The Annals of Applied Statistics
2023, Vol. 17, No. 4, 3241–3256
https://doi.org/10.1214/23-AOAS1759

A PENALIZED COMPLEXITY PRIOR FOR DEEP BAYESIAN TRANSFER LEARNING WITH APPLICATION TO MATERIALS INFORMATICS

BY MOHAMED A. ABBA[a], JONATHAN P. WILLIAMS[b] AND BRIAN J. REICH[c]

Department of Statistics, North Carolina State University, [a]*mabba@ncsu.edu,* [b]*jwilli27@ncsu.edu,* [c]*bjreich@ncsu.edu*

A key task in the emerging field of materials informatics is to use machine learning to predict a material's properties and functions. A fast and accurate predictive model allows researchers to more efficiently identify or construct a material with desirable properties. As in many fields, deep learning is one of the state-of-the art approaches, but fully training a deep learning model is not always feasible in materials informatics due to limitations on data availability, computational resources, and time. Accordingly, there is a critical need in the application of deep learning to materials informatics problems to develop efficient *transfer learning* algorithms. The Bayesian framework is natural for transfer learning because the model trained from the source data can be encoded in the prior distribution for the target task of interest. However, the Bayesian perspective on transfer learning is relatively unaccounted for in the literature and is complicated for deep learning because the parameter space is large and the interpretations of individual parameters are unclear. Therefore, rather than subjective prior distributions for individual parameters, we propose a new Bayesian transfer learning approach based on the penalized complexity prior on the Kullback–Leibler divergence between the predictive models of the source and target tasks. We show via simulations that the proposed method outperforms other transfer learning methods across a variety of settings. The proposed method is applied to predict the properties of a molecular crystal, based on its structural properties, and we show improved precision for estimating the band gap of a material compared to state-of-the-art methods currently used in materials science.

1. Introduction. Materials informatics has fundamentally changed materials science research (e.g., Himanen et al. (2019)). To design or select a material for a particular function, researchers have traditionally relied on intuition and costly experimentation. This process is now supplemented by machine learning to predict a candidate material's properties and triage materials for further experimentation. The addition of machine learning has been shown to improve efficiency, especially when using multiple data sources (e.g., Batra (2021)). We aim to build a predictive model for a material's band gap, defined as the energy differential between the lowest-unoccupied and highest-occupied electronic states (Kittel, McEuen and Wiley (2019)). The band gap governs desirable properties that are useful in industrial sectors such as electric and photovoltaic conductivity. Traditionally, band gap size is computed using methods in quantum mechanics such as density functional theory (DFT; Kohn and Sham (1965)). However, these methods require running costly computer simulations, and so it is *not* feasible to exhaustively search over a broad class of materials. Our objective is to build a statistical model (sometimes called a meta-model, surrogate model, or emulator; O'Hagan (2006)) along with an accompanying transfer learning methodology that is both effective at predicting the output of the DFT simulation and is able to optimize experimentation on future test data sets from related data sources.

Received May 2022; revised December 2022.
Key words and phrases. Kullback–Leibler divergence, materials science, neural networks, variational Bayesian inference.

Deep neural network (DNN) architectures have emerged as leading models in materials informatics. Beyond materials science DNNs have revolutionized the field of machine learning with significant breakthroughs in a wide variety of applications including computer vision (Voulodimos et al. (2018)), natural language processing (Young et al. (2018)), and protein folding (Senior et al. (2020)); see Dargan et al. (2020) for further references to modern applications. Their ability to model complex nonlinear processes enables them to handle a large class of prediction problems. Training a neural network, however, is not an easy task, often requiring intensive computational cost, large quantities of training data, and careful hyperparameter selection (Bengio (2012)). As an overparameterized expressive model, DNNs are prone to overfitting, especially in the case of small data sets. Tan et al. (2018) conclude that the number of parameters in a DNN and the size of the data required for good generalization performance have an almost linear relationship. Since these models tend to have thousands (sometimes millions) of parameters, the size of the data required becomes quickly prohibitive. This is where *transfer learning* plays a central role.

Broadly defined, transfer learning describes a machine learning approach for augmenting the training of a learning task on a *target* population data set with a learning algorithm that has already been trained on a closely related data set from a *source* population, particularly for scenarios where the target and source populations are *not* identical; see Weiss, Khoshgoftaar and Wang (2016) for a recent survey on transfer learning. By developing effective transfer learning strategies, it is possible to reduce the computational burden and the need for large training data sets for training DNN models in applications where complex DNNs have already been trained for similar data sets. For example, Raghu et al. (2019) and Ahishakiye et al. (2021) investigated the use of a pretrained DNN on ImageNet (Deng et al. (2009)) to improve the accuracy in medical imaging tasks where labeled data is usually scarce. Although the potential benefits for transfer learning are significant, there are no generally accepted procedures for how to construct or evaluate a transfer-learning strategy, and this is especially true from the Bayesian perspective. Dube et al. (2020) provides a review of the current approaches to address a variety of important questions of concern for developing transfer-learning methods. For instance, what aspects of the source tasks can be leveraged to improve performance on the target task? In the case of DNN source models, this commonly boils down to isolating the layers that are task agnostic versus those that are task dependent.

1.1. *Data and motivation*. Motivated by the predictive materials science application of estimating the band gap of a material based on its structural properties, we propose a principled statistical approach to transfer learning within a Bayesian framework. In materials science, the band gap governs desirable properties of candidate materials. The conductivity of a solid is closely related to the band gap size. For example, materials with large band gaps are generally insulators, while materials with low band gaps are semiconductors (Van de Walle (2012)). Band gap size (measured in electronvolts, eV) is typically computed using different quantum mechanics methods like the DFT. DFT calculation can be expensive for large molecules and crystals. This, combined with the large possible number of materials, has driven the need for fast screening methods. According to Olsthoorn et al. (2019), there has been a growing interest in the development of accurate machine learning models for material sciences and quantum systems. For small organic molecules, Olsthoorn et al. (2019) reported that DNN models have reached chemical accuracy. For large molecular crystals, however, the task of band gap prediction seems to be much harder. In addition, DFT calculations are more expensive to perform on molecular crystals. For example, Ma et al. (2021) found that the algorithmic complexity of DFT is cubic in the number of particles for a given molecule; hence, it is easier to collect and compute band gap values for molecules than it is for molecular crystals. Given the difficulty of band gap calculation for molecular crystals, we

propose to use transfer-learning methods to borrow information from molecules (source) to molecular crystals (target) to improve prediction. The target data considered here will consist of 50,000 molecular crystals and their band gap values computed using DFT. This data set was provided by the Material Science Research Group at Carnegie Mellon University and required a sustained effort over several years to produce. The source data will be the OE62 data set (Stuke et al. (2019)), which consists of 62,000 molecules and their band gap values. The molecules in the source data were all extracted from organic crystals, and the band gap values were also computed using DFT.

1.2. *Literature review.* Bayesian neural networks have attracted considerable attention in recent years as a result of computational advances (e.g., Wilson (2020)). One of the earliest works on Bayesian transfer learning in DNNs was Wohlert et al. (2018), where a single source task is considered and a DNN with feed forward layers is trained using mean field variational Bayes (VB; Zhang et al. (2019)). The approximate posterior learned on the source task is used as a prior on the parameters for the target task. Wohlert et al. (2018) did not, however, address the problem of freezing or training the transferred layers and surprisingly gave worse performance than training using only the target data. Chandra and Kapoor (2020) introduced a new method for Bayesian multisource transfer learning where they proposed a new Markov chain Monte Carlo (MCMC) sampling scheme with parameter proposals for the target task based on proposals for all the source tasks. At first, all source task parameters are sampled, and then a single source for transfer of knowledge is used for the proposals for the target task. This method requires storing all source tasks data in memory together with the target task and does not fall under the two-stage inductive transfer-learning framework considered in our paper. Bueno et al. (2019) used Bayesian transfer learning for volcano-seismic prediction and uncertainty quantification using variational inference. Yang, Jiao and Sun (2020) focused on Bayesian hyperparameter optimization for transfer learning. Zhou et al. (2020) proposed a new form of regularization that can be viewed as a Gaussian prior on the network weights where the Fisher information from the source task is used as the precision matrix. More generally, Bayesian transfer learning for models other than DNNs has been considered by other articles in the literature, most commonly attributable to the fact that it is natural to borrow strength across data sets via prior distributions (e.g., Karbalayghareh, Qian and Dougherty (2018)).

