A General Class of Transfer Learning Regression without Implementation Cost

Shunya Minami  
The Graduate University for Advanced Studies  
mshunya@ism.ac.jp

Song Liu  
University of Bristol  
song.liu@bristol.ac.uk

Stephen Wu  
The Institute of Statistical Mathematics  
stewu@ism.ac.jp

Kenji Fukumizu  
The Institute of Statistical Mathematics  
fukumizu@ism.ac.jp

Ryo Yoshida  
The Institute of Statistical Mathematics  
yoshidar@ism.ac.jp

Abstract

We propose a novel framework that unifies and extends existing methods of transfer learning (TL) for regression. To bridge a pretrained source model to the model on a target task, we introduce a density-ratio reweighting function, which is estimated through the Bayesian framework with a specific prior distribution. By changing two intrinsic hyperparameters and the choice of the density-ratio model, the proposed method can integrate three popular methods of TL: TL based on cross-domain similarity regularization, a probabilistic TL using the density-ratio estimation, and fine-tuning of pretrained neural networks. Moreover, the proposed method can benefit from its simple implementation without any additional cost; the model can be fully trained using off-the-shelf libraries for supervised learning in which the original output variable is simply transformed to a new output. We demonstrate its simplicity, generality, and applicability using various real data applications.

1 Introduction

Transfer learning (TL) is an increasingly popular machine learning framework that covers a broad range of techniques to which a set of models trained on source tasks is repurposed on another task of interest. It is proven that TL has the potential to significantly improve the prediction performance on the target task, in particular, under a limited supply of training data in which the learning from scratch is less effective. To date, the most outstanding successes of TL have been brought from deep neural networks. One or more layers in pretrained neural networks are refined to the new task with the limited target dataset while the remaining layers are either left frozen (fine-tuning) or almost unchanged during the cross-domain adaptation.

In this study, we aim to establish a new class of TL, which is applicable to any regression models. The proposed class unifies different classes of existing TL methods for regression. To model the transition from a pretrained model to a new model, we introduce a density-ratio reweighting function. The density-ratio function is estimated by conducting a Bayesian inference with a specific prior distribution while keeping the given source model unchanged. Two hyperparameters and the choice of the density-ratio model characterize the proposed class. It can integrate and extend three popular methods of TL within a unified framework, including TL based on the cross-domain similarity...
We employ Bayesian inference to estimate the unknown parameters. We are given a pretrained model $\theta$ that is trained on a source domain $\mathcal{D}_s$. The two building blocks of the likelihood function are modelled by $p_s(y|x, \theta)$ and $p_s(y|x)$, respectively. The normalization constraint is subject to the conditional distribution $p_s(y|x, \theta) = \int p(t|(y,x),\theta)dt$. The density-ratio function $w(y,x)$ depends on the given model $f_s(x)$ and the source model $f_s(x)$. The source distribution is modelled by $p_s(y|x, \theta)$, and the two hyperparameters and a model for the density-ratio function are selected through cross-validation.

Consider that the source distribution is modelled by $p_s(y|x, \theta)$, and the density-ratio method operates with an opposite learning objective that we call the cross-domain dissimilarity regularization; the discrepancy between the source domain and the newly trained model on the discrepancy. These totally different methods can be unified within the proposed framework.

To summarize, the methodological features and contributions of our method are as follows:

- The method can operate with any kinds of regression models.
- The proposed class, which has two hyperparameters, can unify and hybridize three existing methods of TL, including the regularization based on cross-domain similarity and dissimilarity.
- The two hyperparameters and a model for the density-ratio function are selected through cross-validation. With this unified workflow, an ordinary supervised learning without transfer can also be chosen if the previous learning experience interferes with learning in the new task.
- The proposed method can be implemented with no extra cost. With a simple transformation (1), the density-ratio model $w(y,x)$ is associated with a regression model $f_s(x)$.

Practical benefits of bridging totally different methods in the unified workflow are demonstrated on a wide range of prediction tasks in science and engineering applications.

2 Proposed method

We are given a pretrained model $y = f_s(x)$ on the source task, which defines the mapping between any input $x$ to a real-valued output $y \in \mathbb{R}$. The objective is to transform the given $f_s(x)$ into a target model $y = f_t(x)$ by using $n$ instances from the target domain, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$.

Inspired by [7], we apply the probabilistic modeling for the transition from $f_s(x)$ to $f_t(x)$. With the conditional distribution $p_s(y|x)$ of the source task, the one on the target can be written as

$$p_t(y|x) = \frac{w(y,x)p_s(y|x)}{w(y,x) = p_t(y|x)p_s(y|x)}.$$  \hspace{1cm} (1)

Consider that the source distribution is modelled by $p_s(y|x, f_s)$ which involves the pretrained $f_s(x)$. In addition, the density-ratio function $w(y,x)$ is separately modeled as $w(y,x)\theta_w$ with an unknown parameter $\theta_w$, which will be associated with a regression model $f_{\theta_w}(x)$. The target model $p_t(y|x, \theta_w)$ is then

$$p_t(y|x, \theta_w) = w(y,x)\theta_w p_s(y|x, f_s)$$  \hspace{1cm} (2)

where the normalization constraint is subject to the conditional distribution.

We employ Bayesian inference to estimate the unknown $\theta_w$ in the density-ratio model $w(y,x)\theta_w$. The target model $p_t(y|x, \theta_w)$ is given to the likelihood, and a prior distribution $p(\theta_w|f_s)$ is placed on $\theta_w$, which depends on the given $f_s$. The posterior distribution is then

$$p(\theta_w|\mathcal{D}) \propto \prod_{i=1}^n p_t(y_i|x_i, \theta_w) p(\theta_w|f_s).$$  \hspace{1cm} (3)

The two building blocks of the likelihood function are modelled by

$$w(y,x) = \exp \left(-\frac{(y - f_{\theta_w}(x))^2}{\sigma}\right) \quad \text{and} \quad p_s(y|x, f_s) = \exp \left(-\frac{(y - f_s(x))^2}{\eta}\right),$$  \hspace{1cm} (4)

where $\sigma > 0$ and $\eta > 0$. The normalization constant for the product of the two expressions on the right-hand side of Eq. (4) is given as $\exp \left(-(\sigma + \eta)^{-1}(f_s(x) - f_{\theta_w}(x))^2\right)$, which depends on
We consider the MAP (maximum a posteriori) estimation of $\theta$. Any regularization term, such as $\ell$ with the learning algorithms, the resulting class can bridge various methods of TL as described later.

The discrepancy is measured by the sum of their squared distances over $m$ input values $\mathcal{U} = \{u_i\}_{i=1}^{m}$. Hereafter, we use the $n$ observed inputs in $\mathcal{D}$ for $\mathcal{U}$. The posterior distribution involves three hyperparameters $(\sigma, \eta, \lambda)$. Note that $\lambda$ can be either positive or negative to control the degree of deviation, positively or negatively. As described below, this Gaussian-type modeling leads to an analytic workflow that can benefit from less effort on the implementation.

We consider the MAP (maximum a posteriori) estimation of $\theta_w$ and a class of prediction functions $\hat{y}(x)$ that are characterized by two hyperparameters $\tau$ and $\rho$:

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^{n} \{ (y_i - f_{\theta_w}(u_i))^2 - \tau (f_s(x_i) - f_{\theta_w}(x_i))^2 \}, \quad \tau = \frac{\sigma}{\sigma + \eta} - \frac{\sigma}{\lambda} \in (-\infty, 1), \quad (6)$$

$$\hat{y}(x) = (1 - \rho)f_{\theta_w}(x) + \rho f_s(x), \quad \rho = \frac{\sigma}{\sigma + \eta} \in (0, 1). \quad (7)$$

In the training objective Eq. [5], the first term measures the goodness-of-fit with respect to $\mathcal{D}$, and the second term regularizes the training through the discrepancy between $f_{\theta_w}(x)$ and the pretrained $f_s(x)$. The prediction function Eq. [7] arises from $\hat{y}(x) = \arg\max_y p_y(y|x, \hat{\theta}_w)$, which corresponds to the mode of the plug-in predictive distribution Eq. [2]. Note that the original three hyperparameters are reduced to $\tau \in (-\infty, 1)$ and $\rho \in (0, 1)$. With varying $(\tau, \rho)$ and arbitrary model on $f_{\theta_w}(x)$ coupled with the learning algorithms, the resulting class can bridge various methods of TL as described later.

### 3 Implementation cost

By completing the square of Eq. [5] with respect to $f_{\theta_w}(x)$, the objective function can be rewritten as a residual sum of squares on a transformed output variable $z$:

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^{n} (z_i - f_{\theta_w}(x_i))^2, \quad z_i = \frac{y_i - \tau f_s(x_i)}{1 - \tau}. \quad (8)$$

Once the original output $y_i$ is simply converted to $z_i$ with a given $f_s(x)$ and $\tau$, the model can be trained by using a common library for regression that implements the minimization of the $\ell_2$-loss. Any regularization term, such as $\ell_1$- or $\ell_2$-regularization, can also be added. Therefore, the proposed method is implementable at essentially no cost. In the applications shown later, we utilized ridge regression, random forest regression, and neural networks as $f_{\theta_w}(x)$. We simply used the standard libraries of the R language (glmnet, ranger, and MXNet) without any customization or additional coding.

Furthermore, as no source data appear in the objective function, the model is learnable by using only training instances in a target domain as long as a source model is callable. This separately learnable property will be a great advantage in cases where training the source model from scratch is time-consuming, or source data are not disclosed.

### 4 Relations to existing methods

By adjusting $(\tau, \rho)$ coupled with the choice of $f_{\theta_w}(x)$, our method can represent the different types of TL as described below. Their relations in the class are visually overviewed in Figure [1].

**Regularization based on cross-domain similarity**

One of the most natural ideas for model refinement is to use the similarity to the pretrained $f_s(x)$ as a constraint condition. Many studies have been made so far to incorporate such cross-domain
with Eq. 7, we consider without using the source model. The second term represents the discrepancy between the density-ratio model and the source model. The density-ratio TL of [7] was designed to minimize the conditional Kullback-Leibler divergence.

Transfer learning based on the density-ratio estimation

Currently, the most powerful and widely used method of TL relies on deep neural networks. When neural networks are put on both \( f_{\theta_w}(x) \) and \( f_s(x) \) in the objective function Eq. 9, the pretrained source model. Such a newly trained model is directly used as the prediction function without using the source model.

