Transfer learning with affine model transformation

Anonymous authors
Paper under double-blind review

Abstract

Supervised transfer learning (TL) has received considerable attention because of its potential to boost the predictive power of machine learning in cases with limited data. In a conventional scenario, cross-domain differences are modeled and estimated using a given set of source models and samples from a target domain. For example, if there is a functional relationship between source and target domains, only domain-specific factors are additionally learned using target samples to shift the source models to the target. However, the general methodology for modeling and estimating such cross-domain shifts has been less studied. This study presents a TL framework that simultaneously and separately estimates domain shifts and domain-specific factors using given target samples. Assuming consistency and invertibility of the domain transformation functions, we derive an optimal family of functions to represent the cross-domain shift. The newly derived class of transformation functions takes the same form as invertible neural networks using affine coupling layers, which are widely used in generative deep learning. We show that the proposed method encompasses a wide range of existing methods, including the most common TL procedure based on feature extraction using neural networks. We also clarify the theoretical properties of the proposed method, such as the convergence rate of the generalization error, and demonstrate the practical benefits of separately modeling and estimating domain-specific factors through several case studies.

1 Introduction

Transfer learning (TL) is applied to improve the predictive performance of machine learning in a target domain with limited data by reusing knowledge gained from training in related source domains. Its great potential has already been demonstrated in various real-world problems, including computer vision (Krizhevsky et al., 2012; Csurka, 2017), natural language processing (Ruder et al., 2019; Devlin et al., 2019), biology (Sevakula et al., 2019), and materials science (Yamada et al., 2019; Wu et al., 2019; Ju et al., 2021). Notably, most of the outstanding successes of TL to date have relied on naive analytic procedures using deep neural networks. For example, a conventional method reuses feature representations encoded in an intermediate layer of a pre-trained model as an input for the target task, or uses samples from the target domain to fine-tune the parameters of the pre-trained source model (Yosinski et al., 2014). While such methods are operationally plausible and intuitive, they lack methodological principles and remain theoretically unexplored in terms of their learning capability for limited data. This study develops a principled methodology generally applicable to various kinds of TL.

In this study, we focus on supervised TL settings. In particular, we deal with settings where, given feature representations obtained from training in the source domain, we use samples from the target domain to model and estimate their shift to the target. This procedure is called hypothesis transfer learning (HTL); several methods have been proposed, such as using a linear transformation function (Kuzborskij & Orabona, 2013; 2017) or considering a general class of continuous transformation functions (Du et al., 2017). If the transformation function appropriately captures the functional relationship between the source and target domains, only the domain-specific factors need to be additionally learned, which can be done efficiently even with a limited sample size. In other words, the performance of HTL depends strongly on whether the transformation function appropriately represents the cross-domain shift. However, the general methodology for modeling and estimating such domain shifts has been less studied.

This study develops a TL methodology to estimate cross-domain shifts and domain-specific factors simultaneously and separately using given target samples. The HTL framework we employ considers two different transformation functions: one to represent and estimate domain-specific factors and the other to adapt them to the target domain in combination with the source features. For these transformation functions, we derive a class of theoretically optimal transformation functions based on the assumptions of invertibility and differentiability as well as consistency, i.e., that the optimal predictor remains unchanged through the two transformations. The resulting function class takes the form of an affine coupling $g_1 + g_2 \cdot g_3$ of three functions g_1, g_2 and g_3, where the cross-domain shift is represented by the functions g_1 and g_2, and the domain-specific factors are represented by g_3. These functions can be estimated simultaneously using conventional supervised learning algorithms such as kernel methods or deep neural networks. Hereafter, we refer to this framework as the *affine model transfer*.

The affine coupling used in the affine model transfer is the basic model architecture of invertible neural networks, which is widely used in several fields including generative modeling (Papamakarios et al., 2021; Dinh et al., 2014; 2017; Kingma & Dhariwal, 2018). In spite of its simple architecture, invertible neural networks with affine coupling layers are known to have universal approximation ability (Teshima et al., 2020), which means that the proposed model class has the potential to represent a broad class of transformation functions. Furthermore, when we use the intermediate layers of a source neural network as the feature representations in the target domain, the affine model transfer is identical to the ordinary TL based on feature extractions. As described, we can formulate a wide variety of TL algorithms within the affine model transfer, including neural transfer as a special case.

To summarize, the contributions of our study are as follows:

- The affine model transfer is proposed to adapt source features to the target domain by separately estimating cross-domain shift and domain-specific factors.
- Several existing methods of HTL are encompassed in the affine model transfer, including neural network-based TL.
- The affine model transfer can work with any type of source model. For example, non-machine learning models such as physical models can be used. It can also handle multiple source models without loss of generality.
- For each of the three functions g_1, g_2, and g_3, we provide an efficient and stable estimation algorithm when modeled using the kernel method.
- Two theoretical properties of the affine transfer model are shown: the generalization bound and the excess risk bound.

With several applications, we compare the affine model transfer with other TL algorithms, discuss its strengths and weaknesses, and demonstrate the advantage of being able to estimate cross-domain shifts and domain-specific factors separately.

2 Transfer learning via transformation function

2.1 Affine model transfer

This study considers regression problems with squared loss. We assume that the output of the target domain $y \in \mathcal{Y} \subset \mathbb{R}$ follows $y = f_t(x) + \epsilon$, where $f_t : \mathcal{X} \to \mathbb{R}$ is the true model on the target domain, and the observation noise ϵ has mean zero and variance σ^2. We are given n samples $\{(x_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$ from the target domain and the feature representation $f_s(x) \in \mathcal{F}_s$ from one or more source domains. Typically, f_s is given as a vector $f_s(x) = [f_1(x), f_2(x), \ldots, f_M(x)]^\top$, including the output of the source models, observed data in the source domains or learned features in a pre-trained source neural network, and so on, but it can also be a non-vector feature such as a tensor, graph or text. Hereafter, f_s is referred to as the source features.

Du et al. (2017) provided a TL framework using transformation functions as follows:

1. With the source features, perform a variable transformation of the observed outputs as $z_i = \phi(y_i, f_s(x_i))$, using the data transformation function $\phi : \mathcal{Y} \times \mathcal{F}_s \to \mathbb{R}$.
2. Train an intermediate model $\hat{g}(x)$ using the transformed sample set $\{(x_i, z_i)\}_{i=1}^n$ to predict the transformed output z for any given x.
3. Obtain a target model $\hat{f}_t(x) = \psi(\hat{g}(x), f_s(x))$ using the model transformation function $\psi : \mathbb{R} \times \mathcal{F}_s \to \mathcal{Y}$ that combines $\hat{g}$ and f_s to define a predictor.

In particular, Du et al. (2017) considers the case where the model transformation function ψ is equal to the inverse of the data transformation function ϕ^{-1}. We consider a more general case that eliminates this constraint.

The proposed class of TL methods includes several methods proposed in previous studies. For example, Kuzborskij & Orabona (2013; 2017) proposed a learning algorithm consisting of linear data transformation and linear model transformation: $\phi = y - \langle \theta, f_s(x) \rangle$ and $\psi = g(x) + \langle \theta, f_s(x) \rangle$ with pre-defined coefficients θ. In this case, factors unexplained by the linear combination of source features are learned with g, and the target output is predicted additively with the common factor $\langle \theta, f_s(x) \rangle$ and the additionally learned g. In Minami et al. (2021), it is shown that a type of Bayesian TL is equivalent to the following transformation functions; for $\mathcal{F}_s \subset \mathbb{R}$, $\phi = (y - \tau f_s(x))/(1-\tau)$ and $\psi = \rho g(x) + (1-\rho) f_s(x)$ with two varying hyperparameters $\tau < 1$ and $0 \leq \rho \leq 1$. This includes TL using density ratio estimation (Liu & Fukumizu, 2016) and neural network-based fine-tuning as special cases when the two hyperparameters belong to specific regions.

For simplicity, we denote the transformation functions as $\phi_{f_s}(\cdot) = \phi(\cdot, f_s(x))$ and $\psi_{f_s}(\cdot) = \psi(\cdot, f_s(x))$. To derive the optimal class of ϕ and ψ, we make the following assumptions:

Assumption 1 (Differentiability)**.** *The data transformation function ϕ is differentiable with respect to the first argument.*

Assumption 2 (Invetribility)**.** *The model transformation function ψ is invertible with respect to the first argument, i.e., its inverse $\psi_{f_s}^{-1}$ exists.*

Assumption 3 (Consistency)**.** *For any distribution on the target domain $p_t(x, y)$, and for all $x \in \mathcal{X}$,*

$$\psi_{f_s}(g^*(x)) = \mathbb{E}_{p_t}[Y|X = x],$$

where $g^*(x) = \mathbb{E}_{p_t}[\phi_{f_s}(Y)|X = x]$.

The regression function that minimizes the mean squared error is given by the conditional mean. In Assumption 3, g^* is defined to be the best predictor function for the transformed variable $z = \phi_{f_s}(y)$ in terms of the mean squared error. Assumption 3 states that composing the optimal g^* with the model transformation function ψ_{f_s} leads to the best predictor $\mathbb{E}_{p_t}[Y|X = x]$ for the target domain. This assumption corresponds to the unbiased condition of Du et al. (2017).

Under these assumptions, we derive the optimal class of the transformation functions which minimize the mean squared error.

Theorem 1. *Let $g_1, g_2 : \mathcal{F}_s \to \mathbb{R}$ denote arbitrary functions. If Assumptions 1-3 hold, then the transformation functions ϕ and ψ satisfy the following two properties:*

(i) $\psi_{f_s}^{-1} = \phi_{f_s}$.

(ii) $\psi_{f_s}(g) = g_1(f_s) + g_2(f_s) \cdot g$.

Proof. According to Assumption 3, it holds that for any $p_t(y|x)$,

$$\psi_{f_s}\left(\int \phi_{f_s}(y) p_t(y|x) \mathrm{d}y\right) = \int y p_t(y|x) \mathrm{d}y. \tag{1}$$

(i) Let δ_{y_0} be the Dirac delta function supported on y_0. Substituting $p_t(y|x) = \delta_{y_0}$ into Eq. (1), we have

$$\psi_{f_s}(\phi_{f_s}(y_0)) = y_0 \ (\forall y_0 \in \mathcal{Y}).$$

Under Assumption 2, this implies the property (i).

(ii) For simplicity, we assume the inputs x are fixed and $p_t(y|x) > 0$. Applying the property (i) to Eq. (1) yields

$$\int \phi_{f_s}(y) p_t(y|x) \mathrm{d}y = \phi_{f_s}\biggl(\int y p_t(y|x) \mathrm{d}y\biggr).$$

We consider a two-component mixture $p_t(y|x) = (1-\epsilon)q(y|x) + \epsilon h(y|x)$ with a mixing rate $\epsilon \in [0,1]$, where q and h denote arbitrary probability density functions. Then, we have

$$\int \phi_{f_s}(y)\bigl\{(1-\epsilon)q(y|x) + \epsilon h(y|x)\bigr\}\mathrm{d}y = \phi_{f_s}\biggl(\int y\bigl\{(1-\epsilon)q(y|x) + \epsilon h(y|x)\bigr\}\mathrm{d}y\biggr).$$

Taking the derivative at $\epsilon = 0$, we have

$$\int \phi_{f_s}(y)\bigl\{h(y|x) - q(y|x)\bigr\}\mathrm{d}y = \phi'_{f_s}\biggl(\int y q(y|x)\mathrm{d}y\biggr)\biggl(\int y\bigl\{h(y|x) - q(y|x)\bigr\}\mathrm{d}y\biggr),$$

which yields

$$\int \bigl\{h(y|x) - q(y|x)\bigr\}\bigl\{\phi_{f_s}(y) - \phi'_{f_s}\bigl(\mathbb{E}_q[Y|X]\bigr)y\bigr\}\mathrm{d}y = 0. \tag{2}$$

For Eq. (2) to hold for any q and h, $\phi_{f_s}(y) - \phi'_{f_s}\bigl(\mathbb{E}_q[Y|X = x]\bigr)y$ must be independent of y. Thus, the function ϕ_{f_s} and its inverse $\psi_{f_s} = \phi_{f_s}^{-1}$ are limited to affine transformations. □

Theorem 1 implies that the mean squared error is minimized when the data and model transformation functions are given by an affine transformation and its inverse, respectively. In summary, the optimal class for HTL is expressed as follows:

$$\mathcal{H} = \bigl\{x \mapsto g_1(f_s(x)) + g_2(f_s(x)) \cdot g_3(x) \mid g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2, g_3 \in \mathcal{G}_3\bigr\},$$

where $\mathcal{G}_1, \mathcal{G}_2$ and $\mathcal{G}_3$ are the arbitrarily function classes. Here, each of g_1 and g_2 is modeled as a function of f_s that represents common factors across the source and target domains. g_3 is modeled as a function of x, in order to capture the domain-specific factors unexplainable by the source features.

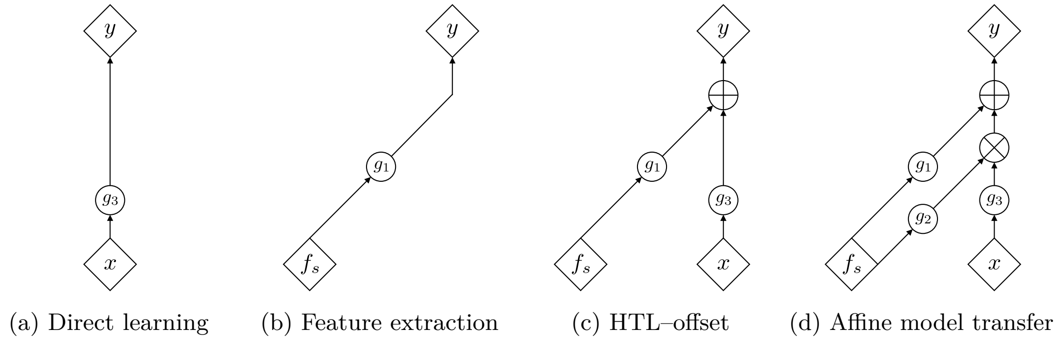
We have derived the optimal form of the transformation functions when the squared loss is employed. Even for general convex loss functions, (i) of Theorem 1 still holds. However, (ii) of Theorem 1 does not generally hold because the optimal transformation function depends on the loss function. Optimal models for various convex loss functions are discussed in Appendix A.1.

Here, the affine transformation is found to be optimal in terms of minimizing the mean squared error. We can also derive the same optimal function by minimizing the upper bound of the estimation error in the HTL procedure, as discussed in Appendix A.2.

2.2 Relation to existing methods

The affine model transfer encompasses a learning scheme without transfer; i.e., by setting $g_1(f_s) = 0$ and $g_2(f_s) = 1$, the prediction model is estimated without using the source features. This corresponds to an ordinary direct learning procedure.

In prior work, Kuzborskij & Orabona (2013) employs a two-step procedure where the source features are combined with pre-defined weights, and then the auxiliary model is additionally learned for the residuals unexplainable by the source features. The affine model transfer can represent this HTL as a special case by setting $g_2(f_s) = 1$, but differs in the following two aspects. First, while the existing method models only the difference (residuals) from the source domains, our model can also consider the cross-domain ratio relationship, i.e., we also consider the function $g_2(f_s)$. Another distinctive feature of affine model transfer lies in the learning procedure that can estimate the data and model transformation functions simultaneously, as described in Section 3.



(a) Direct learning (b) Feature extraction (c) HTL–offset (d) Affine model transfer

Figure 1: Model architectures for the affine model transfer and related procedures. (a) Direct learning predicts outputs using only the original inputs x, while (b) feature extraction-based neural transfer predicts outputs using only the source features f_s. (c) The HTL procedure proposed in Kuzborskij & Orabona (2013) (HTL–offset) constructs the predictor as the sum of $g_1(f_s)$ and $g_3(x)$. (d) The affine model transfer encompasses these procedures, computing g_1 and g_2 as functions of the source features and constructing the predictor as an affine combination with g_3.

The affine model transfer can be naturally expressed as an architecture of network networks. This architecture, called affine coupling layers, is widely used for invertible neural networks in flow-based generative modeling (Dinh et al., 2014; 2017). Neural networks based on affine coupling layers have been proven to have universal approximation ability (Teshima et al., 2020). This implies that the affine transfer model has the potential to represent a wide range of function classes, despite its simple architecture based on the affine coupling of three functions.

When a pre-trained source model is provided as a neural network, TL is usually performed with the intermediate layer as input to the model in the target domain. This is called a feature extractor or frozen featurizer and has been experimentally and theoretically proven to have strong transfer capability as the de facto standard for TL (Yosinski et al., 2014; Tripuraneni et al., 2020). The affine model transfer encompasses the neural feature extractor as a special subclass, which is equivalent to setting $g_2(f_s)g_3(x) = 0$. A performance comparison of the affine model transfer with the neural feature extractor is presented in Section 5.2.

