Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Transferring model structure in Bayesian transfer learning for Gaussian process regression[☆]

Milan Papež[a,c,*], Anthony Quinn[a,b]

[a] Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic
[b] Department of Electronic and Electrical Engineering, Trinity College Dublin, the University of Dublin, Dublin, Ireland
[c] Artifcial Intelligence Center, Czech Technical University, Prague, Czech Republic

ARTICLE INFO

Article history:
Received 23 January 2021
Received in revised form 21 March 2022
Accepted 19 April 2022
Available online 26 April 2022

Keywords:
Bayesian transfer learning (BTL)
Multitask learning
Local and global modelling
Fully probabilistic design
Incomplete modelling
Gaussian process regression

ABSTRACT

Bayesian transfer learning (BTL) is defined in this paper as the task of conditioning a target probability distribution on a transferred source distribution. The target globally models the interaction between the source and target, and conditions on a probabilistic data predictor made available by an independent local source modeller. Fully probabilistic design is adopted to solve this optimal decision-making problem in the target. By successfully transferring higher moments of the source, the target can reject unreliable source knowledge (i.e. it achieves robust transfer). This dual-modeller framework means that the source's local processing of raw data into a transferred predictive distribution – with compressive possibilities – is enriched by (the possible expertise of) the local source model. In addition, the introduction of the global target modeller allows correlation between the source and target tasks – if known to the target – to be accounted for. Important consequences emerge. Firstly, the new scheme attains the performance of fully modelled (i.e. conventional) multitask learning schemes in (those rare) cases where target model misspecification is avoided. Secondly, and more importantly, the new dual-modeller framework is robust to the model misspecification that undermines conventional multitask learning. We thoroughly explore these issues in the key context of interacting Gaussian process regression tasks. Experimental evidence from both synthetic and real data settings validates our technical findings: that the proposed BTL framework enjoys robustness in transfer while also being robust to model misspecification.

1. Introduction

Multitask learning [1] with Gaussian process (GP) regression tasks is of major concern in the statistical signal processing and machine learning communities. Many contemporary instances of this framework have been reported, including methods based on convolved GPs [2–10], convolutional GPs [11–13], product GPs [14,15], deep (nested) GPs [16–18], and deep neural networks [19]. All these methods adopt a single global model to describe the relationship among source and target tasks, involving a joint probability distribution of all latent and output-data processes, which is consistent with the foundational methodology of Bayesian learning [20]. The flexibility of multitask GP regression is evidenced by a wide range of applications, such as reinforcement learning [21], Bayesian optimization [22], Earth observation [23], face recognition [24], and clinical data analysis [25].

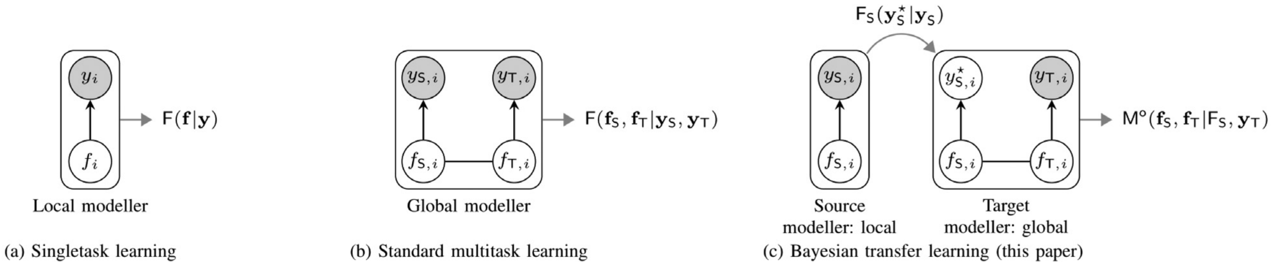
This paper is specifically interested in Bayesian transfer learning (BTL) [26–30], a less explored area compared to non-Bayesian transfer learning, as evidenced by several recent survey papers [31–38]. Without loss of generality, we define BTL as the transfer of probabilistic knowledge from a source task to a target task, a definition which is widely extensible to multitask settings. Under the Bayesian framework, source and target modellers separately adopt distinct probability distributions to model local unknown quantities of interest, i.e. our approach involves *two* stochastically independent modellers without requiring consistency to be established between them, whereas conventional multitask learning involves – as already stated – a single global modeller. These modellers each have exclusive access to their respective local data but not to each other's data. Therefore, the target cannot condition its model on the source data, the essential assumption of conventional global multitask learning. The target's model can be a local model, as in [39–46]. Alternatively, it may itself extend its focus beyond its local target task and adopt a global model involving both source and target tasks, as in the standard multitask setting. This is the setup we adopt in this paper for the first time. To emphasize: the target modeller globally models the interactions between its own task and the source

[☆] The research has been supported by GAČR grant 18-15970S.

[*] Corresponding author.