Transfer learning based on neural networks

The density-ratio TL of [7] was designed to minimize the conditional Kullback-Leibler divergence \( \mathbb{E}_{x \sim q(x)}[\text{KL}(q(y|x) || p_t(y|x, \theta_w))] \) between the true density \( q(y|x) \) and the transferred model \( p_t(y|x, \theta_w) \) based on the density-ratio reweighting as in Eq. 2. As detailed in Supplementary Note A, if the transfer model is parameterized in the same way as Eq. 4, the learning objective derived from an empirical risk on the training set \( D \) takes the form

\[
\hat{\theta}_w = \arg \min_{\theta_w} \sum_{i=1}^{n} \left\{ (y_i - f_{\theta_w}(x_i))^2 + \frac{\sigma}{\lambda} (f_s(x_i) - f_{\theta_w}(x_i))^2 \right\}, \quad \hat{y}(x) = f_{\hat{\theta}_w}(x).
\] (9)

Our method can represent the MAP estimation with the objective function in Eq. 9 by restricting the hyperparameter \( \tau \) to be negative, i.e., \( \tau = -\sigma/\lambda < 0 \). The prediction function in Eq. 9 corresponds to \( \rho = 0 \) in our method. With a negative \( \tau \), the model \( f_{\theta_w}(x) \) is estimated to be closer to the pretrained source model. Such a newly trained model is directly used as the prediction function without using the source model.

Transfer learning based on neural networks

Currently, the most powerful and widely used method of TL relies on deep neural networks. When neural networks are put on both \( f_{\theta_w}(x) \) and \( f_s(x) \) in the objective function Eq. 9, the pretrained source model. Such a newly trained model is directly used as the prediction function without using the source model.

Transfer learning based on the density-ratio estimation

The density-ratio TL of [7] was designed to minimize the conditional Kullback-Leibler divergence \( \mathbb{E}_{x \sim q(x)}[\text{KL}(q(y|x) || p_t(y|x, \theta_w))] \) between the true density \( q(y|x) \) and the transferred model \( p_t(y|x, \theta_w) \) based on the density-ratio reweighting as in Eq. 2. As detailed in Supplementary Note A, if the transfer model is parameterized in the same way as Eq. 4, the learning objective derived from an empirical risk on the training set \( D \) takes the form

\[
\hat{\theta}_w = \arg \min_{\theta_w} \sum_{i=1}^{n} \left\{ (y_i - f_{\theta_w}(x_i))^2 - \rho (f_s(x_i) - f_{\theta_w}(x_i))^2 \right\}, \quad \hat{y}(x) = (1 - \rho) f_{\hat{\theta}_w}(x) + \rho f_s(x).
\] (10)

The second term represents the discrepancy between the density-ratio model and the source model in which the degree of regularization is controlled by \( \rho \in (0, 1) \). For the prediction function, as with Eq. 7, we consider \( \hat{y}(x) = (1 - \rho) f_{\hat{\theta}_w}(x) + \rho f_s(x) \) that corresponds to the plug-in estimator \( \arg \min_{\theta_w} p_t(y|x, \theta_w) \).

In terms of the proposed class of TL, the method in [7] can be considered as a specific choice of \( \tau = \rho \in (0, 1) \) (the blue line in Figure 1). It is noted that the objective function in Eq. 10 resembles Eq. 9 in the cross-domain similarity regularization. These two methods are regularized based on the...
discrepancy between $f_{\theta_w}(x)$ and $f_s(x)$, but their regularization mechanisms work in the opposite directions: the regularization parameter $\tau$ takes a positive value for the method in [7], while a negative value for cross-domain similarity regularization.

Learning without transfer
The proposed class contains two learning schemes without transfer. If the hyperparameters are selected to be $\tau = 0$ and $\rho = 0$ (the black dot in Figure [1]), the density-ratio model $\hat{f}_{\theta_w}(x)$ is estimated without using the source model, and the resulting prediction model becomes $\hat{y}(x) = \hat{f}_{\theta_w}(x)$. This corresponds to an ordinary regression procedure. When negative transfer occurs as the previous learning experience interferes with learning in the new task, cross-validated hyperparameters would be around $\tau = 0$ and $\rho = 0$. In addition, setting $\rho = 1$ (the red line in Figure [1]), the source model alone forms the prediction model as $\hat{y}(x) = f_s(x)$ regardless of any $f_{\theta_w}(s)$. By cross-validating the hyperparameters, the proposed framework will automatically determine when not to transfer without using separate pipelines.

5 Selection of hyperparameters and preference to bias and variance

As described above, our method can hybridize various mechanisms of TL by adjusting $\tau$ and $\rho$. The values of the hyperparameters are adjusted through cross-validation. Inevitably, the optimal combination of the hyperparameters will differ depending on between-task relationships and the choice for the target model.

Here, we show an expression of the mean squared error (MSE) based on the bias-variance decomposition. For simplicity, we restrict $f_{\theta_w}(x)$ to be in the set of all linear predictions taking the form of $\hat{f}_{\theta_w}(x) = x^TSz$. The $n \times n$ smoothing matrix $S$ depends on $n$ samples of $p$ input feature $\phi(x_i) \in \mathbb{R}^p$ ($i = 1, \ldots, n$) with a predefined basis set $\phi$, and $z$ is a vector of $n$ transformed outputs $z_i$ ($i = 1, \ldots, n$). For example, this class of prediction includes the kernel ridge regression.

We assume that $y$ follows $y = f_t(x) + \epsilon$ where $f_t(x)$ denotes the true model and the observation noise $\epsilon$ has mean zero and variance $\sigma_\epsilon^2$. For the prediction function $\hat{y}(x) = (1 - \rho)\hat{f}_{\theta_w}(x) + \rho f_s(x)$, MSE($\hat{y}(x)$) = $\mathbb{E}_y|x|\epsilon - \hat{y}(x)|^2$ can be expressed as:

$$\text{MSE}(\hat{y}(x)) = \left[ \frac{\rho - \tau}{1 - \tau} D(x) + \frac{1 - \rho}{1 - \tau} B_1(x) - \frac{\tau(1 - \rho)}{1 - \tau} B_2(x) \right]^2 + \left( \frac{1 - \rho}{1 - \tau} \right)^2 V(x) + \sigma_\epsilon^2, \quad (11)$$

where

$$D(x) = f_t(x) - f_s(x), \quad B_1(x) = f_t(x) - x^TSf_t, \quad B_2(x) = f_s(x) - x^TSf_s, \quad V(x) = \sigma_\epsilon^2 x^TS^T x. \quad (12)$$

The first term is the squared bias, which consists of three building blocks. $D(x)$ represents the discrepancy between $f_t(x)$ and $f_s(x)$. $B_1(x)$ is a bias of the linear estimator $x^TSf_t$ with respect to the true model $f_t(x)$, assuming that $n$ observations $f_t = (f_t(x_1), \ldots, f_t(x_n))^T$ for the unknown $f_t(x)$ are given. Likewise, $B_2(x)$ is the bias of $x^TSf_s$ with respect to $f_s(x)$. The second term corresponds to the variance of $\hat{y}(x)$. This is proportional to $V(x) = \sigma_\epsilon^2 x^TSS^T x$. The third term is the variance of the observation noise.

As discussed in Supplementary Note C, the relative magnitudes of $\mathbb{E}_x[D(x)^2]$, $\mathbb{E}_x[B_1(x)^2]$, $\mathbb{E}_x[B_2(x)^2]$, and $\mathbb{E}_x[V(x)]$ determine the optimal hyperparameters to the cross-domain similarity regularization, the density-ratio TL, and the learning without transfer. In the numerical experiments shown below, these quantities are approximated by sample averages and examined in relation to the hyperparameters selected by the cross-validation.

6 Results

6.1 Illustrative example

Some intrinsic properties of the proposed method are clarified by presenting numerical examples using artificial data. According to our experience, there is a tendency between the bias and variance magnitudes and the hyperparameters that minimize the MSE. This will be demonstrated.
We assumed the true functions on the source and target tasks to be linear as \( f_t(x) = x^T \theta_t \) and \( f_s(x) = x^T \theta_s \) where \( x \in \mathbb{R}^{300} \). The true parameters were generated as \( \theta_t = \alpha \theta_s + (1 - \alpha) \theta_w \) where \( \theta_s \sim \mathcal{N}(0, 1) \) and \( \theta_w \sim \mathcal{N}(0, 1) \). The output variable was assumed to follow \( y = f_t(x) + \epsilon \) where \( x \sim \mathcal{N}(0, 1) \) and \( \epsilon \sim \mathcal{N}(0, \sigma^2) \). With the given \( \theta_w \) and \( \theta_s \), we generated \( \{x_i, y_i\}_{i=1}^n \) with the sample size set to \( n = 50 \) by randomly generating \( x \) and \( \epsilon \). The discrepancy between the source and target models is controlled by the mixing rate \( \alpha \in [0, 1] \) for any given \( \theta_w \). In particular, if \( \alpha \) is set to zero, the source and target models are the same (\( \forall x: D(x) = 0 \) in Eq. \ref{eq:5}). The variance \( \sigma^2 \) of the observational noises affects the magnitude of the variance \( \mathbb{E}[\mathcal{V}] \) in the model estimation.

We used the linear ridge regression to estimate \( f_{\theta_w}(x) \) with the hyperparameter on the \( \ell_2 \)-regularization that was fixed at \( \lambda = 0.0001 \). The true source model was given to \( f_s(z) \). We then investigated the change of the MSE landscape as a function of the bias \( \alpha \) and the variance \( \sigma^2 \), which are summarized in Figure 2. For any given values of \( \tau \) and \( \rho \), the MSE was approximately evaluated by averaging the \( \ell_2 \)-loss over additionally generated 1,000 samples on \( (x, y) \) and rescaled to the range in \([0, 1]\). For \( \alpha = 0 \) where the source and target models are the same, the MSE became small in the region along \( \rho = 1 \) that corresponds to the use of the pretrained source model as the target model with no modification. As \( \alpha \) increased while keeping \( \sigma^2 \) at smaller values, the region where the MSE becomes small was concentrated around \( \tau = \rho \), indicating the dominance of the density-ratio TL. On the other hand, as both \( \alpha \) and \( \sigma^2 \) became larger, the region with \( \tau < 0 \) and \( \rho = 0 \) tended to be more favored. This region corresponds to the TL with the cross-domain similarity regularization. It was confirmed that the pattern of the MSE landscape varies continuously with respect to the bias and variance components.

In many other applications, we have often observed the same trend on the preference of \( \tau \) and \( \rho \) to the relative magnitude of the bias and variance. Another example of assuming nonlinear models for \( f_s(x) \) and \( f_t(x) \), and random forests for \( f_{\theta_w}(x) \) is shown in Supplementary Note B.

