requiring quadratic space in general, although sparser graph structures can reduce this to a linear scaling.

Co-training regressors trains two kNN regressors with different distance metrics which label each others’ unlabeled data. This works when neighbors in the given input space are meaningful, but cannot adapt the space. As a fully non-parametric method, inference requires retaining the full dataset.

Much of the previous work in semi-supervised learning is in classification and the assumptions do not generally translate to regression. Our experiments show that SSDKL outperforms other adapted semi-supervised methods in a battery of regression tasks.

6 Conclusions

Many important problems are challenging because of the limited availability of training data, making the ability to learn from unlabeled data critical. In experiments with UCI datasets and a real-world poverty prediction task, we find that minimizing posterior variance can be an effective way to incorporate unlabeled data when labeled training data is scarce. SSDKL models are naturally suited for many real-world problems, as spatial and temporal structure can be explicitly modeled through the composition of kernel functions. While our focus is on regression problems, we believe the SSDKL framework is equally applicable to classification problems—we leave this to future work.

A Appendix

A.1 Posterior regularization

Proof of Theorem 1. Let $\mathcal{D} = (X_L, \mathbf{y}_L, X_U)$ be a collection of observed data. Let $X = (X_L, X_U)$ be the observed input data points. As in [8], we assume that \mathcal{F} is a complete separable metric space and Π is a absolutely continuous probability measure (with respect to background measure η) on $(\mathcal{F}, \mathcal{B}(\mathcal{F}))$, where $\mathcal{B}(\mathcal{F})$ is the the Borel σ -algebra, such that a density π exists where $d\Pi = \pi d\eta$. Let Θ be a space of parameters to the model, where we treat $\theta \in \Theta$ as random variables. With regards to the notation in the RegBayes framework, the model is the pair $M = (f, \theta)$. We assume as in [8] that the likelihood function $P(\cdot|M)$ is the likelihood distribution which is dominated by a σ -finite measure λ for all M with positive density, such that a density $p(\cdot|M)$ exists where $dP(\cdot|M) = p(\cdot|M)d\lambda$.

We would like to compute the posterior distribution

$$p(f, \theta|\mathcal{D}) = \frac{p(\mathcal{D}|f, \theta)\pi(f, \theta)}{\int_{f, \theta} p(f, \theta, \mathcal{D})d\eta(f, \theta)}$$

which involves an intractable integral. We introduce a variational approximation $q(f, \theta) \in \mathcal{P}_{prob}$ which approximates $p(f, \theta|\mathcal{D})$, where $\mathcal{P}_{prob} = q : q(f, \theta|\mathcal{D}) = q(f|\theta, \mathcal{D})\delta_{\bar{\theta}}(\theta|\mathcal{D})$ is a family of approximating distributions such that $q(\theta|\mathcal{D})$ is restricted to be a Dirac delta centered on $\bar{\theta}$. We claim that the exact optimal solution of the following optimization problem is precisely the Bayesian posterior $p(f, \theta|\mathcal{D})$:

$$\inf_{q(f, \theta|\mathcal{D})} D_{KL}(q(f, \theta|\mathcal{D})||p(f, \theta|X)) - \int_{f, \theta} q(f, \theta|\mathcal{D}) \log p(\mathbf{y}_L|f, \theta, X)d\eta(f, \theta).$$