E-mail addresses: papez@utia.cas.cz (M. Papež), aquinn@tcd.ie (A. Quinn).

https://doi.org/10.1016/j.knosys.2022.108875
0950-7051/© 2022 Published by Elsevier B.V.


y_i
f_i
$\mathsf{F}(\mathbf{f}|\mathbf{y})$
Local modeller
$y_{\mathsf{S},i}$
$y_{\mathsf{T},i}$
$f_{\mathsf{S},i}$
$f_{\mathsf{T},i}$
$\mathsf{F}(\mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{T}|\mathbf{y}_\mathsf{S}, \mathbf{y}_\mathsf{T})$
Global modeller
$\mathsf{F}_\mathsf{S}(\mathbf{y}_\mathsf{S}^\star|\mathbf{y}_\mathsf{S})$
$y_{\mathsf{S},i}$
$f_{\mathsf{S},i}$
$y^\star_{\mathsf{S},i}$
$y_{\mathsf{T},i}$
$f_{\mathsf{S},i}$
$f_{\mathsf{T},i}$
$\mathsf{M}^\circ(\mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{T}|\mathsf{F}_\mathsf{S}, \mathbf{y}_\mathsf{T})$
Source modeller: local
Target modeller: global

(a) Singletask learning

(b) Standard multitask learning

(c) Bayesian transfer learning (this paper)

Fig. 1. (a) *Singletask learning*: an isolated local Bayesian modeller with complete knowledge of $\mathsf{F}(\mathbf{y}, \mathbf{f})$. (b) *Standard multitask learning*: an interacting global Bayesian modeller with complete knowledge of $\mathsf{F}(\mathbf{y}_\mathsf{S}, \mathbf{y}_\mathsf{T}, \mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{T})$. (c) *Bayesian transfer learning*: an isolated local source Bayesian modeller with complete knowledge of $\mathsf{F}_\mathsf{S}(\mathbf{y}_\mathsf{S}, \mathbf{f}_\mathsf{S})$ and an interacting global target Bayesian modeller with incomplete knowledge of $\mathsf{F}(\mathbf{y}^\star_\mathsf{S}, \mathbf{y}_\mathsf{T}, \mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{T}, \mathsf{F}_\mathsf{S})$. The global target modeller does not have access to $\mathbf{y}_\mathsf{S}$. It compensates for this missing information by utilizing the output-data predictive distribution, F_S, that is made available by the local source modeller. The global target modeller improves its performance based on F_S. The grey and white circles represent observed and unobserved variables, respectively.

task, but is informed *only* by its local target data. Meanwhile, the source modeller, adopting a local Bayesian model, is informed by its local (source) data, and then communicates its knowledge to the target only via a source probability distribution instantiated by its (source) sufficient statistics. Therefore, only the target takes care of interactions and only the source communicates source probabilistic knowledge. The key distinctions between global and local modelling of the source and target tasks are captured in Fig. 1, whose technical details will be explained fully in Section 2. The target modeller has no prescriptive rule of probability calculus with which to condition on the source distribution in this BTL scenario (Fig. 1(c)). The reason for this is that the joint model of the source distribution and the target's quantities of interest is not available. The challenge for the target modeller then lies in deciding how to condition on the source probability distribution in this incompletely modelled case. Fully probabilistic design (FPD) [47,48] – the optimal Bayesian decision-making strategy based on the minimum cross-entropy principle [49] – addresses this problem. Under this framework, the target modeller conditions on the transferred source knowledge by solving a constrained optimization problem.

The principal novelty of this paper, therefore, is that the target modeller is a global modeller (in sympathy with the aforementioned standard multitask learning) that conditions on a local source probability distribution. Recall, from Bayesian foundations, that this latter distribution is a function of both its local source data *and* local source model. What is implied by this BTL framework is that the global target modeller independently adopts a different model from the source. This leads to important advances beyond the current available notions and techniques for BTL, all of which are explored in this paper:

- The introduction of the global target modeller facilitates learning about the correlation structure between the source and target tasks, extending our own previous BTL framework which involved only locally modelled tasks [39–41,44–46]. It also generalizes FPD-based approaches to pooling [50,51], as well as conventional (global) Bayesian multitask learning [2–5,11,14–19]. We demonstrate this formally, and also via extensive experimental evidence.
- We stand to benefit from the fact that the local source modeller is independent of the target and is therefore, potentially, a more informed – i.e. expressive or expert – modeller of its local data. This can provide a major performance dividend over standard multitask learning solutions that adopt only a single global (typically centralized) model.
- The source probability distribution is represented by its sufficient statistics, thereby achieving an optimal encoding (compression) of source knowledge extracted from its (often high-dimensional) data and (possibly complex) model structure. This can avoid transferred message overheads that arise in standard multitask learning, which relies on the transfer of unprocessed source data for its conditioning task (i.e. learning). This is a principal novelty in our BTL framework, when compared to conventional global modelling techniques for knowledge transfer.
- Our previous FPD-based BTL schemes were unable to transfer higher moments of the source distribution, a resource that is necessary in achieving robust transfer (i.e. the rejection of imprecise source knowledge) [39,40,44,52]. By careful design of the FPD-based constrained optimization problem in this paper, we formally solve this problem, obviating the informal adaptations in [39,40,46], and the computationally expensive augmentation strategies in [41,45].