The affine model transfer can also be interpreted as generalizing the feature extractor by adding a product term $g_2(f_s)g_3(x)$. This additional term allows for the inclusion of unknown factors in the transferred model that are unexplainable by source features alone. Furthermore, this encourages the avoidance of a negative transfer. The usual TL based only on $g_1(f_s)$ attempts to explain and predict the data generation process in the target domain using only features from the source domain. However, in the presence of domain-specific factors, a negative transfer can occur owing to a lack of descriptive power. The additional term compensates for this shortcoming. The comparison of behavior for the case with the non-relative source features is described in Section 5.1.

3 Modeling and estimation

In this section, we focus on using kernel methods for the affine transfer model. Let $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ be reproducing kernel Hilbert spaces (RKHSs) with positive-definite kernels k_1, k_2 and k_3, which define the feature mappings $\Phi_1 : \mathcal{F}_s \to \mathcal{H}_1, \Phi_2 : \mathcal{F}_s \to \mathcal{H}_2$ and $\Phi_3 : \mathcal{X} \to \mathcal{H}_3$, respectively. For the proposed model class, the ℓ_2-regularized empirical risk with the squared loss is given as follows:

$$\begin{aligned}F_{\alpha,\beta,\gamma} = \frac{1}{n}\sum_{i=1}^{n}\left\{y_i - \langle \alpha, \Phi_1(f_s(x_i))\rangle_{\mathcal{H}_1} - \langle \beta, \Phi_2(f_s(x_i))\rangle_{\mathcal{H}_2}\langle \gamma, \Phi_3(x_i)\rangle_{\mathcal{H}_3}\right\}^2 \\ + \lambda_1\|\alpha\|^2_{\mathcal{H}_1} + \lambda_2\|\beta\|^2_{\mathcal{H}_2} + \lambda_3\|\gamma\|^2_{\mathcal{H}_3},\end{aligned} \tag{3}$$

Algorithm 1 Block relaxation algorithm (Zhou et al., 2013).

```
Initialize
    $a_0, \quad b_0 \neq 0, \quad c_0 \neq 0$
repeat
    $a_{t+1} = \arg\min_a F(a,\ b_{t+1},\ c_{t+1})$
    $b_{t+1} = \arg\min_b F(a_{t+1},\ b,\ c_t)$
    $c_{t+1} = \arg\min_c F(a_{t+1},\ b_{t+1},\ c)$
until convergence
```

where $\lambda_1, \lambda_2, \lambda_3 > 0$ are hyperparameters for the regularization. According to the representer theorem, the minimizer of $F_{\alpha,\beta,\gamma}$ with respect to the parameters $\alpha \in \mathcal{H}_1$, $\beta \in \mathcal{H}_2$, and $\gamma \in \mathcal{H}_3$ reduces to

$$\alpha = \sum_{i=1}^{n} a_i \Phi_1(f_s(x_i)), \quad \beta = \sum_{i=1}^{n} b_i \Phi_2(f_s(x_i)), \quad \gamma = \sum_{i=1}^{n} c_i \Phi_3(x_i),$$

with the n-dimensional unknown parameter vectors $a, b, c \in \mathbb{R}^n$. Substituting this expression into Eq. (3), we can obtain the objective function as

$$\begin{aligned} F_{\alpha,\beta,\gamma} &= \frac{1}{n}\|y - K_1 a - (K_2 b) \circ (K_3 c)\|_2^2 + \lambda_1 a^\top K_1 a + \lambda_2 b^\top K_2 b + \lambda_3 c^\top K_3 c \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - k_1^{(i)\top} a - b^\top M^{(i)} c\right)^2 + \lambda_1 a^\top K_1 a + \lambda_2 b^\top K_2 b + \lambda_3 c^\top K_3 c \\ &\coloneqq F(a, b, c). \end{aligned} \tag{4}$$

Here, the symbol $\circ$ denotes the Hadamard product. K_I is the Gram matrix associated with the kernel k_I for $I \in \{1, 2, 3\}$. $k_I^{(i)} = [k_I(x_i, x_1) \cdots k_I(x_i, x_n)]^\top$ denotes the i-th column of the Gram matrix. The $n \times n$ matrix $M^{(i)}$ is given by the tensor product $M^{(i)} = k_2^{(i)} \otimes k_3^{(i)}$ of $k_2^{(i)}$ and $k_3^{(i)}$.

Because the model is linear with respect to parameter a and bilinear for b and c, the optimization of Eq. (4) can be solved using well-established techniques for the low-rank tensor regression, such as CP-decomposition (Harshman, 1970), Tucker decomposition (Tucker, 1966), and Tensor-Train decomposition (Oseledets, 2011). In this study, we use the block relaxation algorithm (Zhou et al., 2013) as described in Algorithm 1. It updates a, b, and c by repeatedly fixing two of the three parameters and minimizing the objective function for the remaining one. Fixing two parameters, the resulting subproblem can be solved analytically, because the objective function is expressed in a quadratic form for the remaining parameter. Starting from arbitrary initial values, the algorithm iteratively updates the parameters (a_t, b_t, c_t) at iteration t to $(a_{t+1}, b_{t+1}, c_{t+1})$ as follows:

$$\begin{aligned} a_{t+1} &= (K_1 + n\lambda_1 I_n)^{-1}(y - (K_2 b_t) \circ (K_3 c_t)), \\ b_{t+1} &= (\mathrm{diag}(K_3 c_t)^2 K_2 + n\lambda_2 I_n)^{-1}\mathrm{diag}(K_3 c_t)(y - K_1 a_{t+1}), \\ c_{t+1} &= (\mathrm{diag}(K_2 b_{t+1})^2 K_3 + n\lambda_3 I_n)^{-1}\mathrm{diag}(K_2 b_{t+1})(y - K_1 a_{t+1}), \end{aligned}$$

where y is a vector of n observed outputs, I_n denotes the identity matrix of size n, and $\mathrm{diag}(v)$ is the diagonal matrix whose diagonal element is given by the vector v.

Algorithm 1 alternately estimates the parameters (a, b) of the transformation function and the parameters c in the predictive model of the transformed output with the given transformed dataset $\{(x_i, z_i)\}_{i=1}^n$. The consistency and asymptotic normality of this estimator have been proven in Zhou et al. (2013).

4 Theoretical results

In this section, we present two theoretical properties, the generalization bound and excess risk bound.

Let $(\mathcal{Z}, P)$ be an arbitrary probability space, and set $\{z_i\}_{i=1}^n$ to be independent random variables distributed according to P. For a function $f : \mathcal{Z} \to \mathbb{R}$, define the expectation of f with respect to P and its empirical counterpart as

$$Pf = \mathbb{E}_P f(z), \qquad P_n f = \frac{1}{n} \sum_{i=1}^{n} f(z_i).$$

Let $\ell(y, y')$ be a non-negative loss bounded from above by $L > 0$, such that for any fixed $y' \in \mathcal{Y}$, $y \mapsto \ell(y, y')$ is μ_ℓ-Lipschitz for some $\mu_\ell > 0$.

Recall that the function class proposed in this work is

$$\mathcal{H} = \big\{x \mapsto g_1(f_s(x)) + g_2(f_s(x)) \cdot g_3(x) \mid g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2, g_3 \in \mathcal{G}_3\big\}.$$

In particular, the following discussion in this section assumes that g_1, g_2, and g_3 are represented by linear functions on the RKHSs.

4.1 Generalization bound

The optimization problem is expressed as follows:

$$\min_{\alpha,\beta,\gamma} P_n \ell\big(y,\, \langle \alpha, \Phi_1 \rangle_{\mathcal{H}_1} + \langle \beta, \Phi_2 \rangle_{\mathcal{H}_2} \langle \gamma, \Phi_3 \rangle_{\mathcal{H}_3}\big) + \lambda_\alpha \|\alpha\|_{\mathcal{H}_1}^2 + \lambda_\beta \|\beta\|_{\mathcal{H}_2}^2 + \lambda_\gamma \|\gamma\|_{\mathcal{H}_3}^2, \tag{5}$$

where $\Phi_1 = \Phi_1(f_s(x)), \Phi_2 = \Phi_2(f_s(x))$ and $\Phi_3 = \Phi_3(x)$ denote the feature maps, and $\lambda_\alpha, \lambda_\beta, \lambda_\gamma > 0$ are the regularization parameters. Without loss of generality, it is assumed that $\|\Phi_1\|_{\mathcal{H}_1}^2 \le 1, \|\Phi_2\|_{\mathcal{H}_2}^2 \le 1$, and $\|\Phi_3\|_{\mathcal{H}_3}^2 \le 1$. Hereafter, we will omit the suffixes $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ in the norms if there is no ambiguity.

Let $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ be a solution of Eq. (5). For any α, we have

$$\begin{aligned} \lambda_\alpha \|\hat{\alpha}\|^2 &\le P_n \ell(y, \langle \hat{\alpha}, \Phi_1 \rangle + \langle \hat{\beta}, \Phi_2 \rangle \langle \hat{\gamma}, \Phi_3 \rangle) + \lambda_\alpha \|\hat{\alpha}\|^2 + \lambda_\beta \|\hat{\beta}\|^2 + \lambda_\gamma \|\hat{\gamma}\|^2 \\ &\le P_n \ell(y, \langle \alpha, \Phi_1 \rangle) + \lambda_\alpha \|\alpha\|^2. \end{aligned} \tag{6}$$

The first inequality holds because $\ell(\cdot, \cdot)$ and $\|\cdot\|$ are non-negative. For the second inequality, we use the fact that the parameter set $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ is the minimizer of Eq. (5). Denoting $\hat{R}_s = \inf_\alpha \{P_n \ell(y, \langle \alpha, \Phi_1 \rangle) + \lambda_\alpha \|\alpha\|^2\}$, we obtain $\|\hat{\alpha}\|^2 \le \lambda_\alpha^{-1} \hat{R}_s$. Because the same inequality as Eq. (6) holds for $\lambda_\beta \|\hat{\beta}\|^2, \lambda_\gamma \|\hat{\gamma}\|^2$ and $P_n \ell(y, \hat{h})$, we have $\|\hat{\beta}\|^2 \le \lambda_\beta^{-1} \hat{R}_s, \|\hat{\gamma}\|^2 \le \lambda_\gamma^{-1} \hat{R}_s$, and $P_n \ell(y, \hat{h}) \le \hat{R}_s$. Moreover, we obtain $P\ell(y, \hat{h}) = \mathbb{E}[P_n \ell(y, \hat{h})] \le \mathbb{E}[\hat{R}_s]$. Therefore, it is sufficient to consider the following hypothesis class $\mathcal{H}$ and loss class $\mathcal{L}$:

$$\begin{aligned} \mathcal{H} &= \big\{x \mapsto \langle \alpha, \Phi_1 \rangle + \langle \beta, \Phi_2 \rangle \langle \gamma, \Phi_3 \rangle \\ &\qquad\qquad \mid \ \|\alpha\|^2 \le \lambda_\alpha^{-1} \hat{R}_s,\ \|\beta\|^2 \le \lambda_\beta^{-1} \hat{R}_s,\ \|\gamma\|^2 \le \lambda_\gamma^{-1} \hat{R}_s, P\ell(y, h) \le \mathbb{E}[\hat{R}_s]\big\}, \\ \mathcal{L} &= \big\{(x, y) \mapsto \ell(y, h) \ \mid \ h \in \mathcal{H}\big\}. \end{aligned}$$

Here, we show the generalization bound of the proposed model class. The following theorem is based on Kuzborskij & Orabona (2017), showing that the difference between the generalization error and the empirical error of this hypothesis class can be bounded using the magnitude of the relevance of the source and target domains.

Theorem 2 (Generalization bound)**.** *There exists a constant C depending only on $\lambda_\alpha, \lambda_\beta, \lambda_\gamma$ and L such that, for any $\eta > 0$ and $h \in \mathcal{H}$, with probability at least $1 - e^{-\eta}$,*

$$P\ell(y,\, h) - P_n \ell(y,\, h) = \ \tilde{O}\bigg(\bigg(\sqrt{\frac{R_s}{n}} + \frac{\mu_\ell^2 C^2 + \sqrt{\eta}}{n}\bigg)\Big(\sqrt{L}C + \sqrt{L\eta}\Big) + \frac{C^2 L + L\eta}{n}\bigg),$$

where $R_s = \inf_\alpha \{P\ell(y, \langle \alpha, \Phi_1 \rangle) + \lambda_\alpha \|\alpha\|^2\}$.

The proof is given in Appendix B.1.

Because Φ_1 is the feature map from the source feature space $\mathcal{F}_s$ onto the RKHS $\mathcal{H}_1$, R_s corresponds to the true risk of training in the target domain using only the source features f_s. If this is sufficiently small, e.g., $R_s = \tilde{O}(n^{-1})$, the convergence rate indicated by Theorem 2 becomes n^{-1}, which is an improvement over the naive convergence rate $n^{-1/2}$. This means that if training in the source domain yields feature representations strongly related to the target domain, the convergence of training in the target domain is accelerated. Theorem 2 measures this cross-domain relation using the metric R_s.

Theorem 2 is based on Theorem 11 of Kuzborskij & Orabona (2017) in which the function class $g_1 + g_3$ is considered. Our work differs in the following two points: the source features are modeled not only additively but also multiplicatively, i.e., we consider the function class $g_1 + g_2 \cdot g_3$, and we also consider the estimation of the parameters for the source feature combination, i.e., the parameters of the functions g_1 and g_2. In particular, the latter affects the resulting rate in Theorem 2. Without estimating the source combination parameters, the rate indicated by Theorem 2 improves only up to $n^{-3/4}$. The details are discussed in Appendix B.1.

4.2 Excess risk bound

In this section, we analyze the excess risk, which is the difference between the risk of the estimated function and the smallest possible risk within the function class.

Recall that we consider the functions g_1, g_2 and g_3 to be the elements of the RKHSs $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$ with kernels k_1, k_2 and k_3, respectively. Define the kernel $k^{(1)} = k_1$, $k^{(2)} = k_2 \cdot k_3$ and $k = k^{(1)} + k^{(2)}$. Let $\mathcal{H}^{(1)}, \mathcal{H}^{(2)}$ and $\mathcal{H}$ be the RKHS with $k^{(1)}, k^{(2)}$ and k respectively. For $m = 1, 2$, consider the normalized Gram matrix $K^{(m)} = \frac{1}{n}(k^{(m)}(x_i, x_j))_{i,j=1,\dots,n}$ and its eigenvalues $(\hat{\lambda}_i^{(m)})_{i=1}^n$, arranged in nonincreasing order.

We prepare the following additional assumptions:

Assumption 4. *There exists an $h^* \in \mathcal{H}$ satisfying $P(y - h^*(x))^2 = \inf_{h \in \mathcal{H}} P(y - h(x))^2$. Similarly, there exists an $h^{(m)*} \in \mathcal{H}^{(m)}$ satisfying $P(y - h^{(m)*}(x))^2 = \inf_{h \in \mathcal{H}^{(m)}} P(y - h(x))^2$ for $m = 1, 2$.*

Assumption 5. *For $m = 1, 2$, there exist positive real numbers $a_m > 0$ and $s_m \in (0, 1)$ such that $\hat{\lambda}_j^{(m)} \leq a_m j^{-1/s_m}$.*

Assumption 4 is used in Bartlett et al. (2005) and is not overly restrictive as it holds for many regularization algorithms and convex, uniformly bounded function classes.

In the analysis of kernel methods, Assumption 5 is standard (Steinwart & Christmann, 2008), and is known to be equivalent to the classical covering or entropy number assumption (Steinwart et al., 2009). s_m measures the complexity of the RKHS, with larger values corresponding to more complex function spaces.

Under Assumption 4, we obtain the following excess risk bound for the proposed model class. The proof is based on Bartlett et al. (2005) and shown in Appendix B.2.

Theorem 3. *Let $\hat{h}$ be any element of $\mathcal{H}$ satisfying $P_n \ell(y, \hat{h}(x)) = \inf_{h \in \mathcal{H}} P_n \ell(y, h(x))$. Suppose that Assumption 4 is satisfied, then there exists a constant c depending only on μ_ℓ such that for any $\eta > 0$, with probability at least $1 - 5e^{-\eta}$,*

$$P(y - \hat{h}(x))^2 - P(y - h^*(x))^2 \;\leq\; c\left(\min_{0 \leq \kappa_1, \kappa_2 \leq n}\left\{\frac{\kappa_1 + \kappa_2}{n} + \left(\frac{1}{n}\sum_{j=\kappa_1+1}^{n}\hat{\lambda}_j^{(1)} + \sum_{j=\kappa_2+1}^{n}\hat{\lambda}_j^{(2)}\right)^{\frac{1}{2}}\right\} + \frac{\eta}{n}\right).$$

Theorem 3 is a multiple-kernel version of Corollary 6.7 of Bartlett et al. (2005), and a data-dependent version of Theorem 2 of Kloft & Blanchard (2011) which considers the eigenvalues of the Hilbert-Schmidt operators on $\mathcal{H}$ and $\mathcal{H}^{(m)}$. Theorem 3 concerns the eigenvalues of the Gram matrices $K^{(m)}$ computed from the data.

