Article:
Gardner, P. orcid.org/0000-0002-1882-9728, Bull, L.A., Dervilis, N. orcid.org/0000-0002-5712-7323 et al. (1 more author) (2022) On the application of kernelised Bayesian transfer learning to population-based structural health monitoring. Mechanical Systems and Signal Processing, 167 (Part B). 108519. ISSN 0888-3270

https://doi.org/10.1016/j.ymssp.2021.108519

On the application of kernelised Bayesian transfer learning to population-based structural health monitoring

P. Gardner, L.A. Bull, N. Dervilis, K. Worden

Dynamics Research Group, Department of Mechanical Engineering
University of Sheffield, Sheffield S1 3JD, UK

Abstract

Data-driven approaches to Structural Health Monitoring (SHM) generally suffer from a lack of available health-state data. In particular, for most structures, it is not possible to obtain a comprehensive set of labelled damage data — even covering the most common damage types — due to impracticalities and economic considerations in observing the structure in a range of damage states. One solution to this problem is to utilise labelled data from a set of 'similar' structures. The assumption is that, as a population, the group may have a shared label set that covers a wider range of damage states, which can be used in labelling a different structure of interest. These goals, producing a model that generalises for a population of structures, and transferring label information between structures, are part of a population-based view of SHM — known as population-based SHM (PBSHM). By considering data from a population, it is possible to make data-driven SHM practical in industrial contexts beyond unsupervised learning, i.e. novelty detection. In order to realise the potential of PBSHM, this paper applies a heterogeneous transfer learning method — kernelised Bayesian transfer learning (KBTL) — which is a sparse Bayesian method that infers a discriminative classifier from inconsistent and heterogeneous feature data, i.e. the dataset from each member of the population may refer to different quantities in different dimensions. The technique infers a shared latent space where data from each member of the population are mapped on top of each other, meaning a single classifier can jointly be inferred that generalises to the complete population. As a consequence, label information can be transferred in this shared latent space between members of the population. The ability to infer a mapping from inconsistent and heterogeneous feature data make the approach a *heterogeneous transfer learning* method. To the best of the authors knowledge, this is the first time a heterogeneous transfer learning method has been applied in an SHM context.

Keywords: Heterogeneous transfer learning, kernelised Bayesian transfer learning, sparse Bayesian classification, population-based structural health monitoring

1. Introduction

Conventional data-driven SHM seeks to infer a machine learner using data obtained from an individual structure (or system), with the aim of creating a model that will generalise to future measurements, enabling future predictions of the health-state of the structure. This type of approach to SHM has achieved many successes [1–4]; however, it suffers from several flaws. Firstly, as with any conventional machine learner, the assumption is that the underlying data distribution used in training the algorithm will be the same for any future measurements. This assumption means that if the structure is repaired, or if operational and environmental conditions change the system's

fundamental behaviour, the assumptions in the machine learner are broken, and it will fail to generalise. Secondly, in order to make accurate health statements (beyond stating a particular observation is novel when compared to a given baseline) some form of health-state labels are required. Typically, a variety of labelled data will be costly and potentially infeasible to obtain for operational structures, meaning that there will be, at best, sparsity in the label set, and at worst, a large portion of missing health-state labels. Semi-supervised learning can help with the label sparsity problem [5, 6], expanding the existing dataset by considering any unlabelled data from the structure, but it cannot (on its own) provide new class labels for health states not observed on that structure. This sparsity in label data means that a conventional machine learner is not able to predict classes that have not already been observed on the structure of interest — a significant drawback for conventional data-based approaches to SHM which severely limits their application in real world scenarios. Lastly, even if a computer model was used to train the machine learner (with the large assumption that it is possible to simulate all expected health states without model discrepancy), or if a machine learner was trained on another structure with a more complete label set and applied to the structure of interest, both of these machine learners will fail to generalise well, because of the first problem — machine learners assume the joint distribution from which the data were obtained in training is *exactly* the same as in testing. In the light of these challenges, a new viewpoint is proposed for the problem of SHM. This novel approach to SHM seeks to leverage label information across a population of structures (both physical and simulated) in creating machine learning models that will generalise across the complete population, enabling label information to be transferred from one member of the population to another, and is appropriately named *Population-Based* Structural Health Monitoring (PBSHM), the foundation of which are set out in [7–9].

The first paper in this series on the foundations of PBSHM [7], introduces the field of PBSHM, defining homogeneous populations and proposing the concept of a *Form* as a method for capturing the idealised response from a population, along with the associated variation within the population. The second paper, [8], introduces the concept of heterogeneous populations and the notion of determining similarities between structures in a population using an abstract graphical representation of structures. Within this 'space of structures' metrics can then be used to determine if any two members from a set of structures are similar enough that knowledge could be shared between them. The final paper in the series [9], states that in order to transfer knowledge between structures in a population, transfer learning can be utilised, with an outline of when transfer should be attempted given an SHM problem ($\mathcal{SP}$).

To illustrate the ideas behind PBSHM, a short example is provided. Imagine a scenario in which an asset manager for a train company is interested in performing SHM across their fleet of trains, which forms the population. This population may be composed of trains all of the same model type and specifications, making each member of the population nominally identical and being defined as a *homogeneous population*. More commonly, the population is likely to be formed of a range of trains, with differing model types and specifications, defined as a *heterogeneous population*; however, it is still expected that some failure types will be common across the population. The asset manager is tasked with maintaining and monitoring the complete population; however, for some members of the population, little or no health-state information is available, apart from the normal healthy condition (which is especially true for new trains that have recently been purchased). This sparsity in label data makes conventional approaches to data-based SHM impractical, outside of novelty detection, which still leaves the problem of what 'novel' refers to (be it benign changes or damage related). However, it is likely that the asset manager will have data from across the fleet that cover the majority of common failure types. In fact, they may even have access to relatively good damage simulations for particular trains from physics-based models, meaning simulated data points

of certain health states are attainable. The asset manager, therefore, requires a *population-based* approach to SHM, where the information across the population is used to create a machine learner that will both generalise across the population and will allow label information to be transferred to any train in the fleet, allowing robust health diagnosis for all trains in the population. This paper will seek to address this scenario by utilising a sparse Bayesian discriminative classifier that seeks to pull information from across multiple structures in creating a generalised model, namely Kernelised Bayesian Transfer Learning (KBTL), proposed by Gönen and Margolin [10].

Within the theory of PBSHM, it is critical that algorithms are able to infer a map between features from different structures in the population, enabling information to be transferred from structure to structure. This goal aligns closely with a branch of machine learning called *transfer learning*, that seeks to infer a model even when the feature data distributions for each structure are different, and can include scenarios where the label information is different [11–13]. The approach applied in this paper, KBTL [10], is a sparse Bayesian model that seeks to learn a classifier that generalises for multiple different datasets, which in this context are from members of the population, and has the ability to transfer label information between these structures. The application of this technique is particularly novel in an SHM context, as it allows each member of the population to have a different feature set, i.e. the dimension of each feature set from each structure is different, which is called *feature inconsistency*. This property means that KBTL is a form of *heterogeneous transfer learning*[1] [13–15]. Heterogeneous transfer learning therefore allows a wider range of datasets and structures to be considered in the same population, relaxing the constraint in homogeneous transfer learning that the feature sets for each structure are of the same dimension. For example, in heterogeneous transfer learning, one structure may have a feature set composed of frequency response functions with 1024 spectral lines (i.e. $d = 1024$), whereas another structure has a feature set comprising transmissibilities with 16384 spectral lines (i.e. $d = 16384$), and a third structure could even have a feature set formed from three natural frequencies (i.e. $d = 3$). Clearly in this scenario it is not trivial to map the feature sets from each structure onto a shared latent space, but when possible, heterogeneous transfer learning methods make PBSHM applicable to a wider set of population types — as long as each structure in the population shares some structural similarity with respect to a given SHM problem $\mathcal{SP}$ [9], as denoted by Rytters hierarchy [16]. In the case where no similarities exist between members of the population, in relation to the SHM problem, negative transfer may occur, meaning the transfer learning algorithm will perform worse than a conventional approach [9]. As established in [8], one method for assessing these structural similarities before attempting transfer is to define a metric space over a space of structures, which can be used to objectively quantify whether transfer should be attempted.

Several transfer learning methods have been utilised within the health assessment literature [17–32]; these can be broadly divided into two main categories: those using fine-tuning approaches [17–24], and those performing some form of domain adaptation [25–32]. Fine-tuning approaches seek to leverage pre-trained neural networks (often convolutional neural networks) for new tasks. The approach fixes a set of weights in the network, that are assumed to contain relevant information for the new task, for example the feature extraction from some convolutional layer is expected to be the same on the new task. The remaining subset of weights in the network are re-trained based on data for a new task. In the SHM literature, fine-tuning is typically used to perform image classification tasks [18–24], with Cao *et al.* applying the approach to vibration signals in a condition monitoring

[1]For clarity, heterogeneous transfer learning is not the only way to perform PBSHM for heterogeneous populations, and the two terms should not be confused.

context [17]. Fine-tuning as an approach does not aim to transfer label knowledge from dataset to dataset, but is focused on repurposing expensive-to-train deep neural networks.

The other category of methods in the SHM literature can be classed as using some form of domain adaptation, where the aim is to move the feature spaces between datasets onto a shared latent space to aid the learning of a target task [25–32]. These papers can again be subdivided into those that map each feature space onto a shared latent space and require some additional classifier [26, 30, 32], and those that seek to perform domain adaptation and classification in a single algorithm [25, 27–29, 31], typically using deep neural network architectures [27–29, 31]. Several of these approaches use a maximum mean discrepancy criterion as the mechanism with which the distance between each feature space (particularly between the joint distributions) is minimised in the shared latent space [28–32]. However, all these existing approaches require the feature space from each structure to be consistent, i.e. the dimension of the feature space for each structure is exactly the same, and are therefore all forms of *homogeneous transfer learning*. This limits the application of these methods within an SHM context to scenarios where each structure has the same feature space, e.g. transmissibilities with the same number of spectral lines. The method used in this paper relaxes this constraint, allowing each structure to have a different dimension of feature space, therefore performing *heterogeneous transfer learning*.

Finally, it is noted that there are some similarities between KBTL and multi-task learning algorithms that exist within the SHM literature [33, 34], and again these require consistent feature spaces. Multi-task learning methods aim to infer a model that will generalise to multiple tasks [35], such as predicting the health states of different structures. In particular, it is noted that sparse Bayesian approaches to multi-task learning have been proposed in [34]; however, this method does not handle the inconsistent feature space scenario, and is utilised in performing multi-output regression. To the authors' best knowledge, this is the first time heterogeneous transfer learning has been performed in an SHM context. A MATLAB implementation accompanies this paper: https://github.com/pagard/EngineeringTransferLearning.

The outline of the paper is as follows. Section 2 states the multi-class formulation of KBTL, as proposed by Gönen and Margolin [10]. Application areas are introduced in Section 3, discussing how the approach can be applied to multiple PBSHM scenarios and populations. The following subsections present case studies demonstrating the effectiveness of KBTL in three application areas, where the approach is benchmarked against a variety of conventional and heterogeneous transfer learning approaches. Lastly, conclusions and further research are presented in Section 4.

2. Kernelised Bayesian transfer learning

Kernelised Bayesian Transfer Learning (KBTL) [10] is a supervised learning algorithm that aims to leverage information across multiple datasets from different systems in generating one classification model in a generalised latent space. In order to understand the algorithm it is helpful to define transfer learning, requiring two objects [11]:

> A **domain** $\mathcal{D} = \{\mathcal{X}, p(X)\}$ is an object that consists of a feature space $\mathcal{X}$ and a marginal probability distribution $p(X)$ over a finite sample of feature data $X = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathcal{X}$ from $\mathcal{X}$.
>
> A **task** $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ is the combination of a label space $\mathcal{Y}$ and a predictive function $f : \mathcal{X} \to \mathcal{Y}$.

Transfer learning is the process of improving a target predictive function, for a target task in the target domain, given that knowledge is available for some source domains and tasks; assuming the

source and target domains and/or tasks are different [11–13].

KBTL aims to learn a set of tasks by leveraging the knowledge across all domains [10]. In a multi-class setting, the method also allows information to be transferred between domains, meaning if a domain is missing a class label in training, but that label appears in another domain, the algorithm will leverage this information in creating a classifier that generalises and accurately predicts that label for all domains. More simply, the algorithm can be divided into two steps [10]: 1) a projection and dimensionality reduction step, where a linear projection on a kernel embedding of each domain maps it to a shared latent space $\mathcal{H}$ and 2) a discriminative classifier (for each class in a one-vs-all sense) is inferred in the shared latent space and used to predict each task. The first stage is designed to handle inconsistent feature spaces — at least one feature space $\mathcal{X}_t$ for a domain $\mathcal{D}_t$ is not the same dimension d_t as another feature space $\mathcal{X}_k$ in the domain set such that $d_t \neq d_k$ — by learning a mapping (through a kernel embedding) for each domain from its d_t-dimensional space to one R-dimensional space for all domains; assuming each domain provides useful information in learning the set of tasks, making the approach general to a wide range of problems. This ability to map from inconsistent feature spaces to an R-dimensional space makes the approach a heterogeneous transfer learning method [13–15], which is useful for performing inferences between populations of structures with inconsistent feature data.

Formally, KBTL[2] [10] assumes T domains $\{\mathcal{D}_t\}_{t=1}^{T}$ with inconsistent feature spaces $\{\mathcal{X}_t\}_{t=1}^{T}$ indexed $\{1:T\}$ (where : indicates the set of integers from 1 to T). Each domain has an associated task $\{\mathcal{T}_t\}_{t=1}^{T}$ (i.e. $\mathcal{D}_t$ relates to $\mathcal{T}_t \ \forall \ t \in 1:T$) with consistent label spaces $\mathcal{Y}_t = \mathcal{Y}_k \ \forall \ t, k \in 1:T$ (and hence can be seen as having one global label space $\mathcal{Y}$). It is assumed that each domain has N_t finite feature observations $X_t = \{\boldsymbol{x}_{t,i} \in \mathcal{X}_t\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d_t}$ (where d_t is the dimension of the feature set), that are independent and identically distributed, and correspond to a finite set of label observations $Y_t \in \mathcal{Y}$. In a multi-class setting, these label observations for each domain are stacked into an $N_t \times L$ label matrix $Y_t \ \ \forall \ t \in 1:T$, where for the i^{th} row, only one element in $Y_t[i,:] \in \{-1 \ +1\}$ is given the label $+1$, located in the column that corresponds the class numeric (with all other elements being labelled -1), i.e. each column corresponds to the one-vs-all label set for each class label. For each domain/task pair, there is a specific kernel function $k_t(\cdot,\cdot)$ that defines the correlation between data points for a particular domain (aiding bespoke nonlinear mappings for each domain), moving the data points into a Reproducing Kernel Hilbert Space (RKHS) in the form of a kernel matrix $K_t = k_t(X_t, X_t) \in \mathbb{R}^{N_t \times N_t} \ \forall \ t \in 1:T$.