The Bayesian transfer-learning methodology that we develop sequentially trains a DNN on a target data set by first specifying a prior distribution concentrated on parameter estimates for the DNN trained from a source data set. While this approach is sensible, the primary challenge is that constructing prior densities from pretrained DNN layers is not trivial because individual parameters are not identifiable nor do they have inherent meaning. Accordingly, to solve this issue we extend the fundamental notion of *penalized complexity priors* (PCPs) from Simpson et al. (2017) to the deep-learning setting and specify our *transfer-learning prior distribution* in terms of the Kullback–Leibler divergence between the source and target model. This prior is constructed with the intuition that the predictive model is likely shared (to some extent) across tasks but is flexible enough to disregard the source task if appropriate.

The target data set in our materials science application is small, and so we implement a fully Bayesian analysis with computations via MCMC sampling strategies. However, since the source data set is large, a fully Bayesian analysis is prohibitive, and so we analyze the source data using both optimization and variational-inference (VI; Blundell et al. (2015)) methods. Both methods provide point estimates for each parameter, but VI also provides (approximate) uncertainty quantification. Therefore, in addition to applying to cases where data sets from both the target and the source are available, our proposed method can be applied using source data DNNs trained by other users, such as ImageNet (Deng et al. (2009)). In our

exposition we develop our transfer-learning methods using the optimization-based source model and the VI-based source model in parallel, and we show that both provide improvements over traditional transfer-learning methods for both synthetic and real data.

The remainder of the paper proceeds as follows. Section 2 reviews background material on DNNs and transfer learning. Section 3 introduces the proposed PCP priors for transfer learning in DNNs. The proposed methods are then evaluated using simulated and real data in Sections 4 and 5, respectively. Section 6 concludes. The code for reproducing our empirical results is available at https://github.ncsu.edu/mabba/Pybayes.git.

2. Background material.

2.1. *Deep-learning model.* We begin by briefly reviewing the DNN model; for a more comprehesive review, see Goodfellow, Bengio and Courville (2016). In general, DNNs are comprised of successive layers, starting with an input layer, followed by one or more hidden layers, and finally an output layer. Each layer contains a number of nodes that are connected to the next layer through weights that control the impact of a given layer on the next. For concreteness we describe the Bayesian transfer-learning model in the simple case of a continuous response and fully-connected feed forward neural network. However, the ideas proposed in this paper translate to other structures, including recurrent (Medsker and Jain (2001)) and convolutional (LeCun, Bengio et al. (1995)) networks (see Section 6 for further discussion).

A dense feed forward layer is a function $l : \mathbf{R}^d \longrightarrow \mathbf{R}^k$ characterized by a weight matrix $W \in \mathbf{R}^{k\times d}$, a bias vector $b \in \mathbf{R}^k$, and a real-valued activation function a applied element-wise to a vector argument. For $\mathbf{x} \in \mathbf{R}^d$, the dense layer applies an affine transformation to $\mathbf{x}$ followed by the activation function, $l(\mathbf{x}) = a(W\mathbf{x} + b)$. A DNN with L hidden layers first applies an input layer, typically $l_0(\mathbf{x}) = \mathbf{x}$, then $L-1$ hidden layers, and one output layer; each layer is assigned unique weights, biases, and (potentially) activation functions; that is, $l_i(\mathbf{x}) = a_i(W_i\mathbf{x} + b_i)$ for layer $i \in \{1, \ldots, L\}$. The composition of these L layers defines a function f from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$,

$$
\begin{aligned}
&f(\cdot;\boldsymbol{\theta}) : \mathcal{X} \longrightarrow \mathcal{Y},\\
&f(\mathbf{x};\boldsymbol{\theta}) = l_L\big(l_{L-1}\big(\cdots\big(l_1(\mathbf{x})\big)\big)\big),
\end{aligned}
\tag{1}
$$

where $\boldsymbol{\theta} := \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$ and $\boldsymbol{\theta}_i := \{b_i, W_i\}$ are the parameters for layer i. For a regression task, we consider $f(\mathbf{x};\boldsymbol{\theta})$ to be the mean response given $\mathbf{x}$, whereas for a classification task, $f(\mathbf{x};\boldsymbol{\theta})$ represents a vector of probabilities.

2.2. *Transfer learning.* Before discussing transfer learning for DNNs, we review general definitions and concepts of transfer learning. We adopt notation from Pan and Yang (2009) and Weiss, Khoshgoftaar and Wang (2016) whose early work has been widely adopted in the literature. A *domain* is composed of two parts, a feature space $\mathcal{X}$ and a sampling distribution $p(\mathbf{X})$, where $\mathbf{X}$ denotes a set of instances $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathcal{X}, i = 1, \ldots, n\}$. Accordingly, $\mathcal{D} = \{\mathcal{X}, p\}$. Next, a *task* (e.g., classification or prediction) consists of a label space $\mathcal{Y}$ and a decision function f; that is, $\mathcal{T} = \{\mathcal{Y}, f\}$.

Transfer learning considers two domains: the source and the target. The main objective is to perform the target task in the target domain, and the analysis are supplemented by the related data from the source task (or multiple source tasks, e.g., Maurer, Pontil and Romera-Paredes (2016)). For a source domain and task, denoted $\mathcal{D}_S$ and $\mathcal{T}_S$, respectively, an observed data set is the collection $\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{X}_S, y_i \in \mathcal{Y}_S, 1 \le i \le n_S\}$, and f_S is understood as a conditional distribution or decision function for the instances y_i, given $\mathbf{x}_i$ for $i \in \{1, \ldots, n_S\}$. Similarly, define $\mathcal{D}_T$, $\mathcal{T}_T$, $\{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{X}_T, y_i \in \mathcal{Y}_T, 1 \le i \le n_T\}$ as the domain, task, and data for the target domain. DNNs typically require *inductive transfer learning* where the

feature space is the same across tasks (Pan and Yang (2009)), i.e., $\mathcal{X}_S = \mathcal{X}_T$. Therefore, we will drop the subscript on the feature space and simply use $\mathcal{X}$. Note that this does *not* imply that the conditional decision functions or the sampling distributions of the features are the same for the source and target.

2.3. *Transfer learning with DNN architectures.* As discussed in Section 2.1, DNNs are composed of many layers, each performing a transformation of the data. These transformations can be viewed as learned representations. It has been argued that different layers may learn different concepts relating to the task (Dube et al. (2020)). Specifically, there is evidence that early layers learn general representations of the data, while the the deeper layers are more task specific and learn specialized representations. In Zeiler and Fergus (2013), the authors studied the activation of early layers of DNNs in computer vision problems, where convolutional layers are typically used, and noted that early convolutional layers extract high-level features with the potential to generalize to different image domains. Using this logic, it follows that transfer learning for DNNs amounts to determining how many early layers to share across tasks and how many to assign as task specific.

As defined in (1), in DNNs the tasks are defined by the weight parameters for each layer of the network. Assuming the same network architecture for both tasks, define $\boldsymbol{\theta}_S = \{\boldsymbol{\theta}_{S1}, \ldots, \boldsymbol{\theta}_{SL}\}$ and $\boldsymbol{\theta}_T = \{\boldsymbol{\theta}_{T1}, \ldots, \boldsymbol{\theta}_{TL}\}$ as the parameters for the source and target tasks, respectively. A transfer-learning strategy with DNNs can be defined as the process of learning simultaneously the parameters $\boldsymbol{\theta}_S$ to augment the learning of $\boldsymbol{\theta}_T$ on the target task. The transferred layers can either be held fixed on the target task (referred to as *freezing*) or fine-tuned. Recently, Dube et al. (2020) investigated the empirical performances of freezing and fine-tuning. In their work they focused on a computer vision task with pretrained layers from different architectures that were fit to the Imagenet data set. The conclusion was that freezing pretrained layers, although not optimal, was always better than random initialization. The best performance was obtained when the learning rate for the pretrained layers was significantly lower than the rate of the other layers in the model. They advocate for a factor of 10% as a default choice.