The paper is organized as follows: Section 2 defines the BTL problem, where the central aim is to transfer the source output-data predictor to the target modeller. The new concept of the global target modeller is introduced and the general solution based on the FPD framework is provided. Section 3 instantiates this general setting of BTL in the case of GP regression tasks. Section 4 provides a thorough exploration of these advances via synthetic data experiments, focussing particularly on its ability to capture correlation structure, to achieve robust transfer and to benefit from the transfer of an expressive local model in the source. This will allow us to comment in an informed way on the performance limits of our approach. Section 5 validates our BTL scheme in a real-data context. Section 6 provides further discussion of the key properties of our method, including its source knowledge compression and computational complexity aspects. Section 7 concludes the paper with suggestions for future work.

2. Bayesian transfer learning from a local to global modeller

Consider a (non-parametric) regression task where the aim is to infer unknown (latent) function values, $\mathbf{f} \equiv (f(x_i))_{i=1}^n$, based on known input data (regressors), $\mathbf{x} \equiv (x_i \in \mathbb{R}^{n_x})_{i=1}^n$, and known output data, $\mathbf{y} \equiv (y_i \in \mathbb{R})_{i=1}^n$, where each y_i is an indirect (e.g. noisy) observation of $f(x_i) \in \mathbb{R}$. A Bayesian modeller (Fig. 1(a)) addresses this problem by adopting a complete joint probability model, $\mathsf{F}(\mathbf{y}, \mathbf{f}|\mathbf{x}, \theta)$, and then computing the posterior distribution, $\mathsf{F}(\mathbf{f}|\mathbf{y}, \mathbf{x}, \theta)$, for some fixed and finite-dimensional parameter, $\theta \in \mathbb{R}^{n_\theta}$, given a realization of $\mathbf{y}$. This constitutes standard Bayesian conditioning in the context of the complete joint model,[1] $\mathsf{F}(\mathbf{y}, \mathbf{f}|\mathbf{x}, \theta)$.

[1] In the sequel, we will suppress the details of the conditioning where this is evident. Moreover, F and M denote known (fixed-form) and unknown (variational-form) distributions, respectively. We use "≡" to denote "is defined to be equal to".

This paper addresses the problem of learning in a pair of GP regression tasks, called the source and target learning tasks, respectively. The source task is *local*, being informed by local output data, $\mathbf{y}_S$, available only to the source. Its local processing of knowledge, $F_S(\mathbf{y}_S, \mathbf{f}_S)$, and data, $\mathbf{y}_S$, is to be exploited in order to improve learning at the target task. The standard approach is that a *global* multitask modeller (Fig. 1(b)) elicits a joint model, $F(\mathbf{y}_S, \mathbf{y}_T, \mathbf{f}_S, \mathbf{f}_T)$, capturing interactions between the source and target processes, and conditions on source and target output data, $\mathbf{y}_S$ and $\mathbf{y}_T$, respectively, to which it must have access [2–5,11,14–19]. This yields

$$F(\mathbf{f}_S, \mathbf{f}_T|\mathbf{y}_S, \mathbf{y}_T). \tag{1}$$

This scheme requires the prior elicitation of the joint model of the two-task learning system. We refer to this as *complete modelling*.

In this work, our aim is to relax these restrictive assumptions of the conventional multitask learning framework, as follows (Fig. 1(c)):

(i) The target will not have direct access to the source data, $\mathbf{y}_S$; and
(ii) the target's joint model,[2] $F(\mathbf{y}_S, \mathbf{y}_T, \mathbf{f}_S, \mathbf{f}_T)$, is distinct from the source's local model, $F_S(\mathbf{y}_S, \mathbf{f}_S)$, and each is elicited independently.

Specifically, the goal of the present paper is to substitute the inaccessible source data, $\mathbf{y}_S$, with appropriately specified probabilistic source knowledge, $F_S(\cdot)$. Technically, the inferential objective of the global target modeller is to compute $F(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T)$, conditioned now on the transferred source distribution, F_S, but not on $\mathbf{y}_S$. This conditioning on F_S is incompletely specified because the target does not have a joint model of the type, $F(F_S, \ldots)$, i.e. one that involves the target's hierarchical specification of the source's transferred distribution. Instead, the target's learning task becomes one of optimally choosing its knowledge-conditioned model, $M(\ldots|F_S, \mathbf{y}_T)$, in this *incompletely modelled* case (necessarily, then, M is unknown). The fact that the global target modeller (Fig. 1(c)) processes F_S (a result of learning) is an important progression beyond conventional global multitask learning (Fig. 1(b)), and we reserve the term *Bayesian transfer learning* for the case in (Fig. 1(c)). Among its consequences are the following:

(i) the target learns only from the sufficient statistics of the source, without itself receiving the source data, $\mathbf{y}_S$ (of course, it also processes its local target data, $\mathbf{y}_T$); and
(ii) the target can be enriched by the (possibly expert) local source model.

