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

Zhang Qian

SMILE Lab, UESTC

hnnyzhqian@163.com

December 28, 2018

Contents 目录

1 Gaussian Processes

2 Deep kernel learning

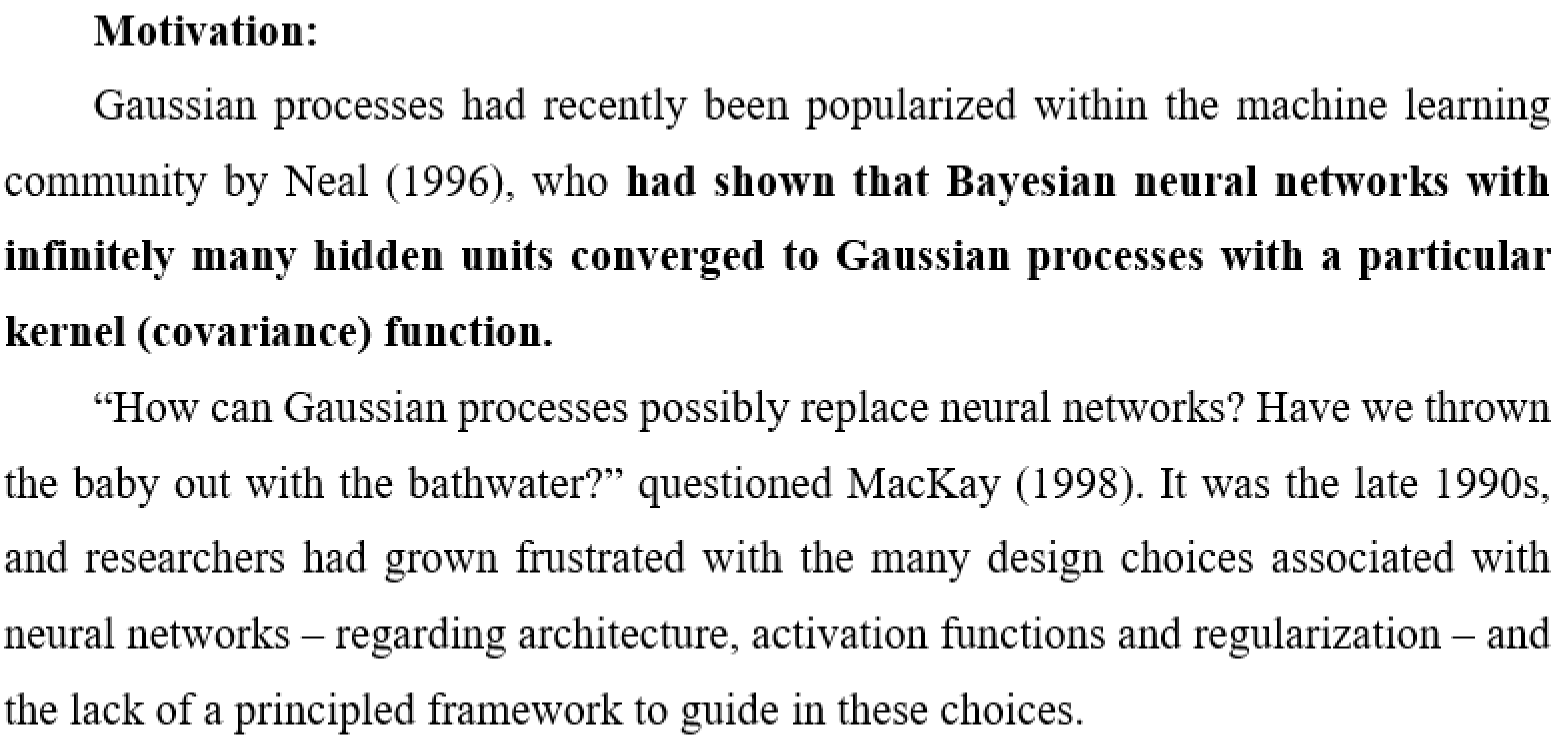
3 Posterior regularization

4 Semi-supervised DKL

NO.1

Gaussian Processes

Gaussian Processes

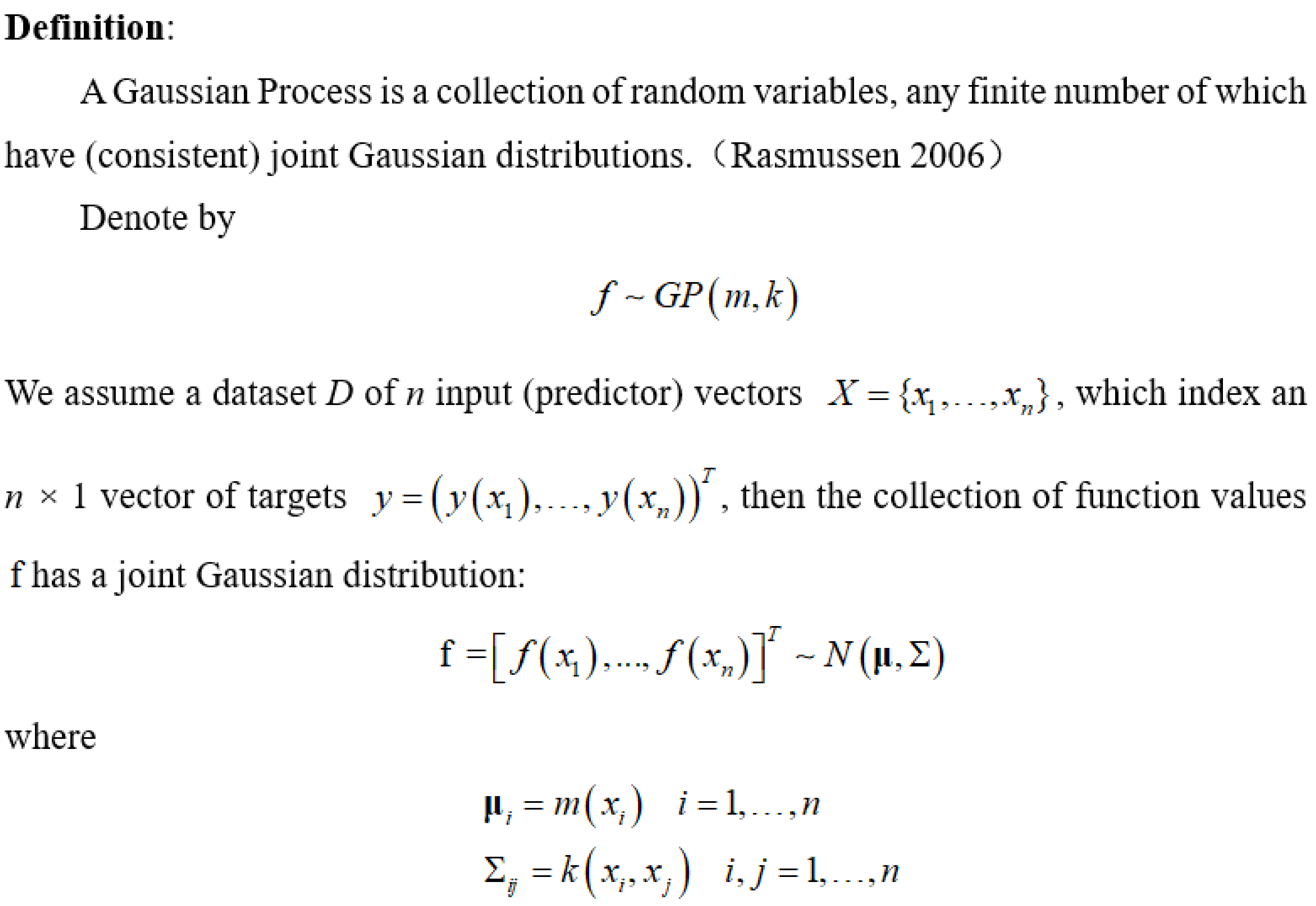


Motivation:

Gaussian processes had recently been popularized within the machine learning community by Neal (1996), who **had shown that Bayesian neural networks with infinitely many hidden units converged to Gaussian processes with a particular kernel (covariance) function.**

“How can Gaussian processes possibly replace neural networks? Have we thrown the baby out with the bathwater?” questioned MacKay (1998). It was the late 1990s, and researchers had grown frustrated with the many design choices associated with neural networks – regarding architecture, activation functions and regularization – and the lack of a principled framework to guide in these choices.

Gaussian Processes



Definition:

A Gaussian Process is a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions.（Rasmussen 2006）

Denote by

$$f \sim GP(m, k)$$

We assume a dataset D of n input (predictor) vectors $X = \{x_1, \ldots, x_n\}$, which index an n × 1 vector of targets $y = (y(x_1), \ldots, y(x_n))^T$, then the collection of function values f has a joint Gaussian distribution:

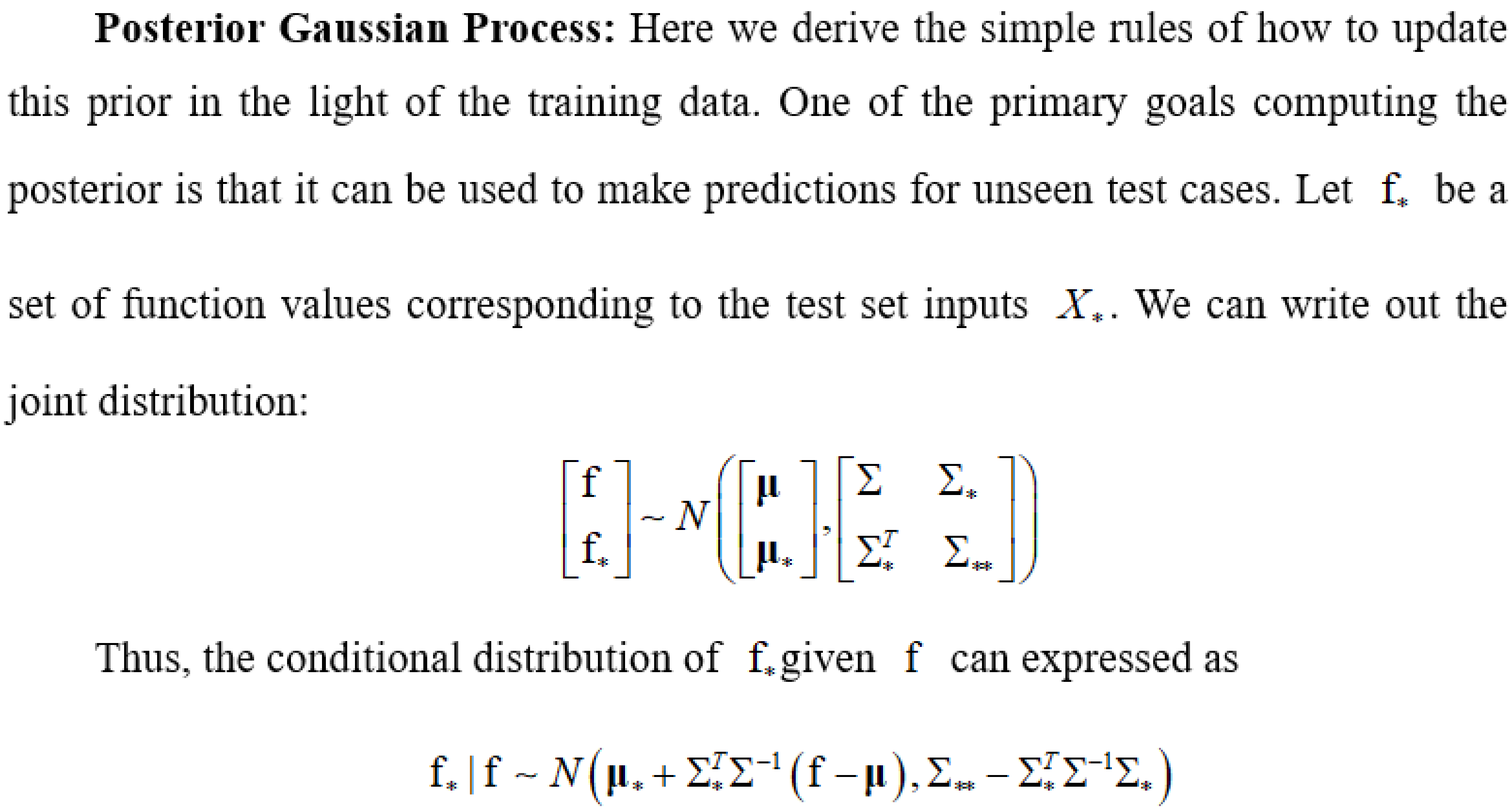
$$\mathrm{f} = [f(x_1), \ldots, f(x_n)]^T \sim N(\boldsymbol{\mu}, \Sigma)$$

where

$$\boldsymbol{\mu}_i = m(x_i) \quad i = 1, \ldots, n$$

$$\Sigma_{ij} = k(x_i, x_j) \quad i, j = 1, \ldots, n$$

Gaussian Processes



Posterior Gaussian Process: Here we derive the simple rules of how to update this prior in the light of the training data. One of the primary goals computing the posterior is that it can be used to make predictions for unseen test cases. Let f_* be a set of function values corresponding to the test set inputs X_*. We can write out the joint distribution:

$$\begin{bmatrix} \mathrm{f} \\ \mathrm{f}_* \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix} \right)$$

Thus, the conditional distribution of f_*given f can expressed as

$$\mathrm{f}_* \,|\, \mathrm{f} \sim N\left(\boldsymbol{\mu}_* + \Sigma_*^T \Sigma^{-1} \left(\mathrm{f} - \boldsymbol{\mu} \right), \Sigma_{**} - \Sigma_*^T \Sigma^{-1} \Sigma_* \right)$$

Gaussian Processes

That is the posterior distribution for a specific set of test cases. It is easy to verify (by inspection) that the corresponding posterior process is:

$$f \mid D \sim GP\left(m_D, k_D\right)$$

$$m_D\left(x\right) = m(x) + \Sigma\left(X, x\right)^T \Sigma^{-1}\left(\mathrm{f}\ \text{-}\ \mathrm{m}\right)$$

$$k_D\left(x, x^{'}\right) = k\left(x, x^{'}\right) - \Sigma\left(X, x\right)^T \Sigma^{-1}\Sigma\left(X, x^{'}\right)$$

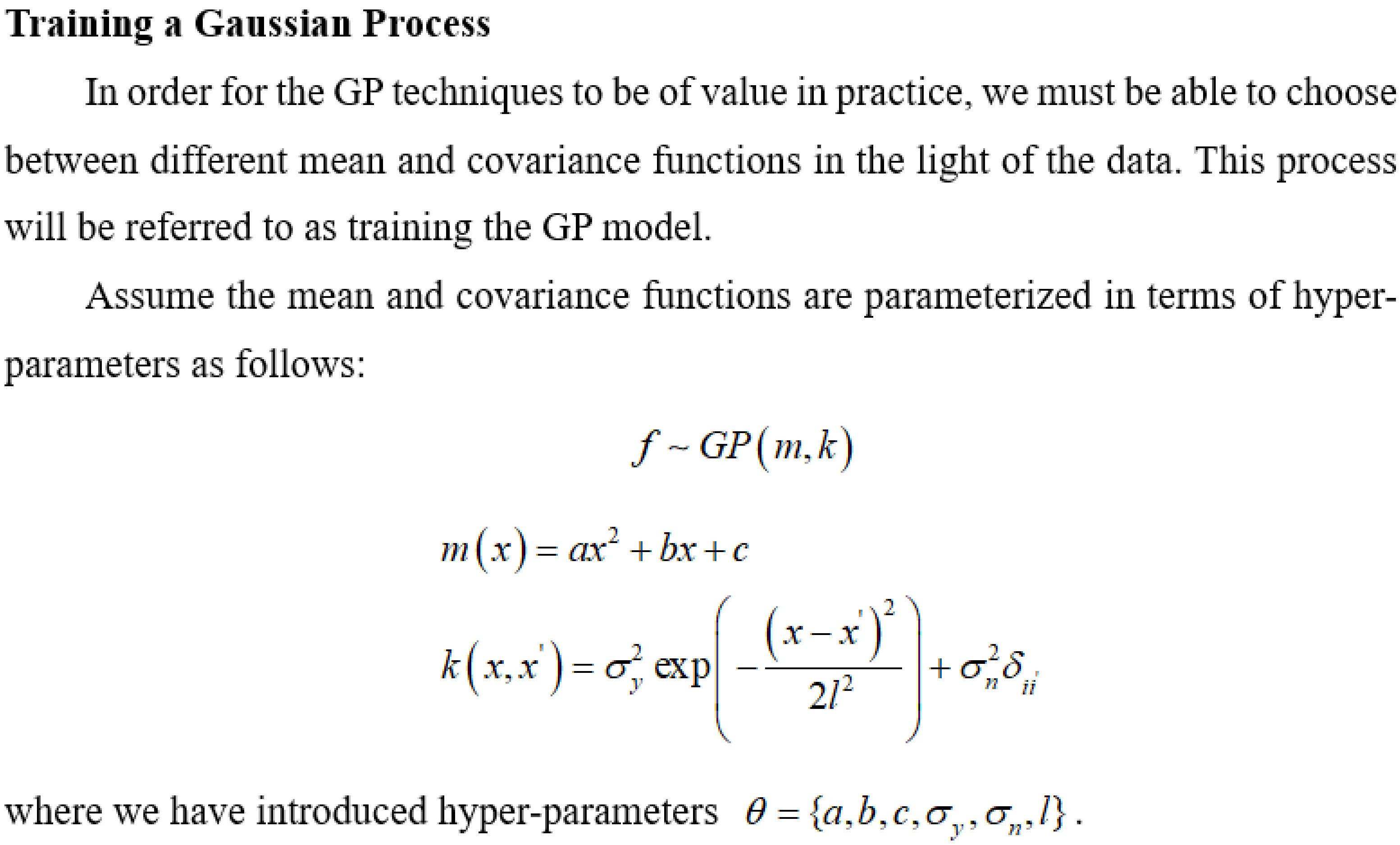
Noise in the training outputs: The most common assumption is that of additive i.i.d. Gaussian noise in the outputs,

$$y\left(x\right) = f\left(x\right) + \varepsilon \quad \varepsilon \sim N\left(0, \sigma_n^2\right)$$

$$f \sim GP\left(m, k\right) \qquad y \sim GP\left(m, k + \sigma_n^2 \delta_{ii^{'}}\right)$$

where $\delta_{ii^{'}} = 1 \; iff \; i = i^{'}$.

Gaussian Processes



Training a Gaussian Process

In order for the GP techniques to be of value in practice, we must be able to choose between different mean and covariance functions in the light of the data. This process will be referred to as training the GP model.

Assume the mean and covariance functions are parameterized in terms of hyper-parameters as follows:

$$f \sim GP(m,k)$$

$$m(x) = ax^2 + bx + c$$

$$k(x,x^{'}) = \sigma_y^2 \exp\left(-\frac{(x-x^{'})^2}{2l^2} \right) + \sigma_n^2 \delta_{ii^{'}}$$

where we have introduced hyper-parameters $\theta = \{a,b,c,\sigma_y,\sigma_n,l\}$.