The first stage of the algorithm is to learn an optimal linear projection matrix $A_t \in \mathbb{R}^{N_t \times R}$ for each domain that maps the kernel embedding onto a shared latent space $\mathcal{H}$ as $H_t = A_t^{\mathsf{T}} K_t \in \mathbb{R}^{R \times N_t}, \ \forall \ t \in 1:T$. This step maps the domains on top of each other in the shared latent space, but also performs dimensionality reduction, aiding classification of high-dimensional feature data.

The second stage of the approach seeks to jointly learn a discriminative classifier (for each class label in a one-vs-all sense) in the shared latent space $\boldsymbol{f}_{t,l} = H_t^{\mathsf{T}} \boldsymbol{w}_l + \mathbf{1} b_l \ \forall \ t \in 1:T$ (where $\mathbf{1}$ is a vector of ones) and $\ \forall \ l \in 1:L$, where the parameters of the classifier $\{b_l \in \mathbb{R}^{1\times 1}, \ \boldsymbol{w}_l \in \mathbb{R}^{R\times 1}\}$ are shared for all tasks. The probability of each class label is subsequently obtained, where the *Maximum A Posteriori* (MAP) estimate can be used to determine the predicted label. It is noted that this part of the method has the same construction as a Relevance Vector Machine (RVM) [36]; however, the relevance vectors now relate to the latent space $\mathcal{H}$ rather than individual feature observations. A visual overview of the complete model for the binary classification setting is shown in Figure 1, with

[2]For an overview of the main nomenclature in KBTL the reader is referred to Appendix A.

a visual example of the discriminative classifier in the shared latent space $\mathcal{H}$ when $R = 2$.

Although Gönen and Margolin define KBTL as a transfer learning technology in [10], it is helpful to note that parallels can be drawn with multi-task learning, where knowledge from *multiple domains* are used to improve the learning of a set of tasks (typically in terms of a better predictive function), where each task is weighted evenly [11, 35]. KBTL can be seen as weighting all tasks evenly, even though it allows knowledge to be transferred across domains in the multi-class setting. In fact, some of the transfer learning literature states that multi-task learning can be viewed as a subcategory of transfer learning [11, 35], with multi-task learning being the special case of transfer learning where all tasks are weighted evenly. However, other authors state that multi-task learning and transfer learning are distinct and different technologies [37].

Finally, KBTL is formed as a hierarchical Bayesian model that differs slightly for the binary and multi-class (Figure 2) classification setting [10]. In the following section, the multi-class formulation from [10] is stated, showing the fully conjugate probabilistic model and the variational inference scheme — the interested reader is referred to [10] for specific details on the binary formulation.

2.1. Multi-class classification

Typically, in discriminative classifiers such as a Support Vector Machine (SVM) or RVM, the multi-class problem (i.e. where $L > 2$) involves training multiple classifiers in either a one-vs-all or one-vs-one manner. However, as the discriminative classifier is jointly inferred in the shared latent space for KBTL (due to the desire to transfer information between domains), a multiple classifier approach cannot be taken; this would lead to a separate latent space for each class. Instead a one-vs-all approach can be formed by learning each classifier in the shared latent space; requiring a plate to be added to the binary graphical model for the class label set, such that a one-vs-all classifier is inferred for each class, as depicted in Figure 2. This formulation means that there are L bias parameters and weight vectors b_l and $\boldsymbol{w}_l \; \forall \; l \in 1 : L$, one associated with each class.

For the multi-class problem, the graphical model in Figure 2 states the conditional relationships[3] between the observed parameters $\{K_t, Y_t\}_{t=1}^{T}$ (where the observed training data $D = \{Y_t\}_{t=1}^{T}$), priors $\Xi = \{\{\Lambda_t\}_{t=1}^{T}, \{\boldsymbol{\eta}_l, \gamma_l\}_{l=1}^{L}\}$, hyperparameters[4] $\boldsymbol{\zeta} = \{\kappa_\lambda, \theta_\lambda, \kappa_\eta, \theta_\eta, \kappa_\gamma, \theta_\gamma, \sigma_h^2, \nu\}$, latent variables and model parameters $\Theta = \{\{b, \boldsymbol{w}\}_{l=1}^{L}, \{\{\boldsymbol{f}_{t,l}\}_{l=1}^{L}, A_t, H_t\}_{t=1}^{T}\}$. The variable that forms the prior precision for the projection matrix A_t is denoted $\Lambda_t \in \mathbb{R}^{N_t \times R}$, where each element in the prior is defined by the hyperparameters $\{\kappa_\lambda, \theta_\lambda\}$. The L weights and bias parameters are also specified by variables that form their prior precisions $\boldsymbol{\eta}_l \in \mathbb{R}^{R \times 1}$ and $\gamma_l \in \mathbb{R}^{1 \times 1}$ respectively, where each element is defined by their set of hyperparameters $\{\kappa_\eta, \theta_\eta\}$ and $\{\kappa_\gamma, \theta_\gamma\}$. The latent space has a hyperparameter that specifies the variance of the latent space data σ_h^2, and a non-negative margin hyperparameter ν is specified for the likelihood function (which is discussed in the following section).

Following these definitions, the probabilistic modelling assumptions for the projection and dimensionality reduction part are:

$$\Lambda_t[i, s] \sim \mathcal{G}(\Lambda_t[i, s] \mid \kappa_\lambda, \theta_\lambda) \tag{1a}$$

[3]Note that the model does not have specific distribution assumptions over each domains kernel embedding (and therefore the feature data for that domain), making the approach flexible for a wide range of SHM problems.

[4]For the sake of clarity, dependencies on any of the hyperparameters $\boldsymbol{\zeta}$ are dropped from the notation in this paper. It is also noted that these are hyperparameters of the hierarchical prior structures, and should be selected given prior beliefs of the problem.

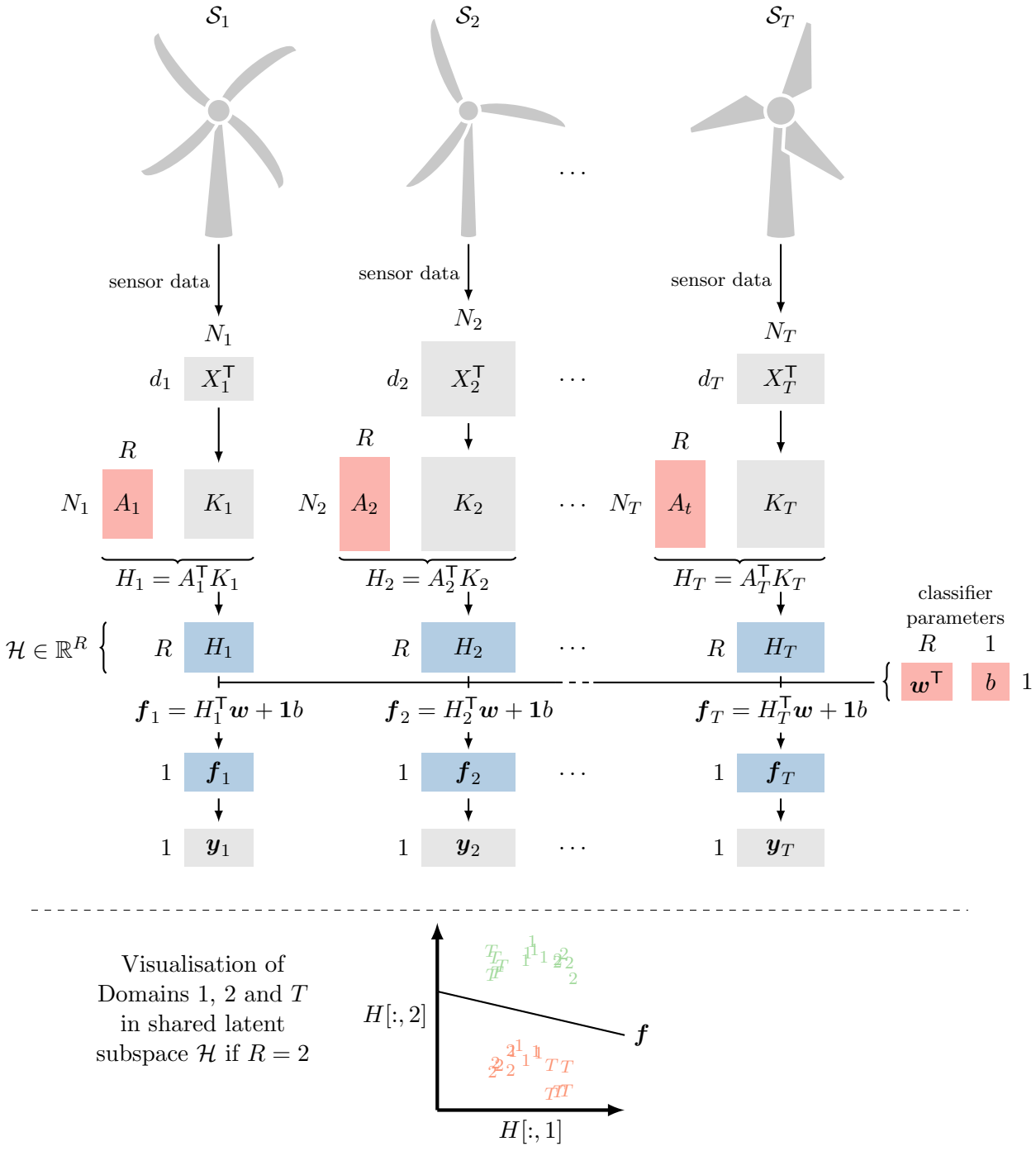

Figure 1: A visual overview of binary KBTL applied to a population of structures $\mathcal{S}_t \; \forall \; t \in 1 : T$. Observed variables are in grey, model parameters are in red, and latent variables are in blue.

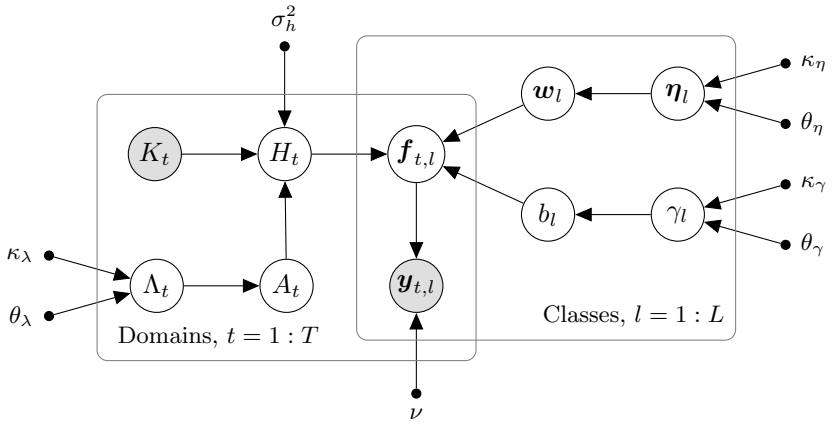

Figure 2: Multi-class classification: graphical model of KBTL (recreated from [10]). The graphical model shows the conditional relationships in the model, where shaded and unshaded nodes represent observed and unobserved variables respectively. Dots represent constants (i.e. hyperparameters in the model), for example $\{\kappa_\lambda, \theta_\lambda\}$ are hyerparameters that define the prior precision of the projection matrix for each domain Λ_t.

$$A_t[i,s] \mid \Lambda_t[i,s] \sim \mathcal{N}(A_t[i,s] \mid 0, (\Lambda_t[i,s])^{-1}) \tag{1b}$$

$$H_t[s,i] \mid A_t[:,s], K_t[:,i] \sim \mathcal{N}(H_t[s,i] \mid A_t[:,s]^\mathsf{T} K_t[:,i], \sigma_h^2) \tag{1c}$$

$\forall\, i \in 1:N_t$, $\forall\, s \in 1:R$, $\forall\, t \in 1:T$, where square brackets denote indices — two elements denote rows and columns of a matrix respectively, one element denotes a vector index, : denotes the set of index values. $\mathcal{G}(\cdot \mid \kappa, \theta)$ refers to a gamma distribution parametrised by shape κ and scale θ parameters, and $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \Sigma)$ refers to a Gaussian distribution parametrised by mean $\boldsymbol{\mu}$ and covariance Σ parameters.

The modelling assumptions for the joint classification problem part are:

$$\gamma_l \sim \mathcal{G}(\gamma_l \,|\, \kappa_\gamma, \theta_\gamma) \tag{2a}$$

$$b_l \mid \gamma_l \sim \mathcal{N}(b \mid 0, \gamma_l^{-1}) \tag{2b}$$

$$\boldsymbol{\eta}_l[s] \sim \mathcal{G}(\boldsymbol{\eta}_l[s]\,;\, \kappa_\eta, \theta_\eta) \tag{2c}$$

$$\boldsymbol{w}_l[s] \mid \boldsymbol{\eta}_l[s] \sim \mathcal{N}(\boldsymbol{w}_l[s] \mid 0, (\boldsymbol{\eta}_l[s])^{-1}) \tag{2d}$$

$$\boldsymbol{f}_{t,l}[i] \mid b_l, \boldsymbol{w}_l, H_t[:,i] \sim \mathcal{N}(\boldsymbol{f}_{t,l}[i] \mid \boldsymbol{w}_l^\mathsf{T} H_t[:,i] + b_l, 1) \tag{2e}$$

$$\boldsymbol{y}_{t,l}[i] \mid \boldsymbol{f}_{t,l}[i] \sim \delta(\boldsymbol{f}_{t,l}[i]\boldsymbol{y}_{t,l}[i] > \nu) \tag{2f}$$

$\forall\, i \in 1:N_t$, $\forall\, s \in 1:R$, $\forall\, l \in 1:L$, $\forall\, t \in 1:T$, where $\delta(\cdot)$ is a Dirac delta function. The non-negative margin parameter ν acts to create a low-density region between the l^{th} class and the rest of the data (in a one-vs-all sense), which aids scaling issues, similar to the concept of a margin in an SVM. In fact, the idea has also been used in semi-supervised learning with Gaussian process models [38].