It remains to decide on the optimal form of the target's F_S-conditioned joint model, $M(\cdot|F_S, \mathbf{y}_T)$. Here, $F_S \equiv F_S(\mathbf{y}_S^\star|\mathbf{y}_S)$, i.e. the source transfers its source-data-conditioned predictor of an unrealized (i.e. unobserved) output, denoted by $\mathbf{y}_S^\star$, for which the known input is $\mathbf{x}_S^\star$. Hence, the target's unspecified joint model must be augmented, to yield

$$M(\mathbf{y}_S^\star, \mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) = M(\mathbf{y}_S^\star|\mathbf{f}_S, \mathbf{f}_T, F_S, \mathbf{y}_T)M(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T), \tag{2}$$

which follows from the chain rule. Fully probabilistic design (FPD) is the axiomatically justified Bayesian decision-making framework [47,53] for optimally choosing the required conditional distribution. FPD is closely related to the minimum cross-entropy principle [49], as further explained in [48]. It solves the following constrained optimization problem:

$$M^{\text{o}}(\mathbf{y}_S^\star, \mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) \equiv \underset{M\in\mathbf{M}}{\operatorname{argmin}}\, \mathcal{D}(M\|M_I), \tag{3}$$

where

$$\mathcal{D}(M\|M_I) = E_M\left[\log\left(\frac{M}{M_I}\right)\right],$$

is the Kullback–Leibler divergence [54] from M to M_I, and E_M is the expected value under M. The FPD-optimal model (3) is the solution (i) that respects the knowledge constraints given by the set of possible models, $M \in \mathbf{M}$, and (ii) that is closest to the ideal model, M_I. The latter must declare the target's preferences about M. These are as follows:

(i) The set of possible models, $\mathbf{M}$, is delineated by restricting the functional form of (2),

$$M(\mathbf{y}_S^\star|\mathbf{f}_S, \mathbf{f}_T, F_S, \mathbf{y}_T) \equiv F(\mathbf{y}_S^\star|\mathbf{f}_S), \tag{4}$$

where we assume that $\mathbf{y}_S^\star$ is conditionally independent of $(\mathbf{f}_T, F_S, \mathbf{y}_T)$ given $\mathbf{f}_S$. (2) then has the form

$$M(\mathbf{y}_S^\star, \mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) \equiv F(\mathbf{y}_S^\star|\mathbf{f}_S)M(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T), \tag{5}$$

where the F_S-conditioned factor remains unknown and unrestricted. Since $F(\mathbf{y}_S^\star|\mathbf{f}_S)$ is fixed, then $M(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T)$ is the only unknown quantity in (5), and the set of possible models is therefore

$$\begin{aligned} M \in \mathbf{M} \equiv \{&\text{models (5) with } F(\mathbf{y}_S^\star|\mathbf{f}_S) \text{ fixed} \\ &\text{and } M(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) \text{ variational}\}. \end{aligned} \tag{6}$$

(ii) Adopting the same chain rule expansion as in (2), the target's ideal model is specified as

$$M_I(\mathbf{y}_S^\star, \mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) \equiv F_S(\mathbf{y}_S^\star|\mathbf{y}_S)F(\mathbf{f}_S, \mathbf{f}_T|\mathbf{y}_T). \tag{7}$$

Comparing with (2), the following ideal declarations have been made by the target:

$$M_I(\mathbf{y}_S^\star|\mathbf{f}_S, \mathbf{f}_T, F_S, \mathbf{y}_T) \equiv F_S(\mathbf{y}_S^\star|\mathbf{y}_S), \tag{8}$$

$$M_I(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) \equiv F(\mathbf{f}_S, \mathbf{f}_T|\mathbf{y}_T). \tag{9}$$

Specifically, in (8), the target assumes as its ideal that $\mathbf{y}_S^\star$ is conditionally independent of $(\mathbf{f}_S, \mathbf{f}_T, \mathbf{y}_T)$ given F_S, and – in order to transfer the source output-data predictor, F_S – adopts the source's predictor of $\mathbf{y}_S^\star$, i.e. $F_S(\mathbf{y}_S^\star|\mathbf{y}_S)$, as its ideal model for $\mathbf{y}_S^\star$. Similarly, in (9), the target ideally models $(\mathbf{f}_S, \mathbf{f}_T)$ conditionally independently of F_S given its own $\mathbf{y}_T$, and adopts its own joint model, $F(\mathbf{f}_S, \mathbf{f}_T|\cdot)$, as its ideal, $M_I(\mathbf{f}_S, \mathbf{f}_T|\cdot)$.

These specifications imply a unique design for the target's source-knowledge-constrained optimal distribution (3), as established in the following proposition.

Proposition 1. *The target modeller constrains the unknown model (2) to belong to the set of possible models, $M \in \mathbf{M}$ (6), and adopts the ideal model M_I (7). Then the target's FPD-optimal model (3) is*

$$M^{\text{o}}(\mathbf{y}_S^\star, \mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) = F(\mathbf{y}_S^\star|\mathbf{f}_S)M^{\text{o}}(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T), \tag{10}$$

where

$$\begin{aligned} M^{\text{o}}(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T) &\propto F(\mathbf{f}_S, \mathbf{f}_T|\mathbf{y}_T) \\ &\times \exp\left\{-\mathcal{D}\big(F(\mathbf{y}_S^\star|\mathbf{f}_S)\|F_S(\mathbf{y}_S^\star\|\mathbf{y}_S)\big)\right\}. \end{aligned} \tag{11}$$

Proof. See Appendix B. □

The design (11) is the optimal Bayesian conditioning mechanism we have been looking for. It forms the update from the pre-posterior, $F(\mathbf{f}_S, \mathbf{f}_T|\mathbf{y}_T)$, to the FPD-optimal, F_S-conditioned, posterior, $M^{\text{o}}(\mathbf{f}_S, \mathbf{f}_T|F_S, \mathbf{y}_T)$. The exponential structure in (11) constitutes the updating term which incorporates F_S. It has the Boltzmann structure that is typical of entropic designs [48,55].