The following corollary follows from Theorem 3 and Assumption 5.

Corollary 4. *Let $\hat{h}$ be any element of $\mathcal{H}$ satisfying $P_n(y - \hat{h}(x))^2 = \inf_{h \in \mathcal{H}} P_n(y - h(x))^2$. Suppose that Assumption 4 and 5 are satisfied, then for any $\eta > 0$, with probability at least $1 - 5e^{-\eta}$,*

$$P(y - \hat{h}(x))^2 - P(y - h^*(x))^2 = O\Big(n^{-\frac{1}{1+\max\{s_1, s_2\}}}\Big).$$

Corollary 4 suggests that the convergence rate of the excess risk depends on the decay rates of the eigenvalues of two Gram matrices $K^{(1)}$ and $K^{(2)}$. s_1 is the decay rate of eigenvalues of $K^{(1)} = \frac{1}{n}(k_1(f_s(x_i), f_s(x_j)))_{i,j=1,\dots,n}$, representing the learning efficiency using only the source features. s_2 is the decay rate of the eigenvalues of the Hadamard product of the Gram matrices $K_2 = \frac{1}{n}(k_2(f_s(x_i), f_s(x_j))_{i,j=1,\dots,n}$ and $K_3 = \frac{1}{n}(k_3(x_i, x_j))_{i,j=1,\dots,n}$. The effect of combining the source features and the original inputs appears here. In general, it is difficult to discuss the relationship between the spectra of two Gram matrices K_2, K_3 and their Hadamard product $K_2 \circ K_3$. Intuitively, the smaller the overlap between the space spanned by source features f_s and by the original input x, the smaller the overlap between $\mathcal{H}_2$ and $\mathcal{H}_3$ because $\mathcal{H}_2$ is defined by the kernel $k_2(f_s(x), f_s(x'))$ and $\mathcal{H}_3$ is defined by the kernel $k_3(x, x')$. In other words, as the source features f_s and the original input x have different information, the tensor product $\mathcal{H}_2 \otimes \mathcal{H}_3$ will be more complex, and the decay rate s_2 is expected to be larger. In Appendix C, we experimentally confirm the relationship between the decay rate s_2 and the overlap of the space spanned by x and f_s.

5 Experimental results

We demonstrate the potential of the affine model transfer through three different case studies: (i) the prediction of feed-forward torque at seven joints of the SARCOS anthropomorphic robot arm (Williams & Rasmussen, 2006), (ii) the prediction of lattice thermal conductivity of inorganic crystalline materials (Yamada et al., 2019), (iii) TL for bridging the gap between experimental and theoretical values of specific heat capacity for organic polymers (Hayashi et al., 2022). Experimental details are described in Appendix D. The Python code is available at `https://github.com/mshunya/AffineTL`.

5.1 Kinematics of the robot arm

We experimentally investigated the learning performance of the affine model transfer, compared to several naive methods. The objective is to predict the feed-forward torques, required to follow the desired trajectory, at seven different joints of the SARCOS anthropomorphic robot arm (Williams & Rasmussen, 2006). Twenty-one features representing the joint position, velocity, and acceleration were used as the input variable $x \in \mathbb{R}^d (d = 21)$. The target task was to predict the torque value at one joint, and the source task was defined as the prediction of torque at the other six joints. The experiments were conducted with seven different tasks (denoted as Torque 1-7) corresponding to the seven different joints. The vector of the source torque values was used as the source feature. For each target region, a training set of size $n \in \{5, 10, 15, 20, 30, 40, 50\}$ was randomly constructed 20 times, and the remaining samples were used as the test sets (Appendix D). The experiment was designed for TL with a fairly small sample size.

For comparison, the following seven procedures were tested, including two existing HTL models (Kuzborskij & Orabona, 2013; Du et al., 2017):

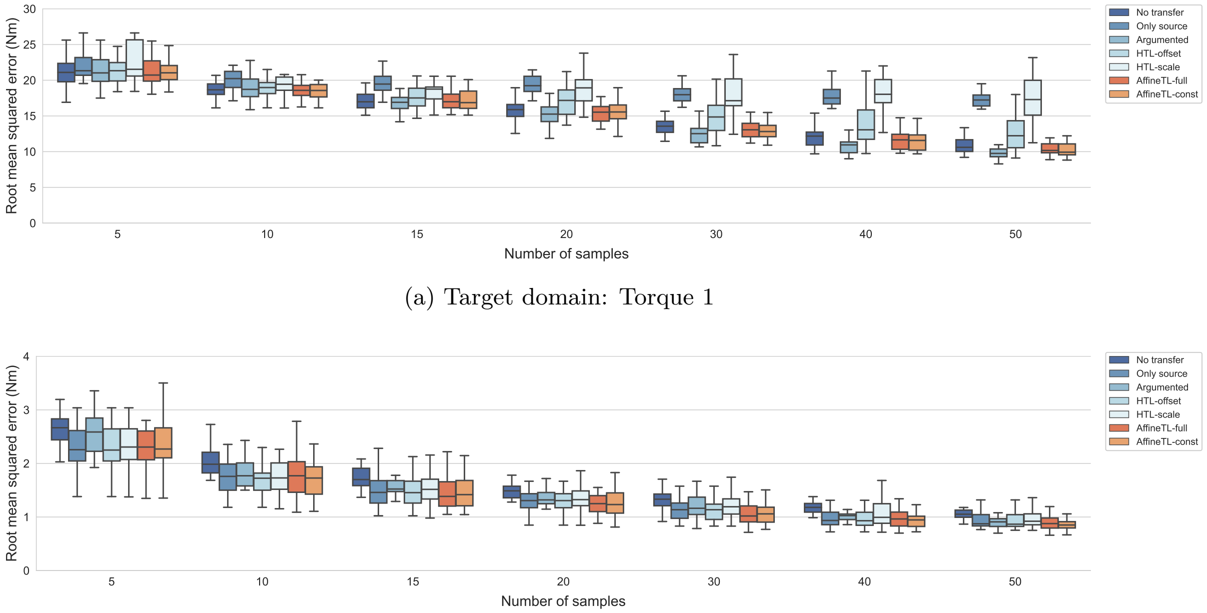
No transfer
Train a model using input x with no transfer.

Only source
Train a model $g_1(f_1)$ using only the source feature f_s as input.

Input augmentation
Perform an ordinary regression with the augmented input vector concatenating x and f_s.

HTL–offset (Kuzborskij & Orabona, 2013)
Calculate the transformed output $z_i = y_i - g_1(f_s)$ where $g_1(f_s)$ is the model pre-trained using **Only source**, and train an additional model with input x_i to predict z_i.

HTL–scale (Du et al., 2017)
Calculate the transformed output $z_i = y_i / g_1(f_s)$ where $g_1(f_s)$ is the model pre-trained using **Only source**, and train an additional model with input x_i to predict z_i.



(a) Target domain: Torque 1

(b) Target domain: Torque 7

Figure 2: Box plots showing the distribution of the root mean squared errors (RMSE) for the seven analysis procedures at (a) Torque 1 and (b) Torque 7. Each box plot showed the median and the first and third quartiles.

Affine transfer (full)
Train the model $g_1 + g_2 \cdot g_3$.

Affine transfer (constrained)
Train the model $g_1 + g_3$.

Kernel ridge regression with the radial basis function (RBF) kernel $\exp(-\|x - x'\|^2/2\ell^2)$ was used to train the model for each procedure. The scale parameter ℓ was fixed to the square root of the dimension of the input. The regularization parameter in the kernel ridge regression and $\lambda_\alpha, \lambda_\beta$ and λ_γ in the affine model transfer were selected through five-fold cross-validation.

Table 1 summarizes the prediction performance of the seven different procedures for varying numbers of training samples in the seven tasks. In most cases, the affine transfer model achieved the best prediction performance in terms of the root mean squared error (RMSE). In several other cases, direct learning without transfer produced the best results; in almost all cases, ordinary TL only using the source features and the two existing HTL models showed no advantage over the affine transfer model. In this experiment, the performance was evaluated in cases where the number of training samples was extremely small. The advantage of the affine transfer model tended to be greater in these circumstances, such as with a sample size of $n = 5$ or 10.

Figure 2 highlights the RMSE values for Torque 1 and Torque 7. The joint of Torque 1 is located closest to the root of the arm. Therefore, the learning task for Torque 1 is less relevant to those for the other joints, and the transfer from Torque 2-6 to Torque 1 would not work. In fact, as shown in Figure 2(a) and Table 1, relying only on the source features (**Only source**) failed to acquire predictive ability. In addition, **HTL–offset** and **HTL–scale** likewise showed poor prediction performance owing to the negative effect of the failure in the variable transformation using the values of the other torques. In particular, the two HTL models achieved lower predictive performance than direct learning (**No transfer**), resulting in the occurrence of negative transfer.

Torque 7 was measured at the joint closest to the end of the arm. Therefore, Torque 7 strongly depends on those at the other six joint positions, and the procedures based on the source features, including **Only**

Target	Model	Number of training samples						
		$n < d$			$n \approx d$	$n > d$		
		5	10	15	20	30	40	50
Torque 1	Direct	**21.2 ± 2.05**	19.0 ± 2.04	17.3 ± 1.66	15.8 ± 1.57	13.7 ± 1.42	12.2 ± 1.56	10.8 ± 1.11
	Only	24.9 ± 11.4	20.3 ± 1.77	19.5 ± 1.42	19.3 ± 1.35	18.3 ± 1.92	18.0 ± 1.76	17.5 ± 1.63
	Augmented	21.3 ± 2.35	18.9 ± 1.71	**16.8 ± 1.27**	**15.3 ± 1.97**	**12.7 ± 1.56**	**11.0 ± 1.58**	**9.93 ± 1.08**
	Offset	24.6 ± 11.5	19.1 ± 2.02	17.7 ± 1.58	17.1 ± 2.15	15.2 ± 3.37	14.3 ± 3.82	12.8 ± 2.97
	Scale	24.9 ± 8.53	20.3 ± 3.32	20.0 ± 7.54	19.1 ± 3.19	17.8 ± 2.83	18.5 ± 3.25	18.1 ± 4.44
	AffineTL-full	21.3 ± 2.10	18.9 ± 2.09	17.4 ± 1.93	15.6 ± 1.68	13.4 ± 1.95	11.6 ± 1.48	10.4 ± 0.925
	AffineTL-const	21.4 ± 1.91	**18.7 ± 1.91**	17.2 ± 1.38	15.7 ± 1.85	13.0 ± 1.43	11.5 ± 1.47	10.3 ± 0.994
Torque 2	Direct	15.8 ± 2.38	13.0 ± 1.42	11.5 ± 0.966	10.4 ± 0.821	9.39 ± 0.978	8.38 ± 0.767	7.72 ± 0.753
	Only	15.3 ± 2.31	13.0 ± 1.51	12.4 ± 2.22	11.9 ± 3.02	12.1 ± 7.43	10.2 ± 1.82	9.45 ± 1.99
	Augmented	15.7 ± 2.48	12.7 ± 1.47	**11.1 ± 1.33**	**9.96 ± 1.41**	**8.65 ± 1.05**	**7.65 ± 0.929**	**6.99 ± 0.615**
	Offset	15.2 ± 2.28	12.8 ± 1.62	12.0 ± 2.45	11.8 ± 3.11	11.9 ± 7.51	9.89 ± 1.87	9.12 ± 2.08
	Scale	15.2 ± 2.29	**12.6 ± 1.59**	12.1 ± 2.32	11.7 ± 3.12	11.9 ± 7.52	9.95 ± 1.86	9.15 ± 2.05
	AffineTL-full	**14.4 ± 1.60**	12.7 ± 1.82	11.5 ± 1.93	10.8 ± 1.68	9.58 ± 1.97	7.96 ± 1.04	7.52 ± 0.674
	AffineTL-const	14.6 ± 1.86	12.8 ± 1.45	11.4 ± 1.48	10.5 ± 1.64	9.39 ± 1.79	8.17 ± 1.06	7.58 ± 0.870
Torque 3	Direct	9.93 ± 1.65	8.17 ± 0.996	7.84 ± 2.60	6.97 ± 1.10	5.97 ± 0.917	5.33 ± 0.942	4.56 ± 0.401
	Only	8.99 ± 2.98	7.62 ± 2.35	6.91 ± 1.65	6.45 ± 1.20	5.66 ± 0.908	5.31 ± 0.968	4.95 ± 0.964
	Augmented	9.66 ± 1.72	7.78 ± 0.978	6.74 ± 1.01	6.25 ± 1.15	5.29 ± 1.27	**4.68 ± 1.24**	**4.03 ± 0.652**
	Offset	8.96 ± 2.98	7.48 ± 2.31	6.87 ± 1.65	6.42 ± 1.21	5.60 ± 0.844	5.23 ± 0.993	4.85 ± 1.01
	Scale	9.06 ± 2.94	7.59 ± 2.29	6.91 ± 1.42	6.69 ± 1.41	5.65 ± 0.964	5.41 ± 1.22	4.98 ± 0.888
	AffineTL-full	**8.64 ± 1.33**	**7.22 ± 1.41**	6.67 ± 1.27	6.07 ± 1.00	5.25 ± 1.17	4.89 ± 1.15	4.28 ± 0.829
	AffineTL-const	8.94 ± 1.40	7.33 ± 1.20	**6.50 ± 1.14**	**5.97 ± 0.918**	**5.18 ± 1.00**	4.76 ± 0.958	4.22 ± 0.745
Torque 4	Direct	14.2 ± 2.30	11.1 ± 2.39	9.49 ± 2.18	7.80 ± 0.978	6.91 ± 0.778	6.06 ± 0.630	5.47 ± 0.653
	Only	12.3 ± 3.60	9.23 ± 2.43	7.81 ± 1.74	6.83 ± 1.45	6.21 ± 1.02	6.19 ± 1.37	5.22 ± 0.629
	Augmented	13.8 ± 2.83	9.69 ± 1.64	8.52 ± 1.80	7.06 ± 1.03	5.97 ± 0.905	**5.16 ± 0.740**	**4.69 ± 0.698**
	Offset	12.3 ± 3.62	9.08 ± 2.35	**7.67 ± 1.68**	6.73 ± 1.40	6.14 ± 1.01	6.16 ± 1.38	5.12 ± 0.582
	Scale	12.3 ± 3.62	9.10 ± 2.36	7.72 ± 1.62	6.74 ± 1.37	6.12 ± 1.04	6.16 ± 1.38	5.14 ± 0.524
	AffineTL-full	**12.0 ± 3.11**	8.95 ± 2.05	7.89 ± 1.92	6.83 ± 1.75	**5.66 ± 1.07**	5.27 ± 1.12	4.87 ± 1.02
	AffineTL-const	12.2 ± 3.40	**8.64 ± 1.95**	7.87 ± 1.87	**6.44 ± 1.09**	5.67 ± 1.12	5.25 ± 1.06	4.78 ± 0.665
Torque 5	Direct	1.08 ± 0.169	0.986 ± 0.0901	0.932 ± 0.165	0.860 ± 0.127	0.737 ± 0.123	0.686 ± 0.0937	**0.608 ± 0.0705**
	Only	1.11 ± 0.155	1.01 ± 0.0894	1.02 ± 0.146	0.964 ± 0.148	0.846 ± 0.125	0.797 ± 0.111	0.739 ± 0.103
	Augmented	1.08 ± 0.160	0.985 ± 0.0898	0.895 ± 0.125	**0.849 ± 0.135**	**0.737 ± 0.129**	**0.679 ± 0.102**	0.623 ± 0.110
	Offset	1.11 ± 0.173	0.998 ± 0.0949	0.982 ± 0.163	0.944 ± 0.152	0.806 ± 0.113	0.738 ± 0.110	0.693 ± 0.0987
	Scale	1.15 ± 0.248	0.993 ± 0.0933	0.970 ± 0.151	0.939 ± 0.124	0.806 ± 0.0966	0.754 ± 0.0842	0.776 ± 0.211
	AffineTL-full	1.03 ± 0.121	**0.935 ± 0.126**	**0.878 ± 0.129**	0.862 ± 0.129	0.762 ± 0.144	0.726 ± 0.117	0.635 ± 0.068
	AffineTL-const	1.04 ± 0.114	0.971 ± 0.0999	0.897 ± 0.113	0.888 ± 0.129	0.739 ± 0.122	0.702 ± 0.092	0.629 ± 0.0672
Torque 6	Direct	1.86 ± 0.246	1.67 ± 0.194	1.50 ± 0.167	1.36 ± 0.156	1.21 ± 0.143	1.11 ± 0.088	1.07 ± 0.0969
	Only	1.95 ± 0.250	1.88 ± 0.407	1.79 ± 0.206	1.80 ± 0.378	1.61 ± 0.216	1.58 ± 0.173	1.55 ± 0.200
	Augmented	1.84 ± 0.171	1.65 ± 0.200	**1.48 ± 0.183**	**1.33 ± 0.207**	**1.17 ± 0.200**	**1.03 ± 0.117**	**0.964 ± 0.115**
	Offset	1.92 ± 0.257	1.84 ± 0.426	1.72 ± 0.262	1.71 ± 0.421	1.44 ± 0.271	1.39 ± 0.245	1.39 ± 0.289
	Scale	1.91 ± 0.256	1.89 ± 0.425	1.81 ± 0.326	1.84 ± 0.398	1.68 ± 0.300	1.59 ± 0.248	1.59 ± 0.242
	AffineTL-full	**1.82 ± 0.229**	**1.64 ± 0.191**	1.58 ± 0.224	1.41 ± 0.248	1.24 ± 0.212	1.13 ± 0.307	0.996 ± 0.0963
	AffineTL-const	1.86 ± 0.202	1.7 ± 0.179	1.55 ± 0.275	1.45 ± 0.276	1.23 ± 0.209	1.09 ± 0.141	1.02 ± 0.0923
Torque 7	Direct	2.67 ± 0.321	2.12 ± 0.420	1.84 ± 0.421	1.53 ± 0.305	1.34 ± 0.203	1.17 ± 0.126	1.05 ± 0.0960
	Only	2.29 ± 0.583	1.76 ± 0.441	1.55 ± 0.407	1.42 ± 0.585	1.16 ± 0.243	0.999 ± 0.231	0.942 ± 0.164
	Augmented	2.55 ± 0.408	1.90 ± 0.433	1.68 ± 0.417	1.39 ± 0.367	1.20 ± 0.236	1.01 ± 0.142	0.901 ± 0.112
	Offset	2.29 ± 0.588	1.71 ± 0.405	1.55 ± 0.408	1.41 ± 0.586	1.15 ± 0.249	0.995 ± 0.233	0.935 ± 0.167
	Scale	2.32 ± 0.58	1.75 ± 0.428	1.59 ± 0.395	1.42 ± 0.569	1.21 ± 0.249	1.06 ± 0.249	0.967 ± 0.161
	AffineTL-full	**2.29 ± 0.533**	1.75 ± 0.447	**1.49 ± 0.380**	1.30 ± 0.327	1.07 ± 0.250	0.975 ± 0.180	0.889 ± 0.145
	AffineTL-const	2.32 ± 0.552	**1.71 ± 0.419**	1.49 ± 0.373	**1.26 ± 0.257**	**1.06 ± 0.220**	**0.950 ± 0.163**	**0.885 ± 0.156**