Gaussian Processes

We compute the probability of the data given the hyper-parameters, log marginal likelihood:

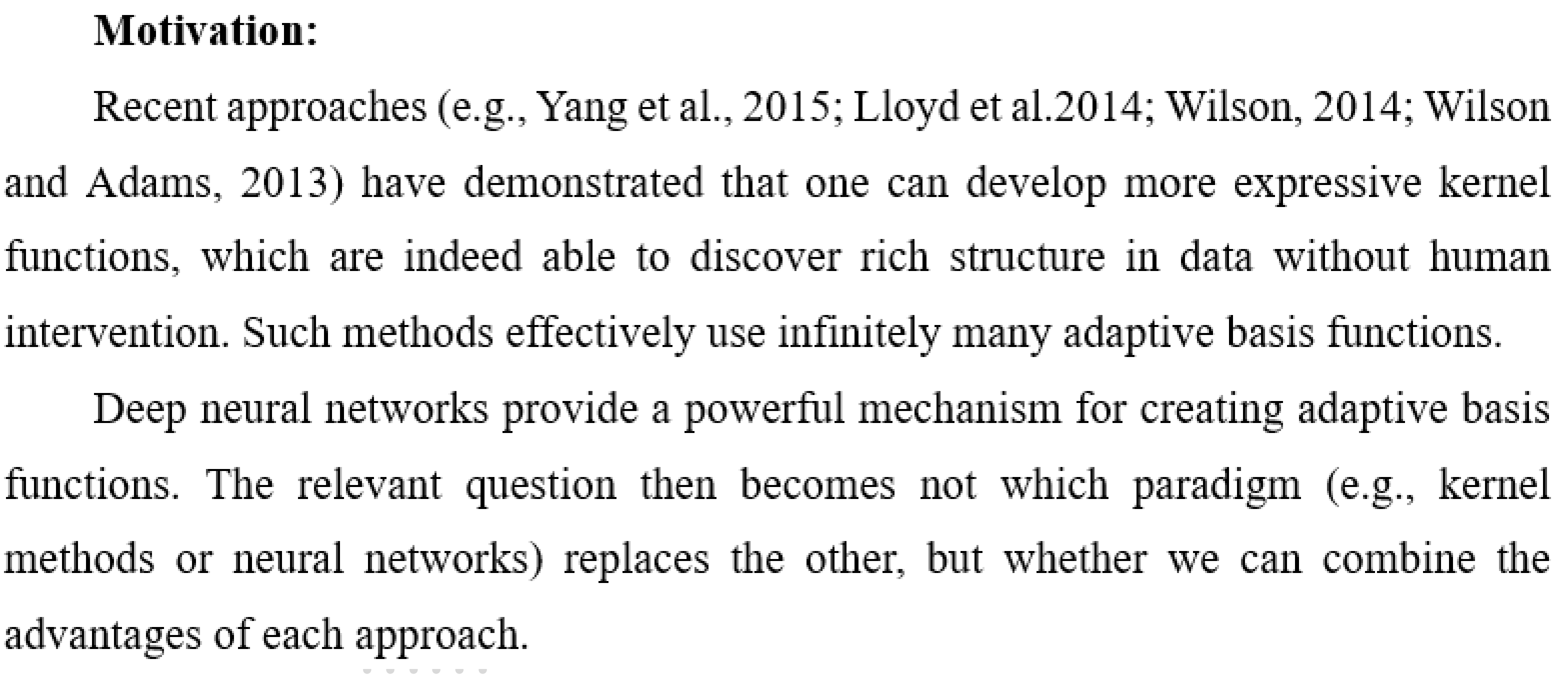
$$L = \log p(y \mid x, \theta) = -\frac{1}{2}\log|\Sigma| - \frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu) - \frac{n}{2}\log(2\pi)$$

The first term, $-\frac{1}{2}\log|\Sigma|$ is a complexity penalty term, which measures and penalizes the complexity of the model. The second term a negative quadratic, and plays the role of a data-fit measure. Note that the tradeoff between penalty and data-fit in the GP model is automatic.

NO.2

Deep kernel learning

Deep kernel learning

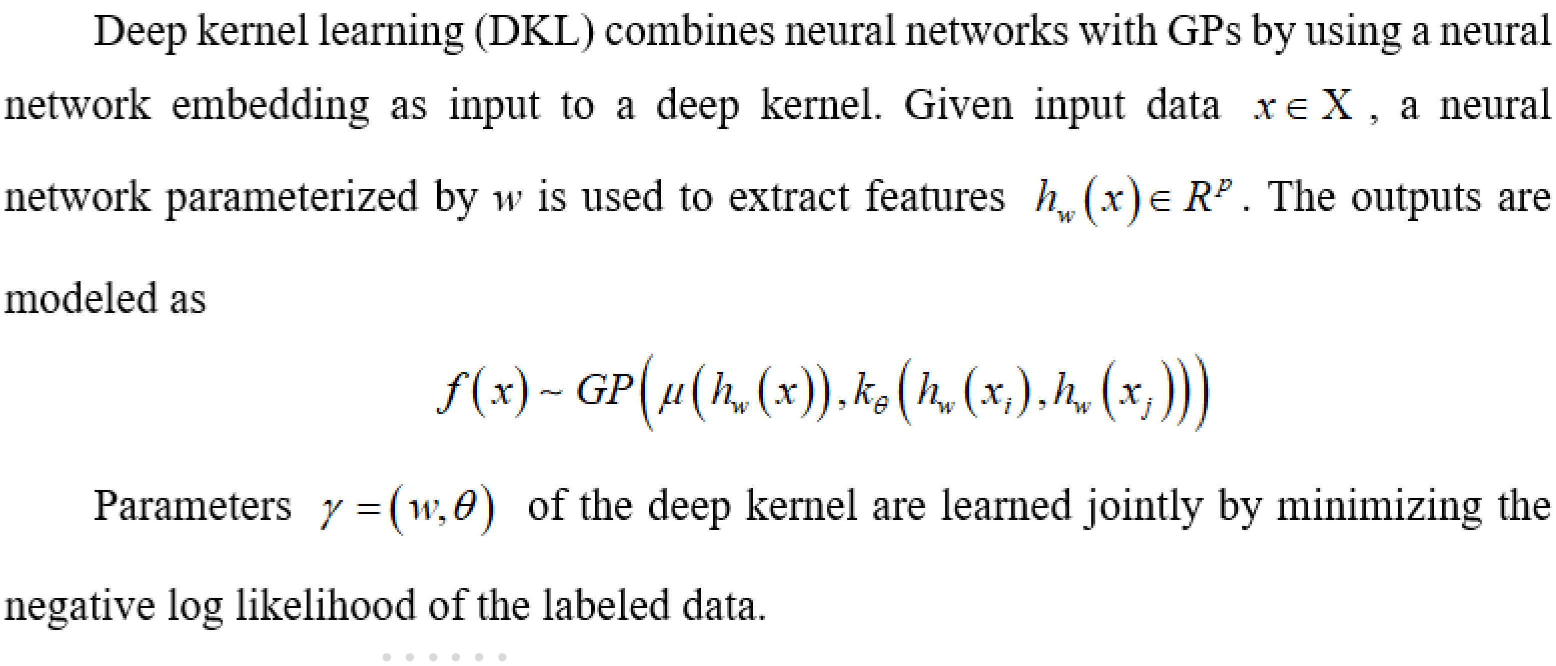


Motivation:

Recent approaches (e.g., Yang et al., 2015; Lloyd et al.2014; Wilson, 2014; Wilson and Adams, 2013) have demonstrated that one can develop more expressive kernel functions, which are indeed able to discover rich structure in data without human intervention. Such methods effectively use infinitely many adaptive basis functions.

Deep neural networks provide a powerful mechanism for creating adaptive basis functions. The relevant question then becomes not which paradigm (e.g., kernel methods or neural networks) replaces the other, but whether we can combine the advantages of each approach.

Deep kernel learning



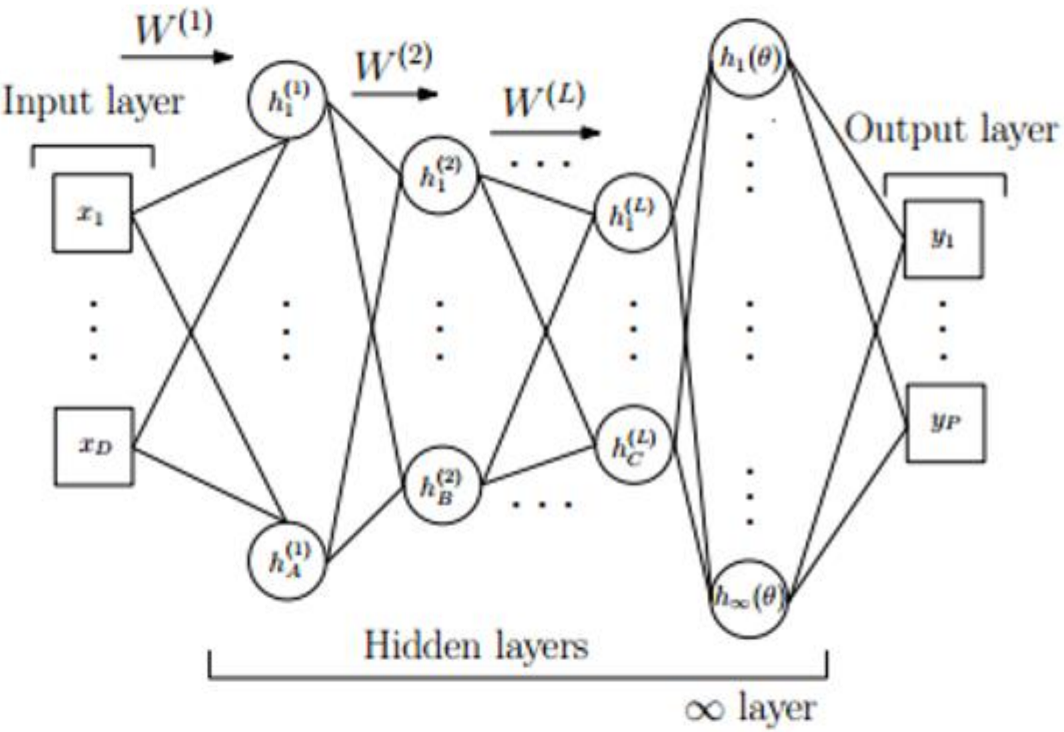
Deep kernel learning (DKL) combines neural networks with GPs by using a neural network embedding as input to a deep kernel. Given input data $x \in X$, a neural network parameterized by w is used to extract features $h_w(x) \in R^p$. The outputs are modeled as

$$f(x) \sim GP\left(\mu\left(h_w(x)\right), k_\theta\left(h_w(x_i), h_w(x_j)\right)\right)$$

Parameters $\gamma = (w, \theta)$ of the deep kernel are learned jointly by minimizing the negative log likelihood of the labeled data.

......