[2] We adopt the following notation: undecorated distributions, F and M, are *target* models, with F_S denoting source distributions.

Algorithm 1: The FPD-optimal global target GP regression modeller

```
Input: $k_{\mathsf{SS}}, k_{\mathsf{ST}}, k_{\mathsf{TT}}, \mathbf{x}_{\mathsf{S}}, \mathbf{x}_{\mathsf{T}}, \bar{\mathbf{m}}_{\mathsf{S}}, \bar{\mathbf{R}}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \sigma_{\mathsf{T}}^2$
For all entries of x, compute
    $m_q^{\circ}(x) = \mathbf{k}_q(\mathbf{K} + \mathrm{blkdiag}(\bar{\mathbf{R}}_{\mathsf{S}}, \sigma_{\mathsf{T}}^2 I_n))^{-1} \begin{bmatrix} \bar{\mathbf{m}}_{\mathsf{S}} \\ \mathbf{y}_{\mathsf{T}} \end{bmatrix},$   (AL1a)
    $k_{qp}^{\circ}(x, x') = k_{qp}(x, x') - \mathbf{k}_q(\mathbf{K} + \mathrm{blkdiag}(\bar{\mathbf{R}}_{\mathsf{S}}, \sigma_{\mathsf{T}}^2 I_n))^{-1} \mathbf{k}_p',$   (AL1b)
    where
    $\mathbf{k}_q \equiv \begin{bmatrix} k_{q\mathsf{S}}(x, \mathbf{x}_{\mathsf{S}}) & k_{q\mathsf{T}}(x, \mathbf{x}_{\mathsf{T}}) \end{bmatrix}, \quad \mathbf{k}_p' \equiv \begin{bmatrix} k_{\mathsf{S}p}(\mathbf{x}_{\mathsf{S}}, x') \\ k_{\mathsf{T}p}(\mathbf{x}_{\mathsf{T}}, x') \end{bmatrix},$   (AL2a)
    $\mathbf{K} \equiv \begin{bmatrix} k_{\mathsf{SS}}(\mathbf{x}_{\mathsf{S}}, \mathbf{x}_{\mathsf{S}}) & k_{\mathsf{ST}}(\mathbf{x}_{\mathsf{S}}, \mathbf{x}_{\mathsf{T}}) \\ k_{\mathsf{TS}}(\mathbf{x}_{\mathsf{T}}, \mathbf{x}_{\mathsf{S}}) & k_{\mathsf{TT}}(\mathbf{x}_{\mathsf{T}}, \mathbf{x}_{\mathsf{T}}) \end{bmatrix}.$   (AL2b)
    Here, $q, p \in (\mathsf{S}, \mathsf{T})$, and $\mathrm{blkdiag}(U, V)$ is the block diagonal matrix
    obtained from $U \in \mathbb{R}^{u \times u}$ and $V \in \mathbb{R}^{v \times v}$.
Output: $\mathbf{m}_q^{\circ} \equiv m_q^{\circ}(\mathbf{x}_q)$, $\mathbf{K}_{qp}^{\circ} \equiv k_{qp}^{\circ}(\mathbf{x}_q, \mathbf{x}_p)$
```

3. Local and global Gaussian process regression modellers

We instantiate the framework in Section 2 to the case of source and target GP regression tasks [56,57]. For this purpose, consider a stochastic process, $f \sim \mathsf{F}$, over a scalar random function, f, such that any n-dimensional collection of evaluation points (i.e. regressors), $\mathbf{x}$, induces a joint probability distribution over the function values, $\mathbf{f}$. The GP, $f(x) \sim \mathcal{GP}(m_{\theta_f}(x), k_{\theta_f}(x, x'))$, where $x \in \mathbb{R}^{n_x}$, is conditioned on known mean function, $m_{\theta_f} : \mathbb{R}^{n_x} \to \mathbb{R}$, assumed (without loss of generality) to be zero. The known covariance (kernel) function is $k_{\theta_f} : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}$. In this way, all joint distributions are specified to be multivariate Gaussian, $\mathsf{F}(\mathbf{f}) \equiv \mathcal{N}(\mathbf{m}, \mathbf{K})$. Here, $\mathbf{m} \equiv m_{\theta_f}(\mathbf{x})$ and $\mathbf{K} \equiv k_{\theta_f}(\mathbf{x}, \mathbf{x})$ are the known, n-dimensional mean vector, and $n \times n$-dimensional covariance matrix, respectively. We suppress θ_f – the parameters of m_{θ_f} and k_{θ_f} – in the sequel for notational simplicity.