3. PCPs for deep Bayesian transfer learning. For the target task, let $\boldsymbol{\theta}_T \equiv \boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$, and denote v_i as the number of parameters in $\boldsymbol{\theta}_i$; for the source task, denote $\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_L\}$ as the estimated values for $\boldsymbol{\theta}$ that are trained from the source data, and when available, let Σ_i be the covariance matrix for $\boldsymbol{\theta}_i$ (e.g., the posterior covariance, given the source data). Our goal is to leverage these estimated parameters to construct an informative prior for the target task. If the tasks were the same, the natural solution would be to set $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. On the other hand, if the tasks were not at all related, $\hat{\boldsymbol{\theta}}$ does not encode any information for the target task and would probably be worse than random initialization. In between these two extreme cases lies the core of transfer learning, where there is a reason to assume that tasks are related but we do not know to which degree.

The focus of our analysis is a judicious choice of the prior distribution for $\boldsymbol{\theta}$. Constructing a prior distribution that uses $\hat{\boldsymbol{\theta}}$ requires, first, the choice of a method for computing $\hat{\boldsymbol{\theta}}$ from the source data, and second, requires specification of the prior distribution for $\boldsymbol{\theta}$, given $\hat{\boldsymbol{\theta}}$. As described in the next subsection, we use the PCP of Simpson et al. (2017) for the prior distribution of $\boldsymbol{\theta}$, given $\hat{\boldsymbol{\theta}}$. The remainder of this section provides an overview the PCP idea of Simpson et al. (2017) and then describes two methods we propose for constructing $\hat{\boldsymbol{\theta}}$. The PCP is derived for both methods of constructing $\hat{\boldsymbol{\theta}}$.

3.1. *Penalized complexity priors.* In the framework of Simpson et al. (2017), the problem of prior choice is considered from a model complexity perspective. The prior on the parameter of interest θ is controlled by a flexibility parameter τ. The base (or simplest) model for θ, which we will denote by $p_0(\theta)$, corresponds to the case where $\tau = 0$, and for larger values of τ, the prior deviates from the base model. Following Occam's razor, simple models should be preferred until there is evidence for more complex ones; in other words, the hyperprior on τ should favor the base model. PCPs, as defined in Simpson et al. (2017), give a principled way of choosing a prior on τ that controls the deviation from the base model by penalizing a scaled version of the Kullback–Liebler divergence from $p_0(\theta)$ to $p(\theta|\tau)$,

$$d_{\mathrm{KL}}(p||p_0) = \sqrt{2\mathrm{KL}(p||p_0)} = \left(2\int_\theta p(\theta|\tau)\log\left(\frac{p(\theta|\tau)}{p_0(\theta)}\right)d\theta\right)^{1/2}. \tag{2}$$

From (2) it is clear that $d_{\mathrm{KL}}(p||p_0)$ is a function of τ. That being so, let

$$h(\tau) = d_{\mathrm{KL}}\big(p(\theta|\tau)||p_0(\theta)\big),$$

and notice that $h(\tau) = 0$ if and only if $\tau = 0$. Furthermore, if $h(\cdot)$ is strictly increasing, penalising $h(\tau)$ for large values induces a prior that penalizes large values of τ and, consequently, puts more mass close to the base model. Simpson et al. (2017) recommend using an exponential distribution, $h(\tau) \sim \mathrm{Exp}(\lambda)$ to ensure a mode at $\tau = 0$ and a constant rate of penalization for larger τ. Let

$$p(\boldsymbol{\theta}|\boldsymbol{\tau}) = \prod_{i=1}^{L} N\big(\boldsymbol{\theta}_i; \hat{\boldsymbol{\theta}}_i, g(\tau_i)\Sigma_i\big),$$

where N is the multivariate Gaussian density function, g is an increasing function, the flexibility parameter τ_i controls the deviation from the base model in layer i, and $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_L)$. The case $\boldsymbol{\tau} = \mathbf{0}$ is used to define the base model. This layerwise specification reflects the fact that some layers can be more transferable than others and hence require more penalization.

If we marginalize over $\boldsymbol{\tau}$, the prior model becomes a scale mixture of Gaussian distributions, where the mixing distribution will be the prior on $\boldsymbol{\tau}$. This latter distribution will be implicitly defined by placing an exponential distribution on $h(\boldsymbol{\tau}) = d_{\mathrm{KL}}(p(\boldsymbol{\theta}_T|\boldsymbol{\tau})||p_0(\boldsymbol{\theta}_T))$. When $L > 1$, the implicit prior on $\boldsymbol{\tau}$ is not identifiable since the mapping $h : \mathbf{R}_+^L \longrightarrow \mathbf{R}_+$ cannot be bijective and hence is not invertible. To circumvent this issue, Simpson et al. (2017) extend the PCP prior to a multivariate parameter $\boldsymbol{\tau}$ by having the prior on $\boldsymbol{\tau}$ uniform on each level set of $h(\cdot)$. Instead, we opt for the simpler approach of constraining $\boldsymbol{\tau}$ to the $(L-1)$-simplex given by any weighted combination of its elements. This is equivalent to $\boldsymbol{\tau} = \tilde{\tau}\mathbf{c}$, where $\mathbf{c} \sim \mathrm{Dirichlet}(\mathbf{1}_L)$ and $\tilde{\tau} \in \mathbf{R}_+$. In this formulation $\tilde{\tau}$ becomes a global flexibility parameter, and the weights c_i control how much deviation each layer is allowed since $\tau_i = \tilde{\tau}c_i$. Thus, given the vector of weights $\mathbf{c}$, we have a one-to-one mapping $h_{\mathbf{c}}(\tilde{\tau})$ through which we have an explicit penalising prior on the global parameter $\tilde{\tau}$.

3.2. *PCP for the VI case.* If the source data are analyzed using VB methods, then $\hat{\boldsymbol{\theta}}_i$ is an approximate posterior mean and Σ_i is an approximate $v_i \times v_i$ posterior covariance matrix. Because standard VB methods approximate the posterior as independent across parameters, Σ_i is a diagonal matrix for each i. We incorporate this information in the prior by setting $g(\tau) = 1 + \tau$ so that the base model with $\boldsymbol{\tau} = \mathbf{0}$ uses the posterior of the source data directly as the prior for the target data, which would be optimal Bayesian learning if the two tasks are the same. If $\tau_i > 0$, then the prior variance increases to reflect uncertainty about the relationship between source and target tasks.

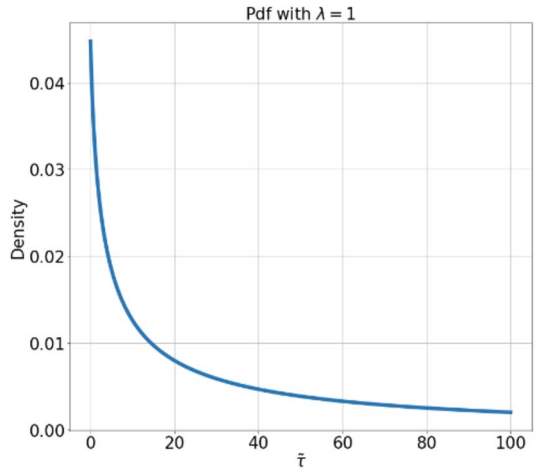

FIG. 1. *The form of the PCP distribution on* $\tilde{\tau}$ *with* $\lambda = 1$, $L = 3$, $c_i = 1/L$, $v_1 = v_2 = 16$, *and* $v_3 = 1$.

Under this prior distribution, the KL divergence is

$$\mathrm{KL}\big(p(\boldsymbol{\theta}|\tilde{\tau}, \mathbf{c})||p_0(\boldsymbol{\theta})\big) = \frac{1}{2}\left(\sum_{i=1}^{L} v_i c_i \tilde{\tau} - v_i \log(1 + c_i \tilde{\tau})\right).$$

So if $d = \sqrt{2\mathrm{KL}(p(\boldsymbol{\theta}|\tilde{\tau}, \mathbf{c})||p_0(\boldsymbol{\theta}))} \sim \mathrm{Exp}(\lambda)$, the one-to-one mapping between d and $\tilde{\tau}$ is

$$d = \sqrt{\sum_{i=1}^{L} v_i \phi(c_i \tilde{\tau})}, \tag{3}$$

where $\phi(x) = x - \log(1 + x)$. Furthermore, ϕ is continuously differentiable and strictly increasing; hence, given $\mathbf{c}$, we have a unique prior on $\tilde{\tau}$ corresponding to the desired prior on the KL divergence from the base model. The induced prior is

$$p(\tilde{\tau}|\mathbf{c}) = p\big(h_{\mathbf{c}}(\tilde{\tau})\big)\left|\frac{dh_{\mathbf{c}}(\tilde{\tau})}{d\tilde{\tau}}\right| = \frac{\lambda e^{-\lambda h_{\mathbf{c}}(\tilde{\tau})}}{2(\sum_{i=1}^{L} v_i \phi(c_i \tilde{\tau}))^{1/2}}\left(\sum_{i=1}^{L} v_i \frac{c_i^2 \tilde{\tau}}{1 + c_i \tilde{\tau}}\right), \tag{4}$$

with $h_{\mathbf{c}}(\tilde{\tau})^2 = \sum_{i=1}^{L} v_i \phi(c_i \tilde{\tau})$. Equation (4), coupled with the Dirichlet prior on the weight $\mathbf{c}$, completely specifies a prior distribution on the vector of scales $\boldsymbol{\tau}$. Using numerical inversion of the mapping in (3), sampling from this prior is straightforward.