Deep kernel learning



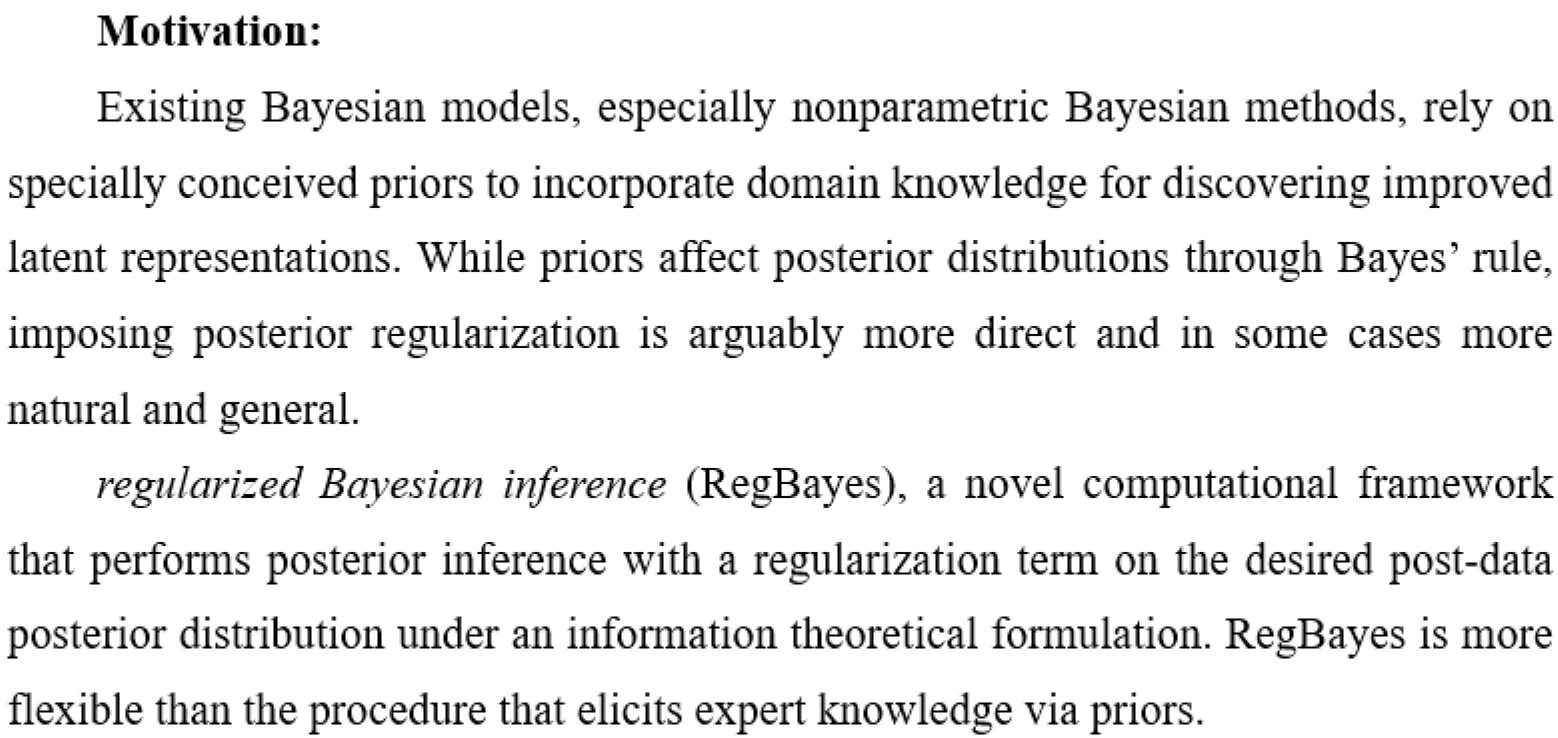
For kernel learning, we use the chain rule to compute derivatives of the log marginal likelihood with respect to the deep kernel hyper-parameters:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial K_\gamma}\frac{\partial K_\gamma}{\partial \theta}, \frac{\partial L}{\partial w} = \frac{\partial L}{\partial K_\gamma}\frac{\partial K_\gamma}{\partial h_w(x)}\frac{\partial h_w(x)}{\partial w}$$

NO.3

Posterior regularization

Posterior regularization

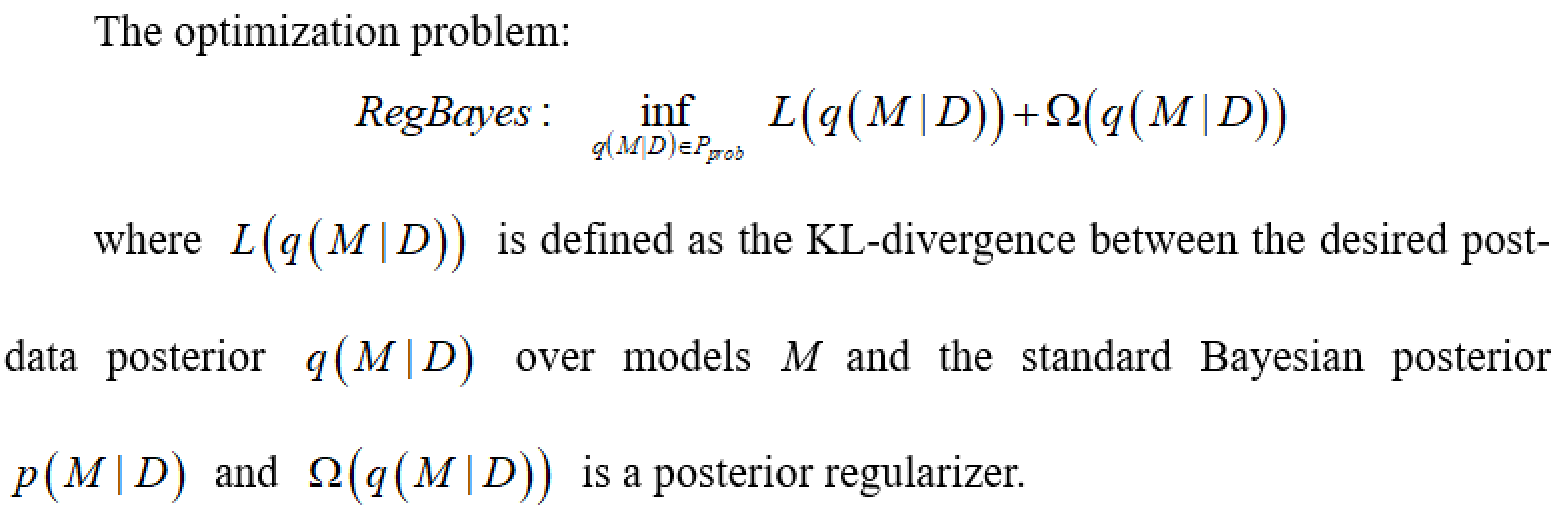


Motivation:

Existing Bayesian models, especially nonparametric Bayesian methods, rely on specially conceived priors to incorporate domain knowledge for discovering improved latent representations. While priors affect posterior distributions through Bayes’ rule, imposing posterior regularization is arguably more direct and in some cases more natural and general.

regularized Bayesian inference (RegBayes), a novel computational framework that performs posterior inference with a regularization term on the desired post-data posterior distribution under an information theoretical formulation. RegBayes is more flexible than the procedure that elicits expert knowledge via priors.

Posterior regularization



The optimization problem:

$$RegBayes: \inf_{q(M|D)\in P_{prob}} L(q(M|D))+\Omega(q(M|D))$$

where $L(q(M|D))$ is defined as the KL-divergence between the desired post-data posterior $q(M|D)$ over models M and the standard Bayesian posterior $p(M|D)$ and $\Omega(q(M|D))$ is a posterior regularizer.

NO.4

Semi supervised DKL

Semi-supervised DKL

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:

$$\begin{aligned} L_{semisup}(\theta) &= \frac{1}{n} L_{likelihood}(\theta) + \frac{\alpha}{m} L_{\mathrm{variance}}(\theta) \\ &= -\frac{1}{n} \log p(y_L \mid X_L, \theta) + \frac{\alpha}{m} \sum_{x \in X_U} Var_{f-p}(f(x)) \end{aligned}$$

where the variance is with respect to $p(f \mid \theta, D)$, the Bayesian posterior given θ and D.

Semi-supervised DKL

Intuition for variance minimization:

The posterior variance acts as a proxy for distance with respect to the kernel function in the deep feature space, and the regularizer is an inductive bias on the structure of the feature space. Since the deep kernel parameters are jointly learned, the neural net is encouraged to learn a feature representation in which the unlabeled examples are closer to the labeled examples, thereby reducing the variance on our predictions.

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.

Semi-supervised DKL

Theorem 1. *Let observed data D, a suitable space of functions F, and parameter space Θ be given. As in [15], we assume that F is a complete separable metric space and Π is an absolutely continuous probability measure (with respect to background measure η) on (F; B(F)), where B(F) is the Borel σ-algebra, such that a density π exists where dΠ = πdη and we have prior density π(f; θ) and likelihood density p(D|f, θ). Then the semi-supervised variance minimization problem (5)*