It is noted that the model setup for the classification part is similar to an RVM [36], where the gamma prior over the weight precision acts to induce sparsity, forming an implicit Student's T prior structure over each weight [39]; this choice of prior means that weights are expected to be zero unless evidence from the likelihood suggests otherwise, leading to sparsity in the posterior weight densities. However, in KBTL, the relevance vectors refer to the dimensions of the latent

space, rather than a particular feature observation; instead the gamma prior over the precision for the projection matrix acts to induce sparsity over each feature observation from each domain. Incorporating sparsity assumptions into the model is useful. Here, sparsity is introduced into the projection matrix ($A_t \; \forall \; t \in 1\!:\!T$) and the classifier weights and bias terms ($\{\boldsymbol{w}_l, b_l\} \; \forall \; l \in 1\!:\!L$). The model must be flexible enough to overcome the domain discrepancies and learn a separating hyperplane between classes; this can result in a large number of parameters. Enforcing sparsity helps avoid overfitting by reducing effectively 'irrelevant' dimensions by forcing their probability mass towards zero, and 'integrating out' these parameters.

2.1.1. Variational inference

The probabilistic model is intractable and cannot be solved in a Bayesian manner in closed form; however, a variational approximation can be formed in order to perform efficient inference. A variational approach means that the log marginal likelihood $\log p(D)$ (log evidence[5]) can be decomposed, by introducing a candidate distribution $q(\{\Theta, \Xi\}) \in Q$,

$$\log p(D) = \mathcal{L}(Q) + \mathcal{KL}(Q||P) \tag{3}$$

where,

$$\mathcal{L}(Q) = \int q(\{\Theta, \Xi\}) \log \frac{p(D, \{\Theta, \Xi\})}{q(\{\Theta, \Xi\})} \, \mathrm{d}\{\Theta, \Xi\} \tag{4}$$

and $\mathcal{KL}(Q||P)$ is the Kullback-Leibler (KL) divergence between $q(\{\Theta, \Xi\})$ and the posterior $p(\{\Theta, \Xi\} \mid D)$,

$$\mathcal{KL}(Q||P) = -\int q(\{\Theta, \Xi\}) \log \frac{p(\{\Theta, \Xi\} \mid D)}{q(\{\Theta, \Xi\})} \, \mathrm{d}\{\Theta, \Xi\} \tag{5}$$

where, as $\mathcal{KL}(Q||P) \geq 0$, $\mathcal{L}(Q)$ is a lower bound on $\log p(D)$. The main concept in variational inference is that maximising the lower bound $\mathcal{L}(Q)$ is equivalent to minimising the KL-divergence, and as $\log p(D)$ is independent of Q, this will make $q(\{\Theta, \Xi\})$ a rigorous approximation of the posterior $p(\{\Theta, \Xi\} \mid D)$. The aim is therefore to choose a suitable form for $q(\{\Theta, \Xi\})$, such that the lower bound $\mathcal{L}(Q)$ is tractable (when the evidence $p(D)$ is not) and easily evaluated, whilst also being flexible, such that the bound is reasonably tight. Typically, variational inference is performed by choosing a meaningful family of Q distributions and maximising the lower bound with respect to Q to find the best approximation in Q.

For the KBTL model, a mean-field approximation can be taken, meaning Q is factorised by partitioning the variables $\{\Theta, \Xi\}$ into independent parts, where each variable in $\{\Theta, \Xi\}$ is governed by its own variational factor. By taking this approach the ensemble approximation of the posterior is,

$$p(\{\Theta, \Xi\} \mid D) \approx q(\{\Theta, \Xi\}) = \prod_{t=1}^{T} [q(\Lambda_t) q(A_t) q(H_t)] \prod_{l=1}^{L} [q(\gamma_l) q(\boldsymbol{\eta}_l) q(b_l, \boldsymbol{w}_l)] \prod_{t=1}^{T} \prod_{l=1}^{L} q(\boldsymbol{f}_{t,l}) \tag{6}$$

[5]Where the evidence $p(D)$ is calculated as $p(D) = \int p(D \mid \{\Theta, \Xi\}) p(\{\Theta, \Xi\}) d\{\Theta, \Xi\}$ when inferring the true posterior, which in this case is intractable.

where,

$$q(\Lambda_t) = \prod_{i=1}^{N_t} \prod_{s=1}^{R} \mathcal{G}(\Lambda_t[i,s] \,|\, \kappa(\Lambda_t[i,s]), \theta(\Lambda_t[i,s])) \tag{7a}$$

$$q(A_t) = \prod_{s=1}^{R} \mathcal{N}(A_t[:,s] \,|\, \mu(A_t[:,s]), \Sigma(A_t[:,s])) \tag{7b}$$

$$q(H_t) = \prod_{i=1}^{N_t} \mathcal{N}(H_t[:,i] \,|\, \mu(H_t[:,i]), \Sigma(H_t[:,i])) \tag{7c}$$

$$q(\gamma_l) = \mathcal{G}(\gamma_l \,|\, \kappa(\gamma), \theta(\gamma)) \tag{7d}$$

$$q(\boldsymbol{\eta}_l) = \prod_{s=1}^{R} \mathcal{G}(\boldsymbol{\eta}_l[s] \,|\, \kappa(\boldsymbol{\eta}_l[s]), \theta(\boldsymbol{\eta}_l[s])) \tag{7e}$$

$$q(b_l, \boldsymbol{w}_l) = \mathcal{N}\left(\begin{bmatrix} b_l \\ \boldsymbol{w}_l \end{bmatrix} \,|\, \mu(b_l, \boldsymbol{w}_l), \Sigma(b_l, \boldsymbol{w}_l) \right) \tag{7f}$$

$$q(\boldsymbol{f}_{t,l}) = \prod_{i=1}^{N_t} \mathcal{TN}(\boldsymbol{f}_{t,l}[i] \,|\, \mu(\boldsymbol{f}_{t,l}[i]), \Sigma(\boldsymbol{f}_{t,l}[i]), \mathrm{a}(\boldsymbol{f}_{t,l}[i]), \mathrm{b}(\boldsymbol{f}_{t,l}[i])) \tag{7g}$$

where $\kappa(\cdot)$, $\theta(\cdot)$, $\mu(\cdot)$ and $\Sigma(\cdot)$ denote shape, scale, mean and covariance parameters for their arguments respectively. Due to the non-negative margin parameter, the factorisation with respect to the discriminative function $\boldsymbol{f}_t$ becomes a truncated Gaussian $\mathcal{TN}(\cdot \,|\, \boldsymbol{\mu}, \Sigma, \mathrm{a}, \mathrm{b})$, fully specified by a mean $\boldsymbol{\mu}$, covariance Σ and two truncation parameters a and b, which define the interval $\mathrm{a} \leq x \leq \mathrm{b}$, outside of which the probability density is zero.

By substituting the approximate posterior in equation (6) into equation (5), the bound on the marginal likelihood becomes (where the dependence on the kernels is dropped for simplicity),

$$\log p(D) \geq \mathbb{E}_{q(\Theta,\Xi)} \left[\log p(D, \{\Theta, \Xi\}) \right] - \mathbb{E}_{q(\Theta,\Xi)} \left[\log q(\Theta, \Xi) \right] \tag{8}$$

which is optimised by iteratively maximising with respect to each factor in equation (6) separately until convergence. The approximate posterior distribution for a particular factor τ is found as,

$$q(\tau) \propto \exp(\mathbb{E}_{q(\{\Theta,\Xi\} \setminus \tau)} \left[\log p(D, \{\Theta, \Xi\}) \right]). \tag{9}$$

Given that the model is constructed from conjugate conditional distributions (gamma-Gaussian and Gaussian-Gaussian), the approximate posterior distributions can be evaluated in closed-form ($\langle f(\cdot) \rangle$ denotes the posterior expectation $\mathbb{E}_{q(\cdot)}[f(\cdot)]$), where the dimensionality reduction part is updated as,

$$\kappa(\Lambda_t[i,s]) = \kappa_\lambda + 1/2 \tag{10a}$$

$$\theta(\Lambda_t[i,s]) = (1/\theta_\lambda + \langle A_t[i,s]^2 \rangle / 2)^{-1} \tag{10b}$$

$$\Sigma(A_t[:,s]) = (\mathrm{diag}(\langle \Lambda_t[:,s] \rangle) + K_t K_t^{\mathsf{T}} / \sigma_h^2)^{-1} \tag{11a}$$

$$\mu(A_t[:,s]) = \Sigma(A_t[:,s])(K_t\langle H_t[s,:]^\mathsf{T}\rangle/\sigma_h^2) \tag{11b}$$

$$\Sigma(H_t[:,i]) = \left(\mathbb{I}/\sigma_h^2 + \sum_{l=1}^{L}\langle \boldsymbol{w}_l\boldsymbol{w}_l^\mathsf{T}\rangle\right)^{-1} \tag{12a}$$

$$\mu(H_t[:,i]) = \Sigma(H_t[:,i])(\langle A_t^\mathsf{T}\rangle K_t[:,i]/\sigma_h^2 + \sum_{l=1}^{L}\langle \boldsymbol{f}_{t,l}[i]\rangle\langle \boldsymbol{w}_l\rangle - \langle b_l\boldsymbol{w}_l\rangle) \tag{12b}$$

meaning that the shared latent space is updated to reflect the performance of *all* L discriminative classifiers. The approximate posterior distributions for the classification part are,

$$\kappa(\gamma_l) = \kappa_\gamma + 1/2 \tag{13a}$$

$$\theta(\gamma_l) = (1/\theta_\gamma + \langle b_l^2\rangle/2)^{-1} \tag{13b}$$

$$\kappa(\boldsymbol{\eta}_l[s]) = \kappa_\eta + 1/2 \tag{13c}$$

$$\theta(\boldsymbol{\eta}_l[s]) = (1/\theta_\eta + \langle \boldsymbol{w}_l[s]^2\rangle/2)^{-1} \tag{13d}$$

$$\Sigma(b_l, \boldsymbol{w}_l) = \begin{bmatrix} \langle\gamma_l\rangle + \sum_{t=1}^{T} N_t & \sum_{t=1}^{T}\mathbf{1}^\mathsf{T}\langle H_t^\mathsf{T}\rangle \\ \sum_{t=1}^{T}\langle H_t\rangle\mathbf{1} & \mathrm{diag}(\langle\boldsymbol{\eta}_l\rangle) + \sum_{t=1}^{T}\langle H_t H_t^\mathsf{T}\rangle \end{bmatrix}^{-1} \tag{14a}$$

$$\mu(b_l, \boldsymbol{w}_l) = \Sigma(b_l, \boldsymbol{w}_l)\begin{bmatrix} \sum_{t=1}^{T}\mathbf{1}^\mathsf{T}\langle \boldsymbol{f}_{t,l}\rangle \\ \sum_{t=1}^{T}\langle H_t\rangle\langle \boldsymbol{f}_{t,l}\rangle \end{bmatrix} \tag{14b}$$

$$\Sigma(\boldsymbol{f}_{t,l}[i]) = 1 \tag{15a}$$

$$\mu(\boldsymbol{f}_{t,l}[i]) = \langle \boldsymbol{w}_l^\mathsf{T}\rangle\langle H_t[:,i]\rangle + \langle b_l\rangle \tag{15b}$$

$$\mathrm{a}(\boldsymbol{f}_{t,l}[i]) = \begin{cases} -\infty, & \text{if } \boldsymbol{y}_{t,l}[i] = -1 \\ \nu, & \text{if } \boldsymbol{y}_{t,l}[i] = +1 \end{cases} \tag{15c}$$

$$\mathrm{b}(\boldsymbol{f}_{t,l}[i]) = \begin{cases} -\nu, & \text{if } \boldsymbol{y}_{t,l}[i] = -1 \\ \infty, & \text{if } \boldsymbol{y}_{t,l}[i] = +1 \end{cases} \tag{15d}$$

where the posterior expectations are defined in Appendix B.

It is noted that KBTL requires the specification of several hyperparameters that are not inferred via these updates. Firstly, $\{\kappa_\lambda, \theta_\lambda, \kappa_\eta, \theta_\eta, \kappa_\gamma, \theta_\gamma\}$ are all hyperprior hyperparameters and must be defined given prior belief about sparsity in the projection matrix, weights and bias respectively. Given the hierarchical model, the posterior will be relatively insensitive to the choice of values for these hyperparameters. A guide for selecting these hyperparameters is as follows: if prior belief suggests the solution should be sparse then κ is small and θ is large, whereas if the problem has a small number of samples then κ and θ should both be small (where both parameters must be non-negative). The remaining hyperparameters are the variance of the latent space σ_h^2, and the margin ν, which can be chosen given prior belief (as used in this paper) or could be updated by numerically optimising the lower bound. Finally, for the model to be fully specified, the dimension

of the latent space R must be chosen. As $R \in \mathbb{Z}^+ \neq 0$, this parameter can be seen more as a model choice, and therefore a model selection technique, or cross-validation, should be used to determine R.

2.1.2. Prediction

The predictive equations are formed by substituting the true posteriors by the approximate posterior distributions. For the projection and dimensionality reduction part, this action leads to substituting $p(A_t \,|\, D)$ by $q(A_t)$. The predictive distribution of the latent space for each domain $\boldsymbol{h}_{t,*}$ for a new data point $\boldsymbol{x}_{t,*}$ (through it's corresponding kernel $k(X_t, \boldsymbol{x}_{t,*}) = \boldsymbol{k}_{t,*}$) is,

$$p(\boldsymbol{h}_{t,*} \mid \boldsymbol{k}_{t,*}, D) = \prod_{s=1}^{R} \mathcal{N}(\boldsymbol{h}_{t,*}[s] \,|\, \mu(A_t[:,s])^{\mathsf{T}} \boldsymbol{k}_{t,*}, \;\; \sigma_h^2 + \boldsymbol{k}_{t,*}^{\mathsf{T}} \Sigma(A_t[:,s]) \boldsymbol{k}_{t,*}) \tag{16}$$

formed from standard Gaussian conditionals. In a similar manner, the predictive distribution for the discriminative classifier can also be solved by approximating $p(b_l, \boldsymbol{w}_l \mid D)$ by $q(b_l, \boldsymbol{w}_l)$,

$$p(f_{t,l,*} \mid \boldsymbol{h}_{t,*}, D) = \mathcal{N}(\boldsymbol{f}_{t,l,*} \,|\, \mu(b_l, \boldsymbol{w}_l)^{\mathsf{T}} [1 \quad \boldsymbol{h}_{t,*}]^{\mathsf{T}}, \;\; 1 + [1 \quad \boldsymbol{h}_{t,*}] \Sigma(b_l, \boldsymbol{w}_l) [1 \quad \boldsymbol{h}_{t,*}]^{\mathsf{T}}) \tag{17}$$

where the probability of an observation belonging to class l is predicted via a truncated Gaussian cumulative density function,

$$p(y_{t,l,*} = +1 \mid f_{t,l,*}, D) = z_{t,l,*}^{-1} \Phi \left(\frac{\mu(f_{t,l,*}) - \nu)}{\Sigma(f_{t,l,*})} \right) \tag{18}$$

where $z_{t,l,*}$ is the normalisation coefficient for a truncated Gaussian distribution, where [a b] (the truncation interval) are for the $y = +1$ case (see Appendix B). This predictive equation for the class label means that $\boldsymbol{y}_{t,*}$ is a $1 \times L$ vector, where each element is the probability of the observation belonging to each class. Finally, a MAP estimate of the class label is formed by finding the element l corresponding to $\max\left(p(\boldsymbol{y}_{t,*} = +1 \mid \boldsymbol{f}_{t,*}, D)\right)$.