It can be shown that the induced prior on the global flexibility parameter in (4) has a mode at zero with a strictly decreasing density, as $\tilde{\tau}$ increases. Also, the tail of the prior behaves like a modified Weibull distribution (Almalki and Nadarajah (2014)) with rate $\lambda \sum v_i c_i$ and shape 0.5. Figure 1 plots the density when $\lambda = 1$.

3.3. *PCP for the point estimate case.* Assume that a point estimate $\hat{\boldsymbol{\theta}}$ is derived from the source data but no measure of uncertainty Σ is provided. In this case the source data does not define a base distribution for $\boldsymbol{\theta}$, and thus we cannot compute the KL divergence between base and full models for $\boldsymbol{\theta}$ without further assumptions. Since DNNs are primarily used for prediction, we place a prior on the KL divergence between the predictive model under the source and target models. Let $f(\mathbf{x}|\boldsymbol{\theta})$ denote the target DNN output for a given input $\mathbf{x}$ and parameter values $(\boldsymbol{\theta})$. Let the regression model likelihood be $p(y) \sim \mathrm{Normal}(f(\mathbf{x}|\boldsymbol{\theta}), \sigma^2)$ and $\boldsymbol{\theta}_i \sim \mathrm{Normal}(\hat{\boldsymbol{\theta}}_i, \tau_i \mathbf{I})$, which induces a prior distribution on the function f. Unfortunately, for

the DNN model it is not available in analytical form. To overcome this, we approximate the DNN output using a first-order Taylor approximation around $\hat{\boldsymbol{\theta}}$

$$f(\mathbf{x}|\boldsymbol{\theta}) \approx f(\mathbf{x}|\hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\mathrm{T}} \nabla_{\hat{\boldsymbol{\theta}}} f(\mathbf{x}|\hat{\boldsymbol{\theta}}). \tag{5}$$

For small values of $\tilde{\tau}$ (i.e., the most relevant values for the prior, assuming the target and source are similar), a first-order approximation is arguably tight enough. Furthermore, if the activation functions used in the model are piecewise linear, like the widely used ReLU function (Agarap (2018)), the output function $f(\mathbf{x}|\boldsymbol{\theta})$ will also be piecewise linear, and we cannot go beyond a first-order Taylor expansion.

Based on the approximation, for the base model with $\boldsymbol{\tau} = \mathbf{0}$, we get the prior distribution

$$p_0(y) \sim \text{Normal}\big(f(\mathbf{x}|\hat{\boldsymbol{\theta}},), \sigma^2\big),$$

and for the flexible model, the prior on $f(\mathbf{y}|\tilde{\tau}, \mathbf{c})$ is

$$p(y|\tilde{\tau}, \mathbf{c}) \dot{\approx} \text{Normal}\big(f(\mathbf{x}|\hat{\boldsymbol{\theta}}), \sigma^2 + \tilde{\tau}\mathbf{c}^{\mathrm{T}}\alpha(\mathbf{x})\big),$$

where $\alpha(\mathbf{x}) = (\|\nabla_{\hat{\boldsymbol{\theta}}_i} f(\mathbf{x}|\hat{\boldsymbol{\theta}},)\|^2)_{1 \le i \le L}$. Now that we have two normal models, an analytic expression for the KL distance is

$$\text{KL}\big(p(y|\tilde{\tau}, \mathbf{c})||p_0\big) = \frac{1}{2}\phi\big[\tilde{\tau}\tilde{\alpha}_{\mathbf{c}}(\mathbf{x})\big] \quad \text{where } \tilde{\alpha}_{\mathbf{c}}(\mathbf{x}) = \frac{\mathbf{c}^{\mathrm{T}}\alpha(\mathbf{x})}{\sigma^2}. \tag{6}$$

In equation (6) $\tilde{\alpha}(\mathbf{x})$ depends on the gradient of $f(\mathbf{x}|\hat{\boldsymbol{\theta}})$ for a given input point $\mathbf{x}$. Assuming that the inputs are identically distributed and we have N target data points, the marginal KL divergence can be approximated by

$$\text{KL}\big(p||p_0\big) \approx \frac{1}{2N}\sum_{i=1}^{N} \phi\big[\tilde{\tau}\tilde{\alpha}_{\mathbf{c}}(\mathbf{x}_i)\big]. \tag{7}$$

Using (7), we have a one-to-one mapping between $d = \sqrt{2\text{KL}(p||p_0)}$ and the global flexibility parameter $\tilde{\tau}$. Hence, an exponential distribution on d induces a prior on $\tilde{\tau}$, which can be obtained with the same simple change of variables as in (4). Let $d_{\mathbf{c}}(\tilde{\tau}) = \sum_{i=1}^{N} \phi[\tilde{\tau}\tilde{\alpha}_{\mathbf{c}}(\mathbf{x}_i)]/N$. Then

$$p(\tilde{\tau}|\mathbf{c}) = p\big(d_{\mathbf{c}}(\tilde{\tau})\big)\left|\frac{d_{\mathbf{c}}(\tilde{\tau})}{d\tilde{\tau}}\right| = \frac{\lambda e^{-\lambda d_{\mathbf{c}}(\tilde{\tau})}}{2(\frac{1}{N}\sum_{i=1}^{N}\phi[\tilde{\tau}\tilde{\alpha}_{\mathbf{c}}(\mathbf{x}_i)])^{1/2}}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{\alpha}_{\mathbf{c}}(\mathbf{x}_i)^2\tilde{\tau}}{1+\tilde{\alpha}_{\mathbf{c}}(\mathbf{x}_i)\tilde{\tau}}\right). \tag{8}$$

The form of the prior in (8) is similar to (4): in both cases the prior induced on $\tilde{\tau}$ has mode at zero, is a strictly decreasing density as $\tilde{\tau}$ increases, and has the same tail behavior.

We evaluate the approximation used in (5) to verify that the derived prior on $\boldsymbol{\tau}$ will result in an approximately exponential distribution on the KL divergence between the two models. We assume $L = 3$ layers with $\{64, 32, 1\}$ nodes, respectively. The source-data estimates $\hat{\boldsymbol{\theta}}_i$ are generated following the Glorot initialization scheme (Glorot and Bengio (2010)). Since we cannot compute the KL divergence analytically, the distribution of KL divergence is approximated with Monte Carlo sampling. The approximation is evaluated for 50 different target data sets generated, as described in Section 4. For each of those data sets, we compute the gradients in (5) and first draw $\mathbf{c}$ and $\tilde{\tau}$ from their priors for different values of the rate parameter λ, then sample $\boldsymbol{\theta}_i|\boldsymbol{\tau} \sim \text{Normal}(\hat{\boldsymbol{\theta}}_i, \tau_i\mathbf{I})$. We then sample 20,000 replications of the model output $f(\mathbf{x}; \boldsymbol{\theta})$ and approximate the KL divergence from the base model $f(\mathbf{x}; \hat{\boldsymbol{\theta}})$ for each replication to get the empirical distribution of the KL divergence. Figure 2 plots the cumulative hazard function of the scaled KL divergence d, which should be linear with slope λ if the distribution is Exponential(λ). The approximation holds near the origin for all λ and is tighter for the entire distribution for larger values of λ.