We note that adding the constant $\log p(\mathcal{D})$ to the objective,

$$\begin{aligned} & \inf_{q(f, \theta|\mathcal{D})} D_{KL}(q(f, \theta|\mathcal{D})||p(f, \theta|X)) - \int_{f, \theta} q(f, \theta|\mathcal{D}) \log p(\mathbf{y}_L|f, \theta, X)d\eta(f, \theta) + \log p(\mathcal{D}) \\ &= \inf_{q(f, \theta|\mathcal{D})} \int_{f, \theta} q(f, \theta|\mathcal{D}) \log \frac{q(f, \theta|\mathcal{D})p(X)}{p(f, \theta, \mathbf{y}_L, X)}d\eta(f, \theta) + \log p(\mathcal{D}) \\ &= \inf_{q(f, \theta|\mathcal{D})} \int_{f, \theta} q(f, \theta|\mathcal{D}) \log \frac{q(f, \theta|\mathcal{D})}{p(f, \theta, \mathcal{D})}d\eta(f, \theta) + \log p(\mathcal{D}) \\ &= \inf_{q(f, \theta|\mathcal{D})} \int_{f, \theta} q(f, \theta|\mathcal{D}) \log \frac{q(f, \theta|\mathcal{D})}{p(f, \theta|\mathcal{D})}d\eta(f, \theta) \\ &= \inf_{q(f, \theta|\mathcal{D})} D_{KL}(q(f, \theta|\mathcal{D})||p(f, \theta|\mathcal{D})) \\ &= \inf_{q(f, \theta|\mathcal{D})} \mathcal{L}(q(f, \theta|\mathcal{D})), \end{aligned}$$

so that the claim holds, and we see that the objective is equivalent to the first term of the RegBayes objective (Section 2.3). When we restrict $q \in \mathcal{P}_{prob}$,

$$\inf_{q(f, \theta | \mathcal{D}) \in \mathcal{P}_{prob}} D_{KL}(q(f, \theta | \mathcal{D}) \| p(f, \theta | X)) - \int_{f, \theta} q(f, \theta | \mathcal{D}) \log p(\mathbf{y}_L | f, \theta, X) d\eta(f, \theta) \quad (6)$$

$$= \inf_{q(f | \theta, \mathcal{D}), \bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \int_f q(f | \theta, \mathcal{D}) \left(\log \frac{q(f | \theta, \mathcal{D}) \delta_{\bar{\theta}}(\theta)}{p(f, \theta | X)} - \log p(\mathbf{y}_L | f, \theta, X) \right) d\eta(f, \theta) \quad (7)$$

$$= \inf_{q(f | \theta, \mathcal{D}), \bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \int_f q(f | \theta, \mathcal{D}) \left(\log \frac{q(f | \theta, \mathcal{D})}{p(f, \theta | X)} - \log p(\mathbf{y}_L | f, \theta, X) \right) d\eta(f, \theta) \quad (8)$$

$$= \inf_{q(f | \theta, \mathcal{D}), \bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \int_f q(f | \theta, \mathcal{D}) \left(\log q(f | \theta, \mathcal{D}) - \log p(f, \theta | X) - \log p(f | \mathbf{y}_L, \theta, X) - \log p(\mathbf{y}_L, \theta, X) + \log p(f, \theta, X) \right) d\eta(f, \theta) \quad (9)$$

$$= \inf_{q(f | \theta, \mathcal{D}), \bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \int_f q(f | \theta, \mathcal{D}) \left(\log q(f | \theta, \mathcal{D}) - \log p(f, \theta | X) - \log p(f | \mathbf{y}_L, \theta, X) - \log p(\mathbf{y}_L, \theta | X) + \log p(f, \theta | X) \right) d\eta(f, \theta) \quad (10)$$

$$= \inf_{q(f | \theta, \mathcal{D}), \bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) (D_{KL}(q(f | \theta, \mathcal{D}) \| p(f | \mathbf{y}_L, \theta, X)) - \log p(\mathbf{y}_L, \theta | X)) \quad (11)$$

where in equation (8) we note that $\int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \log \delta_{\bar{\theta}}(\theta)$ does not vary with $\bar{\theta}$ or q , and can be removed from the optimization, and similarly in equation (10) we remove the constant $\log p(X)$. For every $\bar{\theta}$, the optimizing value is $q^*(f | \theta, \mathcal{D}) = p(f | \theta, X, \mathbf{y}_L) = p(f | \bar{\theta}, \mathcal{D})$, which is the Bayesian posterior given the model parameters. Substituting this optimal value into (11),