3. Case studies

In order to demonstrate the applicability of kernelised Bayesian transfer learning [10] for population-based SHM three applications are considered. The first application considers a population of shear-structures formed from numerical and experimental structures, with each member of the population having a different number of storeys [32], resulting in a heterogeneous population. The feature set for each member of the population is formed from the set of n bending natural frequencies corresponding to the structure's number of storeys, meaning the feature spaces across the population are inconsistent. Binary and multi-class localisation problems are considered for this population, demonstrating the effectiveness of KBTL.

The second application considers an experimental population with one member, namely an aircraft wing from a Gnat trainer aircraft [40, 41]. The aircraft dataset forms two distinct domains, where class distribution shifts have occurred due to changes caused by reattaching inspection panels. In addition, the sensor configuration has changed between the two domains, where a different number of sensors were available in each domain, meaning the feature spaces, formed from transmissibilities

of transducer pairs, are inconsistent. In this application KBTL solves a multi-class localisation problem.

The final application considers a population of five, eight degree-of-freedom systems, formed from numerical and experiential structures, with similar dynamic properties. The feature space for each structure is formed from a frequency response function; however, the measurement properties (i.e. the sample frequency and sample time) are different for each member, leading to inconsistent feature spaces. The SHM problem is classifying the extent of damage located in one of the springs.

For each of the following case studies, KBTL is benchmarked against Heterogeneous Feature Augmentation (HFA) [14] — a supervised heterogeneous transfer learning method that learns a mapping to an augmented (and shared) latent space where a discriminative support vector machine-based classifier is inferred [14] (as with KBTL, and to the authors' best knowledge, HFA also has not been applied in an SHM context before). One main difference between HFA and KBTL is that HFA is constructed for a single source and single target domain. For this reason, all domain combinations are considered in the comparisons. Finally, KBTL is also compared to two single-domain methods (i.e. conventional machine learning with no knowledge transfer): Single-Domain KBTL (SD-KBTL)[6], and an RVM, trained using the algorithm in [36] with a Bernoulli likelihood (in the multi-class setting a one-vs-all approach is used). It is noted that SD-KBTL refers to KBTL trained and tested on a single domain. SD-KBTL may improve performance over an RVM, as the projection inferred by SD-KBTL may lead to a more separable feature space (for the given domain used in training). For each algorithm, a radial-basis kernel is used; the scale hyperparameters for each domain are determined using the median heuristics approach [42] (on the training dataset).

In addition, the total entropy $H(A_t)$ of the projection matrices A_t is calculated for each domain, given their approximate Gaussian posterior distribution.

$$H(A_t) = \sum_{s=1}^{R} - \int p(A_t[:,s]) \log p(A_t[:,s])\, dA_t[:,s] = \sum_{s=1}^{R} \frac{N_t}{2}(1 + \ln(2\pi)) + \frac{1}{2} \ln\left(|\Sigma(A_t[:,s])|\right) \quad (19)$$

The entropy provides a measure of how informative each domain was in learning the latent space $\mathcal{H}$ (given $H_t = A_t^{\mathsf{T}} K_t \; \forall t \in 1:T$) for the complete model. Furthermore, the entropy of each projection matrix gives an indication of whether elements in the matrix have been termed 'irrelevant' from the sparsity assumption on A_t (equations (1a) and (1b)). This measure is particularly useful for PBSHM, as it allows the modellers to assess which structures (and their training datasets) were informative to the overall SHM problem. For example, once a KBTL model has been inferred, domains with low entropy for the projection matrix could be 'pruned', if these structures are only being used to support an SHM task in other particular structures.

3.1. Shear-structures: feature space heterogeneity arising from structural heterogeneity

Two SHM problems are considered for a population of shear-structures, demonstrating the effectiveness of the binary and multi-class forms of KBTL. These two scenarios represent a significant set of PBSHM problems; namely, developing a machine learning model that applies across a set of different structures. Both scenarios consider the same population of seven structures; six are simulated (as

[6]SD-KBTL may offer improved performance over an RVM as it combines the inference of a separating hyperplane in the kernel space with a linear projection, which may aid separability, even when considering a single domain.

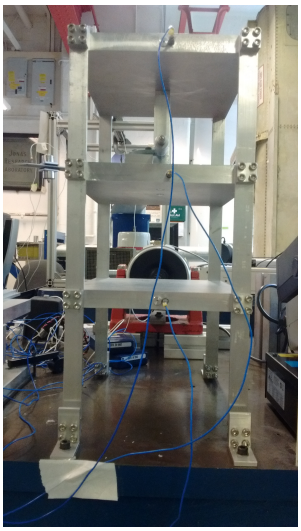

Figure 3: Experimental three-storey shear structure [32].

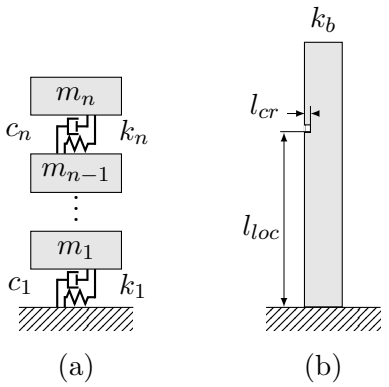

Figure 4: Schematic of the shear structures: panel (a) is a nominal representation of the systems, and panel (b) depicts the cantilever beam component where $\{k_i\}_{i=1}^n = 4k_b$ i.e. the stiffness coefficients in (a) are generated from four times the tip bending stiffness in (b).

lumped-mass models in bending) and one is experimental, as depicted in Figures 3 and 4. Consequently, this particular population highlights the ability of KBTL in leveraging and transferring information from physics-based models to real world structures, even when the simulated structures are *different* to the structure in practice. Each simulated shear structure is represented by n mass $\{m_i\}_{i=1}^n$, stiffness $\{k_i\}_{i=1}^n$ and damping $\{c_i\}_{i=1}^n$ coefficients, assembled via the lumped-mass model in Figure 4. The masses are specified by length l_m, width w_m, thickness t_m and density ρ. The stiffness elements are calculated from four cantilever beams in bending $4k_b = 4(3EI/l_b^3)$, where E is the elastic modulus, I the second moment of area and l_b the length of the beam. The damping coefficients are directly specified and are not derived from a physical model. Each observation, for a particular structure, is composed from random draws from base distributions for E, ρ and c. The properties of the six simulated structures in the population are shown in Table 1[7]. The experimental structure is constructed from aluminium 6082 with dimensions similar to those in Table 1. Observational data, in the form of three natural frequencies, were collected via modal testing, where an electrodynamic shaker applied a 0 to 6553.6Hz broadband white-noise excitation containing 16384 spectral lines (0.2Hz resolution) to the first storey and three uni-axial accelerometers measured the response at each of the three storeys.

In both the binary and multi-class scenarios, damage is introduced to the structure via an open crack (50% of the beam width) at the midpoint of a beam at a particular storey. In the simulated structures, this damage is applied using a reduction in EI (using the model in [43]) across one of the beams i.e. $k_r = 3(3EI/l_b^3) + k_d$, where k_d is the tip stiffness of a cantilever beam subject to an open crack of length l_{cr} at location l_{loc} along the length of the beam. Similarly, a saw cut was applied to one of the beams for the experimental structure (50% of the beam width at the midpoint of the front-right beam at a particular storey). It is noted that all damage scenarios were for single-site damage cases. Lastly, the features in this case study are damped natural frequencies i.e. $X_t = \{\boldsymbol{\omega}_{t,i}\}_{i=1}^{N_t} \in \mathbb{R}^{N_t \times d_t}\ \forall t \in 1\!:\!T$ (where d_t is the number of degrees-of-freedom n for each domain) — leading to a clear need for heterogeneous transfer learning.

[7]The approximate range of mean properties for the undamaged set of structures are $5 \leq m \leq 17$kg, $4.0 \leq c \leq 6.4$Ns/m and $0.5 \times 10^5 \leq k \leq 3 \times 10^5$N/m.

Table 1: Properties of the six simulated structures in the heterogeneous population case study. Degrees-of-freedom (DOF) are denoted by d_t.

$\mathcal{D}$	DOF d_t	Beam geometry $\{l_b, w_b, t_b\}$ mm	Mass geometry $\{l_m, w_m, t_m\}$ mm	Elastic modulus E GPa	Density ρ kg/m^3	Damping c Ns/m
1	4	$\{185, 25, 6.35\}$	$\{350, 254, 25\}$	$\mathcal{N}\left(71, 1.0 \times 10^{-9}\right)$	$\mathcal{N}\left(2700, 10\right)$	$\mathcal{G}\left(50, 0.1\right)$
2	8	$\{200, 35, 6.25\}$	$\{450, 322, 35\}$	$\mathcal{N}\left(70, 1.2 \times 10^{-9}\right)$	$\mathcal{N}\left(2800, 22\right)$	$\mathcal{G}\left(8, 0.8\right)$
3	10	$\{177, 45, 6.15\}$	$\{340, 274, 45\}$	$\mathcal{N}\left(72, 1.3 \times 10^{-9}\right)$	$\mathcal{N}\left(2550, 25\right)$	$\mathcal{G}\left(25, 0.2\right)$
4	3	$\{193, 32, 5.55\}$	$\{260, 265, 32\}$	$\mathcal{N}\left(75, 1.5 \times 10^{-9}\right)$	$\mathcal{N}\left(2600, 15\right)$	$\mathcal{G}\left(20, 0.3\right)$
5	5	$\{165, 46, 7.45\}$	$\{420, 333, 46\}$	$\mathcal{N}\left(73, 1.4 \times 10^{-9}\right)$	$\mathcal{N}\left(2650, 20\right)$	$\mathcal{G}\left(45, 0.1\right)$
6	3	$\{175, 40, 6.05\}$	$\{400, 310, 41\}$	$\mathcal{N}\left(69, 1.1 \times 10^{-9}\right)$	$\mathcal{N}\left(2750, 18\right)$	$\mathcal{G}\left(20, 0.2\right)$

3.1.1. Binary case study

The binary case study considers a scenario where the SHM problem $\mathcal{SP}$ is a damage-detection problem (which can also be seen as a two-class damage localisation problem). For a binary classification problem KBTL can only be used as a multi-task learner, as each domain requires some examples of each class in order to infer the dimensionality reduction matrix $A_t \; \forall \; t \in 1 : T$. As such, the aim of the algorithm is to infer a joint latent space that aids classification for all domains equally. In this scenario the two classes[8] are an undamaged state, denoted by '0' and an open crack applied to the first storey[9], denoted by '1'.

Class imbalance is common in SHM training datasets, as often more undamaged-state observations can be collected than damaged. In order to reflect this general scenario, here each domain has a different degree of class imbalance, presented in Table 2, highlighting that for some domains, such as Domain Four, it would be difficult to create a machine learner that would generalise well, based on such sparse damage observations.

Given the small sample sizes in each domain, the hyperparameters of the projection matrix were $(\kappa_\lambda, \theta_\lambda) = (1 \times 10^{-3}, 1 \times 10^{-3})$, and the hyperparameters of the bias and weights were $(\kappa_\gamma, \theta_\gamma) = (\kappa_\eta, \theta_\eta) = (1 \times 10^{-3}, 1 \times 10^{-3})$ for both KBTL and SD-KBTL. In addition, the standard deviation

[8]These labels are converted into the $[-1 \; +1]$ space for each class before applying the algorithm.

[9]The 'damaged' class for all domains has been kept the same, so that it is clear what information is being transferred across the domains.

Table 2: Shear-structure population, binary case study: number of data points in each class for each domain. * denotes the domain for the experimental structure.

Domain	**Training**		**Testing**	
$\mathcal{D}$	$y = 0$	$y = 1$	$y = 0$	$y = 1$
1	110	25	1000	1000
2	60	20	1000	1000
3	70	50	1000	1000
4	120	10	1000	1000
5	200	25	1000	1000
6	100	10	1000	1000
7*	3	3	2	2

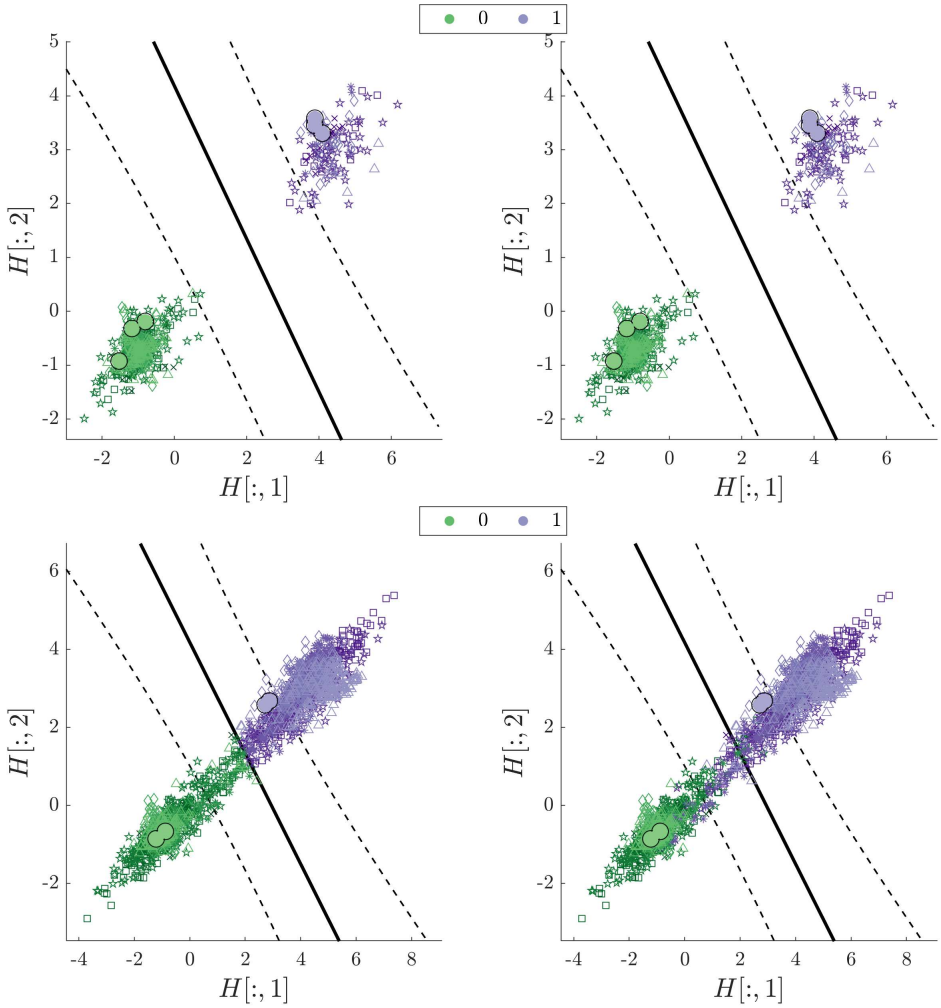

Figure 5: Shear-structure population, binary case study. A visualisation of the expectation of the posterior latent space for the training (top) and testing (bottom) data, where $H[:,1]$ and $H[:,2]$ are the first and second dimensions of the $R = 2$-dimensional shared latent space $\mathcal{H}$. The left panels depict the discriminative classifier (mean (-) and three standard deviations (- -)) as well as the predicted label MAP estimates. The right panels present the true labels. Each domain is denoted by a different symbol, $\{\times, \square, \star, *, \diamond, \triangle, \bullet\}$, for domains $1:7$ respectively.

of the latent space was set to $\sigma_h = 0.25$, reflecting an expectation that the mapping will be relatively certain, given that it is a two-class problem from domains of relatively low dimensions. Furthermore, a margin $\nu = 0$ was chosen, with the latent space dimension $R = 2$ selected to aid visualisation of the algorithm.