### 6.2 Real data applications

#### 6.2.1 Task, data and analysis procedure

The proposed method was applied to five real data analyses in materials science and robotics applications: (i) multiple properties of organic polymers and inorganic compounds \( [11] \), (ii) multiple properties of polymers \( [12] \) and low-molecular-weight compounds (monomers, unpublished data), (iii) properties of donor molecules in organic solar cells \( [13] \) obtained from experiments \( [14] \) and quantum chemical calculations \( [15] \), (iv) formation energies of various inorganic compounds and crystal polymorphisms of SiO2 and CdI2 \( [16] \), and (v) the feed-forward torques required to follow a desired trajectory at seven joints of a SARCOS anthropomorphic robot arm \( [17] \). The model transfers were conducted exhaustively between all task pairs within each application, which resulted in a total of 185 pairs of the source and target tasks with 9 different combinations of \( f_s(x) \) and \( f_{\theta_w}(x) \) (a total of 1,665 cases).
For each task pair, we used three machine learning algorithms, ridge regression using a linear model (LN), random forests (RF), and neural networks (NN) to estimate $f_s(x)$ and $f_{θ_w}(x)$. In the source task, the entire dataset was used to train $f_s(x)$ under default settings of software packages used in our study without adjusting hyperparameters on the model training. In all cases, 50 randomly selected samples were used to train $f_{θ_w}(x)$. The 5-fold cross-validation was performed on this dataset and the learning parameters were adjusted based on the evaluated values of the MSE. The resulting model was used to predict all the remaining data, and the MSE of each $(τ, ρ)$ was evaluated. Details of the datasets and analysis procedure are presented in Supplementary Note C.

6.2.2 Results

Throughout all the 1,665 cases, we investigated how the hyperparameters selected by the cross-validation are distributed (Figure 3). In many cases, the distribution of the selected hyperparameters was concentrated in the neighboring areas of the density-ratio TL $(τ = ρ)$ and the cross-domain similarity regularization $(τ < 0, ρ = 0)$. The density-ratio TL was selected for 609 cases (36.6%) and the cross-domain similarity regularization was selected for 176 cases (10.6%). In particular, there was a significant bias to the neighbors of $τ = ρ$.

The selected hyperparameters and the MSEs for the 1,665 cases are presented in Tables S1-S5 of the Supplementary Note. As an illustrative example, Table 1 shows the result of the TL from one source task (prediction of a dielectric property of small molecules) to five target tasks (prediction of two properties of small molecules and three properties of polymers). This result also indicates the
Figure 4: The MSE landscapes of the hyperparameter space for four different cases that exhibited the best transferability in different hyperparameter sets. Sample estimates on three bias-related quantities ($E_x[D^2]$, $E_x[B_1^2]$, and $E_x[B_2^2]$) and the mean variance ($E_x[V]$) are shown on each plot.

7 Concluding remarks

We proposed a new class of TL that is characterized by two hyperparameters to control the procedure of training and prediction. The class unifies two different types of existing methods that rely on the cross-domain similarity regularization and the density-ratio estimation. If we assume neural networks on the source and target models, the class represents the fine tuning of neural networks. In addition, the selection of specific hyperparameters offers the choice of ordinary regression without transfer or the direct use of a pretrained source model as the target. According to the choice of hyperparameters and models, we can derive various learning methods in which these related methods are hybridized.

The cross-domain similarity regularization and the density-ratio TL follow opposite learning objectives. In the former case, the target model is imposed as being closer to the source model. In the latter case, the difference between the source and target models is estimated to be far away from the source model. Most of the widely used techniques have adopted the former approach that leverages the proximity of the target model to the source model. Interestingly, the empirical results of the present study showed that, in many cases, the cross-domain similarity regularization rarely exhibited the best transferability, and often, the density-ratio estimation or its neighboring areas in the hyperparameter space significantly affects the prediction performance and in other cases it does not.
space showed better performances. Although the idea of the cross-domain similarity regularization is more widely adopted, our results indicate that we should further explore the direction based on the opposite idea, such as the density-ratio estimation.

This study focused on the regression setting. In addition, in the Bayesian framework, we assumed the specific type of the likelihood and prior distribution. The empirical risk derived from this assumption takes the sum of the squared loss. With this formulation, we could perform the model training simply by using an existing library for regression. This allows us to keep the implementation cost to practically zero. However, there are also limitations of using the squared loss. We should consider a wide range of loss functions and learning tasks. The treatment of more general loss functions and discriminant problems is one of the future issues.

Acknowledgments and Disclosure of Funding

Ryo Yoshida acknowledges the financial support received from a Grant-in-Aid for Scientific Research (A) 19H01132 from the Japan Society for the Promotion of Science (JSPS), JST CREST Grant Number JPMJCR1911, Japan, and JSPS KAKENHI Grant Number 19H05820. Stephen Wu acknowledges the financial support received from JSPS KAKENHI Grant Number JP18K18017.

References

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, Transfer Learning. Cambridge University Press, 2020.

[3] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.

[4] R. Jalem, K. Kanamori, I. Takeuchi, M. Nakayama, H. Yamasaki, and T. Saito, “Bayesian-driven first-principles calculations for accelerating exploration of fast ion conductors for rechargeable battery application,” Scientific Reports, vol. 8, no. 1, pp. 1–10, 2018.

[5] Z. Marx, M. T. Rosenstein, L. P. Kaelbling, and T. G. Dietterich, “Transfer learning with an ensemble of background tasks,” in NIPS Workshop on Inductive Transfer, 2005.

[6] R. Raina, A. Y. Ng, and D. Koller, “Constructing informative priors using transfer learning,” in Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 713–720.

[7] S. Liu and K. Fukumizu, “Estimating posterior ratio for classification: transfer learning from probabilistic perspective,” in Proceedings of the 2016 SIAM International Conference on Data Mining, 2016, pp. 747–755.

[8] M. Sugiyama, T. Suzuki, and T. Kanamori, Density Ratio Estimation in Machine Learning. Cambridge University Press, 2012.

[9] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network.” arXiv preprint arXiv:1503.02531, 2015.

[10] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, H. Demis, C. Claudia, K. Dharsan, and H. Raia, “Overcoming catastrophic forgetting in neural networks,” Proceedings of the National Academy of Sciences, vol. 114, no. 13, pp. 3521–3526, 2017.

[11] H. Yamada, C. Liu, S. Wu, Y. Koyama, S. Ju, J. Shiomi, J. Morikawa, and R. Yoshida, “Predicting materials properties with little data using shotgun transfer learning,” ACS Central Science, vol. 5, no. 10, pp. 1717–1730, 2019.

[12] C. Kim, A. Chandrasekaran, T. D. Huan, D. Das, and R. Ramprasad, “Polymer genome: A data-powered polymer informatics platform for property predictions,” The Journal of Physical Chemistry C, vol. 122, no. 31, pp. 17 575–17 585, 2018.
[13] A. Paul, D. Jha, R. Al-Bahrani, W.-k. Liao, A. Choudhary, and A. Agrawal, “Transfer learning using ensemble neural networks for organic solar cell screening,” in 2019 International Joint Conference on Neural Networks, 2019, pp. 1–8.

[14] S. A. Lopez, E. O. Pyzer-Knapp, G. N. Simm, T. Lutzow, K. Li, L. R. Seress, J. Hachmann, and A. Aspuru-Guzik, “The Harvard organic photovoltaic dataset,” Scientific Data, vol. 3, no. 1, pp. 1–7, 2016.

[15] E. O. Pyzer-Knapp, K. Li, and A. Aspuru-Guzik, “Learning from the Harvard clean energy project: the use of neural networks to accelerate materials discovery,” Advanced Functional Materials, vol. 25, no. 41, pp. 6495–6502, 2015.

[16] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, “The Materials Project: A materials genome approach to accelerating materials innovation,” APL Materials, vol. 1, no. 1, p. 011002, 2013.

[17] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning. MIT Press, 2006.
A General Class of Transfer Learning Regression
without Implementation Cost

A Transfer learning based on the density-ratio estimation

In [1], the density-ratio TL was designed to minimize the conditional Kullback-Leibler divergence
\[ \mathbb{E}_{q(x)}[\text{KL}(q(y|x)||p_t(y|x, \theta_w))] \]
between the true density \( q(y|x) \) and the target model \( p_t(y|x, \theta_w) \propto w(y, x|\theta_w)p_s(y|x, f_s) \) using the density-ratio reweighting as in Eq. 2 in the main text:
\[
\mathbb{E}_{q(x)}[\text{KL}(q(y|x)||p_t(y|x, \theta_w))] = -\int q(x) \int q(y|x) \log w(y, x|\theta_w)dydx + \int q(x) \log \int w(u, x|\theta_w)p_s(u|x, f_s)du + \text{const.} \tag{S13}
\]

The right-hand side represents the cross-entropy with respect to \( q(y|x) \) and \( p_t(y|x, \theta_w) \) in which the source density \( p_s(y|x, \theta_s) \) is omitted as a constant. The second term corresponds to the normalizing constant of the unnormalized target model in the right-hand side of \( p_t(y|x, \theta_w) \propto w(y, x|\theta_w)p_s(y|x, f_s) \).

While the original study was developed mainly on classification tasks, we focus on the regression task with the specific form of the target model shown in Eq. 4 in the main text. Substituting Eq. 4 into Eq. [S13], we obtain the normalizing constant as
\[
\int w(u, x|\theta_w)p_s(u|x, f_s)du \propto \int \exp \left( -\frac{(u - f_{\theta_w}(x))^2}{\sigma} - \frac{(u - f_s(x))^2}{\eta} \right) du \\
= \int \exp \left( -\left( \frac{1}{\sigma} + \frac{1}{\eta} \right) \left( u - \frac{\eta f_{\theta_w}(x) + \sigma f_s(x)}{\sigma + \eta} \right)^2 \\
- \frac{(f_{\theta_w}(x) - f_s(x))^2}{\sigma + \eta} \right) du \\
\propto \exp \left( -\frac{(f_{\theta_w}(x) - f_s(x))^2}{\sigma + \eta} \right). \tag{S14}
\]

With this expression, the empirical Kullback-Leibler divergence \( \mathbb{E}_{q(x)}[\text{KL}(\hat{q}(y|x)||p_t(y|x, \theta_w))] \) for a training set \( D \) can be written as
\[
\mathbb{E}_{q(x)}[\text{KL}(\hat{q}(y|x)||p_t(y|x, \theta_w))] = -\frac{1}{n} \sum_{i=1}^{n} \left[ \log w(y_i, x_i|\theta_w) - \log \int w(u, x_i|\theta_w)p_s(u|x_i, f_s)du \right] \\
\propto \frac{1}{n} \sum_{i=1}^{n} \left[ (y_i - f_{\theta_w}(x_i))^2 - \rho(f_s(x_i) - f_{\theta_w}(x_i))^2 \right] + \text{const.} \tag{S15}
\]

where \( \rho = \sigma/(\sigma + \eta) \in (0, 1) \) and all the terms irrelevant to \( \theta_w \) are omitted.