$$\inf_{\bar{\theta}} \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) (D_{KL}(q^*(f | \theta, \mathcal{D}) \| p(f | \mathbf{y}_L, \theta, X)) - \log p(\mathbf{y}_L, \theta | X)) \quad (12)$$

$$= \inf_{\bar{\theta}} - \int_{\bar{\theta}} \delta_{\bar{\theta}}(\theta) \log p(\mathbf{y}_L, \theta | X) \quad (13)$$

$$= \inf_{\bar{\theta}} - \log p(\mathbf{y}_L | \bar{\theta}, X) \quad (14)$$

using that $D_{KL}(q^*(f | \theta, \mathcal{D}) \| p(f | \mathbf{y}_L, \theta, X)) = 0$ in (13). The optimization problem over $\bar{\theta}$ reflects maximizing the likelihood of the data.

The regularization term in the RegBayes framework is expressed variationally as

$$\Omega(q(M | \mathcal{D})) = \inf_{\xi} U(\xi) \text{ s.t. } : q(M | \mathcal{D}) \in \mathcal{P}_{post}(\xi),$$

where ξ are slack variables, $U(\xi)$ is a penalty function, and $\mathcal{P}_{post}(\xi)$ is a subspace of feasible distributions satisfying specified constraints. An equivalent formulation of the RegBayes problem is then

$$\mathbf{RegBayes:} \quad \inf_{q(M | \mathcal{D}) \in \mathcal{P}_{post}(\xi), \xi} \mathcal{L}(q(M | \mathcal{D})) + U(\xi). \quad (15)$$

Let the regularization function be

$$\Omega(q(f, \theta | \mathcal{D})) = \inf_{\xi} U(\xi) \text{ s.t. } : q(f, \theta | \mathcal{D}) \in \mathcal{P}_{post}(\xi),$$

where $U(\xi) = \alpha' \sum_{i=1}^m \xi_i$, $\alpha' = \frac{\alpha n}{m}$, and q is restricted to the family of distributions

$$\mathcal{P}_{post} = \left\{ q : \forall i = 1, \dots, m, \left(\int_f (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 \int_{\theta} q(\theta | \mathcal{D}) p(f | \theta, \mathcal{D}) d\eta(f, \theta) \right) \leq \xi_i, q \in \mathcal{P}_{prob} \right\}$$

where $p(f | \theta, \mathcal{D})$ is the Bayesian posterior from the unregularized objective. Note that given values of q and $\bar{\theta}$, the optimal $\xi_i^* = \int_f (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 \int_{\theta} q(\theta | \mathcal{D}) p(f | \theta, \mathcal{D}) d\eta(\theta)$. Thus the regularization function is

$$\Omega(q(f, \theta | \mathcal{D})) = \alpha' \sum_{i=1}^m \left(\int_f (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 \int_{\theta} q(\theta | \mathcal{D}) p(f | \theta, \mathcal{D}) d\eta(f, \theta) \right).$$

Note that the regularization function only depends on $q(\theta | \mathcal{D}) = \delta_{\bar{\theta}}(\theta)$. Therefore the optimal post-data posterior q in the regularized objective is still in the form $q^*(f, \theta | \mathcal{D}) = p(f | \theta, \mathcal{D}) \delta_{\bar{\theta}}(\theta)$, and q is modified by the regularization function only through $\delta_{\bar{\theta}}(\theta)$.

Thus, augmenting the objective from (11) and using the optimal post-data posterior $q^*(f, \theta | \mathcal{D}) = p(f | \theta, \mathcal{D}) \delta_{\bar{\theta}}(\theta)$, the regularized optimization objective is