$$\inf_{\theta} L_{semi\,\sup}(\theta)$$

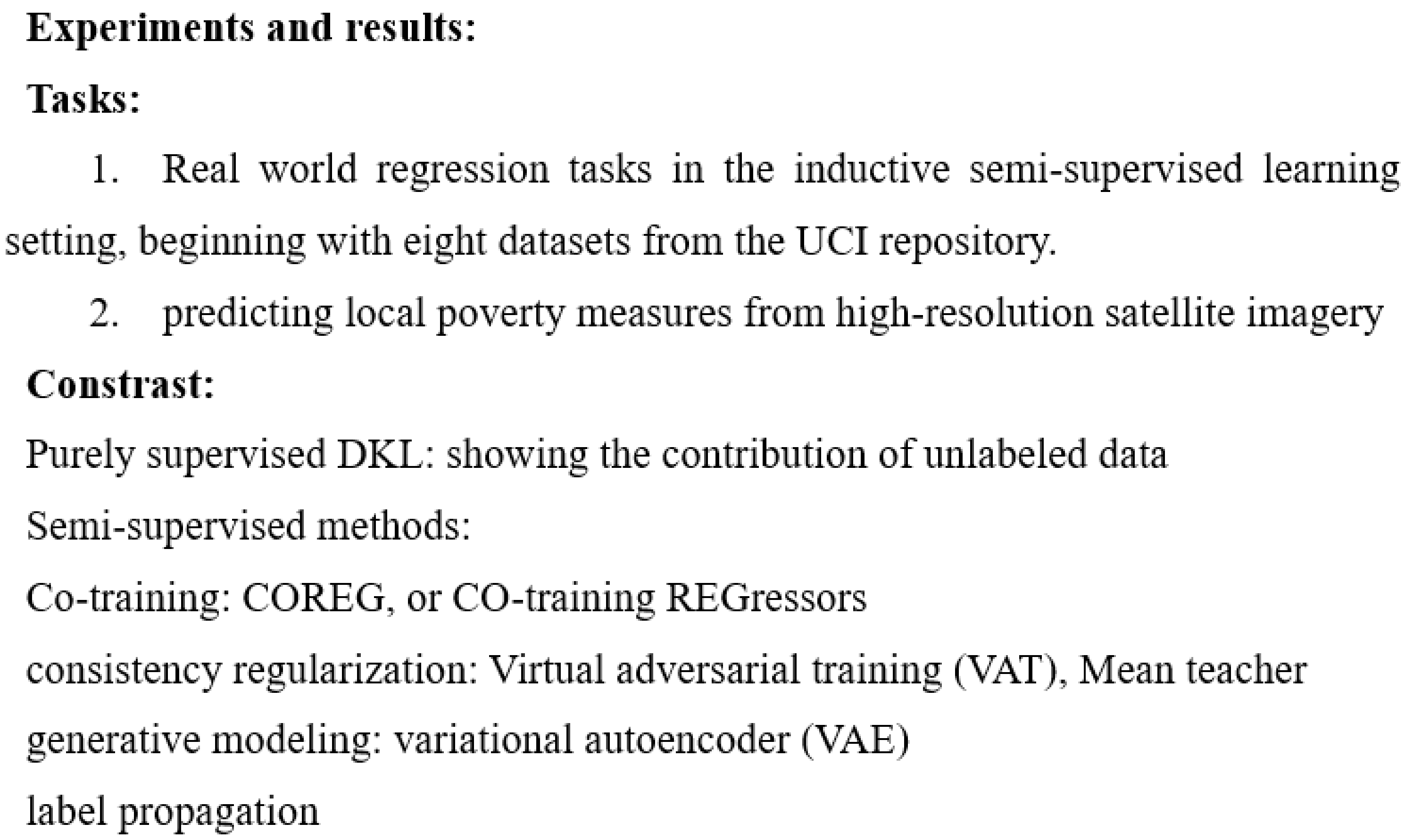
is equivalent to the RegBayes optimization problem(2)

$$\inf_{q(f,\theta|D)\in P_{prob}} \quad L\left(q\left(f,\theta\,|\,D\right)\right)+\Omega\left(q\left(f,\theta\,|\,D\right)\right)$$

$$\Omega\left(q\left(f,\theta\,|\,D\right)\right)=\alpha'\sum_{i=1}^{m}\int_{f,\theta} p\left(f\,|\,\theta,D\right)q\left(\theta\,|\,D\right)\left(f\left(X_u\right)_i-E_p\left[f\left(X_u\right)_i\right]^2\right)d\eta\left(f,\theta\right)$$

where $\alpha' = \dfrac{\alpha n}{m}$ *, and* $P_{prob} = \left\{q : q\left(f,\theta\,|\,D\right) = q\left(f\,|\,\theta,D\right)\delta_{\bar{\theta}}\left(\theta\,|\,D\right), \bar{\theta}\in\Theta\right\}$ *is a variational family of distributions where* $q\left(\theta\,|\,D\right)$ *is restricted to be a Dirac delta centered on* $\bar{\theta}\in\Theta$.

Semi-supervised DKL



Experiments and results:

Tasks:

1. Real world regression tasks in the inductive semi-supervised learning setting, beginning with eight datasets from the UCI repository.
2. predicting local poverty measures from high-resolution satellite imagery

Constrast:

Purely supervised DKL: showing the contribution of unlabeled data

Semi-supervised methods:

Co-training: COREG, or CO-training REGressors

consistency regularization: Virtual adversarial training (VAT), Mean teacher

generative modeling: variational autoencoder (VAE)

label propagation

Semi-supervised DKL

DataSets:

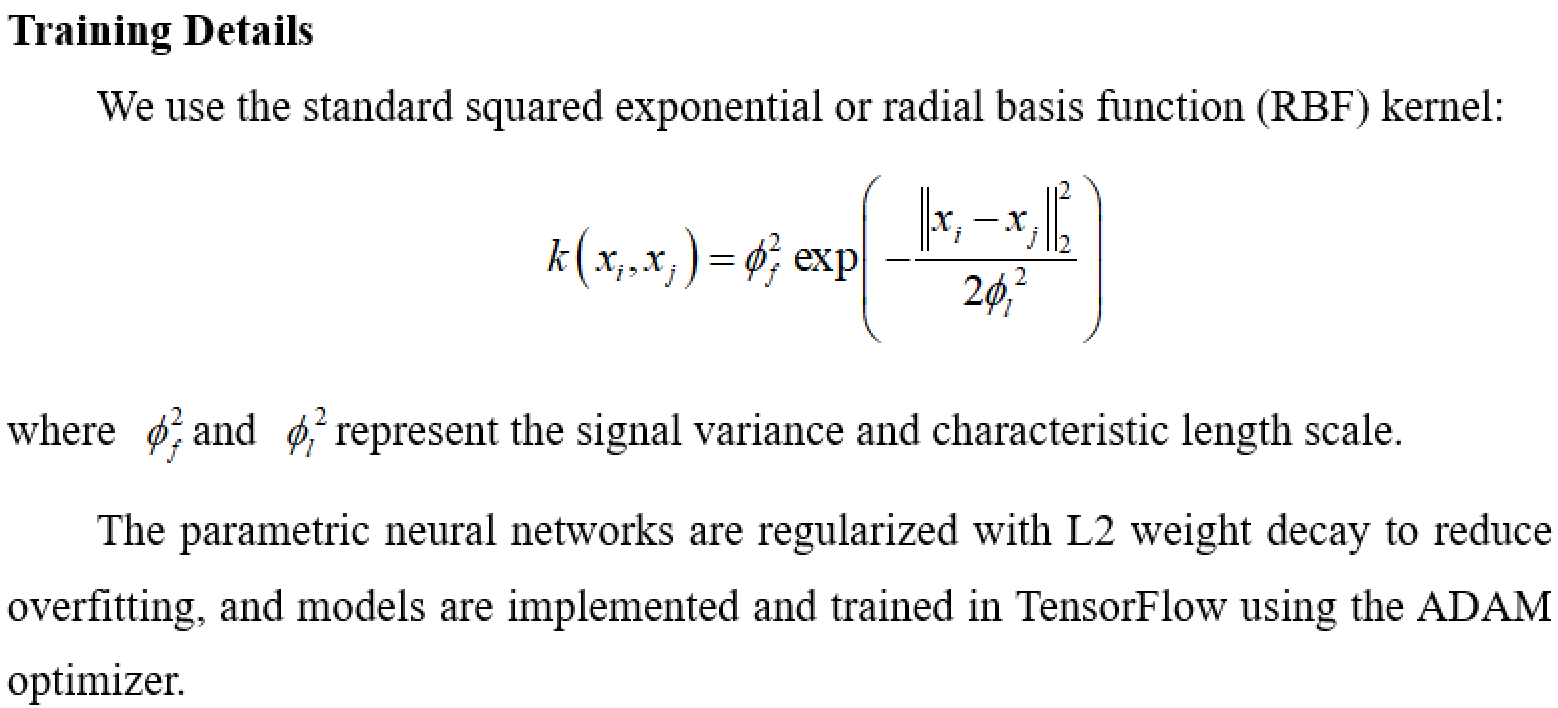
we train on $n = \{50; 100; 200; 300; 400; 500\}$ labeled examples, retain 1000 examples as the hold out test set, and treat the remaining data as unlabeled examples. the labeled data is randomly split 90-10 into training and validation samples.

Network:

We choose a neural network with a similar [*d*-100-50-50-2] architecture and two-dimensional embedding, we use this same base model for all deep models, including SSDKL, DKL, VAT, mean teacher, and the VAE encoder.

The same learning rates and initializations are used across *all* UCI datasets for SSDKL. We use learning rates of $1 \times 10^{-3}$and 0.1 for the neural network and GP parameters respectively and initialize all GP parameters to 1.

Semi-supervised DKL



Training Details

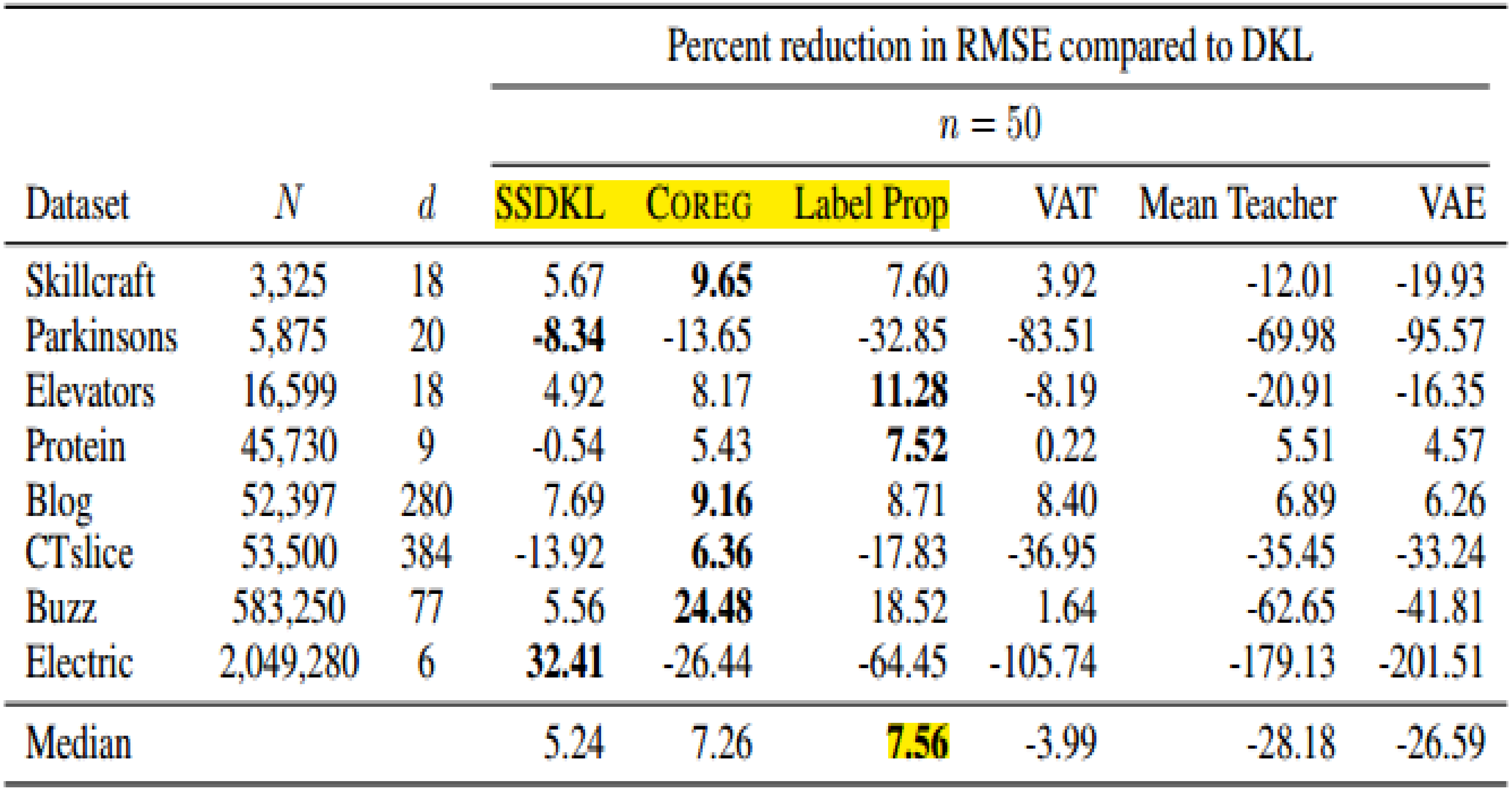
We use the standard squared exponential or radial basis function (RBF) kernel:

$$k\left(x_i, x_j\right) = \phi_f^2 \exp\left(-\frac{\left\| x_i - x_j \right\|_2^2}{2\phi_l^2}\right)$$

where ϕ_f^2 and ϕ_l^2 represent the signal variance and characteristic length scale.

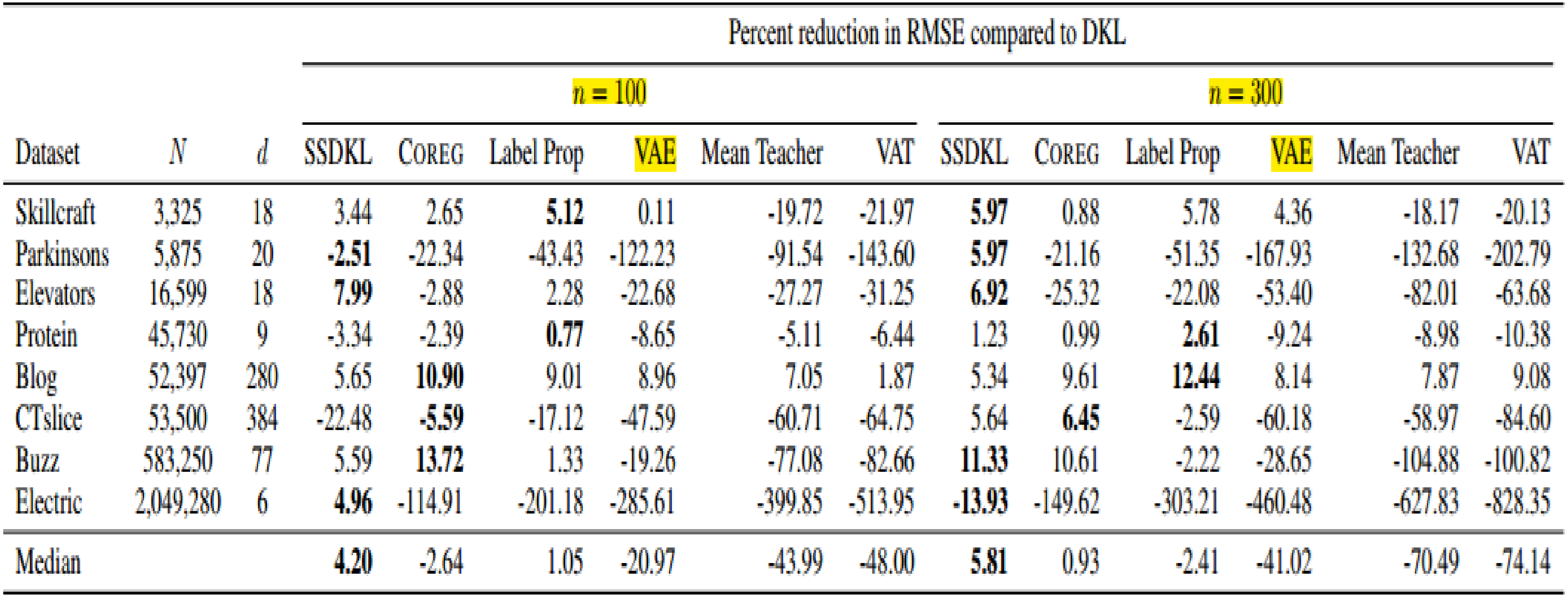
The parametric neural networks are regularized with L2 weight decay to reduce overfitting, and models are implemented and trained in TensorFlow using the ADAM optimizer.

Semi-supervised DKL



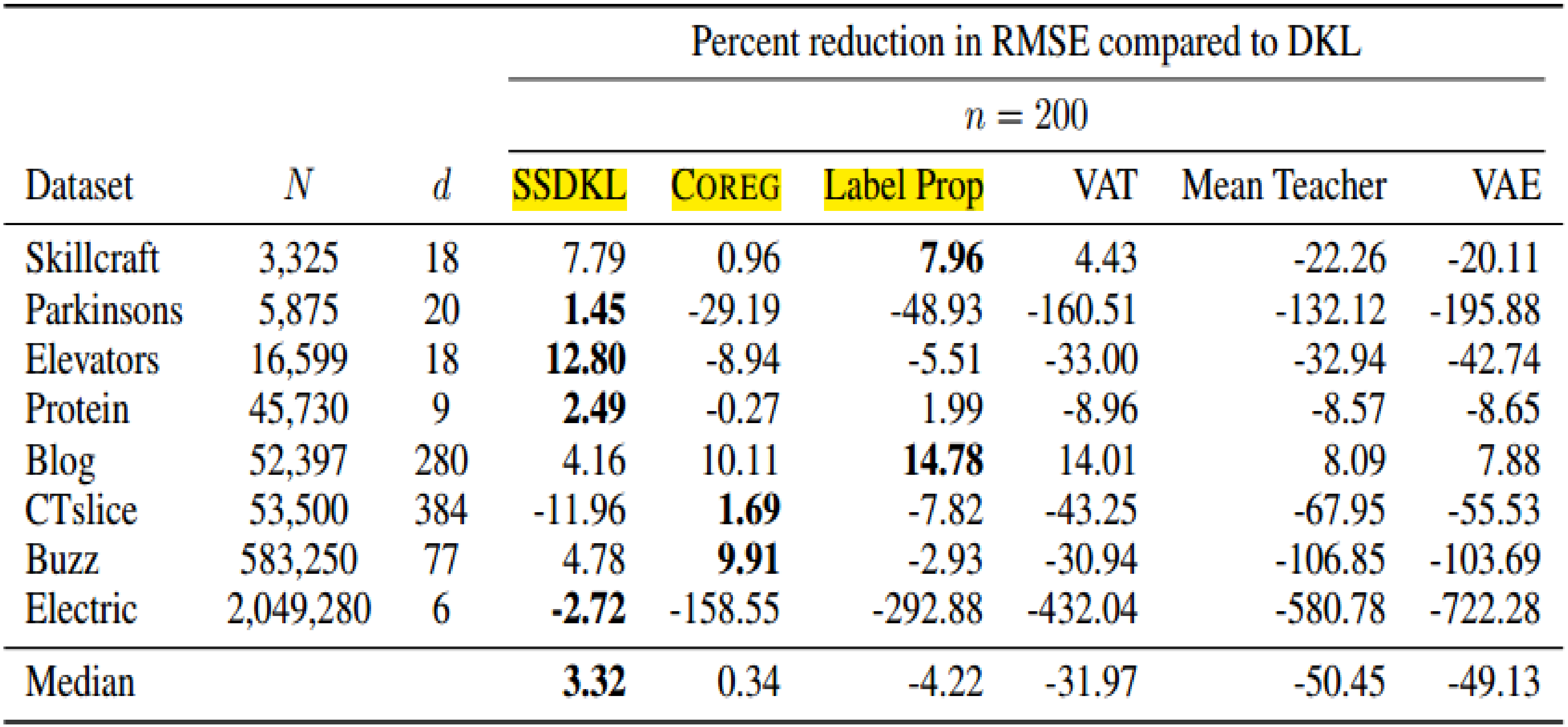
			Percent reduction in RMSE compared to DKL					
			$n = 50$					
Dataset	N	d	SSDKL	COREG	Label Prop	VAT	Mean Teacher	VAE
Skillcraft	3,325	18	5.67	**9.65**	7.60	3.92	-12.01	-19.93
Parkinsons	5,875	20	**-8.34**	-13.65	-32.85	-83.51	-69.98	-95.57
Elevators	16,599	18	4.92	8.17	**11.28**	-8.19	-20.91	-16.35
Protein	45,730	9	-0.54	5.43	**7.52**	0.22	5.51	4.57
Blog	52,397	280	7.69	**9.16**	8.71	8.40	6.89	6.26
CTslice	53,500	384	-13.92	**6.36**	-17.83	-36.95	-35.45	-33.24
Buzz	583,250	77	5.56	**24.48**	18.52	1.64	-62.65	-41.81
Electric	2,049,280	6	**32.41**	-26.44	-64.45	-105.74	-179.13	-201.51
Median			5.24	7.26	**7.56**	-3.99	-28.18	-26.59