The parameter \( \theta_w \) in the density-ratio model should be estimated by maximizing Eq. [S15]. Furthermore, we define the prediction function to be \( \hat{y}(x) = (1 - \rho)f_{\theta_w}(x) + \rho f_s(x) \) that corresponds to the plug-in estimator \( \arg\max_y p_t(y|x, \hat{\theta}_w) \). In terms of our framework, the density-ratio TL of [1] can be considered as a specific choice of \( \tau = \rho \).
B Illustrative example

In Section 6.1 of the main text, we described the MSE landscape as a function of $\tau$ and $\rho$ in the case where a linear model was assumed for $f_{\theta_w}(x)$. In this section, we show the same analysis in cases where nonlinear models are assumed for $f_{\theta_w}(x)$, $f_t(x)$, and $f_s(x)$, respectively. To be specific, we considered three different cases as follows: (a) a random forest is given to $f_{\theta_w}(x)$ where the true models of $f_t(x)$ and $f_s(x)$ are assumed to be linear, (b) a linear model is given to $f_{\theta_w}(x)$ where the true models are assumed to be nonlinear, and (c) a random forest is given to $f_{\theta_w}(x)$ where the true models are assumed to be non-linear.

To generate artificial data with nonlinearity, we assumed single hidden layer neural networks for the source and target models as

$$f_s(x) = B_s \varphi(A_s x), \quad f_t(x) = B_t \varphi(A_t x), \quad \varphi(x) = \max \{0, x\}. \quad (S16)$$

The weight parameters were generated as $A_t = \alpha A_w + (1 - \alpha) A_s$, $B_t = \alpha B_w + (1 - \alpha) B_s$, where $A_s, A_w \in \mathbb{R}^{50 \times 300}$ and $B_s, B_w \in \mathbb{R}^{1 \times 50}$, and each element of $A_s, A_w, B_s, B_w$ was drawn from $\mathcal{N}(0,0.5)$ independently. As in Section 6.1, the output variable was assumed to follow $y = f_t(x) + \epsilon$ where $x \sim \mathcal{N}(0,1)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We generated 50 samples for the training of $f_{\theta_w}(x)$ and 1,000 samples for the evaluation of the MSE.

We used the linear ridge regression and the random forest regression to train $f_{\theta_w}(x)$ with the fixed hyperparameters $\lambda = 0.0001$, $n_{\text{tree}} = 200$ (the number of trees), and $n_{\text{variable}} = 100$ (the number of randomly selected variables at each split). Figure [S5] shows the changes of the MSE landscape for varying $\alpha$ and $\sigma_t$ for each case.

(a) $f_t$ and $f_s$ are linear, $f_{\theta_w}$ is non-linear When assuming the nonlinear model for $f_{\theta_w}(x)$, a similar trend was observed as in the case study shown in Section 6.1, regarding the relationship between hyperparameter preference and the magnitudes of the bias and variance components ($\alpha$ and $\sigma_t$). As $\alpha$ (i.e., $\mathbb{E}_x[D(x)^2]$) was increased while keeping $\sigma_t$ (i.e., $\mathbb{E}_x[V(x)]$) small, the regions with smaller MSEs were concentrated near $\tau = \rho$. On the other hand, as both $\alpha$ and $\sigma_t$ were increased, the regions with $\tau < 0$ and $\rho = 0$ became preferable.

(b) $f_t$ and $f_s$ are non-linear, $f_{\theta_w}$ is linear In this case, the same argument as Section 5 holds because the analysis shown in Section 5 does not place any specific assumption on the mathematical forms of $f_t(x)$ and $f_s(x)$. However, in the lower left figure of Figure [S5] (the case where $\alpha$ is large and $\sigma_t$ is small), the best hyperparameters are located slightly off the diagonal. This would be due to the fact that the linear model $f_{\theta_w}(x)$ could not capture the nonlinearity of $f_t(x)$ and $f_s(x)$, thus $\mathbb{E}_x[B_2(x)^2]$ and $\mathbb{E}_x[B_2(x)^2]$ did not get smaller. Statistical mechanisms on the relationships between the relative magnitude of these two factors to $\mathbb{E}_x[D(x)^2]$ and the preference of hyperparameters are discussed in Section [S5].

(c) $f_t$, $f_s$, and $f_{\theta_w}$ are non-linear As in (a), the pattern in the change of the MSE with respect to $\alpha$ and $\sigma_t$ was similar to the linear case. Assuming the nonlinear model for $f_{\theta_w}(x)$, we could reduce $\mathbb{E}_x[B_2(x)^2]$ and $\mathbb{E}_x[B_2(x)^2]$ more than in the case of assuming the linear model for $f_{\theta_w}(x)$. As a result, the region near the density-ratio TL became more favorable when $\alpha$ was larger and $\sigma_t$ was smaller.
Figure S5: Heatmap display of the MSE landscape on the hyperparameter space ($\tau, \rho$) in the three different settings where the different models were assumed for $f_t(x)$, $f_s(x)$, and $f_{\theta_w}(x)$, respectively. The black dot denotes the lowest MSE.
C Real data applications

C.1 Data and tasks

We performed the proposed method on the five applications using real data as detailed below. The model transfers were conducted exhaustively between all task pairs within each application, which resulted in the 185 pairs of the source and target tasks. For each task pair, we considered the use of three different models (LN, RF, NN) for $f_{\theta_w}(x)$ and $f_s(x)$, which resulted in the 1,665 cases.

**Polymers and inorganic compounds** The task is to make the prediction of five properties (band gap, dielectric constant, refractive index, density, and volume) for inorganic compounds and six properties (band gap, dielectric constant, refractive index, density, volume, and atomization energy) for polymers. The number of the pairs for the source and target tasks to be transferred is $110 = 11 \times 10$. The overall datasets represent the structure-property relationships for 1,056 inorganic compounds and 1,070 polymers, respectively. See [2] for more details on the datasets. For all the materials, any structural information was ignored, only the compositional features were encoded into the 290-dimensional input descriptors, using XenonPy, an open-source platform of materials informatics for Python [7].

**Polymers and small molecules** The task is to predict three properties (band gap, dielectric constant, and refractive index) for polymers and three properties (HOMO-LUMO gap, dielectric constant, and refractive index) for small organic molecules. The number of the paired tasks is $30 = 6 \times 5$. The polymeric data consist of 854 polymers. By performing the quantum chemistry calculation based on density functional theory using the Gaussian09 suite of program codes [8], we produced a dataset on the three properties of 854 small organic molecules that correspond the constitutional repeat units of the 854 polymers. In the DFT calculation, the molecular geometries were optimized at the B3LYP/6-31+G(d) level of theory. The chemical structure of each monomer was encoded into a descriptor vector of 1,905 binary digits using two molecular fingerprinting algorithms referred to as the PubChem and circular fingerprints that are implemented in the rcdk package on R [9].

**CEP and HOPV** The task is to predict the highest occupied molecular orbital (HOMO) energy for donor molecules in an organic solar cell devise. We used two datasets on the HOMO energy levels of 2,322,649 and 351 molecules. The former dataset was obtained from high-throughput quantum chemistry calculations conducted by Harvard clean energy project (CEP) [4] and the latter is a collation of experimental photovoltaic data from the literature, referred to as the Harvard Organic Photovoltaic Dataset (HOPV15) [3]. We used the same fingerprints of the second task to represent input chemical structures.

**Formation energy of SiO$_2$ and all other compounds** We used a dataset in Materials Project [5] that records DFT formation energies of 69,641 inorganic compounds. The input crystal structures were translated by the 441-dimensional descriptors that were obtained by concatenating the 290-dimensional compositional descriptors and the 151-dimensional radial distribution function descriptors in XenonPy. We first derive a pretrained source model using 80% of the 69,358 training instances after removing 283 instances corresponding to SiO$_2$. Such a global model originated from the large dataset was transferred to a localized target model on SiO$_2$ using the remaining small dataset.

**SARCOS robot arm** The task is to predict the feed-forward torques required to follow a desired trajectory at seven joints of a SARCOS anthropomorphic robot arm [6]. The number of the paired tasks is 35. The dataset contains a total of 44,484 and 4,449 instances for training and testing. The 21 input features describe the position, velocity, and acceleration at the seven joints.

C.2 Results

For the 1,665 cases, the selected hyperparameters and the resulting RMSEs on the test sets are presented in Tables S1-S5.
C.3 Remarks: preference of hyperparameters

In the real data applications, we investigated the relationship between the selected hyperparameters and the bias and variance inherent in the data for the 555 (= 3 × 185) cases, out of the total 1,665 cases, where the linear model was assumed for the density-ratio model. If we assume the linearity, as described in the main text, the MSE can be expressed as Eq. 11 in the main text. Here, we focused on the relative magnitudes of $E_x[D(x)^2]$ and $E_x[V(x)]$. The expected value of $E_x[D(x)]^2$ was approximated by the mean of 500 samples randomly selected from the test data. For $E_x[V(x)]$, the variance of the linear predictor function was calculated using 100 bootstrap sets extracted from the training data. We divided the 555 cases into 16 (= 4 × 4) groups according to the quartiles of $E_x[D(x)^2]$ and $E_x[V(x)^2]$ respectively. The thresholds for each interval and the distribution of the selected $\tau$ and $\rho$ for each group are shown in Figure S6. A striking trend was observed, in which the hyperparameters significantly concentrated in the domain of density-ratio TL as $E_x[D(x)^2]$ increased relative to $E_x[V(x)]$ ($E_x[D(x)^2]/E_x[V(x)] \to \infty$). On the other hand, as $E_x[V(x)]$ increased, some of the selected hyperparameters appeared in the domain of the cross-domain similarity regularization. However, many hyperparameters were still distributed in the region of the density-ratio TL. Compared to the case of $E_x[D(x)^2]/E_x[V(x)] \to \infty$, the trend was unclear.