Table 1: Performance on predicting the torque values at seven different joints of the SARCOS anthropomorphic robot arm. The mean and standard deviation of the root mean square error with respect to 20 test sets are reported for varying numbers of training samples and the seven different tasks. Seven different methods were tested: **No transfer** (Direct), **Only source** (Only), **Input augmentation** (Augmented), **HTL-offset** (Offset), **HTL-scale** (Scale), **Affine transfer (full)** (AffineTL-full), and **Affine transfer (constrained)** (AffineTL-const), respectively.

source, were more effective than in the other tasks. In particular, the affine model transfer achieved the best performance among the other methods. This is consistent with the theoretical result that the transfer capability of the affine model transfer could be further improved when the risk of learning using only the source features is sufficiently small.

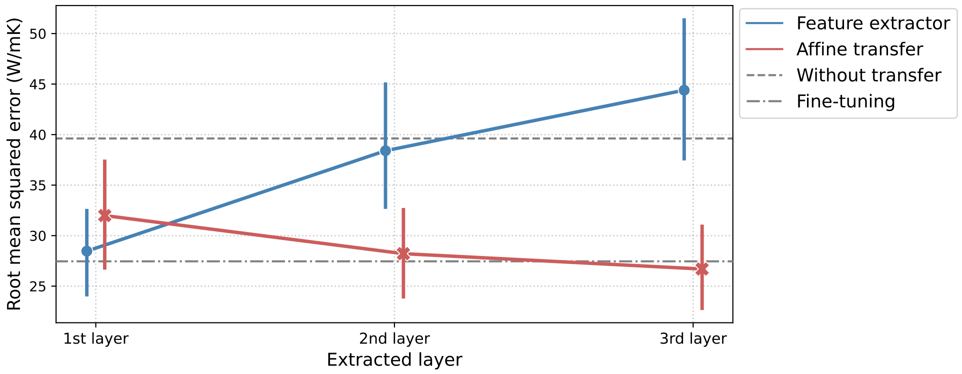


Figure 3: Change of RMSE values between the affine transfer model and the ordinary feature extractor when using different levels of intermediate layers as the source features. The line plot shows the mean and 95% confidence interval. As a baseline, RMSE values for direct learning without transfer and fine-tuned neural networks are shown as dotted and dashed lines, respectively.

5.2 Lattice thermal conductivity of inorganic crystals

Here, we describe the relationship between the qualitative differences in source features and the learning behavior of the affine model transfer, in contrast to ordinary feature extractors using neural networks. The target task is to predict the lattice thermal conductivity (LTC) of inorganic crystalline materials, where the LTC is the amount of vibrational energy propagated by phonons in a crystal. In general, LTC can be calculated ab initio by performing many-body electronic structure calculations based on quantum mechanics. However, it is quite time-consuming to perform the first-principles calculations for thousands of crystals, which will be used as a training sample set to create a surrogate statistical model. Therefore, we perform TL with the source task of predicting an alternative, computationally tractable physical property called scattering phase space (SPS), which is known to be physically related to LTC.

We used the dataset from Ju et al. (2021) that records SPS and LTC for 320 and 45 inorganic compounds, respectively. The input compounds were translated to 290-dimensional compositional descriptors using XenonPy (Liu et al., 2021)[1]. For the preliminary step, neural networks with three hidden layers that predict SPS were trained using 80% of the 320 samples. 100 models with different numbers of neurons were randomly generated and the top 10 source models that showed the highest generalization performance in the source domain were selected. Then, in the target task, an intermediate layer of a source model was used as the feature extractor. A model was trained using 40 randomly chosen samples of LTC, and its performance was evaluated with the remaining 5 samples. For each of the 10 source models, we performed the training and testing 10 times with different sample partitions and compared the mean values of RMSE among four different methods: (i) the affine model transfer using neural networks to model the three functions g_1, g_2 and g_3, (ii) a neural network using the XenonPy compositional descriptors as input without transfer, (iii) a neural network using the source features as input, and (iv) fine-tuning of the pre-trained neural networks. The width of the layers of each neural network, the number of training epochs, and the dropout rate were optimized during five-fold cross-validation looped within each training set. The modeling and learning procedures are detailed in Appendix D.2.

Figure 3 shows the change in prediction performance of TL models using source features obtained from different intermediate layers from the first to the third layers. The affine transfer model and the ordinary feature extractor showed opposite patterns. The performance of the feature extractor improved when the first intermediate layer closest to the input layer was used as the source features and gradually degraded when layers closer to the output were used. When the third intermediate layer was used, a negative transfer occurred in the feature extractor as its performance became worse than that of the direct learning. In contrast, the affine transfer model performs better as the second and third intermediate layers closer to the output

[1] `https://github.com/yoshida-lab/XenonPy`

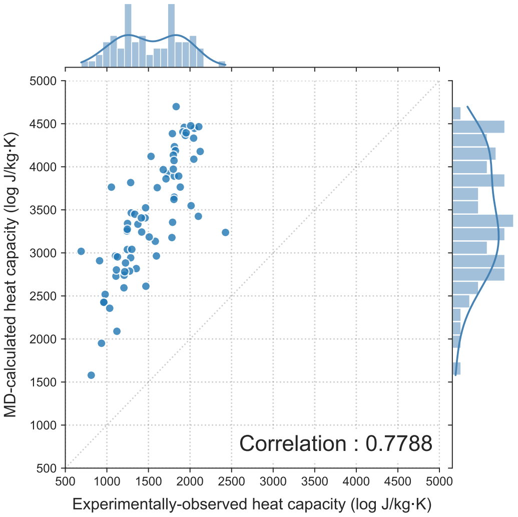


Figure 4: MD-calculated (vertical axis) and experimental values (horizontal axis) of the specific heat capacity at constant pressure for various amorphous polymers.

were used. The affine transfer model using the third intermediate layer reached a level of accuracy slightly better than fine-tuning, which intuitively uses more information to transfer than the extracted features.

In general, the features encoded in an intermediate layer of a neural network are more task-independent as the layer is closer to the input, and the features are more task-specific as the layer is closer to the output (Yosinski et al., 2014). Because the first layer does not differ much from the original input, using both x and f_s in the affine model transfer does not contribute much to performance improvement. However, when using the second and third layers as the feature extractors, the use of both x and f_s contributes to improving the expressive power of the model, because the feature extractors have acquired different representational capabilities from the original input. In contrast, a model based only on f_s from a source task-specific feature extractor could not account for data in the target domain, so its performance would become worse than direct learning without transfer, i.e., a negative transfer would occur.

5.3 Heat capacity of organic polymers

We highlight the benefits of separately modeling and estimating domain-specific factors through a case study in polymer chemistry. The objective is to predict the specific heat capacity at constant pressure C_{P} of any given organic polymer with its chemical structure in the repeating unit. Specifically, we conduct TL to bridge the gap between experimental values and physical properties calculated from molecular dynamics (MD) simulations.

For the target domain, experimental values of C_{P} ($C_{\mathrm{P}}^{\mathrm{exp}}$) for 70 amorphous polymers were obtained from the polymer properties database PoLyInfo (Otsuka et al., 2011). For the source domain, the properties ($C_{\mathrm{P}}^{\mathrm{MD}}$) for 1,077 polymers were calculated from all-atom classical MD simulations in Hayashi et al. (2022). As shown in Figure 4, there was a large systematic bias between experimental and calculated values; the MD-calculated properties $C_{\mathrm{P}}^{\mathrm{MD}}$ exhibited an evident overestimation with respect to their experimental values. As discussed in Hayashi et al. (2022), this observation is inevitable because classical MD calculations do not reflect the presence of quantum effects in the real system: the vibrational energy of the classical harmonic oscillator is significantly larger than that of the quantum harmonic oscillator at the same frequency. Hence, the upward bias was observed in $C_{\mathrm{P}}^{\mathrm{MD}}$, which mainly originated from the lack of quantum effects. According to Einstein's theory for the specific heat in physical chemistry, the logarithmic ratio between $C_{\mathrm{P}}^{\mathrm{exp}}$ and $C_{\mathrm{P}}^{\mathrm{MD}}$

Table 2: Force field parameters that form the General AMBER force field (Wang et al., 2004) version 2 (GAFF2), and their detailed descriptions.

Parameter	Description
mass	Atomic mass
σ	Equilibrium radius of van der Waals (vdW) interactions
ϵ	Depth of the potential well of vdW interactions
charge	Atomic charge of Gasteiger model
r_0	Equilibrium length of chemical bonds
K_{bond}	Force constant of bond stretching
polar	Bond polarization defined by the absolute value of charge difference between atoms in a bond
θ_0	Equilibrium angle of bond angles
K_{angle}	Force constant of bond bending
K_{dih}	Rotation barrier height of dihedral angles

Table 3: Comparison of three prediction models for experimental values of specific heat capacity with and without using the MD-calculated properties as source features (mean and standard deviation of RMSE).

Model	RMSE (log J/kg · K)
$y = \alpha_0 + \alpha_1 f_s + \epsilon$	0.1392 ± 0.04631
$y = f_s + x^\top \gamma + \epsilon$	0.1368 ± 0.04265
$y = \alpha_0 + \alpha_1 f_s - (\beta f_s + 1)x^\top \gamma + \epsilon$	**0.1357 ± 0.04173**

can be calibrated by the following equation (Hayashi et al., 2022):

$$\log C_{\text{P}}^{\text{exp}} = \log C_{\text{P}}^{\text{MD}} + 2\log\left(\frac{\hbar\omega}{k_B T}\right) + \log \frac{\exp\left(\frac{\hbar\omega}{k_B T}\right)}{\left[\exp\left(\frac{\hbar\omega}{k_B T}\right) - 1\right]^2}. \tag{7}$$

where k_B is the Boltzmann constant, $\hbar$ is the Planck constant, ω is the frequency of molecular vibrations, and T is the temperature. The bias is a monotonically decreasing function of frequency ω, which is described as a black-box function of polymers with their molecular features. Hereafter, we consider the calibration of this systematic bias using the affine transfer model.

The log-transformed value of $C_{\text{p}}^{\text{exp}}$ is modeled as

$$y := \log C_{\text{P}}^{\text{exp}} = \underbrace{\alpha_0 + \alpha_1 f_s}_{g_1} - \underbrace{(\beta f_s + 1)}_{g_2} \cdot \underbrace{x^\top \gamma}_{g_3} + \epsilon, \tag{8}$$

where ϵ represents observation noise, and $\alpha_0, \alpha_1, \beta$ and γ are the unknown parameters to be estimated. The input x was given as a 190-dimensional descriptor vector, called the force field descriptor, wihch encodes compositional, structural and physicochemical features of the chemical structure for a given polymer's repeating unit as detailed below. The source feature f_s was given as the log-transformed value of C_{P}^{MD}. Therefore, f_s is no longer a function of x; this modeling was intended for calibrating the MD-calculated properties rather than for conventional TL. When $\alpha_1 = 1$ and $\beta = 0$, Eq. (8) is consistent with the theoretical equation in Eq. (7) in which the quantum effect is linearly modeled as $\alpha_0 + x^\top \gamma$. In addition, for reference, we estimated two partially constrained models: $y = \alpha_0 + \alpha_1 f_s + \epsilon$ and $y = f_s + x^\top \gamma + \epsilon$. The former uses the MD-calculated properties alone. The latter corresponds to the theoretical model of Eq. (7) and is consistent with the full model in Eq .(8) in which $\alpha_1 = 1$ and $\beta = 0$. Of 70 samples for which both MD-calculated and experimental values of C_{P} were available, 60 samples were randomly sampled, and 20 predictive models were trained with different sample splits.

The descriptor x represents the distribution of the ten different force field parameters ($t \in T = \{\text{mass}, \sigma, \epsilon, \text{charge}, r_0, K_{\text{bond}}, \text{polar}, \theta_0, K_{\text{angle}}, K_{\text{dih}}\}$ that make up the empirical potential (i.e., the General AMBER force field (Wang et al., 2004) version 2 (GAFF2)) of the classical MD simulation. The detailed descriptions for each parameter are listed in Table 2. For each t, pre-defined values are assigned to their

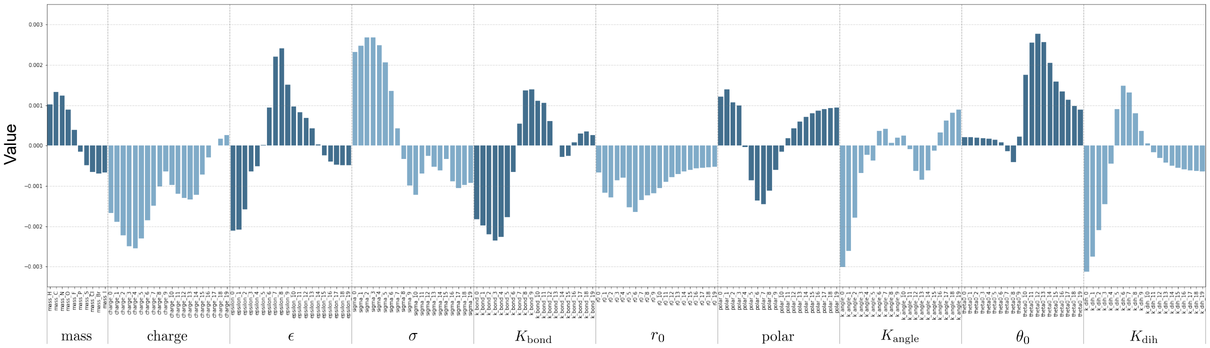


Figure 5: Bar plot of regression coefficients γ of linear calibrator filling the discrepancy between experimental and MD-calculated specific heat capacity of amorphous polymers.

constituent elements in a polymer, such as individual atoms (mass, charge, σ, and ϵ), bonds (r_0, K_{bond}, and polar), angles (θ_0 and K_{angle}), or dihedral angles (K_{dih}), respectively. The probability density function of the assigned values of t is then estimated and discretized into 10 points corresponding to 10 different element species such as hydrogen and carbon for mass, and 20 equally spaced grid points for the other parameters. Thus, the 190-dimensional descriptor vector x consists of nine blocks with 10 or 20 elements corresponding to the different types of force field parameters.