$$\begin{aligned}
& \inf_{q(f, \theta | \mathcal{D}) \in \mathcal{P}_{prob}} \mathcal{L}(q(f, \theta | \mathcal{D})) + \Omega(q(f, \theta | \mathcal{D})) \\
&= \inf_{\bar{\theta}} -\log p(\mathbf{y}_L | \bar{\theta}, X) + \alpha' \sum_{i=1}^m \left(\int_f (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 \int_{\theta} \delta_{\bar{\theta}}(\theta) p(f | \theta, \mathcal{D}) d\eta(f, \theta) \right) \\
&= \inf_{\bar{\theta}} -\log p(\mathbf{y}_L | \bar{\theta}, X) + \alpha' \sum_{i=1}^m \int_f p(f | \bar{\theta}, \mathcal{D}) (f(X_U)_i - \mathbb{E}_p[f(X_U)_i])^2 d\eta(f) \\
&= \inf_{\bar{\theta}} -\log p(\mathbf{y}_L | \bar{\theta}, X) + \alpha' \sum_{i=1}^m \int_f q^*(f | \bar{\theta}, \mathcal{D}) (f(X_U)_i - \mathbb{E}_{q^*}[f(X_U)_i])^2 d\eta(f) \\
&= \inf_{\bar{\theta}} -\log p(\mathbf{y}_L | \bar{\theta}, X) + \alpha' \sum_{i=1}^m \text{Var}_{q^*}(f((X_U)_i)) \\
&= \inf_{\bar{\theta}} L_{semisup}(\bar{\theta}).
\end{aligned}$$

where we use that $p(f | \bar{\theta}, \mathcal{D}) = q^*(f | \bar{\theta}, \mathcal{D})$ in the third-to-last equality. \square

A.2 Virtual Adversarial Training

Virtual adversarial training (VAT) is a general training mechanism which enforces *local distributional smoothness* (LDS) by optimizing the model to be less sensitive to adversarial perturbations of the input [9]. The VAT objective is to augment the marginal likelihood with an LDS objective:

$$\frac{1}{n} \sum_{i=1}^n \log p(\mathbf{y}_L | X_L, \bar{\theta}) + \frac{\lambda}{n} \sum_{i=1}^n \text{LDS}(X, \bar{\theta})$$

where

$$\begin{aligned}
& \text{LDS}(X_i, \theta) = -\Delta_{KL}(r_{v\text{-adv}(i)}, X_i, \theta) \\
& \Delta_{KL}(r, X_i, \theta) = D_{KL}(p(\mathbf{y} | X_i, \theta) \| p(\mathbf{y} | X_i + r, \theta)) \\
& r_{v\text{-adv}} = \arg \max_r \{ \Delta_{KL}(r, X_i, \theta); \|r\|_2 \leq \epsilon \}
\end{aligned}$$

Note that the LDS objective does not require labels, so that unlabeled data can be incorporated. The experiments in the original paper are for classification, although VAT is general. We use VAT for regression by choosing $p(\mathbf{y}_L | X_i, \bar{\theta}) = \mathcal{N}(h_{\theta}(X_i), \sigma^2)$ where $h_{\theta} : \mathbb{R}^d \rightarrow \mathbb{R}^H$ is a parameterized mapping (a neural network), and σ is fixed. Optimizing the likelihood term is then equivalent to minimizing the squared error and the LDS term is the KL-divergence between the model's Gaussian distribution and a perturbed Gaussian distribution, which is also in the form of a squared difference. The adversarial perturbation Δ_{KL} is calculated with the second-order Taylor approximated at each step using a first dominant eigenvector calculation of the Hessian. The eigenvector calculation is done via a finite-difference approximation to the power iteration method. As in [9], one step of the finite-difference approximation was used in all of our experiments.

A.3 Training details

In our reported results, we use the standard squared exponential or radial basis function (RBF) kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi_f^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\phi_l^2}\right),$$

where ϕ_f^2 and ϕ_l^2 represent the signal variance and characteristic length scale. We also experimented with polynomial kernels, $k(\mathbf{x}_i, \mathbf{x}_j) = (\phi_f \mathbf{x}_i^T \mathbf{x}_j + \phi_l)^p$, $p \in \mathbb{Z}_+$, but found that performance generally decreased. To enforce positivity constraints on the kernel parameters and positive definiteness of the covariance, we train these parameters in the log-domain. Although the information capacity

of a non-parametric model increases with the dataset size, the marginal likelihood automatically constrains model complexity without additional regularization [13]. The parametric neural networks are regularized with L2 weight decay to reduce overfitting, and models are implemented and trained in TensorFlow using the ADAM optimizer [34, 35].