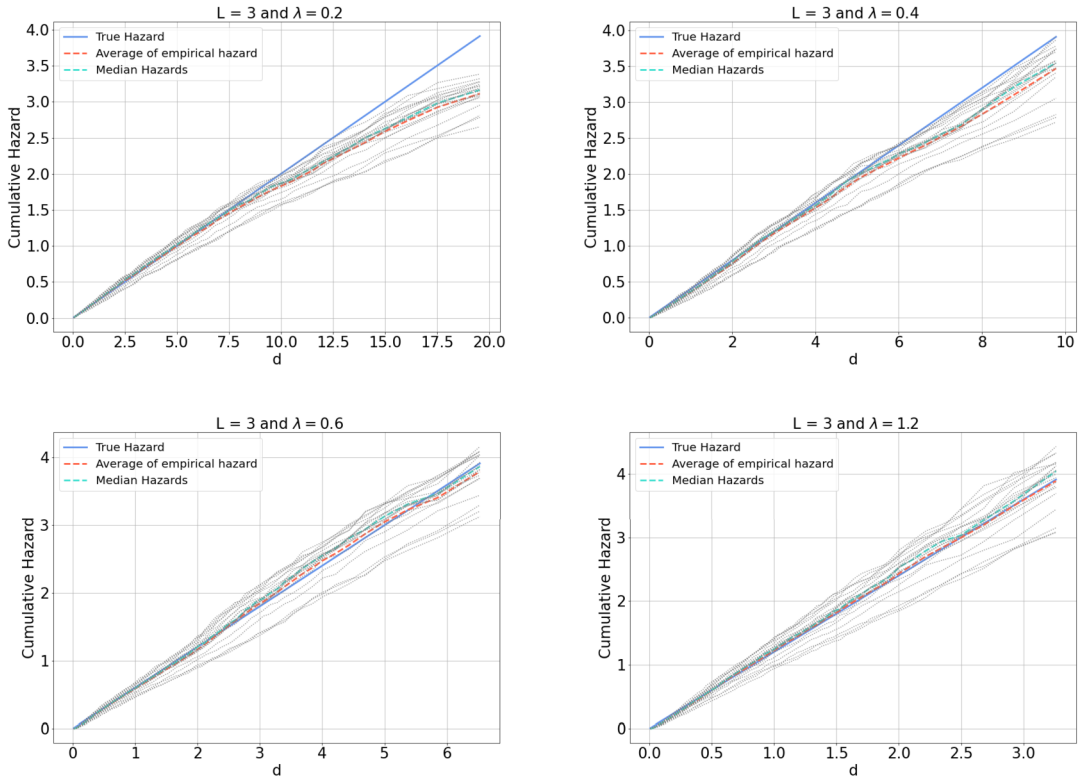

FIG. 2. *True hazard function of the scaled KL divergence* (*blue*), *and cumulative hazard for different generated data sets* (*gray*). *The dashed lines represent the median and mean of the empirical cumulative hazard.*

4. Simulation study. We perform the following simulation experiments to evaluate the predictive performance of the proposed Bayesian transfer-learning methods. The data are generated as follows. For observation i the covariates are generated as $x_{i1} \overset{\text{iid}}{\sim} \text{Normal}(0, 1)$ and $x_{ij}|x_{ij-1} \sim \text{Normal}(\rho x_{ij-1}, 1 - \rho^2)$ for $j \in \{2, \ldots, p\}$. The source data are generated for constants, c, σ, and k_1, as

$$Y_i \sim \text{Normal}\{c\mu_1(\mathbf{x}_i), \sigma^2\} \quad \text{with } \mu_1(\mathbf{x}_i) = \cos\left(2\sum_{j=1}^{k_1} x_{ij}/k_1\right),$$

and the target data are generated, for constant k_2, as

$$Y_i \sim \text{Normal}\{c\mu_2(\mathbf{x}_i), \sigma^2\} \quad \text{with } \mu_2(\mathbf{x}_i) = \cos\left(2\sum_{j=1}^{k_2} x_{ij}/k_2\right).$$

We set $p = 30$ covariates, so $\mathbf{x}_i \in \mathbf{R}^p$, $\rho = 0.5$, $\sigma = 1$, $k_1 = 15$, and $k_2 = k_1 + k$ with $k \in \{0, 5, 10, 15\}$, and we generate 1000 synthetic source training observations with a varying number of target training observations $n \in \{30, 50, 70\}$. The constant c is set so that the signal-to-noise ratio is 4:1 for the target data (i.e., so the variance of $c\mu_2(\mathbf{x}_i)$ is 4). If $k = 0$, then the two data-generating processes are identical, and if k is large, they are dissimilar. For each scenario we simulate 500 data sets and compare predictions on a test set of size $n_{\text{test}} = 200$.

Both the source and target data are analyzed using the fully-connected model with four hidden layers having widths 24, 16, 12, and 8, respectively. The source data are fit using the optimization scheme provided in Section 2 of the Supplementary Material (Abba et al.

(2023)); this produces estimates $\hat{\boldsymbol{\theta}}$ and their variances in the case of VI. We then fit several models to each target data set. The first group are non-Bayesian optimization-based transfer-learning "TL" methods. We consider two methods that fix some layers using the source fit and tune the remaining layers using the target data. The first model "TL1" shares no layers and is fit to the target data with no connection to the source data. The second model "TL2" proceeds in two stages; first, the output layer only is trained on the target data while the others are frozen, and next, in the second stage all layers are fine tuned.

The second group of methods are Bayesian. The first Bayesian model "BNN" ignores the source data by setting the prior $\boldsymbol{\theta}_i|\boldsymbol{\tau} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\tau}_i\mathbf{I})$ and $\tau_i \sim \text{InvGamma}(2, 1)$ so that the marginal standard deviation is 1. Although there is no consensus on the default prior choice in Bayesian deep learning, priors with zero mean and unit standard deviation are the most common (Fortuin (2022)). The second method, "BTL-PCP-PE," fits the Bayesian transfer-learning model with the priors centered at point estimates $\hat{\boldsymbol{\theta}}$ alone, as described in Section 3.3. The third model, "Bayes-PCP-VI," uses the PCP prior based on VI, as described in Section 3.2. For both PCP methods, we assume $\mathbf{c} \sim \text{Dirichlet}(\mathbf{1_L})$ and set λ so that the correlation between $f(\mathbf{x}_i; \boldsymbol{\theta})$ and Y_i is 0.5. All Bayesian method are fit using MCMC, as described in Section 3 of the Supplementary Material (Abba et al. (2023)).

The results are compared using MSE, and for the Bayesian methods, the coverage and width of 95% prediction intervals for $\mu_2(\mathbf{x}_i)$ over a target data test set of 200 observations is provided. The results are reported in Table 1. BLT-PCP-PE outperforms TL1 and TL2 in all cases, perhaps due to the added stability of prior distributions. When the source and target are similar (i.e., $k \in \{0, 5\}$), all the transfer-learning methods work well, especially if the number of training samples is small. When the discrepancy between the source and target domain is high, however, the transfer-learning methods do not improve the performance of the models. In fact, the point estimate method "TL-1" outperforms "TL-2," and the Bayesian methods are comparable. An interesting result is the poor performance of the PCP prior for the VI case compared to the other Bayesian methods. It seems that building a PCP prior, based on the mean-field VI posterior approximation, does not encode enough information for the model to learn from small data sets. However, while VI does not perform as well as MCMC, it is much faster than MCMC and still outperforms TL1 and TL2 in most cases and is thus a viable option for large data sets. The average computational time for each method is reprted in Table 2. Furthermore, when k is large, "BNN" outperforms all the other methods. This is expected since the prior distribution in the PCP has a sharp mode at zero which results in a strong penalizing effect. Thus, when the data sets are dissimilar and the number of target training samples is small, BNN performs better. Perhaps choosing a different distribution for the prior, with heavier tails, on the scaled Kullback–Leibler divergence could reduce this dependence on the source in such cases. That being said, this could also lead to a less sharp mode at the origin and hence less concentration around the true solution when the data sets are similar.

To mimic the the real data analysis in Section 5, we conduct a second set of simulations with the same generative model as above but with $p = 1000$ covariates, $\rho = 0.5$, $\sigma = 0.32$, and the constant $c = 1$ to approximate the 3:1 signal-to-noise in the data analysis. The number of active covariates is determined by $k_1 = 900$ and $k_2 = k_1 + k$ with $k \in \{0, 100\}$; if $k = 0$, then the two data-generating processes are identical, and if k is large, they are similar but not identical. We generate 10,000 synthetic source training observations and $n = 500$ target training observations. For each scenario we simulate 100 data sets and compare predictions on a test set of size $n_{\text{test}} = 10{,}000$. For these larger data sets, we also consider the VI approach of Zhou et al. (2020). The results in Table 3 resemble those in Table 1; that is, the BTL-PCP-PE approach has the smallest MSE and coverage near the nominal level.