The expectation of the posterior latent space, for all domains, is depicted in Figure 5 for both the training $\mathbb{E}\left[p(H_t \mid D, K_t)\right]$ (top panels) and testing $\mathbb{E}\left[p(H_{t,*} \mid D, K_{t,*})\right]$ data (bottom panels). In addition, the discriminative classifier (mean and three standard deviations) in the expected latent space is illustrated in Figure 5 (left panels) along with the MAP estimate of the predicted class labels (where a point belongs to '+1' if $p(y_{t,*} = +1 \mid f_{t,*}, D) \geq 0.5$ and '-1' if < 0.5). The true labels are also shown in Figure 5 (right panels) as a comparison. Furthermore, the testing classification accuracies are presented in Figure 6 as a measure of performance. It is noted that HFA is not applied on the same domain that has been used as the source domain, as the aim of the algorithm is to transfer knowledge to a different target domain. The sensitivity and specificity are stated for

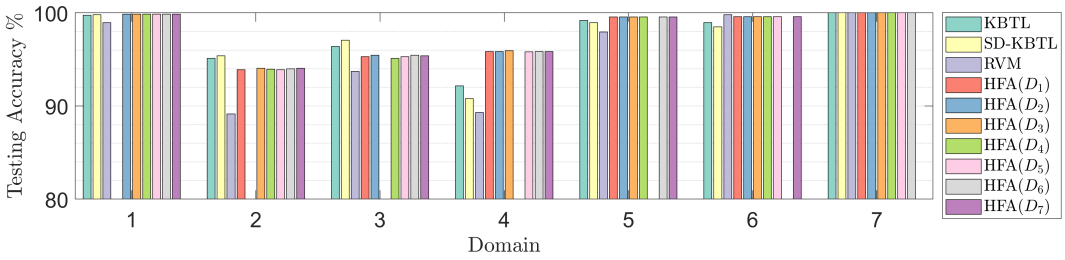

Figure 6: Shear-structure population, binary case study: testing accuracies. HFA(D_i) refers to the results where the i^{th} domain is considered the source domain.

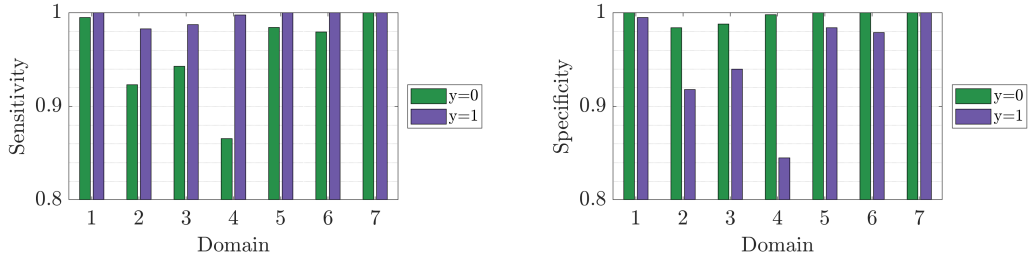

Figure 7: Shear-structure population, binary case study: testing sensitivity and specificity.

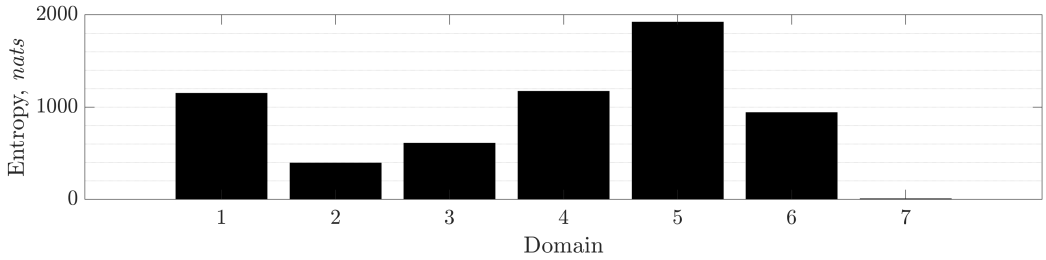

Figure 8: Shear-structure population, binary case study: total entropy of the projection matrix A_t per domain.

KBTL in Figure 7, indicating the models true positive and true negative rates. Finally, Figure 8 presents the entropy of the projection matrices for each domain.

In this example, KBTL outperforms both SD-KBTL and RVM classifiers when considering the average performance across all domains, with an average classification accuracy of 97.4% compared to 96.4% and 95.6% for SD-KBTL and RVM. It also has a high sensitivity and specificity for each class (> 0.8) with a particular high true positive rate for '$y = 1$' across all domains. However, in terms of average accuracy, KBTL is either matched or outperformed by HFA (when trained using any domain as the source domain, with accuracies between 97.4%-98.4%). Considering each domain individually, it can be seen that KBTL does not outperform the other classifiers on a domain-by-domain basis. KBTL is outperformed by HFA in all but two domains, Domain Two and Domain Three, which are the domains with the least number of observations in training. This shows that KBTL may improve performance of domains with a small number of observations, transferring some knowledge, but may reduce performance in other domains to achieve this. An interesting observation is that the entropy of the projection matrices is lowest where classification performance gains are highest (i.e. Domains Two, Three, and Seven), which may evidence that the algorithm is trying to overcome this lack of information in training by slightly reduction performance in more informative domains. Compared to an RVM classifier trained on each domain individually, KBTL has a higher accuracy

in all domains but Domain Six, where an RVM classifier outperforms all the other approaches. The reason for this difference in performance is likely a result of clear class separability in Domain Six (given that the structure has the joint lowest number of stories at three), meaning a classifier trained on Domain Six generalises well. In the comparison with SD-KBTL, KBTL is only outperformed in Domains Two and Three. This outcome is likely a result of the fact that these domains have relatively small amounts of observations for both classes (compared with the other domains), and that they have the largest number of degrees-of-freedom (eight and ten respectively). These two factors will contribute to classes that have a large degree of overlap in their original domain, as changes in natural frequency will be less sensitive to damage at the first storey compared to the other domains (due to a lower amount of strain energy); additionally there is a lower amount of information to explain the class boundary. SD-KBTL can learn a nearer-optimal dimensionality reduction and classifier for these domains individually than KBTL can when considering the joint problem across domains.

3.1.2. Multi-class case study

One of the main aims in PBSHM is the ability to transfer knowledge and create machine learning-based models that generalise well across different structures. In the light of this goal, the multi-class shear-structure case study has been designed to demonstrate these aims. The SHM problem $\mathcal{SP}$ considered in this case study is one of damage localisation, where the label space $\mathcal{Y} = \{$'0', '1' '2' '3'$\}$ is consistent across all domains; class '0' is undamaged and classes '1' to '3' relate to damage at floors one to three respectively. The label space is consistent, as the smallest structure is a three-storey building structure, and therefore, all the class labels can exist in all domains, aiding the potential for positive transfer.

As with the binary case study, the training dataset has been designed to demonstrate several aims of population-based SHM. Namely, knowledge transfer and creating a machine learner that generalises well across all domains, where some may have sparse damage observations. Table 3 displays the number of data points used in the training and testing. It can be seen that Domain One requires knowledge transfer, as no observations of class '3' are present in training. In addition, Domains Four, Five and Six also require a degree of knowledge transfer, as each domain has a class with only one data point. Furthermore, Domains Two and Three, with eight and ten stories, have a larger degree of overlap between the classes in their original feature space, where there is potential for aiding classification by leveraging knowledge in other domains.

Given the small sample sizes in each domain, the hyperparameters for KBTL and SD-KBTL were $(\kappa_\lambda, \theta_\lambda) = (\kappa_\gamma, \theta_\gamma) = (\kappa_\eta, \theta_\eta) = (1 \times 10^{-3}, 1 \times 10^{-3})$ for the projection matrix, bias and weight

Table 3: Shear-structure population, multi-class case study: number of data points in each class for each domain. * denotes the domain for the experimental structure.

Domain	**Training**				**Testing**			
$\mathcal{D}$	$y=0$	$y=1$	$y=2$	$y=3$	$y=0$	$y=1$	$y=2$	$y=3$
1	120	60	60	0	1000	1000	1000	1000
2	50	20	25	20	1000	1000	1000	1000
3	55	30	30	25	1000	1000	1000	1000
4	70	45	1	45	1000	1000	1000	1000
5	140	1	70	70	1000	1000	1000	1000
6	200	50	50	1	1000	1000	1000	1000
7*	3	3	3	0	2	2	2	0

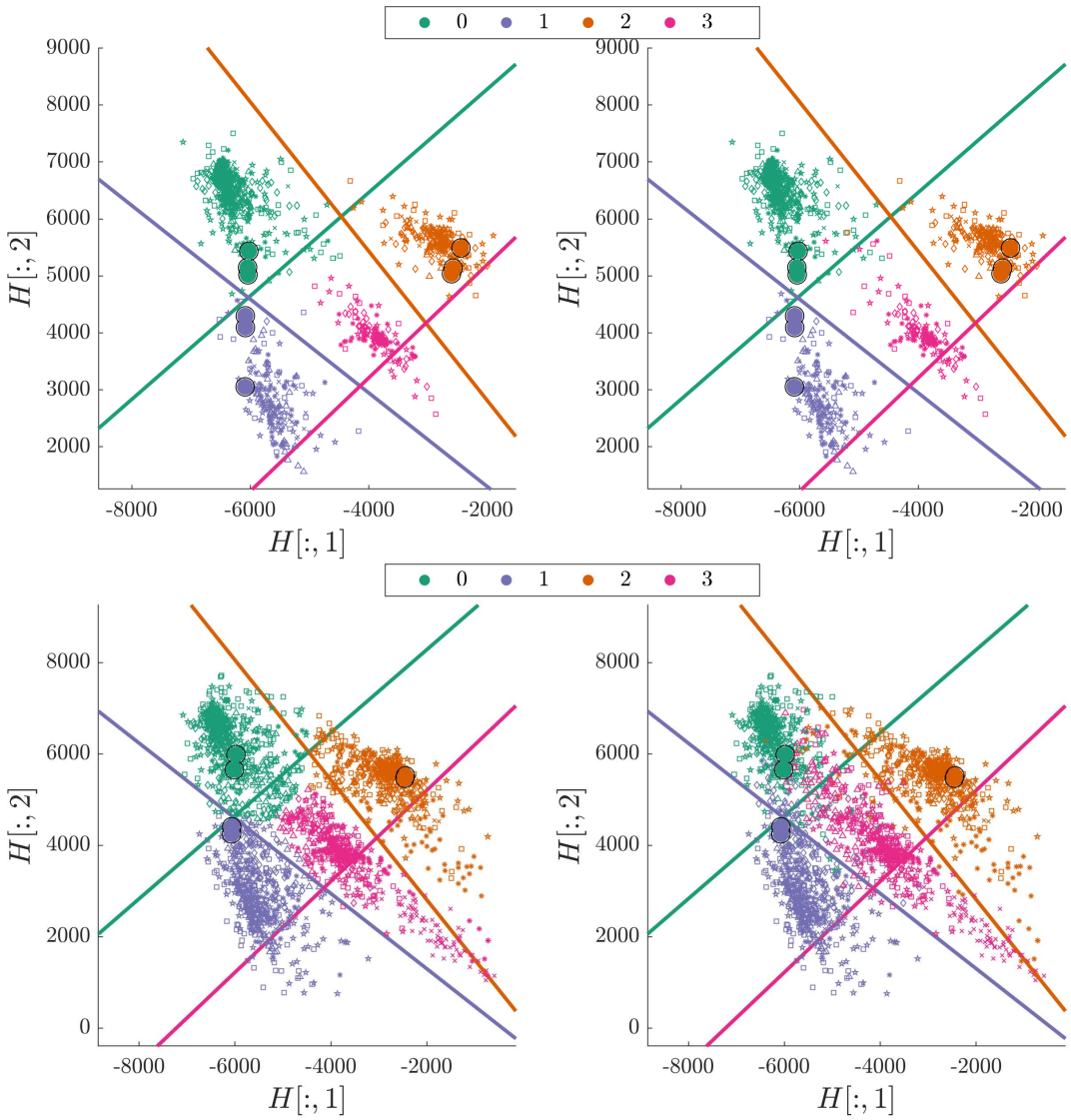

Figure 9: Shear-structure population, multi-class case study. A visualisation of the expectation of the posterior latent space for the training (top) and testing (bottom) data, where $H[:,1]$ and $H[:,2]$ are the first and second dimensions of the $R = 2$-dimensional shared latent space $\mathcal{H}$. The left panels depict the one-vs-all discriminative classifiers (mean (-) and three standard deviations (- -)) for each class (indicated by the colour) as well as the predicted label MAP estimates. The right panels present the true labels. Each domain is denoted by a different symbol, $\{\times, \square, \star, *, \diamond, \triangle, \bullet\}$, for domains $1:7$ respectively.

variances respectively. The latent space standard deviation was $\sigma_h = 6$ reflecting the uncertainty in mapping the four classes in the latent space, which in turn aids the spreading of the classes into the latent space. Finally, the margin $\nu = 1$ was chosen to aid separability between the classes, and the latent space dimensionality was $R = 2$ to aid visualisation. All the discriminative approaches in this case study are trained in a one-vs-all manner.