Let $D = D(x)$, $B_1 = B_1(x)$, $B_2 = B_2(x)$, and $V = V(x)$, respectively. Consider the expectation of the MSE in Eq. 11 with respect to the marginal distribution of $x$: $E_{x \sim q(x)}[MSE(\hat{g}(x))]$. Because the expected MSE is quadratic with respect to $\rho$ for any $\tau$, the minimum under the inequality constraint $0 \leq \rho \leq 1$ is achieved by

$$
\rho(\tau) = \begin{cases} 
0 & \rho_*(\tau) \leq 0 \\
0 < \rho_*(\tau) < 1 & 1 \\
\rho_*(\tau) \geq 1
\end{cases} \quad (S17)
$$

where $\rho_*(\tau)$ denotes the solution for the unconstrained minimization. Taking the derivative of the expected MSE with respect to $\rho$, we have an equation as

$$
\frac{1}{(1-\tau)^2}E[((\rho - \tau)D + (1-\rho)B_1 - \tau(1-\rho)B_2)(D - B_1 + \tau B_2)] - \frac{1 - \rho}{(1-\tau)^2}E[V] = 0. (S18)
$$

Assuming that $\tau \neq 1$, this leads to an expression for the unconstrained solution as

$$
\rho_*(\tau) = \frac{E[(\tau D - B_1 + \tau B_2)(D - B_1 + \tau B_2)] + E[V]}{E[D - B_1 + \tau B_2]^2 + E[V]} . \quad (S19)
$$

Likewise, taking the derivative of the expected MSE with respect to $\tau$, we have

$$
\frac{1 - \rho}{(1-\tau)^3}E[((\rho - \tau)D + (1-\rho)B_1 - \tau(1-\rho)B_2)(D - B_1 + 2B_2)] - \frac{(1-\rho)^2}{(1-\tau)^2}E[V] = 0. \quad (S20)
$$
Combining Eq. S18 and Eq. S20 where $\tau \neq 1$ and $\rho \neq 1$, we obtain an equation

$$(1 - \tau)E[\tau(D + (1 - \rho)B_2)B_2 - (1 - \rho)B_1B_2 + \rho DB_2] = 0,$$

(S21)

then yielding an expression for the solution

$$\tau(\rho) = \frac{(1 - \rho)E[B_1B_2] + \rho E[DB_2]}{(1 - \rho)E[B_3] + E[DB_2]}.$$  

(S22)

According to the two expressions in Eq. S19 and Eq. S22, we can investigate the preference in the hyperparameter selection in regard to the bias and variance components in the data generation process.

Consider a case where the source and target models are significantly different by taking the limit $E[D^2] \to \infty$. For the expectation of $E[DX]$ for the product of $D$ and any $X$, it holds that $E[DX]/E[D^2] \to 0$ as $E[D^2] \to \infty$. This can be seen by considering the Cauchy-Schwarz inequality:

$$-E[D^2]^{\frac{1}{2}}E[X^2]^{\frac{1}{2}} \leq E[DX] \leq E[D^2]^{\frac{1}{2}}E[X^2]^{\frac{1}{2}}$$

$$\Leftrightarrow -\frac{E[X^2]}{E[D^2]} \leq \frac{E[DX]}{E[D^2]} \leq \frac{E[X^2]}{E[D^2]}.$$  

(S23)

In the second line, the upper- and the lower-bounds go to zero as $E[D^2] \to \infty$. Thus, in Eq. S19 all terms except those having $E[D^2]$, which appear in its numerator and denominator, approach asymptotically to zero, which results in

$$\rho_*(\tau) \to \frac{\tau E[D^2]}{E[D^2]} = \tau \text{ as } E[D^2] \to \infty.$$  

(S24)

Furthermore, noting that $E[DX] = O(E[D^2]^{\frac{1}{2}})$, it can been seen that $\tau(\rho)$ in Eq. S22 approaches asymptotically $\rho$:

$$\tau(\rho) \to \frac{\rho E[DB_2]}{E[DB_2]} = \rho \text{ as } E[D^2] \to \infty.$$  

(S25)

Therefore, when $E[D^2]$ dominates the other three quantities, the density-ratio TL ($\tau = \rho$) is preferred. This fact accounts for the experimental observations presented above.

On the other hand, if the source and target models are completely the same ($E[D^2] = 0$), it holds that $\rho_*(\tau) = 1$. Alternatively, if $E[V] \to \infty$, $\rho_*(\tau) = 1$. The direct use of the source model as a prediction function tends to be optimal as the source and target tasks get closer or the variance $E[V]$ becomes larger. As for the cross-domain similarity regularization, statistical mechanisms have not yet been clear, either theoretically or experimentally, on what conditions it is preferred.
| Source task | Target task | $f_a(x)$ | $f_{ac}(x)$ | Hyperparameter |
|-------------|-------------|---------|------------|----------------|
| **Inorganic** | - Density | LN 0.8192 | LN 0.8076 | LN 0.1649 |
|              | RF 1.0142 | RF 0.9452 | RF 0.1654 | RF 0.1604 |
|              | NN 1.0276 | NN 1.6049 | NN 0.2195 | NN 0.1701 |
| **Inorganic** | - Band gap | LN 0.0031 | LN 0.1600 | LN 0.8952 |
|              | RF 0.3042 | RF 0.3456 | RF 1.976 | RF 3.0092 |
|              | NN 0.1649 | NN 0.9335 | NN 0.1273 | NN 0.2195 |
| **Inorganic** | - Volume | LN 0.7324 | LN 0.1592 | LN 19.722 |
|              | RF 0.4507 | RF 0.1872 | RF 0.9392 | RF 3.0319 |
|              | NN 0.1519 | NN 0.1901 | NN 0.8982 | NN 0.1559 |
| **Organic**  | - Band gap | LN 0.9027 | LN 0.1713 | LN 0.7324 |
|              | RF 0.1092 | RF 0.1129 | RF 0.1092 | RF 0.1645 |
|              | NN 0.8882 | NN 0.1318 | NN 0.8882 | NN 0.2075 |
| **Organic**  | - Density | LN 0.1559 | LN 0.1569 | LN 0.9027 |
|              | RF 0.1359 | RF 0.1075 | RF 0.1359 | RF 0.1359 |
|              | NN 0.9627 | NN 0.1359 | NN 0.9627 | NN 0.9627 |
| **Organic**  | - Volume | LN 0.1562 | LN 0.1562 | LN 0.4870 |
|              | RF 0.1759 | RF 0.1617 | RF 1.0549 | RF 0.1759 |
|              | NN 0.1649 | NN 0.1728 | NN 0.1649 | NN 0.1649 |
| **Inorganic** | - Band gap | LN 0.1361 | LN 0.1361 | LN 0.4870 |
|              | RF 1.3270 | RF 1.2897 | RF 1.4060 | RF 1.4060 |
|              | NN 1.2149 | NN 1.2870 | NN 1.4934 | NN 1.4934 |
| **Inorganic** | - Volume | LN 0.1557 | LN 0.1557 | LN 0.4870 |
|              | RF 0.2311 | RF 0.2051 | RF 0.1697 | RF 0.1697 |
|              | NN 0.1243 | NN 0.2032 | NN 0.2882 | NN 0.2882 |
| **Organic**  | - Band gap | LN 0.8232 | LN 0.8232 | LN 0.4870 |
|              | RF 0.3012 | RF 0.8070 | RF 0.8992 | RF 0.8992 |
|              | NN 0.8500 | NN 0.8147 | NN 0.8755 | NN 0.8755 |
| **Organic**  | - Density | LN 0.0724 | LN 0.0724 | LN 0.4870 |
|              | RF 0.0789 | RF 0.1087 | RF 0.0978 | RF 0.0978 |
|              | NN 0.0789 | NN 0.0900 | NN 0.0974 | NN 0.0974 |
| **Organic**  | - Band gap | LN 0.1520 | LN 0.1520 | LN 0.4870 |
|              | RF 0.1581 | RF 0.1429 | RF 0.1697 | RF 0.1697 |
|              | NN 0.1584 | NN 0.1459 | NN 0.1759 | NN 0.1759 |
| **Organic**  | - Volume | LN 35.9712 | LN 35.9712 | LN 48.7003 |
|              | RF 40.9502 | RF 35.9712 | RF 40.9502 | RF 35.9712 |
|              | NN 26.7953 | NN 31.2916 | NN 26.7953 | NN 31.2916 |
| Source task | Target task | $f_s(x)$ | $f_c(x)$ | Hyperparameter |
|------------|-------------|----------|----------|----------------|
|            |              | LN       | RF       | NN             |
| Inorganic  | - Band gap  | LN 1.3808 | RF 1.2466 | NN 1.3438     |
|            |              | (0.8, 0.8) | (0.3, 0.5) | (0.5, 0.6)     |
|            | - Density   | LN 0.8297 | RF 0.9654 | NN 0.9368     |
|            |              | (0.4, 0.4) | (0.5, 0.6) | (0.6, 0.6)     |
| Inorganic  | - Refractive index | LN 1.1117 | RF 0.9892 | NN 0.9051     |
|            |              | (0.7, 0.6) | (0.5, 0.4) | (0.6, 0.6)     |
| Inorganic  | - Volume    | LN 70.6305 | RF 40.0470 | NN 37.1534    |
|            |              | (0.4, 0.4) | (0.2, 0.1) | (0.3, 0.6)     |
| Organic    | - Atomization energy | LN 0.1379 | RF 0.1359 | NN 0.1472     |
|            |              | (0.3, 0.3) | (0.1, 0.1) | (0.1, 0.1)     |
| Organic    | - Band gap  | LN 0.8597 | RF 0.8949 | NN 0.9588     |
|            |              | (0.4, 0.5) | (0.5, 0.4) | (0.6, 0.2)     |
| Organic    | - Density   | LN 0.0914 | RF 0.0895 | NN 0.0889     |
|            |              | (0.2, 0.1) | (0.1, 0.1) | (0.3, 0.3)     |
| Organic    | - Refractive index | LN 0.1465 | RF 0.1445 | NN 0.1500     |
|            |              | (0.7, 0.7) | (0.7, 0.7) | (0.3, 0.3)     |
| Inorganic  | - Band gap  | LN 1.2554 | RF 1.1459 | NN 1.2255     |
|            |              | (0.4, 0.4) | (0.2, 0.5) | (0.3, 0.3)     |
| Inorganic  | - Density   | LN 0.8301 | RF 0.7596 | NN 0.8157     |
|            |              | (0.4, 0.3) | (0.3, 0.3) | (0.3, 0.3)     |
| Inorganic  | - Refractive index | LN 19.4964 | RF 18.7022 | NN 17.9844    |
|            |              | (0.5, 0.5) | (0.5, 0.9) | (0.6, 0.6)     |
| Inorganic  | - Volume    | LN 36.7648 | RF 37.7601 | NN 40.9691    |
|            |              | (0.1, 0.1) | (0.1, 0.1) | (0.3, 0.3)     |
| Organic    | - Atomization energy | LN 0.1209 | RF 0.1325 | NN 0.1319     |
|            |              | (0.0, 0.0) | (0.6, 0.5) | (0.7, 0.7)     |
| Organic    | - Band gap  | LN 0.7972 | RF 0.8060 | NN 0.8810     |
|            |              | (0.2, 0.3) | (0.2, 0.2) | (0.5, 0.5)     |
| Organic    | - Density   | LN 0.0778 | RF 0.0762 | NN 0.0759     |
|            |              | (0.2, 0.2) | (0.3, 0.2) | (0.3, 0.2)     |
| Organic    | - Refractive index | LN 3.8983 | RF 4.0992 | NN 3.9125     |
|            |              | (0.9, 0.9) | (0.5, 0.9) | (0.5, 0.6)     |
| Organic    | - Volume    | LN 43.9842 | RF 50.4987 | NN 52.4927    |
|            |              | (0.5, 0.5) | (0.5, 0.5) | (0.1, 0.1)     |