TABLE 1

Simulation study results: *The non-Bayesian methods (“TL1-TL2”) use transfer learning by fixing different numbers of layers of the network using source data, the Bayesian methods are the Bayesian model without transfer learning (“BNN”) or full Bayesian transfer-learning model with point estimate (“BTL-PCP-PE”) and VI (“BTL-PCP-VI”) in the first stage. Table (a) gives the mean square prediction error (Monte-Carlo standard errors) over the 500 replicated data sets by the difference between the number of active covariates over tasks, k, and size of the target data set, n. Table (b) gives the average coverage (width) for the Bayesian 95% prediction intervals*

(a) Mean squared prediction error (Monte Carlo standard errors)

k	n	TL1	TL2	BNN	BTL-PCP-PE	BTL-PCP-VI
0	30	1.98 (0.06)	1.36 (0.03)	1.87 (0.06)	1.04 (0.03)	1.48 (0.05)
0	50	1.82 (0.09)	1.16 (0.03)	1.60 (0.03)	1.01 (0.04)	1.14 (0.09)
0	70	1.59 (0.08)	1.02 (0.04)	1.06 (0.03)	0.70 (0.02)	0.94 (0.05)
5	30	2.10 (0.05)	1.49 (0.09)	1.81 (0.05)	1.09 (0.02)	1.47 (0.05)
5	50	1.82 (0.05)	1.26 (0.04)	1.64 (0.03)	1.07 (0.04)	1.20 (0.03)
5	70	1.65 (0.07)	1.02 (0.05)	1.01 (0.04)	0.86 (0.02)	0.93 (0.01)
10	30	1.89 (0.10)	2.11 (0.05)	1.85 (0.09)	1.37 (0.03)	1.69 (0.03)
10	50	1.84 (0.06)	1.79 (0.05)	1.61 (0.08)	1.49 (0.06)	1.83 (0.07)
10	70	1.74 (0.10)	1.83 (0.04)	1.05 (0.04)	1.04 (0.05)	1.12 (0.06)
15	30	2.18 (0.09)	2.61 (0.03)	1.89 (0.04)	1.94 (0.02)	2.21 (0.04)
15	50	1.86 (0.07)	2.12 (0.05)	1.46 (0.08)	1.49 (0.07)	1.85 (0.08)
15	70	1.69 (0.05)	1.81 (0.06)	1.19 (0.03)	1.32 (0.02)	1.70 (0.03)

(b) Average coverage (Monte Carlo standard errors, average width) of 95% prediction intervals

k	n	BNN	BTL-PCP-PE	BTL-PCP-VI
0	30	0.86 (0.0089, 0.71)	0.94 (0.0076, 0.65)	0.78 (0.0118, 0.52)
0	50	0.88 (0.0081, 0.69)	0.95 (0.0057, 0.51)	0.82 (0.0095, 0.49)
0	70	0.93 (0.0072, 0.55)	0.98 (0.0044, 0.41)	0.90 (0.0099, 0.38)
5	30	0.86 (0.0096, 0.73)	0.91 (0.0099, 0.68)	0.76 (0.0102, 0.51)
5	50	0.90 (0.0083, 0.70)	0.95 (0.0064, 0.49)	0.83 (0.0094, 0.50)
5	70	0.92 (0.0064, 0.52)	0.97 (0.0056, 0.39)	0.90 (0.0087, 0.40)
10	30	0.85 (0.0099, 0.70)	0.90 (0.0092, 0.67)	0.73 (0.0134, 0.56)
10	50	0.91 (0.0090, 0.61)	0.94 (0.0065, 0.54)	0.89 (0.0106, 0.53)
10	70	0.95 (0.0081, 0.47)	0.97 (0.0057, 0.45)	0.91 (0.0096, 0.43)
15	30	0.82 (0.0097, 0.72)	0.78 (0.0096, 0.56)	0.65 (0.0127, 0.57)
15	50	0.91 (0.0082, 0.64)	0.90 (0.0091, 0.48)	0.80 (0.0119, 0.51)
15	70	0.95 (0.0068, 0.50)	0.94 (0.0074, 0.46)	0.88 (0.0081, 0.41)

5. Real data analysis. We apply the proposed transfer-learning methods for band gap prediction of the molecular crystal target data set. For training we consider only subsets of the data with cardinality 1000, 2500, or 5000 samples to represent a typical data set size, a validation set of 10,000 samples, and finally a testing set of 20,000 samples. Since we are dealing with molecules, we first need to preprocess the data and compute descriptors $\mathbf{x}$ that

TABLE 2

Approximate computational times of the methods used in the simulation study for data sets of size 1000. The Bayesian methods take approximately six times more time to train compared to classical methods

	Computational time per method				
	TL1	TL2	BNN	BTL-PCP-PE	BTL-PCP-VI
Time in minutes	15.12	15.09	98.04	103.41	21.05

TABLE 3

Simulation study results: *The non-Bayesian methods ("TL1-TL2") use transfer learning by fixing different numbers of layers of the network using source data, the Bayesian methods are the Bayesian model without transfer learning ("BNN"), full Bayesian transfer-learning model with point estimate ("BTL-PCP-PE") and VI ("BTL-PCP-VI") in the first stage, and Bayesian transfer learning prior from Zhou et al. (2020) "BTL-VI." Table (a) gives the mean square prediction error by the difference between the number of active covariates over tasks, k, and size of the target data set, n. Table (b) gives the average coverage (width) for the Bayesian 95% prediction intervals*

(a) Mean squared prediction error (standard errors)

k	n	TL1	TL2	BNN	BTL-PCP-PE	BTL-PCP-VI	BTL-VI
0	500	0.68 (0.12)	0.65 (0.08)	0.76 (0.11)	0.44 (0.08)	0.68 (0.09)	0.70 (0.15)
100	500	0.70 (0.08)	0.72 (0.09)	0.72 (0.13)	0.47 (0.11)	0.71 (0.12)	0.73 (0.13)

(b) Average coverage (average width) of 95% prediction intervals

k	n	BNN	BTL-PCP-PE	BTL-PCP-VI	BTL-VI
0	500	0.90 (0.41)	0.94 (0.35)	0.79 (0.22)	0.82 (0.19)
100	500	0.89 (0.35)	0.92 (0.38)	0.73 (0.18)	0.80 (0.24)

can be fed into the DNN. We use the MBTR descriptor method (Huo and Rupp (2018)) to compute $p = 1260$ descriptors for every data point in both the source and target data sets. These descriptors are functions of the structure of the material, for example, the types and configuration of its atoms. The architecture of the model will have two hidden layers with 128 and 64 neurons, respectively. First, the model is trained on the source data with 40,000 samples for training set and 10,000 for the validation. The parameters of the best performing model in terms of mean squared error (MSE) are saved and transferred to the target task as $\hat{\boldsymbol{\theta}}$.

On the target task, we consider four different transfer-learning methods. The non-Bayesian methods differ in the treatment of the hidden layers of the model; we can either freeze the hidden layers and treat them as feature extractors and only fit the output layer to the target task "TL-freeze," or we can fine-tune the hidden layers along with the output layer on the target task "TL-fine-tune." For both Bayesian methods, the priors are $\boldsymbol{\theta}_i|\boldsymbol{\tau} \sim \text{Normal}(\hat{\boldsymbol{\theta}}_i, \tau_i \mathbf{I})$. The two methods differ in the priors on $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$; we compare the PCP with point estimate approach "BTL-PCP-PE" from Section 3.3 with $\mathbf{c} \sim \text{Dirichlet}(1, 1, 1)$ to the uninformative prior approach with $\tau_i \sim \text{InvGamma}(2, 1)$ "BTL-BNN." Since the point estimate approach outperformed the VI approach in the simulation study, we consider only the point estimate approach in this data analysis. To verify that transfer learning provides benefit over using only the target data, we compare to the state-of-the-art DNN model used in materials science where DNNs have been developed as end-to-end models in the sense that they take as input the molecular representation and learn their own embedding. Two of the leading methods are SHNET (Schütt et al. (2017)) and MEGNET (Chen et al. (2019)).