This GP, f, is indirectly (i.e. noisily) observed as y at the evaluation points, $\mathbf{x}$, via an additive, uncorrelated, Gaussian noise process. Hence, the joint distribution is

$$\mathsf{F}(\mathbf{y}|\mathbf{f}, \theta) = \mathcal{N}(\mathbf{f}, \sigma^2 I_n), \tag{12a}$$

$$\mathsf{F}(\mathbf{f}|\theta) = \mathcal{N}(\mathbf{m}, \mathbf{K}), \tag{12b}$$

with the GP as prior (12b). Here, σ^2 is the conditional output-data variance, I_n is the $n \times n$-dimensional identity matrix, and $\theta \equiv (\theta_f, \sigma^2)$ are all known parameters. In the rest of this section, we restore the explicit conditioning on θ, as it is relevant to the calculus.

We now adopt this stochastic structure in the isolated (i.e. unconditioned on transferred knowledge) local source task of BTL (Fig. 1(c)). Our purpose is to transfer an output-data predictor via construction of the GP posterior at unobserved test points, as follows.

Remark 1. The isolated local source modeller adopts $\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}, \mathbf{f}_{\mathsf{S}}|\theta_{\mathsf{S}})$ (12), which is assumed to be strict-sense stationary. Then, the posterior distribution – evaluated at a single test point, $x \in \mathbb{R}^{n_x}$ (now shown explicitly in the conditioning) – is

$$\mathsf{F}_{\mathsf{S}}(f_{\mathsf{S}}|\mathbf{y}_{\mathsf{S}}, \theta_{\mathsf{S}}, x) = \mathcal{N}(\bar{m}_{\mathsf{S}}, \bar{k}_{\mathsf{S}}),$$

where

$$\bar{m}_{\mathsf{S}}(x) = \mathbf{k}_{\mathsf{S}}(\mathbf{K}_{\mathsf{S}} + \sigma_{\mathsf{S}}^2 I_n)^{-1}\mathbf{y}_{\mathsf{S}}, \tag{13a}$$

$$\bar{k}_{\mathsf{S}}(x, x') = k_{\mathsf{S}}(x, x') - \mathbf{k}_{\mathsf{S}}(\mathbf{K}_{\mathsf{S}} + \sigma_{\mathsf{S}}^2 I_n)^{-1}\mathbf{k}_{\mathsf{S}}', \tag{13b}$$

with $\mathbf{k}_{\mathsf{S}} \equiv k_{\mathsf{S}}(x, \mathbf{x}_{\mathsf{S}})$, $\mathbf{k}_{\mathsf{S}}' \equiv k_{\mathsf{S}}(\mathbf{x}_{\mathsf{S}}, x')$, and $\mathbf{K}_{\mathsf{S}} \equiv k_{\mathsf{S}}(\mathbf{x}_{\mathsf{S}}, \mathbf{x}_{\mathsf{S}})$. Here, (x, x') is a second order test pair. The posterior predictor of unobserved outputs, $\mathbf{y}_{\mathsf{S}}^{\star}$, for which the (known) test points are $\mathbf{x}_{\mathsf{S}}^{\star}$, is

$$\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{y}_{\mathsf{S}}, \theta_{\mathsf{S}}) = \mathcal{N}(\bar{\mathbf{m}}_{\mathsf{S}}, \bar{\mathbf{R}}_{\mathsf{S}}), \tag{14}$$

where $\bar{\mathbf{R}}_{\mathsf{S}} = \bar{\mathbf{K}}_{\mathsf{S}} + \sigma_{\mathsf{S}}^2 I_{n^{\star}}$.

Proof. See, for example, [58], Chapter 2. □

Recall that only the source is a local modeller of the kind above. In contrast, the target modeller augments its model to assess the global source–target system (Fig. 1(c)). Therefore, the joint prior distribution, $\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}, \mathbf{y}_{\mathsf{T}}, \mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\theta)$, adopted by the global target modeller before processing $\mathbf{y}_{\mathsf{T}}$ and F_{S}, is specified by

$$\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta) \equiv \mathcal{N}(\mathbf{f}_{\mathsf{S}}, (\sigma^{\star})^2 I_{n^{\star}}), \tag{15a}$$

$$\mathsf{F}(\mathbf{y}_{\mathsf{T}}|\mathbf{f}_{\mathsf{T}}, \theta) \equiv \mathcal{N}(\mathbf{f}_{\mathsf{T}}, \sigma_{\mathsf{T}}^2 I_n), \tag{15b}$$

$$\mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\theta) \equiv \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{15c}$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathsf{SS}} & \mathbf{K}_{\mathsf{ST}} \\ \mathbf{K}_{\mathsf{ST}}^T & \mathbf{K}_{\mathsf{TT}} \end{bmatrix} \tag{16}$$

is the symmetric covariance matrix describing the target's prior beliefs about the interactions between the source and target GP function values, $(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}})$. It is expressed in a block-matrix form, involving three matrix degrees-of-freedom (T denotes matrix transposition). Specific cases will be considered in Section 4. Also, $(\sigma^{\star})^2$ is the target's conditional variance of the source output data at any test point, and σ_{T}^2 is the target's conditional variance of any target output datum.

It remains to compute the target's posterior joint inference of the GP function values, $(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}})$, under its joint model (15), i.e. after it has processed its local data, $\mathbf{y}_{\mathsf{T}}$, and the transferred knowledge, F_{S}, respectively.