Note: The table above represents a portion of the document, focusing on the properties of inorganic and organic materials. Each row corresponds to different properties such as Refractive index, Density, and Volume, with values for both inorganic and organic materials.
| Source task | Target task | $f_{\text{L}}(x)$ | $f_{\text{RF}}(x)$ | $f_{\text{NN}}(x)$ | Hyperparameter |
|-------------|-------------|------------------|------------------|------------------|----------------|
| **Inorganic** | - Band gap | LN 1.3237 1.2427 1.3580 | RF 1.3116 1.2382 1.3712 | NN 1.4462 1.3972 1.7492 | (0.1, 0.2) (0.0) (0.2) |
|             | - Density   | LN 0.8091 0.9130 1.0477 | RF 2.2622 0.9960 0.8985 | NN 2.3083 0.8758 1.1343 | (0.3, 0.3) (0.5, 0.4) (0.4, 0.3) |
| **Inorganic** | - Dielectric constant | LN 18.6567 16.1081 18.5181 | RF 18.8976 16.5782 16.3408 | NN 17.7790 15.5996 15.7704 | (0.0) (0.9, 0.9) (0.5, 0.6) |
|             | - Volume    | LN 1.0419 1.0268 1.0828 | RF 1.0460 1.0360 1.0880 | NN 1.0812 1.0476 1.1245 | (0.0) (0.1, 0.0) (0.2, 0.0) |
| **Organic**  | - Band gap  | LN 0.8450 0.8453 1.0204 | RF 0.9064 0.8897 0.9196 | NN 0.8950 0.9146 1.1818 | (0.3, 0.3) (0.4, 0.4) (0.3, 0.3) |
|             | - Density   | LN 0.0836 0.1116 0.1478 | RF 0.0807 0.1059 0.1144 | NN 0.0809 0.1159 0.1945 | (0.0) (0.2, 0.1) (0.1, 0.1) |
| **Organic**  | - Dielectric constant | LN 2.9277 2.8334 2.8161 | RF 2.9977 2.8376 2.8510 | NN 2.9591 2.7632 3.1282 | (-0.7, 0.3) (-0.1, 0.1) (0.3, 0.3) |
|             | - Volume    | LN 0.1546 0.1505 0.1567 | RF 0.1575 0.1512 0.1569 | NN 0.1580 0.1530 0.1716 | (-0.1, 0.0) (0.4, 0.4) (0.1, 0.1) |
| **Organic**  | - Atomization energy | LN 8.7446 1.2550 1.2405 | RF 8.6841 1.2415 1.2768 | NN 2.4251 1.2506 1.5938 | (-0.0, 0.0) (0.0, 0.0) (0.1, 0.1) |
|             | - Atomization energy | LN 10.8871 1.0194 1.1242 | RF 1.1871 0.9940 1.2816 | NN 1.1301 1.0323 1.6520 | (0.0) (0.1, 0.1) (0.4, 0.4) |
|             | - Volume    | LN 18.7991 17.0726 18.1379 | RF 18.1313 16.7010 17.7502 | NN 18.1296 16.787 18.1288 | (-0.1, 0.0) (0.4, 0.4) (0.1, 0.1) |
| **Organic**  | - Atomization energy | LN 3.1219 8.0088 0.7889 | RF 4.9126 0.7449 0.7135 | NN 5.3636 0.7559 0.8211 | (0.1, 0.1) (0.1, 0.1) (0.5, 0.5) |
|             | - Volume    | LN 55.3176 70.9819 50.3896 | RF 63.6384 57.8758 49.6017 | NN 61.6784 55.2097 100.3145 | (0.0) (0.3, 0.3) (0.5, 0.5) |
| **Organic**  | - Atomization energy | LN 0.9024 0.8380 1.0104 | RF 0.9084 0.8207 0.8893 | NN 0.8928 0.8362 1.2863 | (-0.4, 0.0) (0.4, 0.0) (0.1, 0.1) |
|             | - Atomization energy | LN 0.1907 0.1420 0.1830 | RF 0.1820 0.1545 0.1496 | NN 0.1338 0.1683 0.2405 | (0.0) (0.5, 0.5) (0.4, 0.4) |
| **Organic**  | - Dielectric constant | LN 2.2701 2.1462 2.2037 | RF 2.2590 2.1377 2.2022 | NN 2.6567 2.1436 2.6324 | (0.0) (0.1, 0.2) (0.1, 0.1) |
|             | - Volume    | LN 0.1358 0.1260 0.1558 | RF 0.1526 0.1388 0.1906 | NN 0.1445 0.1270 0.1901 | (0.0) (0.4, 0.4) (0.3, 0.3) |
| **Organic**  | - Atomization energy | LN 71.7697 52.8113 121.1668 | RF 62.1807 42.9842 73.9654 | NN 63.2865 41.9871 156.4083 | (0.3, 0.3) (0.2, 0.2) (0.0, 0.0) |

$\theta_w$
| Source task | Target task | $f_s(x)$ | $f_{sp}(x)$ | Hyperparameter |
|-------------|-------------|----------|-------------|----------------|
| Inorganic  | - Band gap  | LN  | 1.5506  | RF  | 1.2365  | NN  | 1.4975  | (0.0) | (0.0) | (-0.2, 0.1) |
|            |             | LN  | 1.4018  | RF  | 1.2530  | NN  | 1.3354  | (0.1, 0.6) | (0.0) | (0.9, 0.9) | (-0.3, 0.0) |
| Inorganic  | - Density   | LN  | 1.0126  | RF  | 1.0133  | NN  | 1.5513  | (-0.2, 0) | (0.2, 0.2) | (0.1, 0.3) |
| Inorganic  | - Dielectric constant | LN  | 13.6250  | RF  | 12.6773  | NN  | 16.4276  | (-0.6, 0.1) | (-1.0) | (-1.4, 0.1) |
| Organic    | - Band gap  | LN  | 2.6326  | RF  | 0.9505  | NN  | 0.9413  | (0.0, 0) | (0.1, 0) | (0.0, 0.1) |
| Organic    | - Volume    | LN  | 52.1059  | RF  | 49.9397  | NN  | 68.7418  | (0.0) | (0.1, 0) | (0.2, 0.2) |
| Organic    | - Atomization energy | LN  | 0.1398  | RF  | 0.1892  | NN  | 0.2064  | (0.1, 0.1) | (0.3, 0.3) | (0.5, 0.5) |
| Organic    | - Density   | LN  | 0.0911  | RF  | 0.0905  | NN  | 0.0937  | (0.5, 0.5) | (0.1, 0.1) | (0.0) |
| Organic    | - Dielectric constant | LN  | 3.0899  | RF  | 3.0213  | NN  | 3.0405  | (0.1, 0) | (0.2, 0.2) | (-0.2, 0) |
| Organic    | - Refractive index | LN  | 0.1486  | RF  | 0.1219  | NN  | 0.1544  | (-0.1, 0) | (0.3, 0.4) | (0.0, 0.1) |
| Organic    | - Volume    | LN  | 55.6166  | RF  | 33.6603  | NN  | 57.9079  | (0.0) | (0.1, 0.1) | (-0.2, 0) |
| Organic    | - Band gap  | LN  | 1.3926  | RF  | 1.2412  | NN  | 1.3545  | (0.2, 0.2) | (0.1, 0) | (0.0) |
| Organic    | - Density   | LN  | 0.8424  | RF  | 0.0139  | NN  | 0.9342  | (0.3, 0.3) | (0.0, 0.3) | (0.2, 0.2) |
| Organic    | - Dielectric constant | LN  | 14.9834  | RF  | 14.8942  | NN  | 14.6026  | (0.2, 0.1) | (0.3, 0.3) | (0.0, 0.4) |
| Organic    | - Refractive index | LN  | 14.5941  | RF  | 14.8685  | NN  | 15.3146  | (0.0) | (0.1, 0) | (0.8, 0.8) |
| Organic    | - Volume    | LN  | 14.9711  | RF  | 14.9001  | NN  | 14.6675  | (0.2, 0.2) | (0.3, 0.3) | (0.0) |
| Organic    | - Band gap  | LN  | 24.8515  | RF  | 1.1719  | NN  | 1.1069  | (0.6, 0.6) | (0.2, 0.1) | (0.2, 0.1) |
| Organic    | - Density   | LN  | 1.3566  | RF  | 1.1051  | NN  | 1.2385  | (0.0) | (0.3, 0.3) | (0.3, 0.2) |
| Organic    | - Dielectric constant | LN  | 52.8522  | RF  | 48.5924  | NN  | 56.4913  | (0.0) | (0.2, 0.2) | (0.0) |
| Organic    | - Volume    | LN  | 50.0095  | RF  | 52.3725  | NN  | 44.0897  | (0.1, 0.1) | (0.1, 0.1) | (0.6, 0.6) |
| Organic    | - Atomization energy | LN  | 0.1477  | RF  | 0.1887  | NN  | 0.1667  | (0.0) | (0.6, 0.6) | (0.6, 0.6) |
| Organic    | - Density   | LN  | 1.1492  | RF  | 0.8384  | NN  | 0.8873  | (0.3, 0.3) | (0.0) | (0.1, 0.1) |
| Organic    | - Dielectric constant | LN  | 2.9771  | RF  | 2.8895  | NN  | 3.0364  | (-0.1, 0) | (0.1, 0) | (0.4, 0.3) |
| Organic    | - Refractive index | LN  | 3.1880  | RF  | 2.8822  | NN  | 3.0437  | (-0.1, 0) | (0.1, 0) | (0.2, 0.1) |
| Organic    | - Volume    | LN  | 47.6037  | RF  | 40.0016  | NN  | 93.6506  | (0.3, 0.3) | (0.2, 0.1) | (0.3, 0.2) |
| Source task | Target task | $f_s(x)$ | $f_{DN}(x)$ | Hyperparameter |
|-------------|-------------|----------|-------------|----------------|
| Inorganic | Band gap | LN 1.2983 | RF 1.3186 | NN 1.3605 | (-0.5, 0.2) | (0.4, 0.3) | (-0.1, 0.1) |
| Inorganic | Density | LN 0.8087 | RF 2.3837 | NN 0.9526 | (0.9, 0.9) | (0.7, 0.7) | |
| Inorganic | Dielectric constant | LN 18.2690 | RF 15.5712 | NN 15.9667 | (-1.7, 0) | (0.3, 0.4) | |
| Inorganic | Refractive index | LN 1.1526 | RF 1.0292 | NN 0.9963 | (0.1, 0.2) | (0.5, 0.3) | (0.0) |
| Organic | Dielectric constant | LN 51.6332 | RF 144.0797 | NN 50.3164 | (-1.0, 0) | (0.0) | (0.0) |
| Organic | Volume | LN 0.1183 | RF 0.7828 | NN 0.7734 | (0.1, 0.0) | (0.4, 0.4) | |
| Organic | Atomization energy | LN 0.7845 | RF 0.7928 | NN 0.7734 | (0.3, 0.4) | (0.3, 0.3) | |
| Organic | Band gap | LN 0.1085 | RF 0.0703 | NN 0.0697 | (0.4, 0.3) | (0.4, 0.3) | |
| Organic | Density | LN 0.1451 | RF 0.1382 | NN 0.1322 | (0.1, 0.0) | (0.4, 0.4) | |
| Organic | Refractive index | LN 1.6684 | RF 1.3032 | NN 1.2803 | (0.3, 0.3) | (0.3, 0.3) | |
| Inorganic | Band gap | LN 1.2117 | RF 1.4225 | NN 1.2029 | (0.0, 0.0) | (0.0, 0.0) | |
| Inorganic | Density | LN 22.8981 | RF 22.1033 | NN 21.9499 | (0.1, 0.0) | (0.2, 0.2) | |
| Inorganic | Dielectric constant | LN 1.7616 | RF 1.5027 | NN 1.0733 | (0.6, 0.6) | (0.3, 0.3) | |
| Inorganic | Refractive index | LN 52.3688 | RF 52.1246 | NN 84.2979 | (0.0, 0.0) | (0.0, 0.0) | |
| Organic | Atomization energy | LN 0.1647 | RF 0.1608 | NN 0.1851 | (0.2, 0.2) | (0.3, 0.3) | |
| Organic | Band gap | LN 0.7513 | RF 0.7413 | NN 0.7353 | (0.4, 0.4) | (0.4, 0.4) | |
| Organic | Density | LN 0.1054 | RF 0.0744 | NN 0.0764 | (0.0, 0.0) | (0.0, 0.0) | |
| Organic | Dielectric constant | LN 2.7461 | RF 2.9687 | NN 3.4959 | (-0.6, 0.0) | (0.2, 0.2) | |
| Organic | Volume | LN 66.1096 | RF 65.6943 | NN 60.4089 | (0.0, 0.0) | (0.0, 0.0) | |