The target data are split into groups of size 5000, 10,000, and 20,000 for training so that each group contains materials of different classes and the band gap distribution stays roughly the same. We train the model for samples of size $n \in \{1000, 2500, 5000\}$. Figure 3 plots the MSE on a test set of size 20,000 for each method and training set, and Figure 4 plots the observed and predicted values for the BTL-PCP-PE method. As expected, MSE decreases for each method as the size training set increases, and in all cases the BTL-PCP-PE method gives the smallest MSE.

With a fully Bayesian treatment of all the parameters, we can look at the summary statistics of $\mathbf{c}$, the parameter that controls which layers contain the useful information from the source

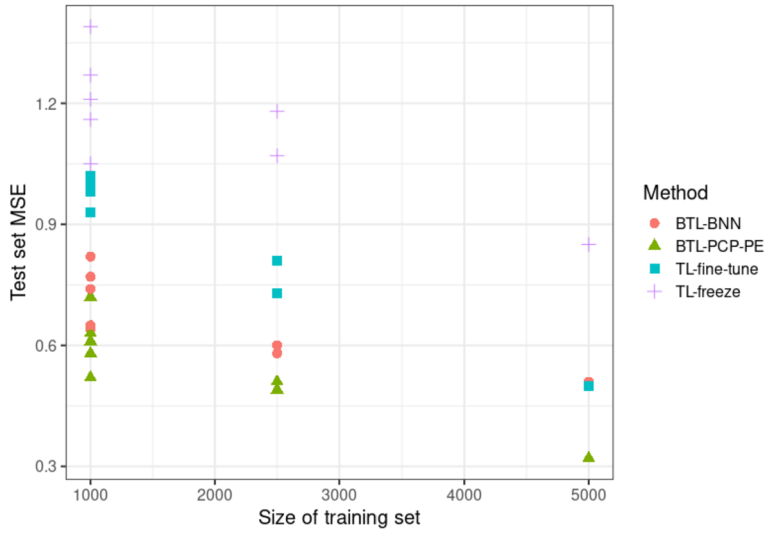

FIG. 3. *Test set mean squared error (MSE; eV^2) for different transfer-learning methods applied to band-gap prediction. The methods are plotted against the size of the target data training set. Each point represents the MSE for one fold in the cross validation. For comparison, the sample variance of the response is* 1.12 eV^2.

data and which are irrelevant. For training set size 5000, the posterior means (standard deviation) for c_1, c_2, and c_3 are 0.05 (0.02), 0.05 (0.03), and 0.90 (0.15), respectively. Therefore, the first two layers are shrunk toward the source data fits while the output layer varies from the source-model fit. The Bayesian methods also provide prediction intervals. The empirical coverage and average width of 95% prediction intervals for the test set are, respectively, 0.91 and 1.24 for BTL-BNN and 0.94 and 1.16 for BTL-PCP-PE.

These results are competitive with state-of-the-art methods in the materials community. For a training set of size 5000 (the same 5000 as in Figure 3), the MSE for SHNET and MEGNET are 0.48 eV^2 and 1.01 eV^2, respectively. In fact, using the same data except with a much larger training set of 30,000 observations, the methods achieved MSE of 0.59 eV^2 for SHNET and 0.34 eV^2 for MEGNET.

6. Conclusions. For molecular crystals the computational cost of band gap values can be orders of magnitude higher in comparison to organic molecules. Recently, DNN models reached chemical accuracy levels for band gap prediction on small organic molecules and can now be reliably used as a fast screening method. However, these models often require large data sets for training and can have unstable performance on small data sets. Using transfer

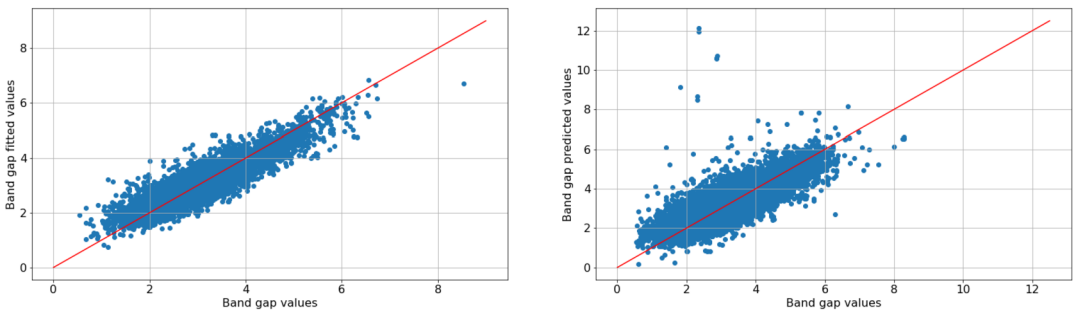

FIG. 4. *Predicted vs. band-gap for the training (left) and test (right) sets for the BLT-PCP-PE methods with* $n = 5000$ *training observations. The units of the plot are electronvolts.*

learning, we leverage the accuracy of neural networks models on organic molecules to build a data efficient model with accurate predictions and calibrated uncertainty quantification for molecular crystals band gap prediction. Our new transfer-learning methods provide regularization by centering the prior distribution on estimates from an auxiliary source data set, that is, organic molecules. We use the PCP framework to ensure that the prior concentrates around the source model but is allowed to deviate if appropriate. We develop two methods for cases where the source data analysis does and does not provide uncertainty measures for the parameter estimates. We show via simulation that the proposed methods reduce prediction error, compared to standard transfer-learning methods, and unlike standard methods, the Bayesian approach gives reasonable coverage for prediction intervals.

The accuracy of our method for large molecular crystals matches state-of-the-art models while only requiring a sixth of the training data. The performance on both simulated and real data sets indicate that the methods can be used to guide and optimize experimentation and selection of materials with specific properties without the need for a large training sample, hence reducing the cost associated with DFT calculations. The potential scope of the application of the developed transfer-learning methods goes beyond material science and includes areas of applied statistical learning where data labelling is costly and auxiliary source tasks are available.

There are many areas of future work. We have developed our method in the simplest case of a continuous response and feed forward network. The prior based on VI in Section 3.2 would apply directly to other networks and response distributions, as the prior is only a function of the approximate Gaussian posterior distribution from the source data analysis. Similarly, the prior based on point estimates in Section 3.3 can be applied without modification to non-Gaussian responses and richer architecture. Changing the network architecture would change the form of the gradients in (5), but the method itself can be used without modification. The method also applies for noncontinuous responses if we view f as a process that spans the real line that is related to the response distribution. For example, for binary data, f in (5) could be the logistic function of the success probability, which spans the real line, and thus the proposed methods could be applied on this scale. It may also be possible to extend the PCP prior from neural networks to other frameworks such as tree-based methods (e.g., Chen and Guestrin (2016), Chipman, George and McCulloch (2010)).

Next, our method applies an informative prior on the overall variance parameter but a uniform prior on the distribution of the variance across levels, $\mathbf{c}$. If prior information about the transferability of different layers is available, it could be incorporated in the prior for $\mathbf{c}$. For example, it may be reasonable to assume the prior mean of c_i increases from the input to the output layer. Lastly, another interesting area of future work is to compare the two-stage analysis that sequentially analyzes the source then target data with a simultaneous analysis of all data sources. Here we have assumed that building a hierarchical model for all data sources is computationally prohibitive, but it would be interesting to compare the efficiency of these two approaches when possible, especially when the source data set is small and thus the parameter estimates are uncertain.

Acknowledgments. The authors thank Noa Marom and Xingyu Liu of Carnegie Mellon University for assistance with the data collection and interpretation.

Funding. The authors thank the National Science Foundation (CMMT-1844484, CMMT-2022254, DGE-1633587, DMS-2152887), King Abdullah University of Science and Technology (3800.20), and the National Institutes of Health (R56HL155373) for supporting this research.

SUPPLEMENTARY MATERIAL

Appendices (DOI: 10.1214/23-AOAS1759SUPPA; .pdf). The Appendix is divided into three sections. The first section details the complete procedure to evaluate the approximation in (5). The second section provides all the details of the optimization procedure for all the non-Bayesian methods considered in the paper. Finally, the last section covers the details of the MCMC details for the Bayesian ethods.