Proposition 2. *The global target modeller elicits the joint prior distribution (15) and processes the source output-data predictor (14). Then, the FPD-optimal posterior distribution (11) – evaluated at a test point x – is*

$$\mathsf{M}^{\circ}(f_{\mathsf{S}}, f_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta, x) = \mathcal{N}(m^{\circ}, k^{\circ}), \tag{17}$$

where

$$m^{\circ} = \begin{bmatrix} m_{\mathsf{S}}^{\circ}(x) \\ m_{\mathsf{T}}^{\circ}(x) \end{bmatrix}, \qquad k^{\circ} = \begin{bmatrix} k_{\mathsf{SS}}^{\circ}(x, x') & k_{\mathsf{ST}}^{\circ}(x, x') \\ k_{\mathsf{TS}}^{\circ}(x, x') & k_{\mathsf{TT}}^{\circ}(x, x') \end{bmatrix},$$

with m_q° and k_{qp}° being specified in (AL1) for $q, p \in (\mathsf{S}, \mathsf{T})$.

Proof. See Appendix C. □

The proposed global target GP regression modeller, supported by FPD-optimal BTL from the local source GP regression modeller, is summarized in Algorithm 1. Note that there is no $(\sigma^{\star})^2$ – i.e. the target's conditional variance of the source output data at any test point – in the input list, since it is cancelled out in the derivation of the KLD term (C.2) in the proof of Proposition 2. This is an important feature of our framework, consistently making the target modeller independent of the source parameters.

4. Synthetic data experiments

In this section, we investigate the performance of our algorithm (Algorithm 1) against a number of alternative transfer learning algorithms, in particular focussing on distinctions between the *analysis model* underlying each algorithm and the *synthesis model* used to generate the synthetic data. We illustrate the following key properties of Algorithm 1:

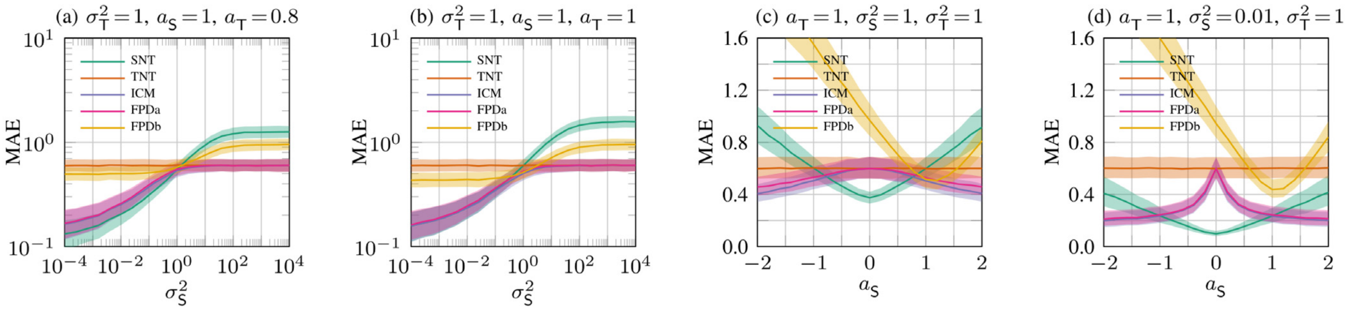


Fig. 2. Synthetic data experiments. The MAE (21) vs. the conditional output-data variance of the source, σ_S^2, for the target weight (a) $a_T = 0.8$, (b) $a_T = 1$. The MAE vs. the source weight, a_S, for the conditional output-data variance of the source (c) $\sigma_S^2 = 1$ and (d) $\sigma_S^2 = 0.01$. The results are averaged over 5000 independent simulation runs, with the solid line and the shaded area being the median and the interquartile range, respectively.

- the ability to achieve robust knowledge transfer – i.e. rejection of imprecise source knowledge – due to successful transfer of all moments of F_S (14);
- the ability to process known correlation between the source and target latent functions, $(\mathbf{f}_S, \mathbf{f}_T)$, via the target's specification of the covariance structure, $\mathbf{K}$ (16);
- the experimental bounding of the performance of Algorithm 1, which holds under the ideal condition when the synthesis and analysis models are equal; and
- transfer of source model structure via F_S (14), leading to improved positive transfer of Algorithm 1 under mismatch between the source and target models.

The first three issues are explored in Section 4.1, and the fourth in Section 4.2. Throughout, the synthesis model is the (rank-1) intrinsic coregionalization model (ICM) [5],

$$\begin{aligned} u &\sim \mathcal{GP}(0, k_{\theta_u}), \\ y_S &\sim \mathcal{N}(a_S u(x), \sigma_S^2), \\ y_T &\sim \mathcal{N}(a_T u(x), \sigma_T^2), \end{aligned} \tag{18}$$

and so $x_S = x_T = x$. Here, $f_S(x) = a_S u(x)$ and $f_T(x) = a_T u(x)$ share a common function, $u(x)$, modelled as a GP with zero mean and covariance function, k_{θ_u}. It follows that the covariance matrix (16) is