Semi-supervised DKL



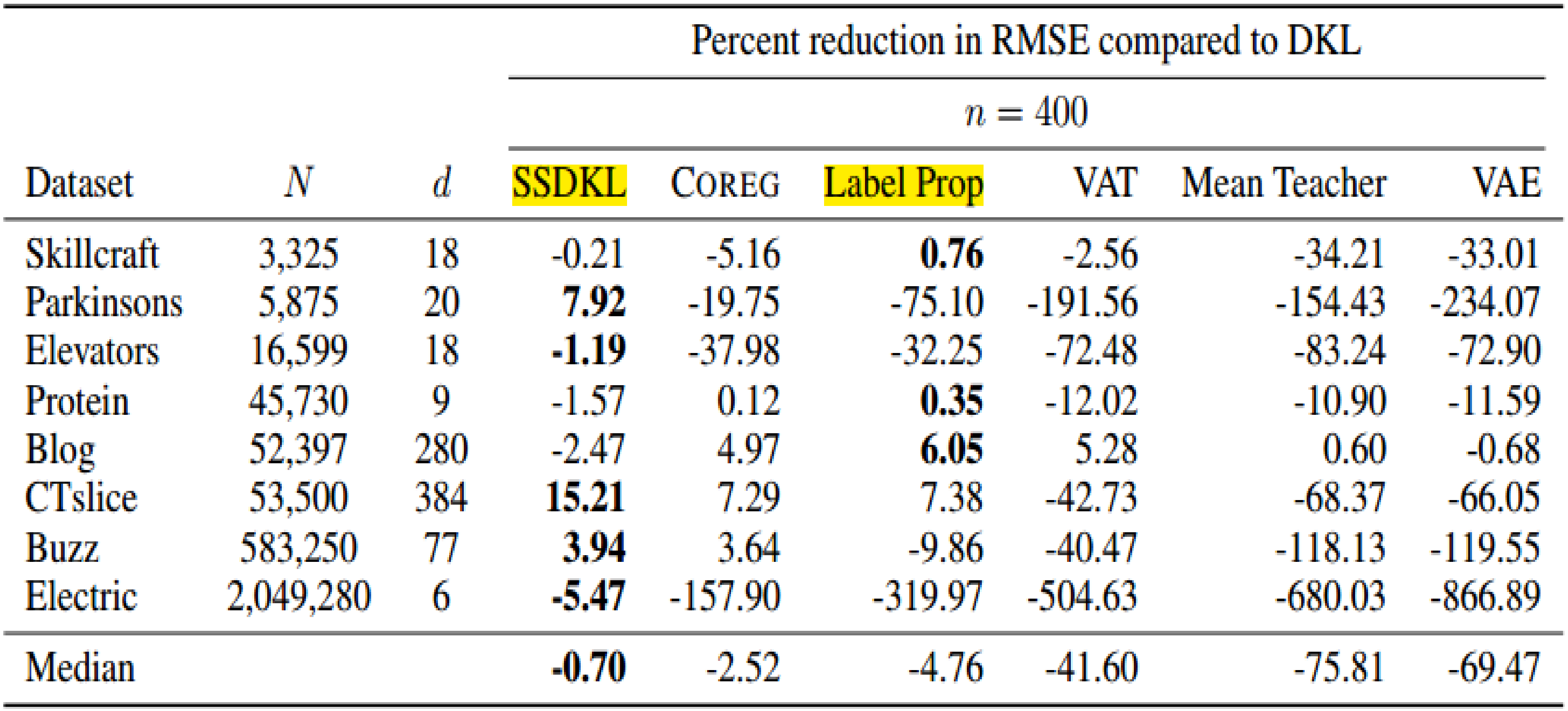
			Percent reduction in RMSE compared to DKL											
			n = 100						n = 300					
Dataset	N	d	SSDKL	CoReg	Label Prop	VAE	Mean Teacher	VAT	SSDKL	CoReg	Label Prop	VAE	Mean Teacher	VAT
Skillcraft	3,325	18	3.44	2.65	**5.12**	0.11	-19.72	-21.97	**5.97**	0.88	5.78	4.36	-18.17	-20.13
Parkinsons	5,875	20	**-2.51**	-22.34	-43.43	-122.23	-91.54	-143.60	**5.97**	-21.16	-51.35	-167.93	-132.68	-202.79
Elevators	16,599	18	**7.99**	-2.88	2.28	-22.68	-27.27	-31.25	**6.92**	-25.32	-22.08	-53.40	-82.01	-63.68
Protein	45,730	9	-3.34	-2.39	**0.77**	-8.65	-5.11	-6.44	1.23	0.99	**2.61**	-9.24	-8.98	-10.38
Blog	52,397	280	5.65	**10.90**	9.01	8.96	7.05	1.87	5.34	9.61	**12.44**	8.14	7.87	9.08
CTslice	53,500	384	-22.48	**-5.59**	-17.12	-47.59	-60.71	-64.75	5.64	**6.45**	-2.59	-60.18	-58.97	-84.60
Buzz	583,250	77	5.59	**13.72**	1.33	-19.26	-77.08	-82.66	**11.33**	10.61	-2.22	-28.65	-104.88	-100.82
Electric	2,049,280	6	**4.96**	-114.91	-201.18	-285.61	-399.85	-513.95	**-13.93**	-149.62	-303.21	-460.48	-627.83	-828.35
Median			**4.20**	-2.64	1.05	-20.97	-43.99	-48.00	**5.81**	0.93	-2.41	-41.02	-70.49	-74.14

Semi-supervised DKL



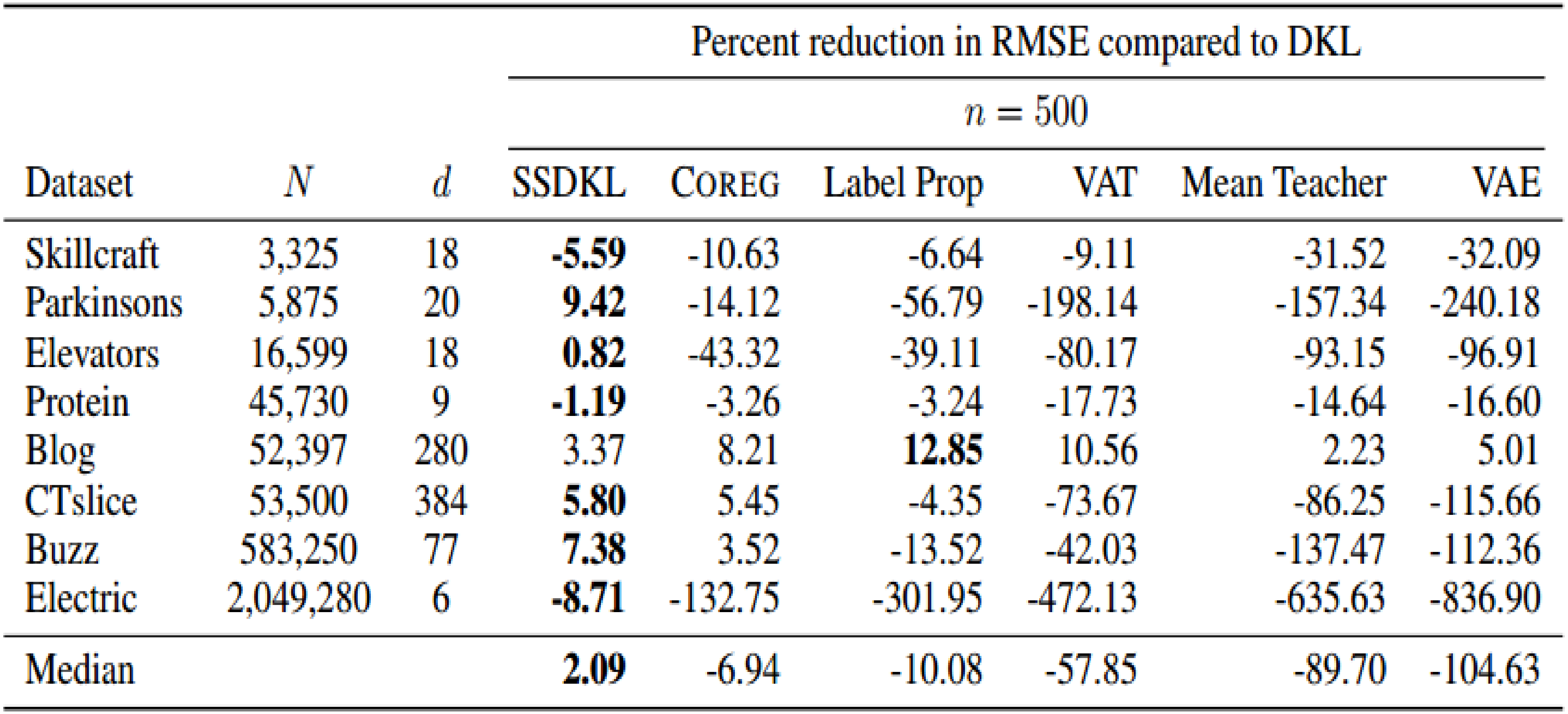
			Percent reduction in RMSE compared to DKL					
			$n = 200$					
Dataset	N	d	SSDKL	COREG	Label Prop	VAT	Mean Teacher	VAE
Skillcraft	3,325	18	7.79	0.96	**7.96**	4.43	-22.26	-20.11
Parkinsons	5,875	20	**1.45**	-29.19	-48.93	-160.51	-132.12	-195.88
Elevators	16,599	18	**12.80**	-8.94	-5.51	-33.00	-32.94	-42.74
Protein	45,730	9	**2.49**	-0.27	1.99	-8.96	-8.57	-8.65
Blog	52,397	280	4.16	10.11	**14.78**	14.01	8.09	7.88
CTslice	53,500	384	-11.96	**1.69**	-7.82	-43.25	-67.95	-55.53
Buzz	583,250	77	4.78	**9.91**	-2.93	-30.94	-106.85	-103.69
Electric	2,049,280	6	**-2.72**	-158.55	-292.88	-432.04	-580.78	-722.28
Median			**3.32**	0.34	-4.22	-31.97	-50.45	-49.13