For each block in x describing the discretized density function of a force field parameter, the corresponding parameters in γ should be estimated smoothly. To this end, fused regularization was introduced in the objective function to be minimized:

$$\lambda_1 \|\gamma\|_2^2 + \lambda_2 \sum_{t \in T} \sum_{j=2}^{J_t} \left(\gamma_{t,j} - \gamma_{t,j-1} \right)^2,$$

where $J_t = 10$ for $t =$ mass and $J_t = 20$ otherwise. $\gamma_{t,j}$ denotes the coefficient for the j-th grid point of the force field parameter t. The hyperparameters λ_1 and λ_2 respectively penalize the increasing norm of γ and the increasing discrepancy between the regression coefficients of adjacent grid points. Both λ_1 and λ_2 were optimized through five-fold cross-validation.

Table 3 summarizes the prediction performance (RMSE) of the three models. The ordinary linear model $y = \alpha_0 + \alpha_1 f_s + \epsilon$, which ignores the force-field descriptors, exhibited the lowest prediction performance. The other two calibration models $y = f_s + x^\top \gamma + \epsilon$ and the full model in Eq. (8) reached almost the same accuracy, but the latter had achieved slightly better prediction accuracy. The estimated parameters of the full model were $\alpha_1 \approx 0.889$ and $\beta \approx -0.004$. The model form is highly consistent with the theoretical equation in Eq. (7) as well as the restricted model ($\alpha_1 = 1, \beta = 0$). This supports the validity of the theoretical model in Hayashi et al. (2022) that explains the discrepancy between experimental and calculated values owing to the presence or absence of quantum effects.

It is expected that physicochemical insights can be obtained by examining the estimated coefficient parameter γ in the term of $x^\top \gamma$, which would capture the contribution of the force field parameters to the quantum effects yielding the systematic bias in the MD calculations. Figure 5 shows the mean values of the estimated parameter γ for the full calibration model. The magnitude of the quantum effect is a monotonically increasing function of the frequency of the harmonic oscillator ω, and is known to be highly related to the depth of the potential well in van der Waals interaction (ϵ) and in bond rotation (K_{dih}), the force constants of bond-stretching (K_{bond}) and -bending (K_{angle}), and the mass of the atoms (mass). According to physicochemical intuition, it is considered that as ϵ, K_{bond}, K_{angle}, and K_{dih} decrease, their potential energy surface becomes shallow. This leads to the decrease of the frequency ω, resulting in the decrease of quantum effects for C_{P}. Further, theoretically, because the molecular vibration of light-weight atoms is faster than that of heavy atoms, ω and quantum effects for C_{P} should increase with decreasing mass. These physical relationships could be captured consistently with the estimated coefficients. The coefficients in lower regions of ϵ, K_{bond}, K_{angle}

and K_{dih} showed large negative values, indicating that polymers containing more atoms, bonds, angles, and dihedral angles with lower values of these force field parameters will have smaller quantum effects. Conversely, the coefficients in lower regions of mass showed positive large values, meaning that polymers containing more atoms with smaller masses will have larger quantum effects. Separately including the domain-common and domain-specific factors in the transfer model, we could infer the features relevant to the cross-domain differences.

6 Conclusions

In this study, we defined a general class of TL based on affine model transformations and clarified their learning capability and applicability. We considered a procedure consisting of two stages: first, the source features and target samples are given and transformed, and then the domain transfer model is estimated using the transformed data. The affine model transformation was shown to minimize the expected squared loss in the class of two-stage transfer learning. The affine transfer model is structurally common to a low-rank tensor regression model and an invertible neural network model with affine-coupling layers. In the context of TL, the model can be used to represent and estimate the cross-domain shift and domain-specific factors simultaneously and separately.

Currently, the most widely applied methods of TL reuse features acquired by pre-trained neural networks in the source domain. Such procedural approaches, including feature extractors and fine-tuning, are intuitively plausible but lack a theoretical foundation. In addition, existing methods are designed to describe the target domain using only features acquired in the source domain, and thus cannot adequately deal with cases where domain-specific factors are present. Our affine model transfer is a principled methodology based on the minimum expected square loss. It also has the ability to handle domain common and unique factors simultaneously and separately.

The present methodology provides a general framework that can handle any model including neural networks and any pre-defined features. As described, using intermediate layers of a pre-trained neural network as the source features in the affine transfer model, we can represent ordinary TL based on feature extraction. Furthermore, as shown in the case studies, the affine transfer model can be used as a calibrator between computational models and real-world systems by defining predicted values from physics simulators as the source features. By designing the source features and the three coupling functions that make up the optimal form of transfer models, we expect to be able to formulate various kinds of TL.

References

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, 33:1497–1537, 2005.

Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *ArXiv*, abs/1702.05374, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *ArXiv*, abs/1410.8516, 2014.

Laurent Dinh, Jascha Narain Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *International Conference on Learning Representations*, 2017.

Simon S Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. Hypothesis transfer learning via transformation functions. *Advances in Neural Information Processing Systems*, 30, 2017.

Richard A. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-model factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

Yoshihiro Hayashi, Junichiro Shiomi, Junko Morikawa, and Ryo Yoshida. Radonpy: Automated physical property calculation using all-atom classical molecular dynamics simulations for polymer informatics. *ArXiv*, abs/2203.14090, 2022.

Sheng Ju, Ryo Yoshida, Chang Liu, Kenta Hongo, Terumasa Tadano, and Junichiro Shiomi. Exploring diamond-like lattice thermal conductivity crystals via feature-based transfer learning. *Physical Review Materials*, 5(5):053801, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*, 31, 2018.

Marius Kloft and Gilles Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. *Advances in Neural Information Processing Systems*, 2011.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pp. 942–950. PMLR, 2013.

Ilja Kuzborskij and Francesco Orabona. Fast rates by transferring from auxiliary hypotheses. *Machine Learning*, 106(2):171–195, 2017.

Chang Liu, Erina Fujita, Yukari Katsura, Yuki Inada, Asuka Ishikawa, Ryuji Tamura, Kaoru Kimura, and Ryo Yoshida. Machine learning to predict quasicrystals from chemical compositions. *Advanced Materials*, 33(36):2102507, 2021.

Song Liu and Kenji Fukumizu. Estimating posterior ratio for classification: Transfer learning from probabilistic perspective. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 747–755. SIAM, 2016.

Shunya Minami, Song Liu, Stephen Wu, Kenji Fukumizu, and Ryo Yoshida. A general class of transfer learning regression without implementation cost. *Proceedings of AAAI Conference on Artificial Intelligence*, 35:8992–8999, 2021.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet S. Talwalkar. *Foundations of Machine Learning.* MIT press, 2018.

Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. Polyinfo: Polymer database for polymeric materials design. In *2011 International Conference on Emerging Intelligent Data and Web Technologies*, pp. 22–29. IEEE, 2011.

George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22:57:1–57:64, 2021.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, 2019.

Rahul Kumar Sevakula, Vikas Singh, Nishchal Kumar Verma, Chandan Kumar, and Yan Cui. Transfer learning for molecular cancer classification using deep neural networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16:2089–2100, 2019.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. *Advances in Neural Information Processing Systems*, 23, 2010.

Ingo Steinwart and Andreas Christmann. Support vector machines. In *Information science and statistics*, 2008.

Ingo Steinwart, Don R. Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pp. 79—-93, 2009.

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33:7852–7862, 2020.

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *Journal of Computational Chemistry*, 25, 2004.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Stephen Wu, Yukiko Kondo, Masa-aki Kakimoto, Bin Yang, Hironao Yamada, Isao Kuwajima, Guillaume Lambard, Kenta Hongo, Yibin Xu, Junichiro Shiomi, et al. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *npj Computational Materials*, 5(1): 1–11, 2019.

Hironao Yamada, Chang Liu, Stephen Wu, Yukinori Koyama, Sheng Ju, Junichiro Shiomi, Junko Morikawa, and Ryo Yoshida. Predicting materials properties with little data using shotgun transfer learning. *ACS Central Science*, 5:1717–1730, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27, 2014.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

A Other perspectives on affine model transfer

A.1 Transformation functions for general loss functions

Here we discuss the optimal transformation function for general loss functions.

Let $\ell(y, y') \geq 0$ be a convex loss function that returns zero if and only if $y = y'$, and let $g^*(x)$ be the optimal predictor that minimizes the expectation of ℓ with respect to the distribution p_t followed by x and y transformed by ϕ:

$$g^*(x) = \arg\min_g \mathbb{E}_{p_t}\big[\ell(g(x), \phi_{f_s}(y))\big].$$

The function g that minimizes the expected loss

$$\mathbb{E}_{p_t}\big[\ell(g(x), \phi_{f_s}(y))\big] = \iint \ell(g(x), \phi_{f_s}(y)) p_t(x, y) \mathrm{d}x\mathrm{d}y$$

should be a solution to the Euler-Lagrange equation

$$\frac{\partial}{\partial g(x)} \int \ell(g(x), \phi_{f_s}(y)) p_t(x, y) \mathrm{d}y = \int \frac{\partial}{\partial g(x)} \ell(g(x), \phi_{f_s}(y)) p_t(y|x) \mathrm{d}y \, p_t(x) = 0. \tag{9}$$

Denote the solution of Eq. (9) by $G(x; \phi_{f_s})$. While G depends on the loss ℓ and distribution p_t, we omit those from the argument for notational simplicity. Using this function, the minimizer of the expected loss $\mathbb{E}_{x,y}[\ell(g(x), y)]$ can be expressed as $G(x; \mathrm{id})$, where id represents the identity function.

Here, we consider the following assumption to hold, which generalizes Assumption 3:

Assumption 3(b). *For any distribution on the target domain $p_t(x, y)$ and all $x \in \mathcal{X}$, the following relationship holds:*

$$\psi_{f_s}(g^*(x)) = \arg\min_g \mathbb{E}_{x,y}[\ell(g(x), y)].$$

This implies that the transformation functions ϕ_{f_s} and ψ_{f_s} satisfy

$$\psi_{f_s}\big(G(x; \phi_{f_s})\big) = G(x; \mathrm{id}). \tag{10}$$

Assumption 3(b) states that if the best predictor $G(x; \phi_{f_s})$ for the data transformed by ϕ is given to the model transformation function ψ, it is consistent with the overall best predictor $G(x; \mathrm{id})$ in the target region in terms of the loss function ℓ. We consider all pairs of ψ and ϕ that satisfy this consistency condition.

Here, let us consider the following proposition:

Proposition 5. *Under Assumption 1-2 and 3(b), it holds that $\psi_{f_s}^{-1} = \phi_{f_s}$.*

Proof. The proof is analogous to that of Theorem 1. For any $y_0 \in \mathcal{Y}$, let $p_t(y|x) = \delta_{y_0}$. Combining this with Eq. (9) leads to

$$\frac{\partial}{\partial g(x)} \ell(g(x), \phi_{f_s}(y_0)) = 0 \ (\forall y_0 \in \mathcal{Y}).$$

Because $\ell(y, y')$ returns the minimum value zero if and only if $y = y'$, we obtain $G(x; \phi_{f_s}) = \phi_{f_s}(y_0)$. Similarly, we have $G(x; \mathrm{id}) = y_0$. From these two facts and Assumption 3(b), we have $\psi_{f_s}(\phi_{f_s}(y_0)) = y_0$, proving that the proposition is true. □

Proposition 5 indicates that the first statement of Theorem 1 holds for general loss functions. However, the second claim of Theorem 1 generally depends on the type of loss function. Through the following examples, we describe the optimal class of transformation functions for several loss functions.

Example 1 (Squared loss)**.** Let $\ell(y, y') = |y - y'|^2$. As a solution of Eq. (9), we can see that the optimal predictor is the conditional expectation $\mathbb{E}_{p_t}[\phi_{f_s}(Y)|X = x]$. As discussed in Section 2, the transformation functions ϕ_{f_s} and ψ_{f_s} should be affine transformations.

Example 2 (Absolute loss)**.** Let $\ell(y, y') = |y - y'|$. Substituting this into Eq. (9), we have

$$\begin{aligned}
0 &= \int \frac{\partial}{\partial g(x)} \big|g(x) - \phi_{f_s}(y)\big| p_t(y|x) \mathrm{d}y \\
&= \int \operatorname{sign}\big(g(x) - \phi_{f_s}(y)\big) p_t(y|x) \mathrm{d}y \\
&= \int_{\phi_{f_s}(y) \geq g(x)} p_t(y|x) \mathrm{d}y - \int_{\phi_{f_s}(y) < g(x)} p_t(y|x) \mathrm{d}y.
\end{aligned}$$

Assuming that ϕ_{f_s} is monotonically increasing, we have

$$0 = \int_{y \geq \phi_{f_s}^{-1}(g(x))} p_t(y|x) \mathrm{d}y - \int_{y < \phi_{f_s}^{-1}(g(x))} p_t(y|x) \mathrm{d}y.$$

This yields,

$$\int_{\phi_{f_s}^{-1}(g(x))}^{\infty} p_t(y|x) \mathrm{d}y = \int_{-\infty}^{\phi_{f_s}^{-1}(g(x))} p_t(y|x) \mathrm{d}y.$$

The same result is obtained even if ϕ_{f_s} is monotonically decreasing. Consequently,

$$\phi_{f_s}^{-1}(g(x)) = \mathrm{Median}[Y|X = x],$$

which results in

$$G(x; \phi_{f_s}) = \phi_{f_s}\big(\mathrm{Median}[Y|X = x]\big).$$

This implies that Eq. (10) holds for any ϕ_{f_s} including an affine transformation.

Example 3 (Exponential-squared loss)**.** As an example where the affine transformation is not optimal, consider the loss function $\ell(y, y') = |e^y - e^{y'}|^2$. Substituting this into Eq. (9), we have

$$\begin{aligned}
0 &= \int \frac{\partial}{\partial g(x)} \big|\exp(g(x)) - \exp(\phi_{f_s}(y))\big|^2 p_t(y|x) \mathrm{d}y \\
&= 2 \exp(g(x)) \int \big(\exp(g(x)) - \exp(\phi_{f_s}(y))\big) p_t(y|x) \mathrm{d}y.
\end{aligned}$$

Therefore,

$$G(x; \phi_{f_s}) = \log \mathbb{E}\big[\exp(\phi_{f_s}(Y))|X = x\big].$$

Consequently, Eq. (10) becomes

$$\log \mathbb{E}\big[\exp(\phi_{f_s}(Y))\big] = \phi_{f_s}\big(\log \mathbb{E}\big[\exp(Y)\big]\big).$$

If ϕ_{f_s} is an affine transformation, this equation does not generally hold.

A.2 Analysis of the optimal function class based on the upper bound of the estimation error

Here, we discuss the optimal class for the transformation function based on the upper bound of the estimation error.

In addition to Assumptions 1 and 2, we assume the following in place of Assumption 3:

Assumption 6. *The transformation functions ϕ and ψ are Lipschitz continuous with respect to the first argument, i.e., there exist constants μ_ϕ and μ_ψ such that,*

$$\phi(a, c) - \phi(a', c) \leq \mu_\phi \|a - a'\|_2, \quad \psi(b, c) - \psi(b', c) \leq \mu_\psi \|b - b'\|_2,$$

for any $a, a' \in \mathcal{Y}$ and $b, b' \in \mathbb{R}$ with any given $c \in \mathcal{F}_s$.