Figure 9 illustrates the expected posterior latent space for both the training $\mathbb{E}\left[p(H_t \mid D, K_t)\right]$ (top panels) and testing $\mathbb{E}\left[p(H_{t,*} \mid D, K_{t,*})\right]$ data (bottom panels). As in the binary case study, the discriminative classifiers are depicted in this latent space (left panels), where the MAP estimate of the labels are shown (i.e. the label is assigned to the class with the largest probability compared to

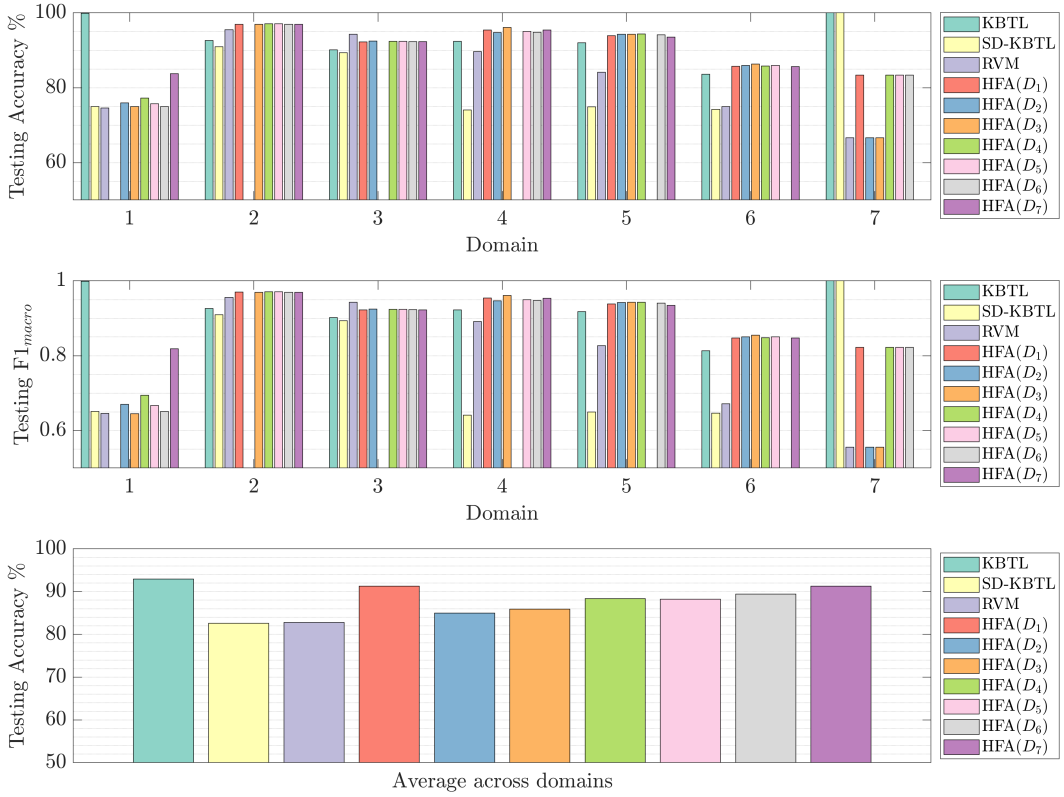

Figure 10: Shear-structure population, multi-class case study: top and middle panels display the testing accuracies and macro F1-scores and the bottom panel shows the average accuracy across all domains. HFA(D_i) refers to the results where the i^{th} domain is considered the source domain.

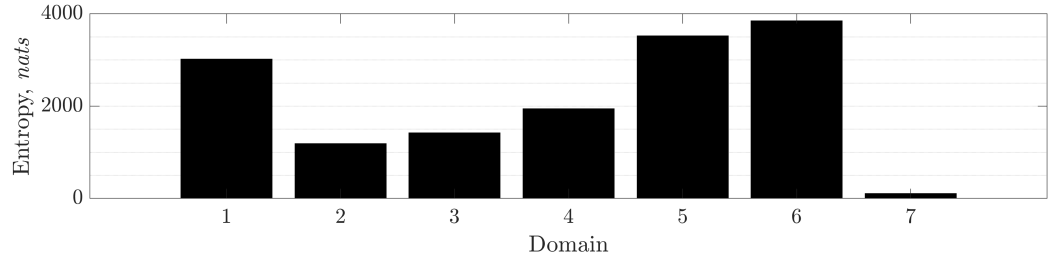

Figure 11: Shear-structure population, multi-class case study: total entropy of the projection matrix A_t per domain.

the other classes). The true labels are also displayed (right panels) as a visual comparison. The testing accuracies and macro F1-scores are presented in Figure 10, along with the average test accuracy across all domains and Figure 11 states the entropy of the projection matrices.

In this case study KBTL performs best when considering the average performance across all domains, with an average classification accuracy of 92.9% compared to the highest HFA accuracy (where Domain One was the source domain) of 91.3%, and the SD-KBTL and RVM classifiers, 82.6% and 82.8%, respectively. Comparing KBTL and HFA a pattern emerges. In domains where a class label is missing (Domains One and Seven), KBTL significantly outperforms HFA, with 100% classification accuracies compared to all HFA accuracies, that are below 84%, demonstrating that KBTL has successfully transferred label knowledge to that particular domain. For example, in

Domain One, knowledge about three of the known classes helps to anchor the inferred projection matrix in training, meaning KBTL is able to place class '3' in the correct part of the latent space for Domain One. However, in domains where some knowledge about all the class labels exists, HFA consistently outperforms KBTL, although very slightly. This effect is similar to that shown in the binary case study, where KBTL may reduce performance slightly in domains where knowledge is most complete in order to improve performance in domains with sparse data. This observation, to a degree, is supported by the entropy of the projection matrices, where Domains Five and Six in particular are being leveraged (i.e. they have high entropy) to infer the shared latent space; however it is noted that the projection matrix for Domain One also has high entropy. Furthermore, KBTL outperforms the other two approaches in all domains apart from Two and Three, where the RVM classifier is best; in fact in Domain Three the RVM classifier outperforms all other methods. This outcome is likely a result of Domains Two and Three having the most even balance of data points for each class. It is interesting to note that the macro F1-score demonstrates that SD-KBTL has failed to capture the behaviour of Domains Four, Five and Six. In training, the SD-KBTL projection for these domains has little information about the classes with one data point, this affects the inferred mapping to the latent space. As a consequence, the inferred mapping is unlikely to be optimal for that class, and therefore does not generalise well for the class with one data point, reflected in testing, leading to accuracies all below 75%. Similarly, the RVM also performs worse than KBTL in these domains, but unlike SD-KBTL, it is able to capture partial information about the class with one data point in training. Finally, it is noted that KBTL achieves 100% classification accuracy on the experimental data — something that is not achieved by HFA or the RVM classifier.

3.2. Gnat aircraft wing: feature space heterogeneity arising from sensor configurations

One reason that feature spaces may be heterogeneous and inconsistent is due to different sensor configurations. The case study in this application considers a scenario where two different sensor configurations, shown in Figure 12, were attached to the same aircraft wing (part of Gnat trainer aircraft), which form two domains. In addition, domain shift occurs between the two domains, with the data distributions changing due to slight difference in the boundary conditions of inspection panels following replacement, meaning some form of transfer learning is required.

The Gnat dataset has been well-studied [40, 41, 44–47], and is an experimental dataset in which an aircraft wing was excited with a band-limited Gaussian noise and the response measured at several locations via uni-axial accelerometers. Inspection panels were subsequently removed, giving pseudo-damage scenarios. The feature set was formed from transmissibilities, calculated from accelerometer pairs, with 1024 spectral lines between 1024-2048Hz; for more information about the dataset, the interested reader is referred to [41].

This case study considers part of the dataset, where a local group of three inspection panels (P1, P2 and P3) are removed sequentially (as seen in Figure 12), i.e. $Y = 0$ when no panels are removed, $Y = 1$ when P1 is removed, $Y = 2$ when P2 is removed and $Y = 3$ when P3 is removed. 100 measurements were collected for each class, and the measurement sequence repeated with the inspection panels being reattached, i.e. data were collected for classes '0' to '3' before the panels were reattached and a second sequence of data collected for classes '0' to '3', with fastener torque being controlled [41]. In this case study, the two data sequences are categorised as two different domains, each with 400 observations. Furthermore, in Domain One two transmissibilities are available, $T1 = A1/AR$ and $T2 = A2/AR$, whereas in Domain Two an additional sensor is added, meaning that an extra transmissibility path $T3 = A3/AR$ is included in the set $\{T1, T2, T3\}$. The two features spaces for each domain are therefore inconsistent and heterogeneous, with $\mathcal{X}_1 = \{T1, T2\} \in \mathbb{R}^{N_1 \times 2048}$ and $\mathcal{X}_2 = \{T1, T2, T3\} \in \mathbb{R}^{N_2 \times 3072}$. The training datasets are:

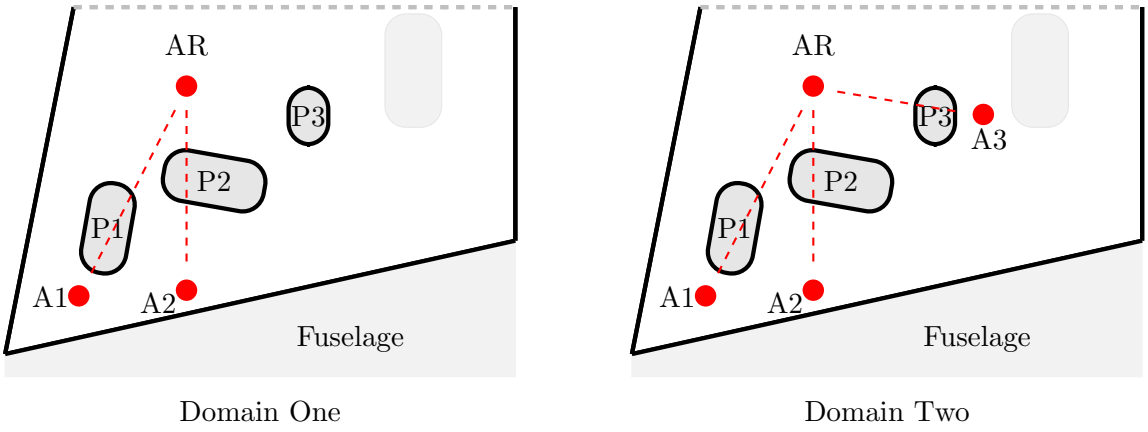

Figure 12: Schematics of the Gnat aircraft wing with two sensor configurations (not to scale).

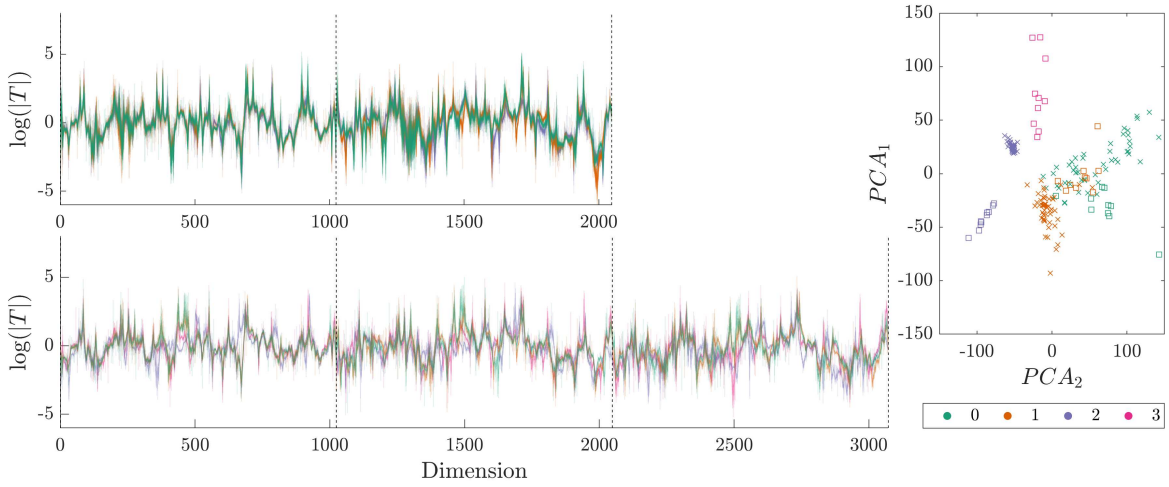

Figure 13: Visualisation of the Gnat aircraft training feature spaces for Domains One and Two. The top left and bottom left panels show the stacked log transmissibilities of Domains One and Two respectively, with the vertical lines indicating the start of a new transmissibility. The right panel presents a comparison of the first two principal components for each domain: Domain One (×) and Domain Two (□).

- Domain One consists of 50 observations of each class, apart from $Y = 3$ where *no* observations are available, i.e. $N_1^{train} = 150$.
- Domain Two has 10 observations of each class, i.e. $N_2^{train} = 40$.

The remaining data points are used as independent test sets for the two domains. Figure 13 presents a visualisation of the training data feature spaces for Domains One and Two, stating their stacked transmissibilities for each observation, along with a comparison of each domain's first two principal components. If no domain shift had occurred, then one might expect that (up to a certain dimension), the PCA components would be well aligned (i.e. close in Euclidean space); this would then allow a conventional classification approach, where a classifier is inferred in a PCA subspace for Domain One and applied to Domain Two. As seen in Figure 13 the PCA subspaces change significantly between the two domains, even for the first two principal components, meaning that domain shift has occurred, i.e. the underlying data distributions have changed between Domains One and Two.