Hyperparameter values are given as ranges in parentheses.
## Table S3: Transfer between monomeric and polymeric properties

| Source task | Target task | \( f_s(x) \) | \( f_o(x) \) | Hyperparameter |
|-------------|-------------|----------------|----------------|----------------|
| Monomer - Band gap | | | | |
| LN | 0.8292 | 0.7435 | 0.8823 | (0.3, 0.6) | (0.1, 0.3) |
| RF | 0.8326 | 0.7143 | 0.8256 | (0.5, 0.3) | (0.3, 0.8) |
| NN | 0.8250 | 0.7372 | 0.7644 | (0.2, 0.3) | (0.4, 0.4) |
| Monomer - Refractive index | | | | |
| LN | 0.0436 | 0.0424 | 0.0439 | (0.8, 0.9) | (0.8, 0.9) |
| RF | 0.0463 | 0.0415 | 0.0415 | (0.9, 0.9) | (0.4, 0.8) |
| NN | 0.0365 | 0.0355 | 0.0350 | (0.3, 0.5) | (0.4, 0.4) |
| Polymer - Band gap | | | | |
| LN | 2.3829 | 2.4687 | 2.4327 | (0.8, 0.9) | (0.6, 0.6) |
| RF | 2.3103 | 2.3687 | 2.3716 | (0.3, 0.4) | (0.8, 0.8) |
| NN | 2.3489 | 2.3736 | 2.5189 | (0.1, 0.1) | (0.1, 0.1) |
| Monomer - Dielectric constant | | | | |
| LN | 0.0631 | 0.0535 | 0.0676 | (0.5, 0.6) | (0.6, 0.7) |
| RF | 0.0598 | 0.0578 | 0.0608 | (0.2, 0.3) | (0.4, 0.4) |
| NN | 0.0614 | 0.0578 | 0.0761 | (0.1, 0.2) | (0.2, 0.2) |
| Monomer - HOMO-LUMO gap | | | | |
| LN | 0.0278 | 0.0296 | 0.0325 | (0.9, 0.9) | (0.4, 0.4) |
| RF | 0.0267 | 0.0289 | 0.0318 | (0.3, 0.3) | (0.3, 0.3) |
| NN | 0.0271 | 0.0296 | 0.0318 | (0.2, 0.2) | (0.4, 0.4) |
| Polymer - Refractive index | | | | |
| LN | 0.0726 | 0.0724 | 0.0774 | (0.6, 0.5) | (0.4, 0.4) |
| RF | 0.0823 | 0.0794 | 0.0912 | (0.1, 0.1) | (0.4, 0.4) |
| NN | 0.0825 | 0.0820 | 0.0858 | (0.0, 0.0) | (0.1, 0.1) |
| Polymer - Band gap | | | | |
| LN | 0.7679 | 0.7431 | 0.7771 | (0.8, 0.9) | (0.9, 0.9) |
| RF | 0.6330 | 0.6593 | 0.6366 | (0.9, 0.9) | (0.8, 0.8) |
| NN | 0.7416 | 0.8077 | 0.7816 | (0.5, 0.5) | (0.8, 0.7) |
| Polymer - Dielectric constant | | | | |
| LN | 0.5053 | 0.4960 | 0.4746 | (0.1, 0.1) | (0.6, 0.5) |
| RF | 0.4981 | 0.4972 | 0.4982 | (0.1, 0.1) | (0.6, 0.5) |
| NN | 0.4900 | 0.5233 | 0.5251 | (0.2, 0.2) | (0.9, 0.9) |
| Polymer - HOMO-LUMO gap | | | | |
| LN | 0.3202 | 0.3230 | 0.3943 | (0.0, 0.0) | (-0.2, 0) |
| RF | 0.3212 | 0.3305 | 0.4305 | (0.1, 0.1) | (0.5, 0.4) |
| NN | 0.3209 | 0.3308 | 0.3393 | (0.0, 0.0) | (0.1, 0.1) |
Table S4: Transfer between theoretical and experimental energy levels of HOMO for the OPV molecules (CEP and HOPV15)

| Source task | Target task | $f_r(x)$ | $f_{r_0}(x)$ | Hyperparameter |
|-------------|-------------|----------|--------------|----------------|
| Monomer     | LN          | 0.1986   | 0.1739       | (0.7, 0.8)     |
|             | RF          | 0.1970   | 0.1449       | (0.3, 0.6)     |
|             | NN          | 0.1482   | 0.1208       | (0.6, 0.7)     |

| Polymer     | - Dielectric constant | LN          | 0.7613   | 0.6881       | (0.1, 0.3)   |
|             |                       | RF          | 0.7645   | 0.6574       | (0.0, 0.3)   |
|             |                       | NN          | 0.7470   | 0.6939       | (0.1, 0.2)   |

| Polymer     | - HOMO-LUMO gap       | LN          | 1.0144   | 0.8676       | (0.3, 0.1)   |
|             |                       | RF          | 0.9307   | 0.7581       | (0.2, 0.1)   |
|             |                       | NN          | 0.8978   | 0.8347       | (0.1, 0.2)   |

| Polymer     | - Band gap            | LN          | 0.5930   | 0.5395       | (0.8, 0.8)   |
|             |                       | RF          | 0.5814   | 0.5457       | (0.2, 0.2)   |
|             |                       | NN          | 0.5767   | 0.5405       | (0.3, 0.3)   |

| Polymer     | - Refractive index    | LN          | 0.3552   | 0.3368       | (0.8, 0.8)   |
|             |                       | RF          | 0.3392   | 0.3475       | (0.0, 0.0)   |
|             |                       | NN          | 0.3452   | 0.3363       | (0.4, 0.2)   |

| Monomer     | LN          | 0.3016   | 0.3284       | (0.3, 0.4)   |
|             | RF          | 0.2971   | 0.3578       | (0.1, 0.2)   |
|             | NN          | 0.3059   | 0.3282       | (0.1, 0.2)   |

| Monomer     | LN          | 0.7779   | 0.7101       | (0.6, 0.7)   |
|             | RF          | 0.6711   | 0.6118       | (0.2, 0.4)   |
|             | NN          | 0.7652   | 0.6664       | (0.9, 0.9)   |

| Monomer     | LN          | 0.0768   | 0.0833       | (0.2, 0.2)   |
|             | RF          | 0.0784   | 0.0815       | (0.0, 0.1)   |
|             | NN          | 0.0768   | 0.0764       | (0.3, 0.3)   |

| Polymer     | LN          | 0.4789   | 0.4988       | (0.6, 0.6)   |
|             | RF          | 0.5349   | 0.5018       | (0.2, 0.4)   |
|             | NN          | 0.4988   | 0.5053       | (0.2, 0.2)   |

| Polymer     | LN          | 0.4311   | 0.4263       | (-0.9, 0)    |
|             | RF          | 0.3735   | 0.3935       | (-0.1, 0)    |
|             | NN          | 0.3789   | 0.4322       | (-0.6, 0)    |

| Monomer     | LN          | 0.2924   | 0.3133       | (0.5, 0.5)   |
|             | RF          | 0.3005   | 0.3179       | (0.6, 0.6)   |
|             | NN          | 0.2853   | 0.3113       | (0.2, 0.1)   |

| Monomer     | LN          | 0.9456   | 0.8603       | (0.2, 0.2)   |
|             | RF          | 0.8252   | 0.8266       | (0.1, 0.1)   |
|             | NN          | 0.8960   | 0.9528       | (0.1, 0.1)   |

| Monomer     | LN          | 0.0912   | 0.0859       | (-0.3, 0)    |
|             | RF          | 0.0879   | 0.0869       | (0.0, 0.0)   |
|             | NN          | 0.0843   | 0.0874       | (0.0, 0.0)   |