Note that each Lipschitz constant is a function of the second argument f_s, i.e., $\mu_\phi = \mu_\phi(f_s)$ and $\mu_\psi = \mu_\psi(f_s)$. Under Assumptions 1-2 and 6, the estimation error is upper bounded as follows:

$$\begin{aligned}
\mathop{\mathbb{E}}_{x,y}\Big[|f_t(x) - \hat{f}_t(x)|^2\Big] &= \mathop{\mathbb{E}}_{x,y}\Big[\big|\psi(g(x), f_s(x)) - \psi(\hat{g}(x), f_s(x))\big|^2\Big] \\
&\leq \mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big|g(x) - \hat{g}(x)\big|^2\Big] \\
&\leq 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big(\big|g(x) - \phi(f_t(x), f_s(x))\big|^2 \\
&\qquad\qquad + \big|\phi(f_t(x), f_s(x)) - \phi(y, f_s(x))\big|^2 \\
&\qquad\qquad + \big|\phi(y, f_s(x)) - \hat{g}(x)\big|^2\big)\Big] \\
&\leq 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big|\psi^{-1}(f_t(x), f_s(x)) - \phi(f_t(x), f_s(x))\big|^2\Big] \\
&\qquad\qquad + 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\mu_\phi(f_s(x))^2\big|f_t(x) - y\big|^2\Big] \\
&\qquad\qquad + 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big|z - \hat{g}(x)\big|^2\Big] \\
&= 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big|\psi^{-1}(f_t(x), f_s(x)) - \phi(f_t(x), f_s(x))\big|^2\Big] \\
&\qquad\qquad + 3\sigma^2\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\mu_\phi(f_s(x))^2\Big] \\
&\qquad\qquad + 3\mathop{\mathbb{E}}_{x,y}\Big[\mu_\psi(f_s(x))^2\big|z - \hat{g}(x)\big|^2\Big].
\end{aligned}$$

The derivation of this inequality is based on Du et al. (2017). We use the Lipschitz property of ψ and ϕ for the first and third inequalities, and the second inequality comes from the numeric inequality $(a-d)^2 \leq 3(a-b)^2 + 3(b-c)^2 + 3(c-d)^2$ for $a, b, c, d \in \mathbb{R}$.

According to this inequality, the upper bound of the estimation error is decomposed into three terms: the discrepancy between the two transformation functions, the variance of the noise, and the estimation error for the transformed data. Although it is intractable to find the optimal solution of ϕ, ψ, $\hat{g}$ that minimizes all these terms together, it is possible to find a solution that minimizes the first and second terms expressed as the functions of ϕ and ψ only. Obviously, the first term, which represents the discrepancy between the two transformation functions, reaches its minimum (zero) when $\phi_{f_s} = \psi_{f_s}^{-1}$. The second term, which is related to the variance of the noise, is minimized when the differential coefficient $\frac{\partial}{\partial u}\psi_{f_s}(u)$ is a constant, i.e., when ψ_{f_s} is a linear function. This is verified as follows. From $\phi_{f_s} = \psi_{f_s}^{-1}$ and the continuity of ψ_{f_s}, it follows that

$$\mu_\psi = \max\Big|\frac{\partial}{\partial u}\psi_{f_s}(u)\Big|, \quad \mu_\phi = \max\Big|\frac{\partial}{\partial u}\psi_{f_s}^{-1}(u)\Big| = \frac{1}{\min|\frac{\partial}{\partial u}\psi_{f_s}(u)|},$$

and thus the product $\mu_\phi\mu_\psi$ takes the minimum value (one) when the maximum and minimum of the differential coefficient are the same. Therefore, we can write

$$\phi(y, f_s(x)) = \frac{y - g_1(f_s(x))}{g_2(f_s(x))}, \quad \psi(g(x), f_s(x)) = g_1(f_s(x)) + g_2(f_s(x))g(x),$$

where $g_1, g_2 : \mathcal{F}_s \to \mathbb{R}$ are arbitrarily functions. Thus, the minimization of the third term in the upper bound of the estimation error can be expressed as

$$\min_{g_1, g_2, g}\ \mathop{\mathbb{E}}_{x,y}|y - g_1(f_s(x)) + g_2(f_s(x))g(x)|^2.$$

As a result, the suboptimal function class for the upper bound of the estimated function is given as

$$\mathcal{H} = \big\{x \mapsto g_1(f_s(x)) + g_2(f_s(x)) \cdot g_3(x) \mid g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2, g_3 \in \mathcal{G}_3\big\}.$$

This is the same function class derived in Section 2.

B Proofs

B.1 Proof of Theorem 2

To bound the generalization error, we use the empirical and population Rademacher complexity $\hat{\mathfrak{R}}_S(\mathcal{F})$ and $\mathfrak{R}(\mathcal{F})$ of hypothesis class $\mathcal{F}$, defined as:

$$\hat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_\sigma \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i), \qquad \mathfrak{R}(\mathcal{F}) = \mathbb{E}_S \hat{\mathfrak{R}}_S(\mathcal{F}),$$

where $\{\sigma_i\}_{i=1}^n$ is a set of Rademacher variables that are independently distributed and each take one of the values in $\{-1,+1\}$ with equal probability, and S denotes a set of samples. The following proof is based on the one of Theorem 11 shown in Kuzborskij & Orabona (2017).

Proof of Theorem 2. For any hypothesis class $\mathcal{F}$ with feature map Φ where $\|\Phi\|^2 \le 1$, the following inequality holds:

$$\mathbb{E}_\sigma \sup_{\|\theta\|^2\le\Lambda} \frac{1}{n}\sum_{i=1}^n \sigma_i\langle\theta, \Phi(x_i)\rangle \le \sqrt{\frac{\Lambda}{n}}.$$

The proof is given, for example, in Theorem 6.12 of Mohri et al. (2018). Thus, the empirical Rademacher complexity of $\mathcal{H}$ is bounded as

$$\begin{aligned}
\hat{\mathfrak{R}}_S(\mathcal{H}) &= \mathbb{E}_\sigma \sup_{\|\alpha\|^2_{\mathcal{H}_1}\le\lambda_\alpha^{-1}\hat{R}_s, \|\beta\|^2_{\mathcal{H}_2}\le\lambda_\beta^{-1}\hat{R}_s, \|\gamma\|^2_{\mathcal{H}_3}\le\lambda_\gamma^{-1}\hat{R}_s} \frac{1}{n}\sum_{i=1}^n \sigma_i\Big\{\langle\alpha, \Phi_1(f_s(x_i))\rangle_{\mathcal{H}_1} + \langle\beta, \Phi_2(f_s(x_i))\rangle_{\mathcal{H}_2}\langle\gamma, \Phi(x_i)\rangle_{\mathcal{H}_3}\Big\} \\
&\le \mathbb{E}_\sigma \sup_{\|\alpha\|^2_{\mathcal{H}_1}\le\lambda_\alpha^{-1}\hat{R}_s} \frac{1}{n}\sum_{i=1}^n \sigma_i\langle\alpha, \Phi_1(f_s(x_i))\rangle_{\mathcal{H}_1} \\
&\qquad\qquad + \sup_{\|\beta\|^2_{\mathcal{H}_2}\le\lambda_\beta^{-1}\hat{R}_s, \|\gamma\|^2_{\mathcal{H}_3}\le\lambda_\gamma^{-1}\hat{R}_s} \frac{1}{n}\sum_{i=1}^n \sigma_i\langle\beta\otimes\gamma, \Phi_2(f_s(x_i))\otimes\Phi(x_i)\rangle_{\mathcal{H}_2\otimes\mathcal{H}_3} \\
&\le \mathbb{E}_\sigma \sup_{\|\alpha\|^2_{\mathcal{H}_1}\le\lambda_\alpha^{-1}\hat{R}_s} \frac{1}{n}\sum_{i=1}^n \sigma_i\langle\alpha, \Phi_1(f_s(x_i))\rangle_{\mathcal{H}_1} \\
&\qquad\qquad + \sup_{\|\beta\otimes\gamma\|^2_{\mathcal{H}_2\otimes\mathcal{H}_3}\le\lambda_\beta^{-1}\lambda_\gamma^{-1}\hat{R}_s^2} \frac{1}{n}\sum_{i=1}^n \sigma_i\langle\beta\otimes\gamma, \Phi_2(f_s(x_i))\otimes\Phi(x_i)\rangle_{\mathcal{H}_2\otimes\mathcal{H}_3} \\
&\le \sqrt{\frac{\hat{R}_s}{\lambda_\alpha n}} + \sqrt{\frac{\hat{R}_s^2}{\lambda_\beta\lambda_\gamma n}} \qquad (11) \\
&\le \sqrt{\frac{\hat{R}_s}{n}}\Bigg\{\sqrt{\frac{1}{\lambda_\alpha}} + \sqrt{\frac{L}{\lambda_\beta\lambda_\gamma}}\Bigg\}.
\end{aligned}$$

The first inequality follows from the subadditivity of supremum. The last inequality follows from the fact that $\hat{R}_s \le P_n\ell(y, \langle 0, \Phi_1\rangle) + \lambda_\alpha\|0\|^2 \le L$.

Let $C = \sqrt{\frac{1}{\lambda_\alpha}} + \sqrt{\frac{L}{\lambda_\beta\lambda_\gamma}}$, and applying Talagrand's lemma (Mohri et al., 2018) and Jensen's inequality, we obtain

$$\mathfrak{R}(\mathcal{L}) = \mathbb{E}\hat{\mathfrak{R}}_S(\mathcal{L}) \le \mu_\ell\mathbb{E}\hat{\mathfrak{R}}_S(\mathcal{H}) \le C\mu_\ell\mathbb{E}\sqrt{\frac{\hat{R}_s}{n}} \le C\mu_\ell\sqrt{\frac{\mathbb{E}\hat{R}_s}{n}}.$$

To apply Corollary 3.5 of Bartlett et al. (2005), we should solve the equation

$$r = C\mu_\ell\sqrt{\frac{r}{n}}, \qquad (12)$$

and obtain $r^* = \frac{\mu_\ell^2 C^2}{n}$. Thus, for any $\eta > 0$, with probability at least $1 - e^{-\eta}$, there exists a constant $C' > 0$ that satisfies

$$P_n\ell(y,\, h) \le C'\biggl(\mathbb{E}\hat{R}_s + \frac{\mu_\ell^2 C^2}{n} + \frac{\eta}{n}\biggr) \le C'\biggl(R_s + \frac{\mu_\ell^2 C^2}{n} + \frac{\eta}{n}\biggr). \tag{13}$$

Note that, for the last inequality, because $\hat{R}_s \le P_n\ell(y, \langle\alpha, \Phi_1\rangle) + \lambda_\alpha\|\alpha\|^2$ for any α, taking the expectation of both sides yields $\mathbb{E}\hat{R}_s \le P\ell(y, \langle\alpha, \Phi_1\rangle) + \lambda_\alpha\|\alpha\|^2$, and this gives $\mathbb{E}\hat{R}_s \le \inf_\alpha\{P\ell(y, \langle\alpha, \Phi_1\rangle) + \lambda_\alpha\|\alpha\|^2\} = R_s$. Consequently, applying Theorem 1 of Srebro et al. (2010), we have

$$P\ell(y,\, h(x)) \le P_n\ell(y,\, h(x)) +\ \tilde{O}\biggl(\biggl(\sqrt{\frac{R_s}{n}} + \frac{\mu_\ell C + \sqrt{\eta}}{n}\biggr)\Bigl(\sqrt{L}C + \sqrt{L\eta}\Bigr) + \frac{C^2 L + L\eta}{n}\biggr). \tag{14}$$

Here, we use $\hat{\mathfrak{R}}_S(\mathcal{H}) \le C\sqrt{\frac{\hat{R}_s}{n}} \le C\sqrt{\frac{L}{n}}$. □

Remark 1. As in Kuzborskij & Orabona (2017), without the estimation of the parameters α and β, the right-hand side of Eq. (11) becomes $\frac{1}{\sqrt{n}}\Bigl(c_1 + c_2\sqrt{\hat{R}_s}\Bigr)$ with some constant $c_1 > 0$ and $c_2 > 0$, and Eq. (12) becomes

$$r = \frac{1}{\sqrt{n}}(c_1 + c_2\sqrt{r}).$$

This yields the solution

$$r^* = \left(\frac{c_2}{2\sqrt{n}} + \sqrt{\left(\frac{c_2}{2\sqrt{n}}\right)^2 + \frac{c_1}{\sqrt{n}}}\right)^2 \le \frac{c_2^2}{n} + \frac{c_1}{\sqrt{n}},$$

where we use the inequality $\sqrt{x} + \sqrt{x+y} \le \sqrt{4x + 2y}$. Thus, Eq. (13) becomes

$$P_n\ell(y,\, h) \le C'\biggl(R_s + \frac{c_2^2}{n} + \frac{c_1}{\sqrt{n}} + \frac{\eta}{n}\biggr).$$

Consequently, we have the following result:

$$P\ell(y,\, h(x)) \le P_n\ell(y,\, h(x)) +\ \tilde{O}\biggl(\biggl(\sqrt{\frac{R_s}{n}} + \frac{\sqrt{c_1}}{n^{3/4}} + \frac{c_2 + \sqrt{\eta}}{n}\biggr)\Bigl(c_1 + c_2\sqrt{L} + \sqrt{L\eta}\Bigr) + \frac{(c_1 + c_2\sqrt{L})^2 + L\eta}{n}\biggr).$$

This means that even if $R_s = \tilde{O}(n^{-1})$, the resulting rate only improves to $\tilde{O}(n^{-3/4})$.

B.2 Proof of Theorem 3

Recall that loss function $\ell(\cdot, \cdot)$ is assumed to be μ_ℓ-Lipschitz for the first argument. In addition, we impose the following assumption.

Assumption 7. *There exists a constant $B \ge 1$ such that for every $h \in \mathcal{H}$, $P(h - h^*) \le BP(\ell(y, h) - \ell(y, h^*))$.*

Because we consider $\ell(y, y') = (y - y')^2$ in Theorem 3, Assumption 3 holds as stated in Bartlett et al. (2005).

First, we show the following corollary, which is a slight modification of Theorem 5.4 of Bartlett et al. (2005).

Corollary 6. *Let $\hat{h}$ be any element of $\mathcal{H}$ satisfying $P_n\ell(y, \hat{h}) = \inf_{h\in\mathcal{H}} P_n\ell(y, h)$, and let $\hat{h}^{(m)}$ be any element of $\mathcal{H}^{(m)}$ satisfying $P_n\ell(y, \hat{h}^{(m)}) = \inf_{h\in\mathcal{H}^{(m)}} P_n\ell(y, h)$. Define*

$$\hat{\psi}(r) = c_1\hat{\mathfrak{R}}_S\{h \in \mathcal{H} : \max_{m\in\{1,2\}} P_n(h^{(m)} - \hat{h}^{(m)})^2 \le c_3 r\} + \frac{c_2\eta}{n},$$

where c_1, c_2 and c_3 are constants depending only on B and μ_ℓ. Then, for any $\eta > 0$, with probability at least $1 - 5e^{-\eta}$,

$$P\ell(y, \hat{h}) - P\ell(y, h^*) \le \frac{705}{B}\hat{r}^* + \frac{(11\mu_\ell + 27B)\eta}{n},$$

where $\hat{r}^*$ is the fixed point of $\hat{\psi}$.