A.4 Poverty prediction

High-resolution satellite imagery offers the potential for cheap, scalable, and accurate tracking of changing socioeconomic indicators. The United Nations has set 17 Sustainable Development Goals (SDGs) for the year 2030—the first of these is the worldwide elimination of extreme poverty, but a lack of reliable data makes it difficult to distribute aid and target interventions effectively. Traditional data collection methods such as large-scale household surveys or censuses are slow and expensive, requiring years to complete and costing billions of dollars [36]. Because data on the outputs that we care about are scarce, it is difficult to train models on satellite imagery using traditional supervised methods. In this task, we attempt to predict local poverty measures from satellite images using limited amounts of poverty labels. As described in [30], the dataset consists of 3,066 villages across five Africa countries: Nigeria, Tanzania, Uganda, Malawi, and Rwanda. These countries include some of the poorest in the world (Malawi, Rwanda) as well as regions of Africa that are relatively better off (Nigeria), making for a challenging and realistically diverse problem. The raw satellite inputs consist of 400×400 pixel RGB satellite images downloaded from Google Static Maps at zoom level 16, corresponding to 2.4 m ground resolution. The target variable that we attempt to predict is a wealth index provided in the publicly available Demographic and Health Surveys (DHS) [37, 38].

References

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [2] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- [3] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2004.
- [4] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, pages 57–64, 2005.
- [5] Aarti Singh, Robert Nowak, and Xiaojin Zhu. Unlabeled data: Now it helps, now it doesn’t. In *Advances in neural information processing systems*, pages 1513–1520, 2009.
- [6] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. *The Journal of Machine Learning Research*, 2015.
- [7] Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049, 2010.
- [8] Jun Zhu, Ning Chen, and Eric P Xing. Bayesian inference with posterior regularization and applications to infinite latent svms. *Journal of Machine Learning Research*, 15(1):1799–1847, 2014.
- [9] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.
- [10] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1195–1204. Curran Associates, Inc., 2017.
- [11] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. *Proc. Seventh IEEE Int’l Conf. Data Mining Workshops*, 2007.
- [12] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 370–378, 2016.

- [13] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.
- [14] Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (KISS-GP). In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1775–1784, 2015.
- [15] Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016.
- [16] Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu, and Eric P Xing. Learning scalable deep kernels with recurrent structure. *arXiv preprint arXiv:1610.08936*, 2016.
- [17] M. Lichman. UCI machine learning repository, 2013.
- [18] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *AAAI Conference on Artificial Intelligence*, 2016.
- [19] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005.
- [20] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [21] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ICLR 2017*.
- [22] Augustus Odena, Avital Oliver, Colin Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of semi-supervised learning algorithms. 2018.
- [23] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017.
- [24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [25] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.
- [26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [28] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.
- [29] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.
- [30] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.
- [31] Andreas C. Damianou and Neil D. Lawrence. Deep gaussian processes. *The Journal of Machine Learning Research*, 2013.
- [32] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. *The International Conference on Machine Learning (ICML)*, 2006.
- [33] Chenyang Zhao and Shaodan Zhai. Minimum variance semi-supervised boosting for multi-label classification. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1342–1346. IEEE, 2015.
- [34] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

- [35] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, 2015.
- [36] Morten Jerven. *Poor numbers: how we are misled by African development statistics and what to do about it*. Cornell University Press, 2013.
- [37] ICF International. Demographic and health surveys (various) [datasets], 2015.
- [38] Deon Filmer and Lant H Pritchett. Estimating wealth effects without expenditure data—or tears: An application to educational enrollments in states of india*. *Demography*, 38(1):115–132, 2001.