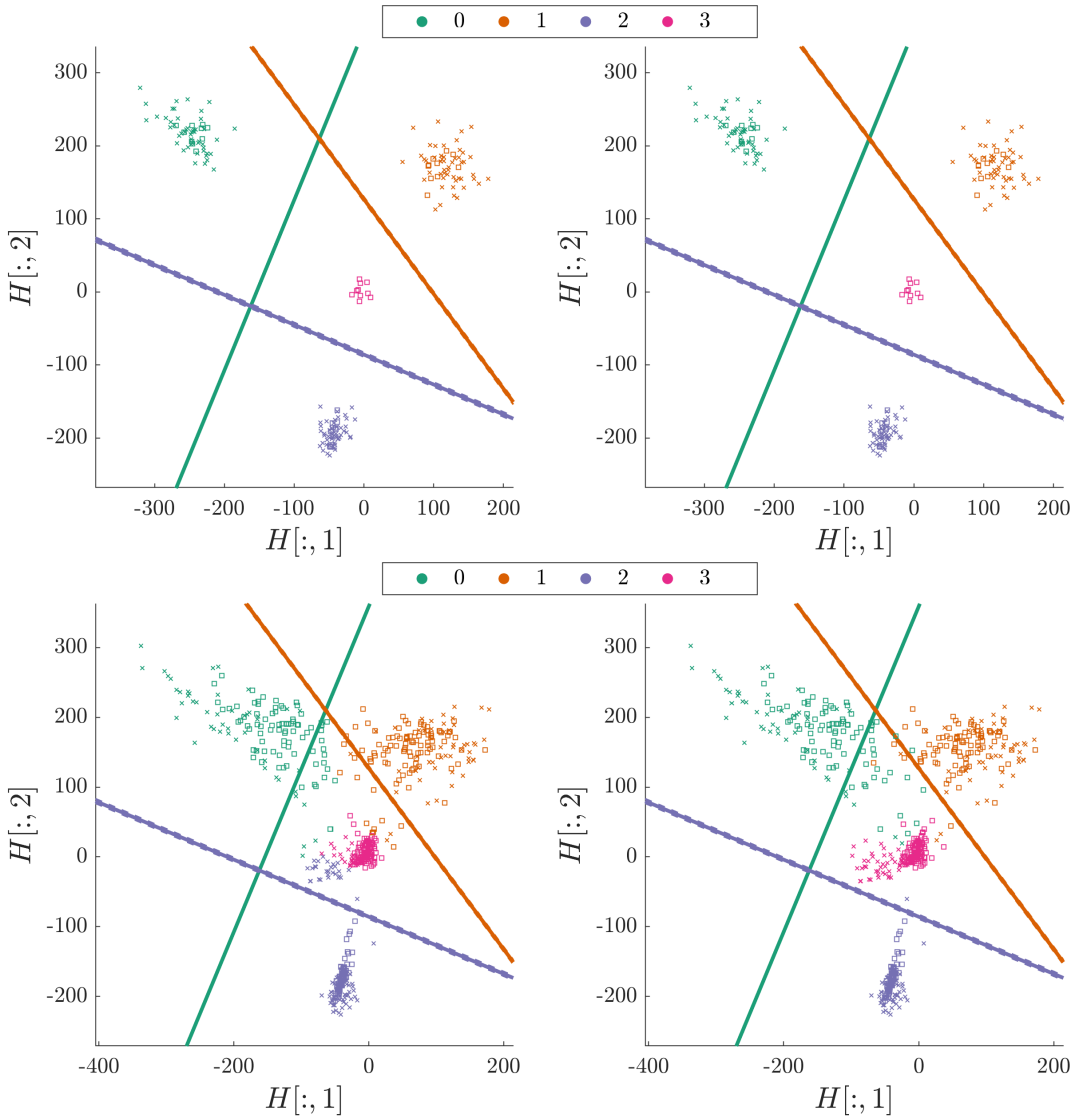

Figure 14: Gnat aircraft case study. A visualisation of the expectation of the posterior latent space for the training (top) and testing (bottom) data, where $H[:,1]$ and $H[:,2]$ are the first and second dimensions of the $R = 2$-dimensional shared latent space $\mathcal{H}$. The left panels depict the one-vs-all discriminative classifiers (mean (-) and three standard deviations (- -)) for each class (indicated by the colour) as well as the predicted label MAP estimates. The right panels present the true labels. Each domain is denoted by a different symbol for Domains One (×) and Two (□).

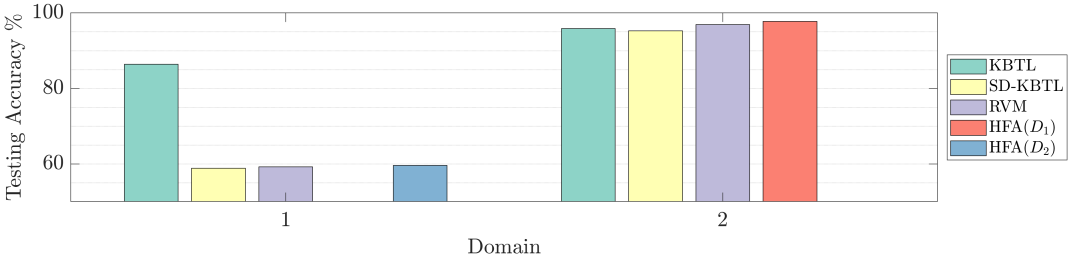

Figure 15: Gnat aircraft case study: testing accuracies. HFA(D_i) refers to the results where the i^{th} domain is considered the source domain.

KBTL was used to infer a joint classification model for the two domains with the aim of aiding classification of $Y = 3$ on Domain One (where no observations of $Y = 3$ were available in training). Once more, there were a small number of observations in each domain, meaning that the hyper-parameters were set to $(\kappa_\lambda, \theta_\lambda) = (\kappa_\gamma, \theta_\gamma) = (\kappa_\eta, \theta_\eta) = (1 \times 10^{-3}, 1 \times 10^{-3})$. The latent space standard deviation was $\sigma_h = 1$ reflecting a degree of uncertainty in mapping, with a small margin of $\nu = 0.1$ used to try and aid separability. For visualisation purposes $R = 2$ was used.

The inferred expected posterior latent space, classifier functions and label MAP estimates (left panels) are shown in Figure 14, where the top panels show the training data and the bottom panels the testing data. The right panels show the expected posterior latent space where the data points are coloured according to their true labels. It can be seen that class $Y = 3$ is placed at the centre of the latent space around the origin, reflecting that little information is known about the class, with only ten observations in Domain Two. The testing accuracies are presented in Figure 15.

The testing accuracies demonstrate that KBTL has managed to transfer knowledge from Domain Two to Domain One, which can be observed in the latent space in Figure 14, although it is noted that the uncertainty in the mapping has led to some observations being mislabelled. Compared to HFA and the single domain approaches, KBTL has provided significant benefits to Domain One. However, in Domain Two, KBTL is slightly outperformed by both HFA (where Domain One was the source domain) and by the RVM, 95.8% compared to 96.9% and 97.8%. In this case study, the entropy for the projection matrix for Domain One was 778 *nats* compared to 183 *nats* in Domain Two. The entropy indicates that their is likely to be lower information content in Domain Two for learning the shared latent space than from Domain One, which may be due to the lower dimension of Domain One's original feature space, and higher total number of data points across classes '0' to '2'.

3.3. Eight degree-of-freedom systems: feature space heterogeneity arising from measurement properties

The third application considers a population of eight degree-of-freedom systems, one experimental, displayed in Figure 16 and four numerical. Heterogeneous transfer learning is required, as the feature spaces for each domain are formed from frequency response functions (FRFs) where changes in the measurement properties, namely the sample frequencies and sample time, have led to the FRFs having a different number of frequency bins with different spacing within the same frequency range.

The four simulated eight degree-of-freedom systems in the population were generated using random mass, damping and stiffness coefficients, drawn from Gaussian distributions where the mean values were defined as those identified by Bull *et al.* from the experimental structure in [7]; where the interested reader can find more details about the analysis and the model structure. The experimental dataset comes from experiments performed at Los Alamos National laboratory [4]. The sampled coefficients for the i^{th} member of the population are defined as $m_j^{(i)} \sim \mathcal{N}(m_j, 0.05 \times m_j)$, $c_j^{(i)} \sim \mathcal{N}(c_j, 0.05 \times c_j)$ and $k_j^{(i)} \sim \mathcal{N}(k_j, 0.05 \times k_j)\ \forall\, j = 1 : 8$. The measurement properties for each member of the population are defined in Table 4 (numerical and experimental). Frequency response functions were obtained in a similar manner to the experimental structure [4] (Figure 16), where a Gaussian noise input was applied as a force to the first mass and the acceleration response measured at the eighth mass, with the simulated noise being normally distributed with zero mean and a signal-to-noise ratio (in terms of variance) of 40dB. The FRFs for each domain were truncated to 0-128Hz, as this is where the main structural resonances occurred.

The structural health monitoring problem $\mathcal{SP}$ was classifying damage extents, where damage was

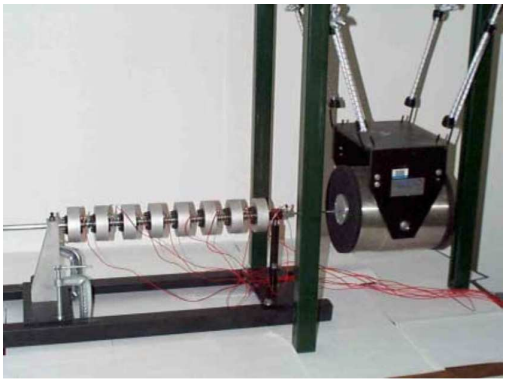

Figure 16: Experimental eight degree-of-freedom system [4].

Table 4: Measurement properties of the eight degree-of-freedom structures. * denotes the domain for the experimental structure.

Domain, $\mathcal{D}$	1	2	3	4	5*
Sample frequency, Hz	512	1024	512	1024	400.25
Sample time, sec	7	8	10	15	8
Feature space dimension	897	1025	1281	1921	1025

simulated through a reduction in stiffness at spring k_5 [7], as simulated in the experimental case study [4]. The damage extents correspond to a reduction in stiffness of 7%, 14% and 24%, i.e. $Y = 0$ when undamaged, $Y = 1$ when $k_5^d = 0.93k_5$, $Y = 2$ when $k_5^d = 0.86k_5$ and $Y = 3$ when $k_5^d = 0.76k_5$. The experiments only considered class labels $Y = 0$ and $Y = 3$, where four observations were obtained for each class. In contrast, 50 observation were simulated for each class label for the numerical systems. In the following case study, Domains One to Four are simulated, where the training data for each domain comprises of nine observations of $Y = 0$, eight of observations of $Y = 1$, eight of observations of $Y = 2$ and nine observations of $Y = 3$. For the experimental domain — Domain Five — only a single observation of $Y = 0$ and $Y = 1$ are used in training. An independent validation dataset was also constructed with the same number of different observations as the training set, and this was used to cross-validate the parameter R, the dimension of the latent space in KBTL. Finally, the remaining data in each domain are used as an independent test set.

The KBTL hyperparameters were defined as $(\kappa_\lambda, \theta_\lambda) = (\kappa_\gamma, \theta_\gamma) = (\kappa_\eta, \theta_\eta) = (1 \times 10^{-3}, 1 \times 10^{-3})$, reflecting the small sample sizes in each domain. The latent space standard deviation was $\sigma_h = 5$ due to the expected uncertainty in mapping, with a small margin of $\nu = 1$ to aid separability. All the discriminative approaches in this case study are trained in a one-vs-all manner. Cross-validation was performed on the independent validation dataset in order to select R, where the optimal $R = 10$. The classification results for each approach are shown in Figure 17, and the entropy of the projection matrices in Figure 18.

In terms of average accuracy KBTL outperforms all the other methods, with an average testing accuracy of 98.5%. HFA, trained using Domain Five as the source domain, is the closest in terms of average performance, with an accuracy of 98.3%. However, this model does not make any predictions for Domain Five, where observations of classes one and two are not present, as this would mean the source and target domains are the same. In cases where HFA is trying to make classification predictions on Domain Five, the performance drops to a maximum of 92.8% (when Domain One is considered the source domain). On a domain-by-domain basis, KBTL achieves

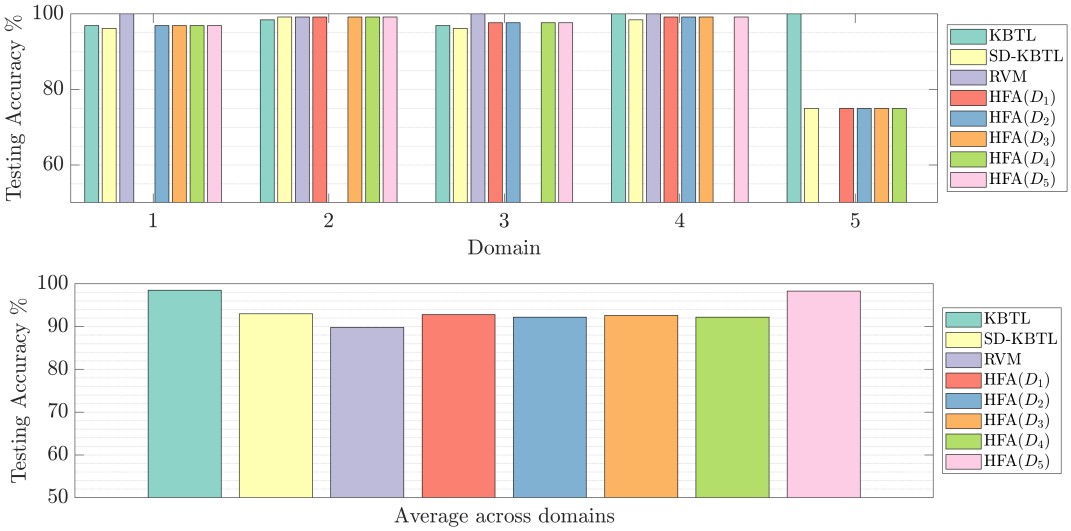

Figure 17: Eight degree-of-freedom case study: top panel displays the testing accuracies and the bottom panel shows the average accuracy across all domains. HFA(D_i) refers to the results where the i^{th} domain is considered the source domain.

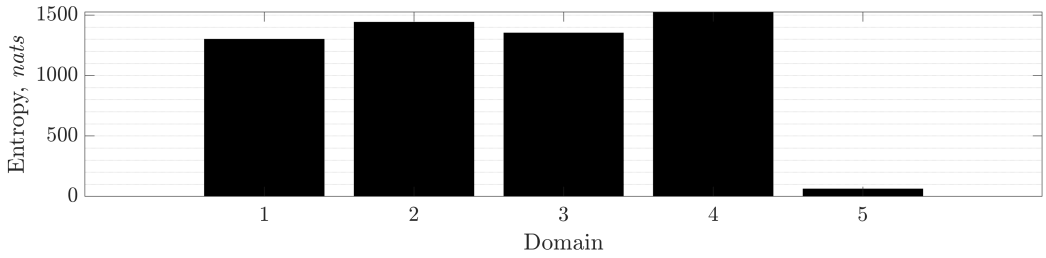

Figure 18: Eight degree-of-freedom case study: total entropy of the projection matrix A_t per domain.

100% accuracy on two domains, Domain Four and Five, with comparable performance to HFA on Domains One to Three. In this case study the entropy for the projection matrix for all four numerical domains are relatively equal, showing that they have all contributed to learning the latent space. As expected Domain Five has the lowest entropy, where Domains One to Four have provided benefits in classifying Domain Five. It is interesting to note that the RVM classifier achieves near perfect performance on Domains One to Four, showing that there is enough information in each simulated domain to construct a conventional classifier. However, the RVM classifier only achieves an accuracy of 50% on the experimental Domain Five, demonstrating the need for transfer learning.

4. Conclusions

The ability to perform classification for populations with inconsistent feature spaces means a wider range of population types (and feature types) can be considered in population-based SHM. A sparse Bayesian approach to heterogeneous transfer learning [10], as applied in this paper, allows label information to be shared and transferred, within, and between such populations of structures. This advance in technology means that a lack of available health state data for a particular structure of interest can be overcome by considering the shared label set in a population, even when the features from one structure are different to others in the group, e.g. one having ten natural frequencies and the other six.