Proof. Define the function ψ as

$$\psi(r) = \frac{c_1}{2}\mathfrak{R}\{h \in \mathcal{H} : \mu_\ell^2 \max P(h^{(m)} - h^{(m)*})^2 \le r\} + \frac{(c_2 - c_1)\eta}{n}.$$

Because $\mathcal{H}, \mathcal{H}^{(1)}$ and $\mathcal{H}^{(2)}$ are all convex and thus star-shaped around each of its points, Lemma 3.4 of Bartlett et al. (2005) implies that ψ is a sub-root. Also, define the sub-root function ψ_m as

$$\psi_m(r) = \frac{c_1^{(m)}}{2}\mathfrak{R}\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P(h^{(m)} - h^{(m)*})^2 \le r\} + \frac{(c_2 - c_1)\eta}{n}.$$

Let r_m^* be the fixed point of $\psi_m(r_m)$. Now, for $r_m \ge \psi_m(r_m)$, Corollary 5.3 of Bartlett et al. (2005) and the condition on the loss function imply that, with probability at least $1 - e^{-\eta}$,

$$\mu_\ell^2 P(\hat{h}^{(m)} - h^{(m)*})^2 \le B\mu_\ell^2 P(\ell(y, \hat{h}^{(m)}) - \ell(y, \hat{h}^{(m)*})) \le 705\mu_\ell^2 r_m + \frac{(11\mu_\ell + 27B)B\mu_\ell^2\eta}{n}.$$

Denote the right-hand side by s_m, and define $r = \max r_m$ and $s = \max s_m$. Because $s \ge s_m \ge r_m \ge r_m^*$, we obtain $s \ge \psi_m(s)$ according to Lemma 3.2 of Bartlett et al. (2005), and thus,

$$s \ge 10\mu_\ell^2\mathfrak{R}\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P(h^{(m)} - h^{(m)*})^2 \le s\} + \frac{11\mu_\ell^2\eta}{n}.$$

Therefore, applying Corollary 2.2 of Bartlett et al. (2005) to the class $\mu_\ell\mathcal{H}^{(m)}$, it follows that with probability at least $1 - e^{-\eta}$,

$$\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P(h^{(m)} - h^{(m)*})^2 \le s\} \subseteq \{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P_n(h^{(m)} - h^{(m)*})^2 \le 2s\}.$$

This implies that with probability at least $1 - 2e^{-\eta}$,

$$\begin{aligned} P_n(\hat{h}^{(m)} - h^{(m)*})^2 &\le 2\left(705r + \frac{(11\mu_\ell + 27B)B\eta}{n}\right) \\ &\le 2\left(705 + \frac{(11\mu_\ell + 27B)B}{n}\right)r, \end{aligned}$$

where the second inequality follows from $r \ge \psi(r) \ge \frac{c_2\eta}{n}$. Define $2\left(705 + \frac{(11\mu_\ell+27B)B}{n}\right) = c'$. According to the triangle inequality in $L_2(P_n)$, it holds that

$$\begin{aligned} P_n(h^{(m)} - \hat{h}^{(m)})^2 &\le \left(\sqrt{P_n(h^{(m)} - h^{(m)*})^2} + \sqrt{P_n(h^{(m)*} - \hat{h}^{(m)})^2}\right)^2 \\ &\le \left(\sqrt{P_n(h^{(m)} - h^{(m)*})^2} + \sqrt{c'r}\right)^2. \end{aligned}$$

Again, applying Corollary 2.2 of Bartlett et al. (2005) to $\mu_\ell\mathcal{H}^{(m)}$ as before, but now for $r \ge \psi_m(r)$, it follows that with probability at least $1 - 4e^{-\eta}$,

$$\begin{aligned} \{h \in \mathcal{H} : \mu_\ell^2 \max P(h^{(m)} - h^{(m)*})^2 \le r\} &= \bigcap_{m=1}^{2}\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P(h^{(m)} - h^{(m)*})^2 \le r\} \\ &\subseteq \bigcap_{m=1}^{2}\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P_n(h^{(m)} - h^{(m)*})^2 \le 2r\} \\ &\subseteq \bigcap_{m=1}^{2}\{h^{(m)} \in \mathcal{H}^{(m)} : \mu_\ell^2 P_n(h^{(m)} - \hat{h}^{(m)})^2 \le (\sqrt{2r} + \sqrt{c'r})^2\} \\ &= \{h \in \mathcal{H} : \mu_\ell^2 \max P_n(h^{(m)} - \hat{h}^{(m)})^2 \le c_3 r\}, \end{aligned}$$

where $c_3 = (\sqrt{2}+\sqrt{c'})^2$. Combining this with Lemma A.4 of Bartlett et al. (2005) leads to the following inequality: with probability at least $1-5e^{-x}$

$$\begin{aligned}\psi(r) &= \frac{c_1}{2}\mathfrak{R}\{h \in \mathcal{H} : \mu_\ell^2 \max P(h^{(m)} - h^{(m)*})^2 \le r\} + \frac{(c_2 - c_1)\eta}{n}\\ &\le c_1\hat{\mathfrak{R}}_S\{h \in \mathcal{H} : \mu_\ell^2 \max P(h^{(m)} - h^{(m)*})^2 \le r\} + \frac{c_2\eta}{n}\\ &\le c_1\hat{\mathfrak{R}}_S\{h \in \mathcal{H} : \mu_\ell^2 \max P_n(h^{(m)} - \hat{h}^{(m)})^2 \le c_3 r\} + \frac{c_2\eta}{n}\\ &= \hat{\psi}(r).\end{aligned}$$

Letting $r = r^*$ and using Lemma 4.3 of Bartlett et al. (2005), we obtain $r^* \le \hat{r}^*$, thus proving the statement. □

Proof of Theorem 3. Define $R = \max_m \sup_{h\in\mathcal{H}^{(m)}} P_n(y-h(x))^2$. For any $h \in \mathcal{H}^{(m)}$, we obtain

$$P_n(h^{(m)}(x) - \hat{h}^{(m)}(x))^2 \le 2P_n(y - h^{(m)}(x))^2 + 2P_n(y - \hat{h}^{(m)}(x))^2 \le 4 \sup_{h\in\mathcal{H}^{(m)}} P_n(y - h(x))^2 \le 4R.$$

From the symmetry of the σ_i and the fact that $\mathcal{H}^{(m)}$ is convex and symmetric, we obtain the following:

$$\begin{aligned}\hat{\mathfrak{R}}_S\{h \in \mathcal{H} : \max P_n(h^{(m)} - \hat{h}^{(m)})^2 \le 4R\} &= \mathbb{E}_\sigma \sup_{\substack{h^{(m)}\in\mathcal{H}^{(m)}\\ P_n(h^{(m)}-\hat{h}^{(m)})^2\le 4R}} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^2 h^{(m)}(x_i)\\ &= \mathbb{E}_\sigma \sup_{\substack{h^{(m)}\in\mathcal{H}^{(m)}\\ P_n(h^{(m)}-\hat{h}^{(m)})^2\le 4R}} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^2 (h^{(m)}(x_i) - \hat{h}^{(m)}(x_i))\\ &\le \mathbb{E}_\sigma \sup_{\substack{h^{(m)},g^{(m)}\in\mathcal{H}^{(m)}\\ P_n(h^{(m)}-g^{(m)})^2\le 4R}} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^2 (h^{(m)}(x_i) - g^{(m)}(x_i))\\ &= 2\mathbb{E}_\sigma \sup_{\substack{h^{(m)}\in\mathcal{H}^{(m)}\\ P_n h^{(m)2}\le R}} \frac{1}{n}\sum_{i=1}^n \sigma_i \sum_{m=1}^2 h^{(m)}(x_i)\\ &\le 2\sum_{m=1}^2 \mathbb{E}_\sigma \sup_{\substack{h^{(m)}\in\mathcal{H}^{(m)}\\ P_n h^{(m)2}\le R}} \frac{1}{n}\sum_{i=1}^n \sigma_i h^{(m)}(x_i)\\ &\le 2\sum_{m=1}^2 \left\{\frac{2}{n}\sum_{j=1}^n \min\{R, \hat{\lambda}_j^{(m)}\}\right\}^{\frac{1}{2}}\\ &\le \left\{\frac{16}{n}\sum_{m=1}^2\sum_{j=1}^n \min\{R, \hat{\lambda}_j^{(m)}\}\right\}^{\frac{1}{2}}.\end{aligned}$$

The second inequality comes from the subadditivity of supremum and the third inequality follows from Theorem 6.6 of Bartlett et al. (2005). To obtain the last inequality, we use $\sqrt{x}+\sqrt{y} \le \sqrt{2(x+y)}$. Thus, we have

$$\begin{aligned}&2c_1\hat{\mathfrak{R}}_S\{h \in \mathcal{H} : \max P_n(h^{(m)} - \hat{h}^{(m)})^2 \le 4R\} + \frac{(c_2+2)\eta}{n}\\ &\qquad\le 4c_1\left\{\frac{16}{n}\sum_{m=1}^2\sum_{j=1}^n \min\left\{R, \hat{\lambda}_j^{(m)}\right\}\right\}^{\frac{1}{2}} + \frac{(c_2+2)\eta}{n},\end{aligned}$$

for some constants c_1 and c_2. To apply Corollary 6, we should solve the following inequality for r

$$r \leq 4c_1 \left\{ \frac{16}{n} \sum_{m=1}^{2} \sum_{j=1}^{n} \min \left\{ r, \hat{\lambda}_j^{(m)} \right\} \right\}^{\frac{1}{2}}.$$

For any integers $\kappa_m \in [0, n]$, the right-hand side is bounded as

$$\begin{aligned} 4c_1 \left\{ \frac{16}{n} \sum_{m=1}^{2} \sum_{j=1}^{n} \min \left\{ r, \hat{\lambda}_j^{(m)} \right\} \right\}^{\frac{1}{2}} &\leq 4c_1 \left\{ \frac{16}{n} \sum_{m=1}^{2} \left(\sum_{j=1}^{\kappa_m} r + \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right) \right\}^{\frac{1}{2}} \\ &= \left\{ \left(\frac{256c_1^2}{n} \sum_{m=1}^{2} \kappa_m \right) r + \frac{256c_1^2}{n} \sum_{m=1}^{2} \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right\}^{\frac{1}{2}}, \end{aligned}$$

and we obtain the solution r^* as

$$\begin{aligned} r^* &\leq \frac{128c_1^2}{n} \sum_{m=1}^{2} \kappa_m + \left(\left\{ \frac{128c_1^2}{n} \sum_{m=1}^{2} \kappa_m \right\}^2 + \frac{256c_1^2}{n} \sum_{m=1}^{2} \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right)^{\frac{1}{2}} \\ &\leq \frac{256c_1^2}{n} \sum_{m=1}^{2} \kappa_m + \left(\frac{256c_1^2}{n} \sum_{m=1}^{2} \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right)^{\frac{1}{2}}. \end{aligned}$$

Optimizing the right-hand side with respect to κ_1 and κ_2, we obtain the solution as

$$r^* \leq \min_{0 \leq \kappa_1, \kappa_2 \leq n} \left\{ \frac{256c_1^2}{n} \sum_{m=1}^{2} \kappa_m + \left(\frac{256c_1^2}{n} \sum_{m=1}^{2} \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right)^{\frac{1}{2}} \right\}.$$

Furthermore, according to Corollary 6, there exists a constant c such that with probability at least $1 - 5e^{-\eta}$,

$$P(y - \hat{h}(x))^2 - P(y - h^*(x))^2 \leq c \left(\min_{0 \leq \kappa_1, \kappa_2 \leq n} \left\{ \frac{1}{n} \sum_{m=1}^{2} \kappa_m + \left(\frac{1}{n} \sum_{m=1}^{2} \sum_{j=\kappa_m+1}^{n} \hat{\lambda}_j^{(m)} \right)^{\frac{1}{2}} \right\} + \frac{\eta}{n} \right).$$

□

Proof of Theorem 4. Using the inequality $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for $x \geq 0, y \geq 0$, we have

$$\begin{aligned} P(y - \hat{h}(x))^2 - P(y - h^*(x))^2 &= O\left(\min_{0 \leq \kappa_1, \kappa_2 \leq n} \left\{ \frac{\kappa_1 + \kappa_2}{n} + \left(\frac{1}{n} \sum_{j=\kappa_1+1}^{n} \hat{\lambda}_j^{(1)} + \frac{1}{n} \sum_{j=\kappa_2+1}^{n} \hat{\lambda}_j^{(2)} \right)^{\frac{1}{2}} \right\} + \frac{\eta}{n} \right) \\ &\leq O\left(\min_{0 \leq \kappa_1, \kappa_2 \leq n} \left\{ \frac{\kappa_1 + \kappa_2}{n} + \left(\frac{1}{n} \sum_{j=\kappa_1+1}^{n} j^{-\frac{1}{s_1}} + \frac{1}{n} \sum_{j=\kappa_2+1}^{n} j^{-\frac{1}{s_2}} \right)^{\frac{1}{2}} \right\} + \frac{\eta}{n} \right) \\ &\leq O\left(\min_{0 \leq \kappa_1, \kappa_2 \leq n} \left\{ \frac{\kappa_1 + \kappa_2}{n} + \left(\frac{1}{n} \sum_{j=\kappa_1+1}^{n} j^{-\frac{1}{s_1}} \right)^{\frac{1}{2}} + \left(\frac{1}{n} \sum_{j=\kappa_2+1}^{n} j^{-\frac{1}{s_2}} \right)^{\frac{1}{2}} \right\} + \frac{\eta}{n} \right). \end{aligned}$$

Because it holds that

$$\sum_{j=\kappa_m+1}^{n} j^{-\frac{1}{s_m}} < \int_{\kappa_m}^{\infty} x^{-\frac{1}{s_m}} \mathrm{d}x < \left[\frac{1}{1-\frac{1}{s_m}} x^{1-\frac{1}{s_m}} \right]_{\kappa_m}^{\infty} = \frac{s_m}{1-s_m} {\kappa_m}^{1-\frac{1}{s_m}},$$

for $m = 1, 2$, we should solve the following minimization problem:

$$\min_{0 \le \kappa_1, \kappa_2 \le n} \left\{ \frac{\kappa_1 + \kappa_2}{n} + \left(\frac{1}{n} \frac{s_1}{1-s_1} {\kappa_1}^{1-\frac{1}{s_1}} \right)^{\frac{1}{2}} + \left(\frac{1}{n} \frac{s_1}{1-s_1} {\kappa_2}^{1-\frac{1}{s_1}} \right)^{\frac{1}{2}} \right\} \equiv g(\kappa).$$

Taking the derivative, we have

$$\frac{\partial g(\kappa)}{\partial \kappa_1} = \frac{1}{n} + \frac{1}{2} \left(\frac{1}{n} \frac{s_1}{1-s_1} {\kappa_1}^{1-\frac{1}{s_1}} \right)^{-\frac{1}{2}} \left(-\frac{{\kappa_1}^{-\frac{1}{s_1}}}{n} \right).$$

Setting this to zero, we find the optimal κ_1 as

$$\kappa_1 = \left(\frac{s_1}{1-s_1} \frac{4}{n} \right)^{\frac{s_1}{1+s_1}}.$$

Similarly, we have

$$\kappa_2 = \left(\frac{s_2}{1-s_2} \frac{4}{n} \right)^{\frac{s_2}{1+s_2}},$$

and

$$\begin{aligned}
&P(y - \hat{h}(x))^2 - P(y - h^*(x))^2 \\
&\le O\left(\frac{1}{n} \left(\frac{s_1}{1-s_1} \frac{4}{n} \right)^{\frac{s_1}{1+s_1}} + \frac{1}{n} \left(\frac{s_2}{1-s_2} \frac{4}{n} \right)^{\frac{s_2}{1+s_2}} + 2^{\frac{1-s_1}{1+s_1}} \left(\frac{s_1}{1-s_1} \frac{1}{n} \right)^{\frac{1}{1+s_1}} + 2^{\frac{1-s_2}{1+s_2}} \left(\frac{s_2}{1-s_2} \frac{1}{n} \right)^{\frac{1}{1+s_2}} + \frac{\eta}{n} \right) \\
&= O\Big(n^{-\frac{1}{1+s_1}} + n^{-\frac{1}{1+s_2}} \Big) \\
&= O\Big(n^{-\frac{1}{1+\max\{s_1, s_2\}}} \Big).
\end{aligned}$$

□

C Eigenvalue decay of the Hadamard product of two Gram matrices

We experimentally investigated how the decay rate s_2 in Corollary 4 is related to the overlap degree in the spaces spanned by the original input x and source features f_s.

For the original input $x \in \mathbb{R}^{100}$, we randomly constructed a set of 10 orthonormal bases, and then generated 100 samples from their spanning space. For the source features $f_s \in \mathbb{R}^{100}$, we selected d bases randomly from the 10 orthonormal bases selected for x and the remaining $10 - d$ bases from their orthogonal complement space. We then generated 100 samples of f_s from the space spanned by these 10 bases. The overlap number d can be regarded as the degree of overlap of two spaces spanned by the samples of x and f_s. We generated the 100 different sample sets of x and f_s.