Codes for the simulation and real data analysis (DOI: 10.1214/23-AOAS1759SUPPB; .zip). All the codes used for the simulation study and the real data analysis are provided in one zip file.

REFERENCES

ABBA, M. A., WILLIAMS, J. P. and REICH, B. J. (2023). Supplement to "A penalized complexity prior for deep Bayesian transfer learning with application to materials informatics." https://doi.org/10.1214/23-AOAS1759SUPPA, https://doi.org/10.1214/23-AOAS1759SUPPB

AGARAP, A. F. (2018). Deep learning using rectified linear units (RELU). Available at arXiv:1803.08375.

AHISHAKIYE, E., VAN GIJZEN, M. B., TUMWIINE, J., WARIO, R. and OBUNGOLOCH, J. (2021). A survey on deep learning in medical image reconstruction. *Intelligent Medicine*.

ALMALKI, S. J. and NADARAJAH, S. (2014). Modifications of the Weibull distribution: A review. *Reliab. Eng. Syst. Saf.* **124** 32–55.

BATRA, R. (2021). Accurate machine learning in materials science facilitated by using diverse data sources. *Nature* **589** 524–525. https://doi.org/10.1038/d41586-020-03259-4

BENGIO, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks*: *Tricks of the Trade* 437–478. Springer, Berlin.

BLUNDELL, C., CORNEBISE, J., KAVUKCUOGLU, K. and WIERSTRA, D. (2015). Weight Uncertainty in Neural Networks.

BUENO, A., BENITEZ, C., DE ANGELIS, S., MORENO, A. D. and IBÁÑEZ, J. M. (2019). Volcano-seismic transfer learning and uncertainty quantification with Bayesian neural networks. *IEEE Trans. Geosci. Remote Sens.* **58** 892–902.

CHANDRA, R. and KAPOOR, A. (2020). Bayesian neural multi-source transfer learning. *Neurocomputing* **378** 54–64.

CHEN, C., YE, W., ZUO, Y., ZHENG, C. and ONG, S. P. (2019). Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.* **31** 3564–3572.

CHEN, T. and GUESTRIN, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining* 785–794.

CHIPMAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2010). BART: Bayesian additive regression trees. *Ann. Appl. Stat.* **4** 266–298. MR2758172 https://doi.org/10.1214/09-AOAS285

DARGAN, S., KUMAR, M., ROHIT AYYAGARI, M. and KUMAR, G. (2020). A survey of deep learning and its applications: A new paradigm to machine learning. *Arch. Comput. Methods Eng.* **27** 1071–1092. MR4138449 https://doi.org/10.1007/s11831-019-09344-w

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K. and FEI-FEI, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 *IEEE Conference on Computer Vision and Pattern Recognition* 248–255. IEEE, New York.

DUBE, P., BHATTACHARJEE, B., PETIT-BOIS, E. and HILL, M. (2020). Improving transferability of deep neural networks. In *Domain Adaptation for Visual Understanding* 51–64. Springer, Berlin.

FORTUIN, V. (2022). Priors in Bayesian deep learning: A review. *Int. Stat. Rev.* **90** 563–591. MR4524825

GLOROT, X. and BENGIO, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (*AISTATS'*10). *Society for Artificial Intelligence and Statistics*.

GOODFELLOW, I., BENGIO, Y. and COURVILLE, A. (2016). *Deep Learning*. *Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR3617773

HIMANEN, L., GEURTS, A., FOSTER, A. S. and RINKE, P. (2019). Data-driven materials science: Status, challenges, and perspectives. *Adv. Sci.* **6** 1900808. https://doi.org/10.1002/advs.201900808

HUO, H. and RUPP, M. (2018). Unified representation of molecules and crystals for machine learning. Available at arXiv:1704.06439.

KARBALAYGHAREH, A., QIAN, X. and DOUGHERTY, E. R. (2018). Optimal Bayesian transfer learning. *IEEE Trans. Signal Process.* **66** 3724–3739. MR3841740 https://doi.org/10.1109/TSP.2018.2839583

KITTEL, C., MCEUEN, P. and WILEY, J. (2019). Introduction to solid state physics.

KOHN, W. and SHAM, L. J. (1965). Self-consistent equations including exchange and correlation effects. *Phys. Rev.* (2) **140** A1133–A1138. MR0189732

LECUN, Y., BENGIO, Y. et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks* **3361** 1995.

MA, S., MA, Y., ZHANG, B., TIAN, Y. and JIN, Z. (2021). Forecasting system of computational time of DFT/TDDFT calculations under the multiverse ansatz via machine learning and cheminformatics. *ACS Omega* **6** 2001–2024.

MAURER, A., PONTIL, M. and ROMERA-PAREDES, B. (2016). The benefit of multitask representation learning. *J. Mach. Learn. Res.* **17** Paper No. 81. MR3517104

MEDSKER, L. R. and JAIN, L. (2001). Recurrent neural networks. *Design and Applications* **5** 64–67.

O'HAGAN, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliab. Eng. Syst. Saf.* **91** 1290–1300.

OLSTHOORN, B., GEILHUFE, R. M., BORYSOV, S. S. and BALATSKY, A. V. (2019). Band gap prediction for large organic crystal structures with machine learning. *Advanced Quantum Technologies* **2** 1900023.

PAN, S. J. and YANG, Q. (2009). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22** 1345–1359.

RAGHU, M., ZHANG, C., KLEINBERG, J. M. and BENGIO, S. (2019). Transfusion: Understanding Transfer Learning with Applications to Medical Imaging. *CoRR*. Available at arXiv:1902.07208.

SCHÜTT, K. T., KINDERMANS, P.-J., SAUCEDA, H. E., CHMIELA, S., TKATCHENKO, A. and MÜLLER, K.-R. (2017). Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. ArXiv preprint. Available at arXiv:1706.08566.

SENIOR, A. W., EVANS, R., JUMPER, J., KIRKPATRICK, J., SIFRE, L., GREEN, T., QIN, C., ŽÍDEK, A., NELSON, A. W. et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature* **577** 706–710.

SIMPSON, D., RUE, H., RIEBLER, A., MARTINS, T. G. and SØRBYE, S. H. (2017). Penalising model component complexity: A principled, practical approach to constructing priors. *Statist. Sci.* **32** 1–28. MR3634300 https://doi.org/10.1214/16-STS576

STUKE, A., KUNKEL, C., GOLZE, D., TODOROVIĆ, M., MARGRAF, J. T., REUTER, K., RINKE, P. and OBERHOFER, H. A. (2019). "OE62-dataset" of molecular orbital energies.

TAN, C., SUN, F., KONG, T., ZHANG, W., YANG, C. and LIU, C. (2018). A survey on deep transfer learning. In *International Conference on Artificial Neural Networks* 270–279. Springer, Berlin.

VAN DE WALLE, C. G. (2012). *Wide-Band-Gap Semiconductors*. Elsevier, Amsterdam.

VOULODIMOS, A., DOULAMIS, N., DOULAMIS, A. and PROTOPAPADAKIS, E. (2018). Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018** 7068349. https://doi.org/10.1155/2018/7068349

WEISS, K., KHOSHGOFTAAR, T. M. and WANG, D. (2016). A survey of transfer learning. *J. Big Data* **3** 1–40.

WILSON, A. G. (2020). The case for Bayesian deep learning. ArXiv preprint. Available at arXiv:2001.10995.

WOHLERT, J., MUNK, A., SENGUPTA, S. and LAUMANN, F. (2018). Bayesian transfer learning for deep networks. *ViXra*.

YANG, H., JIAO, S. and SUN, P. (2020). Bayesian-convolutional neural network model transfer learning for image detection of concrete water-binder ratio. *IEEE Access* **8** 35350–35367.

YOUNG, T., HAZARIKA, D., PORIA, S. and CAMBRIA, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **13** 55–75.

ZEILER, M. D. and FERGUS, R. (2013). Visualizing and Understanding Convolutional Networks. *CoRR*. Available at arXiv:1311.2901.

ZHANG, C., BUTEPAGE, J., KJELLSTROM, H. and MANDT, S. (2019). Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **41** 2008–2026. https://doi.org/10.1109/TPAMI.2018.2889774

ZHOU, C., ZHANG, J., LIU, J., ZHANG, C., SHI, G. and HU, J. (2020). Bayesian transfer learning for object detection in optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **58** 7705–7719.