The sparse Bayesian method applied in this paper — KBTL [10] — extends ideas from relevance vector machines, with the addition of a projection and dimensionality reduction step, mapping the data from each structure onto a shared latent space. This projection is inferred in a sparse manner, meaning that relevance vectors in the mapping are identified. In the shared latent space, a joint classifier is inferred, meaning that label information from the population is shared in inferring one set of classifier parameters. As a result, label information can also be transferred to structures where certain label classes were missing in training.

The method has been demonstrated as effective in both binary and multi-class settings. In addition, the approach has been demonstrated on three applications; a population of simulated and experimental shear-structures with different numbers of storeys, an experimental Gnat aircraft structure where the number of sensors was different between domains, and a population of simulated and experimental eight degree-of-freedom systems where differing measurement properties led to a inconsistent feature space. In each application, KBTL has been able to increase the classification performance in domains with sparse label observations, particularly if observations of a class label were not available in training. In addition, the approach produced the highest average classification accuracy on all the multi-class classification case studies, when compared to Heterogeneous Feature Augmentation (HFA), and either SD-KBTL or RVM classifiers. However, it is noted that improved average performance, and increased classification accuracy in particularly sparsely-labelled domains, was often countered by a slight reduction in classification performance in other domains when compared to HFA. However, KBTL has the ability to learn one classification model for all domains (i.e. it is a multi-domain method), reducing the number of classification models that require training, and simplifying the application of heterogeneous transfer learning for the PBSHM context. In contrast, HFA is a single source domain to single target domain approach, meaning that selection of the most appropriate source domain would be required, or some ensemble-based approach, leading to technical challenges in implementing the method over a large population of structures. Furthermore, HFA often struggled to perform well in scenarios where label observation sparsity occurred in a particular domain, meaning its ability to *transfer* label knowledge was less optimal than KBTL. Overall, the results show that KBTL is a technique capable of creating a classifier that generalises across a population, shares label information, and allows labels to be transferred between members of the population — all goals of PBSHM.

A significant advantage of this approach is that physics-based simulations can be used to create label information that can be transferred to operational structures. Both case studies have demonstrated the potential for using data from a range of simulated structures, which may not be exactly the same as the experimental structure of interest, in labelling a real world structure with limited label knowledge. This ability to robustly use simulation data, is a powerful advancement in PBSHM, as typically simulations will be difficult to validate, or may not match the real system closely, but can be useful in transferring information to a structure of interest. As a result, simulations can be used in a more robust manner, overcoming the lack of available health-state data in data-driven SHM.

5. Acknowledgements

The authors would like to acknowledge the support of the UK Engineering and Physical Sciences Research Council via grants EP/R006768/1, EP/R003645/1 and EP/R004900/1.

Appendix A. Kernelised Bayesian transfer learning nomenclature

An overview of the main nomenclature used in kernelised Bayesian transfer model are summarised below:

Objects and Spaces

$\mathcal{D}$	Domain
$\mathcal{T}$	Task
$\mathcal{X}$	Feature space
$\mathcal{Y}$	Label space
$\mathcal{H}$	Shared latent space
R	Dimension of the shared latent space $\mathcal{H}$

Data, Latent Variables and Model Parameters

X	A finite set of feature observations from a feature space $\mathcal{X}$
Y	A finite set of label observations from a label space $\mathcal{Y}$
D	A finite set of observed data, i.e. label observations
K	Kernel matrix
Θ	Set of latent variables and model parameters
A	Projection matrix
H	Matrix in the shared latent space $\mathcal{H}$, formed from $H = A^\mathsf{T} K$
b	Bias scalar for the discriminative function
$\boldsymbol{w}$	Vector of weights for the discriminative function
$\boldsymbol{f}$	Discriminative function vector in the shared latent space, formed from $\boldsymbol{f} = H^\mathsf{T}\boldsymbol{w} + \mathbf{1}b$

Priors and Hyperparameters

Ξ	Set of priors
Λ	Prior precision matrix for the projection matrix A
η	Prior precision vector for the weight vector $\boldsymbol{w}$
γ	Prior precision scalar for the bias b
$\boldsymbol{\zeta}$	Set of hyperparameters
κ_λ	Shape hyperparameter for the prior precision matrix Λ
θ_λ	Scale hyperparameter for the prior precision matrix Λ
κ_η	Shape hyperparameter for the prior precision vector η
θ_η	Scale hyperparameter for the prior precision vector η
κ_γ	Shape hyperparameter for the prior precision scalar γ
θ_γ	Scale hyperparameter for the prior precision scalar γ
σ_h^2	Variance of the latent space
ν	Non-negative margin

Appendix B. Posterior expectations

Each of the posterior expectations (i.e. $\langle f(\cdot)\rangle$) are,

$$\langle A_t[:,s]^2\rangle = \mu(A_t[:,s])^2 + \mathrm{diag}(\Sigma(A_t[:,s])) \tag{B.1a}$$

$$\langle A_t\rangle = \mu(A_t) \tag{B.1b}$$

$$\langle H_t H_t^{\mathsf{T}} \rangle = \mu(H_t)\mu(H_t)^{\mathsf{T}} + \Sigma(H_t) \tag{B.1c}$$

$$\langle H_t \rangle = \mu(H_t) \tag{B.1d}$$

$$\langle \boldsymbol{w}_l \boldsymbol{w}_l^{\mathsf{T}} \rangle = \mu(\boldsymbol{w}_l)\mu(\boldsymbol{w}_l)^{\mathsf{T}} + \Sigma(\boldsymbol{w}_l) \tag{B.1e}$$

$$\langle \boldsymbol{w}_l \rangle = \mu(\boldsymbol{w}_l) \tag{B.1f}$$

$$\langle b_l^2 \rangle = \mu(b_l)^2 + \Sigma(b_l) \tag{B.1g}$$

$$\langle b_l \rangle = \mu(b_l) \tag{B.1h}$$

$$\langle b_l \boldsymbol{w}_l \rangle = \mu(b_l)\mu(\boldsymbol{w}_l) + \Sigma(b_l, \boldsymbol{w}_l) \tag{B.1i}$$

$$\langle \boldsymbol{\eta}_l \rangle = \kappa(\boldsymbol{\eta}_l)\theta(\boldsymbol{\eta}_l) \tag{B.1j}$$

$$\langle \gamma_l \rangle = \kappa(\gamma_l)\theta(\gamma_l) \tag{B.1k}$$

$$\langle \boldsymbol{f}_{t,l}[i] \rangle = \mu(\boldsymbol{f}_{t,l}[i]) + \frac{\phi(\boldsymbol{\alpha}_{t,l}[i]) - \phi(\boldsymbol{\beta}_{t,l}[i])}{\boldsymbol{z}_{t,l}[i]} \Sigma(\boldsymbol{f}_{t,l}[i]) \tag{B.1l}$$

$$\boldsymbol{\alpha}_{t,l}[i] = \frac{\mathrm{a}(\boldsymbol{f}_{t,l}[i]) - \mu(\boldsymbol{f}_{t,l}[i])}{\Sigma(\boldsymbol{f}_{t,l}[i])} \tag{B.1m}$$

$$\boldsymbol{\beta}_{t,l}[i] = \frac{\mathrm{b}(\boldsymbol{f}_{t,l}[i]) - \mu(\boldsymbol{f}_{t,l}[i])}{\Sigma(\boldsymbol{f}_{t,l}[i])} \tag{B.1n}$$

$$\boldsymbol{z}_{t,l}[i] = \Phi(\boldsymbol{\beta}_{t,l}[i]) - \Phi(\boldsymbol{\alpha}_{t,l}[i]) \tag{B.1o}$$

where $\Sigma(b, \boldsymbol{w})$ denotes the cross covariance and $\phi(\cdot)$ and $\Phi(\cdot)$ are standard Gaussian probability and cumulative density functions respectively.

References

[1] K. Worden and J. M. Dulieu-Barton. An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1):85–98, 2004.

[2] K. Worden and G. Manson. The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):515–537, 2007.

[3] E. Figueiredo, G. Park, C. R. Farrar, K. Worden, and J. Figueiras. Machine learning algorithms for damage detection under operational and environmental variability. *Structural Health Monitoring*, 10(6):559–572, 2011.

[4] C. R. Farrar and K. Worden. *Structural Health Monitoring: a Machine Learning Perspective*. John Wiley & Sons, Ltd, Chichester, UK, 2012.

[5] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT press, 2006.

[6] L. A. Bull, K. Worden, and N. Dervilis. Towards semi-supervised and probabilistic classification in structural health monitoring. *Mechanical Systems and Signal Processing*, 140:106653, 2020.

[7] L. A. Bull, P. Gardner, J. Gosliga, N. Dervilis, E. Papatheou, A. E. Maguire, C. Campos, T. J. Rogers, E. J. Cross, and K. Worden. Foundations of population-based structural health monitoring, Part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing*, 148:107141, 2021.

[8] J. Gosliga, P. Gardner, L. A. Bull, N. Dervilis, and K. Worden. Foundations of population-based structural health monitoring, Part II: Heterogeneous populations and structures as graphs, networks, and communities. *Mechanical Systems and Signal Processing*, 148:107144, 2021.

[9] P. Gardner, L. A. Bull, J. Gosliga, N. Dervilis, and K. Worden. Foundations of population-based structural health monitoring, Part III: Heterogeneous populations, transfer and mapping. *Mechanical Systems and Signal Processing*, 149:107142, 2021.

[10] M. Gönen and A. A. Margolin. Kernelized Bayesian transfer learning. In *Proceedings of the National Conference on Artificial Intelligence*, 2014.

[11] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[12] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3:29, 2017.

[13] O. Day and T. M. Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4:29, 2017.

[14] L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29^{th} International Conference on Machine Learning, ICML 2012*, 2012.

[15] W. Li, L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1134–1148, 2014.

[16] A. Rytter. *Vibrational Based Inspection of Civil Engineering Structures*. PhD thesis, Aalborg University, Denmark, 1993.

[17] P. Cao, S. Zhang, and J. Tang. Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning. *IEEE Access*, 6:26241–26253, 2018.

[18] S. Dorafshan, R. J. Thomas, and M. Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186:1031–1045, 2018.

[19] A. Vetrivel, M. Gerke, N. Kerle, F. Nex, and G. Vosselman. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:45–59, 2018.

[20] Y. Gao and K. M. Mosalam. Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):748–768, 2018.

[21] C. Feng, H. Zhang, S. Wang, Y. Li, H. Wang, and F. Yan. Structural damage detection using deep convolutional neural network and transfer learning. *KSCE Journal of Civil Engineering*, (23):4493–4502, 2019.

[22] K. Jang, N. Kim, and Y. An. Deep learning-based autonomous concrete crack evaluation through hybrid image scanning. *Structural Health Monitoring*, page 147592171882171, 2019.

[23] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud. A very deep transfer learning model for vehicle damage detection and localization. In *2019 31st International Conference on Microelectronics (ICM)*, pages 158–161, 2019.

[24] M. Azimi, A. D. Eslamlou, and G. Pekcan. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, 20(10), 2020.

[25] D. Chakraborty, N. Kovvali, B. Chakraborty, A. Papandreou-Suppappola, and A. Chattopadhyay. Structural damage detection with insufficient data using transfer learning techniques. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, page 798147, 2011.

[26] J. Ye, T. Kobayashi, H. Tsuda, and M. Murakawa. Robust hammering echo analysis for concrete assessment with transfer learning. In *Proceedings of the the 11th International Workshop on Structural Health Monitoring*, pages 943–949, 2017.

[27] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2), 2017.

[28] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun. Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Processing*, 157:180–197, 2019.

[29] Q. Wang, G. Michau, and O. Fink. Domain adaptive transfer learning for fault diagnosis. In *2019 Prognostics and System Health Management Conference (PHM-Paris)*, pages 279–285, 2019.

[30] P. Gardner and K. Worden. On the application of domain adaptation for aiding supervised SHM methods. In *Proceedings of the 12th International Workshop on Structural Health Monitoring*, pages 3347–3357, Stanford, USA, 2019.

[31] S.-X. Chen, L. Zhou, Y.-Q. Ni, and X.-Z. Liu. An acoustic-homologous transfer learning approach for acoustic emission–based rail condition evaluation. *Structural Health Monitoring*, page 1475921720976941, 2020.

[32] P. Gardner, X. Liu, and K. Worden. On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing*, 138:106550, 2020.

[33] H. Wan and Y. Ni. Bayesian multi-task learning methodology for reconstruction of structural health monitoring data. *Structural Health Monitoring*, 18:1282–1309, 2019.

[34] Y. Huang, J. L. Beck, and H. Li. Multitask sparse Bayesian learning with applications in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 34(9): 732–754, 2019.

[35] Y. Zhang and Q. Yang. An overview of multi-task learning. *National Science Review*, 5:30–43, 2018.

[36] M. E. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems*, pages 652–658. MIT Press, 2000.

[37] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, pages 1–34, 2020.

[38] N. D. Lawrence and M. I. Jordan. Semi-supervised learning via Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 753–760. MIT Press, 2005.

[39] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211—244, 2001.

[40] G. Manson, K. Worden, and D. Allman. Experimental validation of a structural health monitoring methodology: Part II. Novelty detection on a gnat aircraft. *Journal of Sound and Vibration*, 259(2):345–363, 2003.

[41] G. Manson, K. Worden, and D. Allman. Experimental validation of a structural health monitoring methodology: Part III. Damage location on an aircraft wing. *Journal of Sound and Vibration*, 259(2):365–385, 2003.

[42] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, and M. Pontil. Optimal kernel choice for large-scale two-sample tests. In *Neural Information Processing Systems*, pages 1205–1213, 2012.

[43] S. Christides and A. D. S. Barr. One-dimensional theory of cracked Bernoulli-Euler beams. *International Journal of Mechanical Sciences*, 26(11-12):639–648, 1984.

[44] K. Worden, G. Manson, and D. Allman. Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure. *Journal of Sound and Vibration*, 259(2):323–343, 2003.

[45] K. Worden, G. Manson, G. Hilson, and S. G. Pierce. Genetic optimisation of a neural damage locator. *Journal of Sound and Vibration*, 309(3):529–544, 2008.

[46] L. A. Bull, K. Worden, R. Fuentes, G. Manson, E. J. Cross, and N. Dervilis. Outlier ensembles: A robust method for damage detection and unsupervised feature extraction from high-dimensional data. *Journal of Sound and Vibration*, 453:126–150, 2019.

[47] G. Tsialiamanis, D. J. Wagg, P. Gardner, N. Dervilis, and K. Worden. On partitioning of an SHM problem and parallels with transfer learning. In *Proceedings of IMAC XXXVIII International Conference on Modal Analysis*, Houston, USA, 2020.