| Polymer     | LN          | 0.7835   | 0.7310       | (0.3, 0.5)   |
|             | RF          | 0.8140   | 0.7811       | (0.6, 0.7)   |
|             | NN          | 0.8146   | 0.7304       | (0.7, 0.7)   |

| Polymer     | LN          | 0.3037   | 0.3035       | (0.3, 0.3)   |
|             | RF          | 0.3051   | 0.3091       | (0.8, 0.8)   |
|             | NN          | 0.3289   | 0.3032       | (0.2, 0.2)   |

| Monomer     | LN          | 0.2649   | 0.2996       | (0.4, 0.3)   |
|             | RF          | 0.2745   | 0.3235       | (0.2, 0.2)   |
|             | NN          | 0.2621   | 0.3071       | (0.1, 0.1)   |

| Monomer     | LN          | 0.8649   | 0.8294       | (0.3, 0.3)   |
|             | RF          | 1.0292   | 0.8434       | (0.5, 0.5)   |
|             | NN          | 0.9574   | 0.8242       | (0.2, 0.1)   |

| Monomer     | LN          | 0.0745   | 0.0836       | (0.3, 0.3)   |
|             | RF          | 0.0795   | 0.0810       | (0.0, 0.1)   |
|             | NN          | 0.0758   | 0.0822       | (0.3, 0.3)   |

| Monomer     | LN          | 0.9409   | 1.0573       | (0.1, 0.2)   |
|             | RF          | 0.9511   | 1.0695       | (0.3, 0.3)   |
|             | NN          | 1.0031   | 1.0513       | (0.5, 0.5)   |

| Polymer     | LN          | 0.5058   | 0.4930       | (0.8, 0.8)   |
|             | RF          | 0.4338   | 0.4669       | (0.3, 0.3)   |
|             | NN          | 0.4782   | 0.5008       | (0.3, 0.2)   |

Table S4: Transfer between theoretical and experimental energy levels of HOMO for the OPV molecules (CEP and HOPV15)
Table S5: Formation energy of SiO$_2$/CdI$_2$ and all other inorganic compounds in the Materials Project database

| Source task | Target task | $f_s(x)$ | $f_{target}(x)$ | Hyperparameter |
|-------------|-------------|---------|----------------|----------------|
|             | SiO$_2$     | LN      | RF             | NN             |
| All (without SiO$_2$) | LN | 0.0183 | 0.0026 | 0.0002 | (-1.7, 0.8) |
|             | RF | 0.0012 | 0.0016 | 0.0023 | (0.2, 0.2) |
|             | NN | 0.0013 | 0.0021 | 0.0013 | (0.5, 0.5) |
| All (without CdI$_2$) | LN | 0.2017 | 0.1825 | 0.1938 | (0, 0) |
|             | RF | 0.2055 | 0.1831 | 0.2181 | (0.1, 0) |
|             | NN | 0.2006 | 0.1951 | 0.2966 | (0.8, 0.8) |

Table S6: Transfer on the prediction of torques at the seven joints in a SARCOS robot arm

| Source task | Target task | $f_s(x)$ | $f_{target}(x)$ | Hyperparameter |
|-------------|-------------|---------|----------------|----------------|
|             | 1st - torque | LN      | RF             | NN             |
|             | 2nd - torque | LN      | RF             | NN             |
|             | 3rd - torque | LN      | RF             | NN             |
|             | 4th - torque | LN      | RF             | NN             |
|             | 5th - torque | LN      | RF             | NN             |
|             | 6th - torque | LN      | RF             | NN             |
|             | 7th - torque | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |
|             |             | LN      | RF             | NN             |

24
| Source task | Target task | $f_{x}(x)$ | $f_{y}(x)$ | Hyperparameter |
|------------|-------------|------------|------------|----------------|
| **1st - torque** | LN | 8.0995 | 12.9936 | (0.1, 0) | (0.4, 0.4) | (0.9, 0.9) |
| | RF | 8.5322 | 13.1431 | (0.1, 0.1) | (0.3, 0.3) | (0.9, 0.9) |
| | NN | 6.5680 | 11.8912 | (0.4, 0.4) | (0.4, 0.4) | (0.2, 0.1) |
| **2nd - torque** | LN | 5.6931 | 9.2281 | (0.2, 0.2) | (0.4, 0.4) | (0.1, 0.1) |
| | RF | 5.7954 | 9.1785 | (0.1, 0.2) | (0.1, 0.2) | (0.7, 0.6) |
| | NN | 5.6663 | 8.9926 | (0.0, 0.0) | (0.2, 0.3) | (0.1, 0.1) |
| **3rd - torque** | LN | 0.4547 | 0.7286 | (0.0, 0.0) | (0.4, 0.4) | (0.1, 0.1) |
| | RF | 0.4870 | 0.6988 | (0.0, 0.0) | (0.3, 0.3) | (0.7, 0.7) |
| | NN | 0.5083 | 0.7514 | (0.2, 0.2) | (0.3, 0.3) | (0.1, 0.1) |
| **4th - torque** | LN | 3.7396 | 6.8087 | (0.2, 0.2) | (0.3, 0.3) | (0.3, 0.3) |
| | RF | 3.9574 | 6.7297 | (0.4, 0.4) | (0.4, 0.4) | (0.2, 0.1) |
| | NN | 4.1679 | 6.9722 | (0.6, 0.6) | (0.5, 0.5) | (0.2, 0.1) |
| **5th - torque** | LN | 0.4547 | 0.7286 | (0.0, 0.0) | (0.4, 0.4) | (0.1, 0.1) |
| | RF | 0.4870 | 0.6988 | (0.0, 0.0) | (0.3, 0.3) | (0.7, 0.7) |
| | NN | 0.5083 | 0.7514 | (0.2, 0.2) | (0.3, 0.3) | (0.1, 0.1) |
| **6th - torque** | LN | 6.8177 | 15.6776 | (0.1, 0.1) | (0.3, 0.3) | (0.3, 0.3) |
| | RF | 6.0471 | 11.3694 | (0, 0) | (0.2, 0.2) | (0.5, 0.5) |
| | NN | 6.0471 | 11.3694 | (0, 0) | (0.2, 0.2) | (0.5, 0.5) |
| **7th - torque** | LN | 0.8748 | 0.7716 | (0.7, 0.7) | (0.8, 0.8) | (0.8, 0.8) |
| | RF | 0.8717 | 0.9576 | (0.7, 0.8) | (0.8, 0.9) | (0.7, 0.8) |
| | NN | 0.6443 | 0.6389 | (0.9, 0.9) | (0.9, 0.9) | (0.8, 0.8) |
| Source task | Target task | $f_y(x)$ | $f_{y_{	ext{pred}}}(x)$ | Hyperparameter |
|-------------|-------------|---------|----------------|----------------|
| LN          | RF          | NN      | LN  | RF  | NN  | LN  | RF  | NN  |
| 1st - torque| LN 8.2293  | 13.3603 | 17.8432 | (0.5, 0.5) | (0.5, 0.4) | (0.4, 0.4) |
|             | RF 6.6954  | 14.7708 | 15.3444 | (0.2, 0.2) | (0.4, 0.5) | (0.9, 0.9) |
|             | NN 6.9347  | 13.6454 | 19.2643 | (0.6, 0.6) | (0.6, 0.6) | (0.0, 0) |
| 2nd - torque| LN 6.3065  | 7.7745  | 9.9898  | (0.3, 0.3) | (0.4, 0.4) | (0.8, 0.8) |
|             | RF 6.4229  | 9.5041  | 10.7557 | (0.2, 0.2) | (0.3, 0.2) | (0.8, 0.7) |
|             | NN 6.9347  | 13.6454 | 19.2643 | (0.6, 0.6) | (0.6, 0.6) | (0.0, 0) |
| 3rd - torque| LN 3.8986  | 3.7911  | 4.1658  | (0.3, 0.3) | (0.8, 0.8) | (0.9, 0.9) |
|             | RF 5.1348  | 5.6002  | 5.8848  | (0.3, 0.3) | (0.8, 0.8) | (0.9, 0.9) |
|             | NN 3.7655  | 3.9798  | 10.0019 | (0.1, 0.1) | (0.7, 0.7) | (0.2, 0.3) |
| 4th - torque| LN 0.4950  | 0.6895  | 0.8496  | (0.0, 0)   | (0.2, 0.2) | (0.3, 0.3) |
|             | RF 0.5150  | 0.7038  | 0.8149  | (0.2, 0.2) | (0.5, 0.3) | (0.7, 0.6) |
|             | NN 0.5170  | 0.7280  | 1.0479  | (-0.1, 0)  | (-0.1, 0)  | (0.2, 0) |
| 5th - torque| LN 1.1954  | 1.2089  | 2.1451  | (0.0, 0)   | (0.0, 0)   | (0.9, 0.9) |
|             | RF 1.1798  | 1.2039  | 1.5165  | (-0.1, 0)  | (0.4, 0.2) | (0.6, 0.3) |
|             | NN 1.1419  | 1.1475  | 1.6955  | (0.1, 0.2) | (0.0, 0)   | (0.4, 0.4) |

References

[1] S. Liu and K. Fukumizu, “Estimating posterior ratio for classification: transfer learning from probabilistic perspective,” in Proceedings of the 2016 SIAM International Conference on Data Mining, 2016, pp. 747–755.

[2] H. Yamada, C. Liu, S. Wu, Y. Koyama, S. Ju, J. Shiomi, J. Morikawa, and R. Yoshida, “Predicting materials properties with little data using shotgun transfer learning,” ACS Central Science, vol. 5, no. 10, pp. 1717–1730, 2019.

[3] S. A. Lopez, E. O. Pyzer-Knapp, G. N. Simm, T. Lutzow, K. Li, L. R. Seress, J. Hachmann, and A. Aspuru-Guzik, “The Harvard organic photovoltaic dataset,” Scientific Data, vol. 3, no. 1, pp. 1–7, 2016.

[4] E. O. Pyzer-Knapp, K. Li, and A. Aspuru-Guzik, “Learning from the Harvard clean energy project: the use of neural networks to accelerate materials discovery,” Advanced Functional Materials, vol. 25, no. 41, pp. 6495–6502, 2015.

[5] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, “The Materials Project: A materials genome approach to accelerating materials innovation,” APL Materials, vol. 1, no. 1, p. 011002, 2013.

[6] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning. MIT Press, 2006.

[7] “Xenonpy,” https://xenonpy.readthedocs.io/en/latest

[8] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. Montgomery, J. A., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. P. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, “Gaussian 09,” 2016, Gaussian, Inc., Wallingford CT.

[9] R. Guha, “Chemical informatics functionality in R,” Journal of Statistical Software, vol. 18, no. 5, pp. 1–16, 2007.