We calculated the Hadamard product of the Gram matrices K_2 and K_3 using the samples of x and f_s, respectively. For the computation of K_2 and K_3, all combinations of the following five kernels were tested:

Linear kernel $k(x, x') = \dfrac{x^\top x}{2\gamma^2} + 1,$

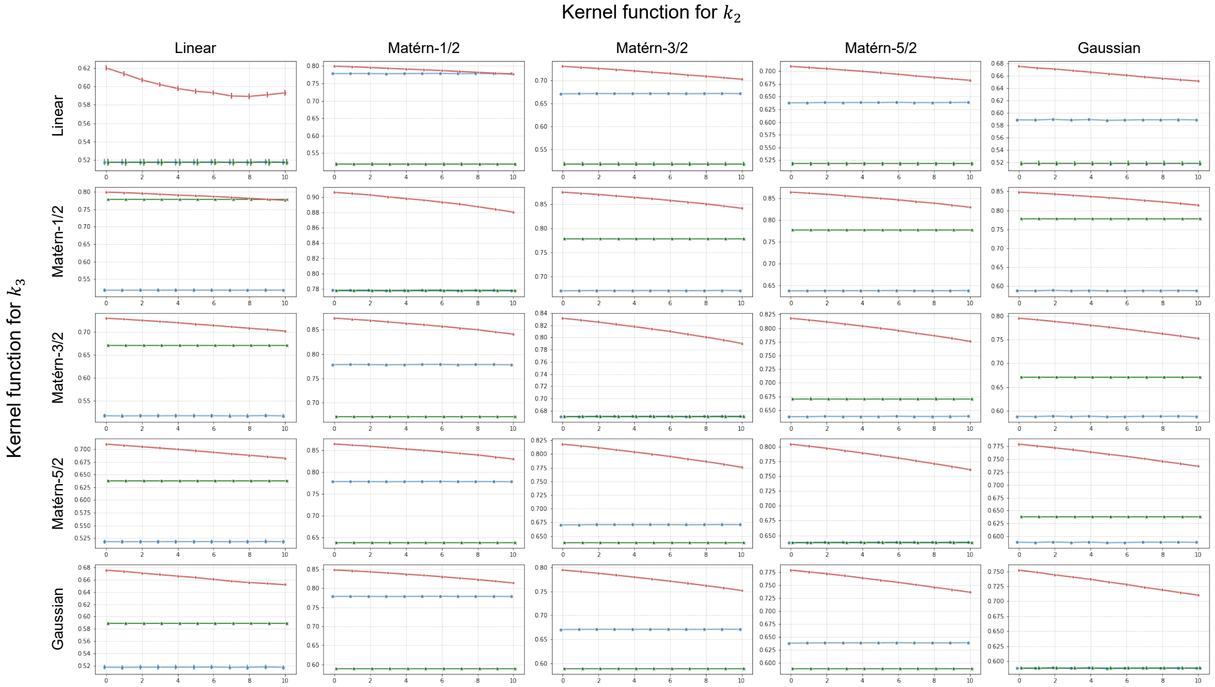


Figure 6: Decay rates of eigenvalues of K_2 (blue lines), K_3 (green lines) and $K_2 \circ K_3$ (red lines) for all combinations of the five different kernels. The vertical axis represents the decay rate, and the horizontal axis represents the overlap dimension d in the space where x and f_s are distributed.

$$\textbf{Matérn kernel}\ \ k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\|x - x'\|_2}{\gamma}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\|x - x'\|_2}{\gamma}\right) \ \text{ for } \nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \infty,$$

where $K_\nu(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is the gamma function. Note that for $\nu = \infty$, the Matérn kernel is equivalent to the Gaussian RBF kernel. The scale parameter γ of both kernels was set to $\gamma = \sqrt{\dim(x)} = \sqrt{10}$. For a given matrix K, the decay rate of the eigenvalues was estimated as the smallest value of s that satisfies $\lambda_i \leq \|K\|_F^2 \cdot i^{-\frac{1}{s}}$ where $\|\cdot\|_F$ denotes the Frobenius norm. Note that this inequality holds for any matrices K with $s = 1$ (Vershynin, 2018).

Figure 6 shows the change of the decay rates with respect to varying d for various combinations of the kernels. In all cases, the decay rate of $K_2 \circ K_3$ showed a clear trend of monotonically decreasing as the degree of overlap d increases. In other words, the greater the overlap between the spaces spanned by x and f_s, the smaller the decay rate, and the smaller the complexity of the RKHS $\mathcal{H}_2 \otimes \mathcal{H}_3$.

D Experimental details

The Python code to reproduce all experiments is available at `https://github.com/mshunya/AffineTL`. Instructions for obtaining the datasets used in the experiments are described in the code.

D.1 Kinematics of the robot arm

D.1.1 Dataset

We used the SARCOS dataset in Williams & Rasmussen (2006). The task is to predict the feed-forward torque required to follow the desired trajectory in the seven joints of the SARCOS anthropomorphic robot arm. The twenty one features representing the joints' position, velocity, and acceleration were used as x. The observed values of six torques other than the torque at the joint in the target domain were given to

Algorithm 2 Block relaxation algorithm for **Affine transfer (full)**.

```
**Initialize**
    $a_0 \leftarrow (K_1 + \lambda_1 I_n)^{-1} y, \quad b_0 \sim \mathcal{N}(\mathbf{0}, I_n), \quad c_0 \sim \mathcal{N}(\mathbf{0}, I_n), \ \ d_0 \leftarrow 0.5$
**repeat**
    $a \leftarrow (K_1 + \lambda_1 I_n)^{-1}(y - (K_2 b + \mathbf{1}) \circ (K_3 c))$
    $b \leftarrow (\mathrm{Diag}(K_3 c)^2 K_2 + \lambda_2 I_n)^{-1}((K_3 c) \circ (y - K_1 a - K_3 c))$
    $c \leftarrow (\mathrm{Diag}(K_2 b + \mathbf{1}) + \lambda_3 I_n)^{-1}((K_2 b + \mathbf{1}) \circ (y - K_1 a))$
    $d \leftarrow \langle y - K_1 a - (K_2 b + \mathbf{1}) \circ (K_3 c), \mathbf{1} \rangle / n$
**until** convergence
```

the source features f_s. The dataset includes 44,484 training samples and 4,449 test samples. We selected $\{5, 10, 15, 20, 30, 40, 50\}$ samples randomly from the training set. The prediction performances of the trained models were evaluated using the 4,449 test samples. Repeated experiments were conducted 20 times with different independently sampled datasets.

D.1.2 Model definition and hyperparameter search

No transfer, Only source, Augmented, HTL–offset, HTL–scale For each procedure, we used kernel ridge regression with the RBF kernel $k(x, x') = \exp(-\frac{1}{2\ell^2}\|x - x'\|_2^2)$. The scale parameter ℓ was set to the square root of the input dimension as $\ell = \sqrt{21}$ for **No transfer**, **HTL–offset** and **HTL–scale**, $\ell = \sqrt{6}$ for **Only source** and $\ell = \sqrt{27}$ for **With source**. The regularization parameter λ was selected in five-fold cross-validation in which the grid search was performed over 50 grid points in the interval $[10^{-4}, 10^2]$.

Affine transfer (full), Affine transfer (constrained) We considered the following kernels:

$$k_1(f_s(x), f_s(x')) = \exp\Big(-\frac{1}{2\ell^2}\|f_s(x) - f_s(x')\|_2^2\Big) \ (\ell = \sqrt{6}),$$
$$k_2(f_s(x), f_s(x')) = \exp\Big(-\frac{1}{2\ell^2}\|f_s(x) - f_s(x')\|_2^2\Big) \ (\ell = \sqrt{6}),$$
$$k_3(x, x') = \exp\Big(-\frac{1}{2\ell^2}\|x - x'\|_2^2\Big) \ (\ell = \sqrt{27}),$$

for g_1, g_2 and g_3 in the affine transfer model, respectively.

Hyperparameters to be optimized are the three regularization parameters λ_1, λ_2 and λ_3. We performed five-fold cross-validation to identify the best hyperparameter set from the candidate points; $[10^{-3}, 10^{-2}, 10^{-1}, 1]$ for λ_1 and $[10^{-2}, 10^{-1}, 1, 10]$ for each of λ_2 and λ_3.

To learn the **Affine transfer (full)** and **Affine transfer (constrained)**, we used the following objective functions:

Affine transfer (full) $\|y - (K_1 a + (K_2 b + \mathbf{1}) \circ (K_3 c) + d)\|_2^2 + \lambda_1 a^\top K_1 a + \lambda_2 b^\top K_2 b + \lambda_3 c^\top K_3 c,$

Affine transfer (constrained) $\frac{1}{n}\|y - (K_1 a + K_3 c + d)\|_2^2 + \lambda_1 a^\top K_1 a + \lambda_3 c^\top K_3 c.$

Algorithm 2 summarizes the block relaxation algorithm for **Affine transfer (full)**. For **Affine transfer (constrained)**, we found the optimal parameters as follows:

$$\begin{bmatrix} \hat{a} \\ \hat{c} \\ \hat{d} \end{bmatrix} = \left(\begin{bmatrix} K_1 \\ K_3 \\ \mathbf{1}^\top \end{bmatrix} \begin{bmatrix} K_1 & K_3 & \mathbf{1} \end{bmatrix} + \begin{bmatrix} \lambda_1 K_1 & & \\ & \lambda_3 K_3 & \\ & & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} K_1 \\ K_3 \\ \mathbf{1}^\top \end{bmatrix} y$$

The stopping criterion of the algorithm was set as

$$\max_{\theta \in \{a,b,c\}} \frac{\max_i \left|\theta_i^{(\mathrm{new})} - \theta_i^{(\mathrm{old})}\right|}{\max_i \left|\theta_i^{(\mathrm{old})}\right|} < 10^{-4}, \tag{15}$$

where θ_i denotes the i-th element of the parameter θ. This convergence criterion is employed in several existing machine learning libraries, e.g., scikit-learn [2].

D.2 Lattice thermal conductivity of inorganic crystals

D.2.1 Datasets

Two datasets were used, consisting of the lattice thermal conductivity (LTC) and scattering phase space (SPS) of 45 and 320 inorganic crystals, respectively, that were obtained by performing first principle calculations (Ju et al., 2021). To obtain the input descriptor x, XenonPy[3] was used to calculate 290 compositional and physicochemical features of a given material (Liu et al., 2021).

D.2.2 Model definition and hyperparameter search

Fully connected neural networks were used for both the source and target models, with a LeakyReLu activation function with $\alpha = 0.01$. The model training was conducted using the Adam optimizer (Kingma & Ba, 2015). Hyperparameters such as the width of the hidden layer, learning rate, number of epochs, and regularization parameters were adjusted with five-fold cross-validation. For more details on the experimental conditions and procedure, refer to the provided Python code.

Source model As a preliminary step, we trained 100 neural networks to predict SPS. The hidden layer width L was randomly chosen from the range $[50, 100]$, and we trained a neural network with a structure of (input)-L-L-L-1. Each of the three hidden layers of the source model was used as an input to the transfer models, and we examined the difference in prediction performance for the three layers.

Affine transfer The functions g_1, g_2, and g_3 in the affine transfer model were modeled by neural networks. We used neural networks with one hidden layer for g_1, g_2 and g_3.

D.3 Heat capacity of organic polymers

D.3.1 Dataset

The task is to bridge the specific heat capacity of a given polymeric material calculated by the classical MD simulation, including bias and variance, to its experimental value. Experimental values of the specific heat capacity of the 70 polymers were collected from PoLyInfo (Otsuka et al., 2011). The MD simulation was also applied to calculate their heat capacities. For models to predict the log-transformed heat capacity, a given polymer with its chemical structure was translated into the 190-dimensional force field descriptors, using RadonPy[4] (Hayashi et al., 2022). We randomly sampled 60 training polymers and tested the prediction performance of a trained model on the remaining 10 polymers 20 times. The PoLyInfo sample identifiers for the selected polymers are listed in the code.

D.3.2 Model definition and hyperparameter search

As described in Section 5.3, the 190-dimensional force field descriptor consists of nine blocks corresponding to different types of features. The J_t features that make up block t represent discretized values of the density function of the force field parameters assigned to the atoms, bonds, or dihedral angles that constitute the

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html`

[3] `https://github.com/yoshida-lab/XenonPy`

[4] `https://github.com/RadonPy/RadonPy`

Algorithm 3 Block relaxation algorithm for the model in Eq. (16).

```
Initialize
    $\alpha_0 \leftarrow \hat{\alpha}_{0,\mathrm{olr}}, \quad \alpha_1 \leftarrow \hat{\alpha}_{1,\mathrm{olr}}, \quad \beta \leftarrow 0, \;\; \gamma \leftarrow\leftarrow \hat{\gamma}_{\mathrm{diff}}$
repeat
    $\alpha \leftarrow \arg\min_\alpha F_{\alpha,\beta,\gamma}$
    $\beta \leftarrow \arg\min_\beta F_{\alpha,\beta,\gamma}$
    $\gamma \leftarrow \arg\min_\gamma F_{\alpha,\beta,\gamma}$
until convegence
```

given polymer. Therefore, the regression coefficients of the features within a block should be estimated smoothly. To this end, we imposed fused regularization on the parameters as

$$\lambda_1 \|\gamma\|_2^2 + \lambda_2 \sum_{t \in T} \sum_{j=2}^{J_t} \big(\gamma_{t,j} - \gamma_{t,j-1}\big)^2,$$

where $T = \{\mathrm{mass}, \mathrm{charge}, \epsilon, \sigma, K_{\mathrm{bond}}, r_0, K_{\mathrm{angle}}, \theta, K_{\mathrm{dih}}\}$, and $J_t = 10$ for $t =$ mass and $J_t = 20$ otherwise. The regression coefficient $\gamma_{t,j}$ corresponds to the j-th feature of block t.

Ordinary linear regression The experimental heat capacity $y = \log C_{\mathrm{P}}^{\mathrm{exp}}$ was regressed on the MD-calculated property, without regularization, as $\hat{y} = \alpha_0 + \alpha_1 f_s$ where $\hat{y}$ denotes the conditional expectation and $f_s = \log C_{\mathrm{P}}^{\mathrm{MD}}$.

Learning the log-difference We calculated the log-difference $\log C_{\mathrm{P}}^{\mathrm{exp}} - \log C_{\mathrm{P}}^{\mathrm{MD}}$ and trained the linear model with the ridge penalty. The hyperparameters λ_1 and λ_2 for the scale- and smoothness-regularizers were determined based on five-fold cross validation across 25 equally space grids in the interval $[10^{-2}, 10^2]$ for λ_1 and in $\{50, 100, 150\}$ for λ_2.

Affine transfer We used the affine transfer model as

$$\hat{y} = \alpha_0 + \alpha_1 f_s(x) - (\beta f_s(x) + 1) \cdot x^\top \gamma, \tag{16}$$

where $f_s(x)$ is the log-transformed MD-calculated heat capacity $\log C_{\mathrm{P}}^{\mathrm{MD}}$. In the model training, the objective function was given as follows:

$$F_{\alpha,\beta,\gamma} = \frac{1}{n} \sum_{i=1}^{n} \big\{y_i - (\alpha_0 + \alpha_1 f_s(x_i) - (\beta f_s(x_i) + 1) x^\top \gamma)\big\}^2 \\ + \lambda_\beta \beta^2 + \lambda_{\gamma,1} \|\gamma\|_2^2 + \lambda_{\gamma,2} \sum_{t \in T} \sum_{j=2}^{J_t} \big(\gamma_{t,j} - \gamma_{t,j-1}\big)^2,$$

where $\alpha = [\alpha_0 \;\; \alpha_1]^\top$. With a fixed $\lambda_\beta = 1$, the remaining hyperparameters $\lambda_{\gamma,1}$ and $\lambda_{\gamma,2}$ were optimized through five-fold cross validation over 25 equally space grids in the interval $[10^{-2}, 10^2]$ for $\lambda_{\gamma,1}$ and in $\{50, 100, 150\}$ for $\lambda_{\gamma,2}$.

The algorithm to estimate the parameters α, β and γ is described in Algorithm 3, where $\alpha_{0,\mathrm{olr}}$ and $\alpha_{1,\mathrm{olr}}$ are the estimated parameters of the ordinary linear regression model, and $\hat{\gamma}_{\mathrm{diff}}$ is the estimated parameter of the log-difference model. For each step, the full conditional minimization of $F_{\alpha,\beta,\gamma}$ with respect to each parameter can be made analytically as

$$\begin{aligned}
\arg\min_\alpha F_{\alpha,\beta,\gamma} &= (F_s^\top F_s)^{-1} y_s^\top (y + (\beta f_s(X) + 1) \circ (X\gamma)), \\
\arg\min_\beta F_{\alpha,\beta,\gamma} &= -(f_s(X)^\top \mathrm{diag}(X\gamma)^2 f_s(X) + n\lambda_2)^{-1} f_s(X)^\top \mathrm{diag}(X\gamma)(y - F_s\alpha + X\gamma), \\
\arg\min_\gamma F_{\alpha,\beta,\gamma} &= -(X^\top \mathrm{diag}(f_s(X)\beta + 1)^2 X + \Lambda)^{-1} X^\top \mathrm{diag}(f_s(X)\beta + 1)(y - F_s\alpha),
\end{aligned}$$

where X denote the matrix in which the i-th row is x_i, $y = [y_1 \cdots y_n]^\top$, $f_s(X) = [f_s(x_1) \cdots f_s(x_n)]^\top$, $F_s = [f_s(X) \;\; \mathbf{1}]$, and $d = 190$. Λ is a matrix including the two regularization parameters $\lambda_{\gamma,1}$ and $\lambda_{\gamma,2}$ as

$$\Lambda = D^\top D, \text{ where } D = \left[\begin{array}{c} \lambda_{\gamma,1} I_d \\ \lambda_{\gamma,2} M \end{array} \right], \; M = \left[\begin{array}{cccccc} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow m\text{-th rows} \\ \\ \\ \end{array},$$

where $m \in \{10, 30, 50, 70, 90, 110, 130, 150, 170\}$. Note that the matrix M is the same as the matrix $[\mathbf{0} \;\; I_{189}] - [I_{189} \;\; \mathbf{0}]$ except that the m-th row is all zeros. Note also that $M \in \mathbb{R}^{189 \times 190}$, and therefore $D \in \mathbb{R}^{279 \times 190}$ and $\Lambda \in \mathbb{R}^{190 \times 190}$.

The stopping criterion for the algorithm was the same as Eq. (15).