Semi-supervised DKL



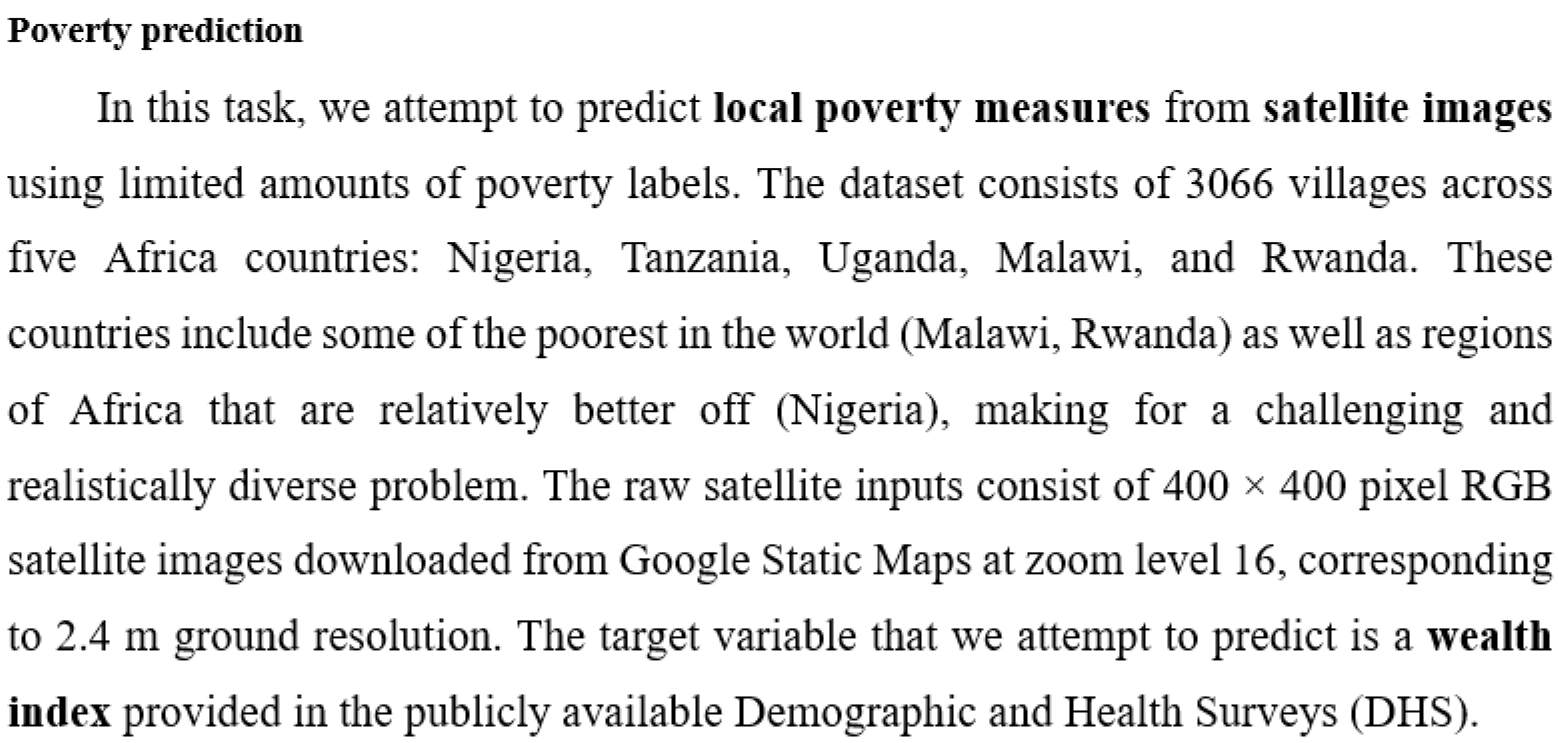
			Percent reduction in RMSE compared to DKL					
			$n = 400$					
Dataset	N	d	SSDKL	COREG	Label Prop	VAT	Mean Teacher	VAE
Skillcraft	3,325	18	-0.21	-5.16	**0.76**	-2.56	-34.21	-33.01
Parkinsons	5,875	20	**7.92**	-19.75	-75.10	-191.56	-154.43	-234.07
Elevators	16,599	18	**-1.19**	-37.98	-32.25	-72.48	-83.24	-72.90
Protein	45,730	9	-1.57	0.12	**0.35**	-12.02	-10.90	-11.59
Blog	52,397	280	-2.47	4.97	**6.05**	5.28	0.60	-0.68
CTslice	53,500	384	**15.21**	7.29	7.38	-42.73	-68.37	-66.05
Buzz	583,250	77	**3.94**	3.64	-9.86	-40.47	-118.13	-119.55
Electric	2,049,280	6	**-5.47**	-157.90	-319.97	-504.63	-680.03	-866.89
Median			**-0.70**	-2.52	-4.76	-41.60	-75.81	-69.47

Semi-supervised DKL



			Percent reduction in RMSE compared to DKL					
			$n = 500$					
Dataset	N	d	SSDKL	CoReg	Label Prop	VAT	Mean Teacher	VAE
Skillcraft	3,325	18	**-5.59**	-10.63	-6.64	-9.11	-31.52	-32.09
Parkinsons	5,875	20	**9.42**	-14.12	-56.79	-198.14	-157.34	-240.18
Elevators	16,599	18	**0.82**	-43.32	-39.11	-80.17	-93.15	-96.91
Protein	45,730	9	**-1.19**	-3.26	-3.24	-17.73	-14.64	-16.60
Blog	52,397	280	3.37	8.21	**12.85**	10.56	2.23	5.01
CTslice	53,500	384	**5.80**	5.45	-4.35	-73.67	-86.25	-115.66
Buzz	583,250	77	**7.38**	3.52	-13.52	-42.03	-137.47	-112.36
Electric	2,049,280	6	**-8.71**	-132.75	-301.95	-472.13	-635.63	-836.90
Median			**2.09**	-6.94	-10.08	-57.85	-89.70	-104.63

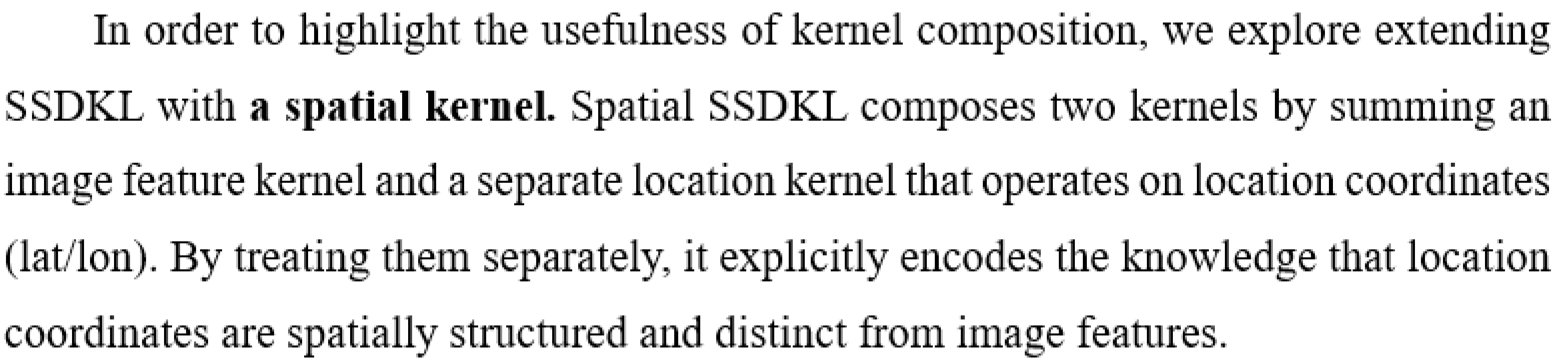
Semi-supervised DKL



Poverty prediction

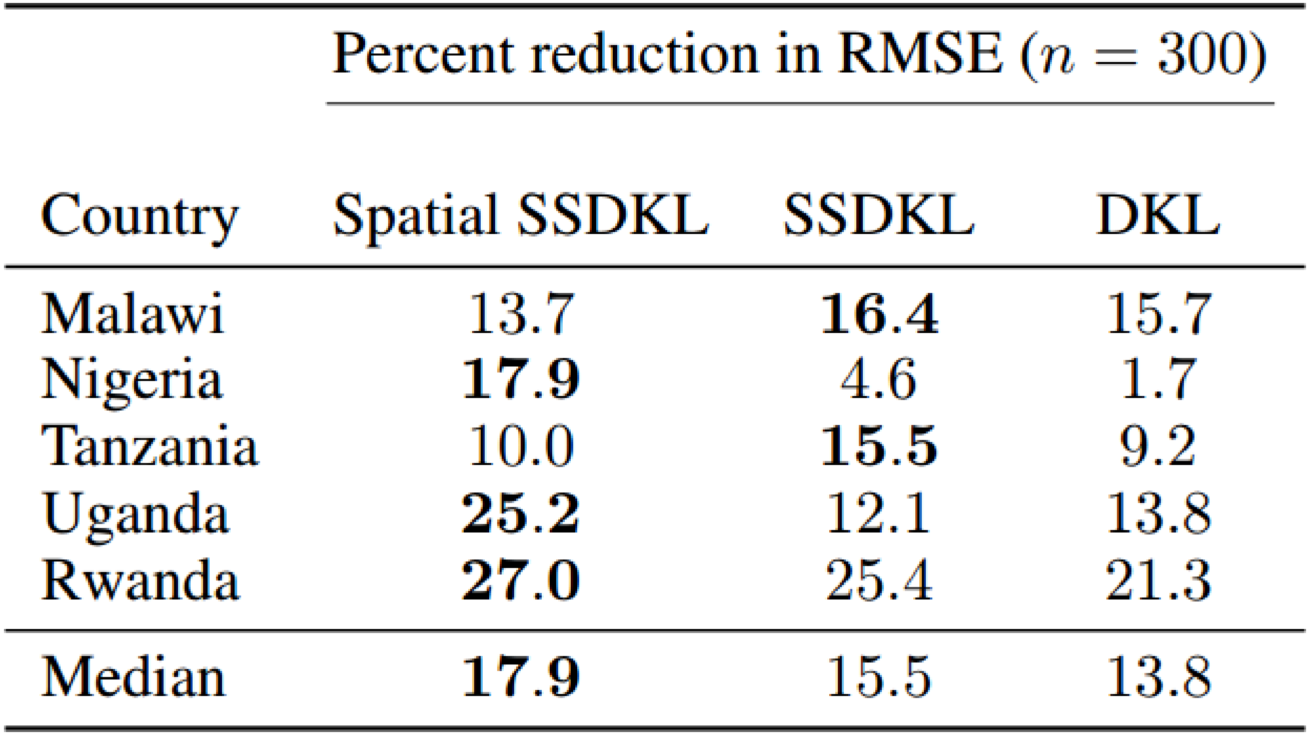
In this task, we attempt to predict **local poverty measures** from **satellite images** using limited amounts of poverty labels. The dataset consists of 3066 villages across five Africa countries: Nigeria, Tanzania, Uganda, Malawi, and Rwanda. These countries include some of the poorest in the world (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).

Semi-supervised DKL



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.

Semi-supervised DKL



	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
Median	**17.9**	15.5	13.8

Thanks