$$\mathbf{K} = B \otimes k_{\theta_u}(\mathbf{x}, \mathbf{x}), \tag{19}$$

where

$$B = \begin{bmatrix} a_S \\ a_T \end{bmatrix} \begin{bmatrix} a_S \\ a_T \end{bmatrix}^T \equiv \begin{bmatrix} b_{SS} & b_{ST} \\ b_{ST} & b_{TT} \end{bmatrix} \in \mathbb{R}^{2\times 2} \tag{20}$$

is the coregionalization matrix [59], and $\otimes$ denotes the Kronecker product of the matrix arguments. We adopt the mean absolute error (MAE) as the performance measure:

$$\mathrm{MAE} = \frac{1}{n^*} \sum_{i=1}^{n^*} |f_T(x_i^*) - m_T^o(x_i^*)|, \tag{21}$$

where $f_T(x_i^*)$ is the true target function value (Fig. 1), $m_T^o(x_i^*)$ is the lower sub-vector of the posterior predictive mean estimate (AL1a), $|\cdot|$ is the absolute value, and $(x_i^*)_{i=1}^{n^*}$ are the $n^* \geq 1$ test points.

4.1. The bounding performance of our BTL algorithm (Fig. 2)

Settings: We choose k_{θ_u} in (19) as the squared exponential (covariance) kernel function with the parameters $\theta_u \equiv (\sigma_u^2, l_u^2)$ (the full list of kernel functions involved in Section 4 are defined in Table 2). The complete parameterization of (18) is therefore $\theta = (\sigma_S^2, \sigma_T^2, a_S, a_T, \sigma_u^2, l_u^2) = (1, 1, 0.8, 1, 2, 0.4)$. Furthermore, $n = 64$ scalar input data (i.e. $n_x = 1$) are generated via $x \sim \mathcal{U}(-3.5, 3.5)$, where $\mathcal{U}(a, b)$ is the uniform distribution on the open interval (a, b). The $n^* = 200$ test points, $(x_i^*)_{i=1}^{n^*}$, are placed on a uniform grid in the interval $(-5, 5)$.

Table 1
Algorithms compared in Sections 4 and 5. FPDa is the newly proposed BTL algorithm.

Algorithm		Description	
Source No Transfer	(SNT)	Remark 1	(Fig. 1a)
Target No Transfer	(TNT)	Remark 1	(Fig. 1a)
Intrinsic Coregionalization Model	(ICM)	[5]	(Fig. 1b)
Fully Probabilistic Design	***(FPDa)***	**Algorithm 1**	(Fig. 1c)
Fully Probabilistic Design	(FPDb)	[44]	

Table 2
Kernel functions adopted in this paper, see [58]. Here, σ^2 is the signal variance, γ is the offset, d is the polynomial degree, l^2 is the length-scale, α is the fluctuation parameter, and $r(x, x') \equiv ||x - x'||$ is the Euclidean distance between x and x'. Note that the Matérn kernel is assessed only in the case of the smoothness parameter $\nu = 3/2$.

Covariance function		Expression
Constant	(C)	$k(x, x') = \sigma^2$
Linear	(L)	$k(x, x') = \sigma^2 x x'$
Polynomial	(P)	$k(x, x') = (\sigma^2 x x' + \gamma)^d$
Cosine	(Co)	$k(x, x') = \sigma^2 \cos\left(2\pi \sum_{i=1}^{n_x} (x_i - x_i')/l^2\right)$
Squared Exponential	(SE)	$k(x, x') = \sigma^2 \exp\left\{-\frac{1}{2}\frac{r(x,x')^2}{l^2}\right\}$
Rational Quadratic	(RQ)	$k(x, x') = \sigma^2 \left(1 + \frac{1}{2}\frac{r(x,x')^2}{l^2\alpha}\right)^{-\alpha}$
Matérn ($\nu = 3/2$)	(M)	$k(x, x') = \sigma^2(1 + \sqrt{3}r(x, x'))\exp\{-\sqrt{3}r(x, x')\}$

Performance: We compare the performance of the algorithms in Table 1 as we adapt the source knowledge quality via σ_S^2 (18) and the correlation between the source and target latent functions via a_S in (20), holding all other parameters in θ constant. In this experiment, we eliminate misspecification between the ICM synthesis model (18) and the analysis models of the various algorithms in Table 1; i.e. they all have perfect knowledge of θ. Fig. 2(a) and (b) show the MAE for target weights, $a_T = 0.8$ and 1.0, respectively, as a function of source variance, σ_S^2. The TNT algorithm (i.e. isolated target task) defines the baseline performance since it does not depend on the source knowledge, and so it is invariant to all source settings. If any of the BTL algorithms (FPDa and FPDb) yields an MAE below or above this level, it is said to deliver *positive* or *negative* transfer, respectively. If any algorithm saturates at this level, it is said to achieve *robust* transfer. The source MAE of the SNT algorithm (also an isolated task) is computed in terms of the quantities f_S and m_S^o, and, therefore, is unrelated to the target MAE (21). Indeed, we present this source MAE to track the influence of changing σ_S^2 and a_S on the performance of the SNT algorithm. The latter provides the transferred source output-data predictor to the FPDa and FPDb algorithms. Consequently, the performance of SNT will impact FPDa and FPDb, as explained in the next paragraph.

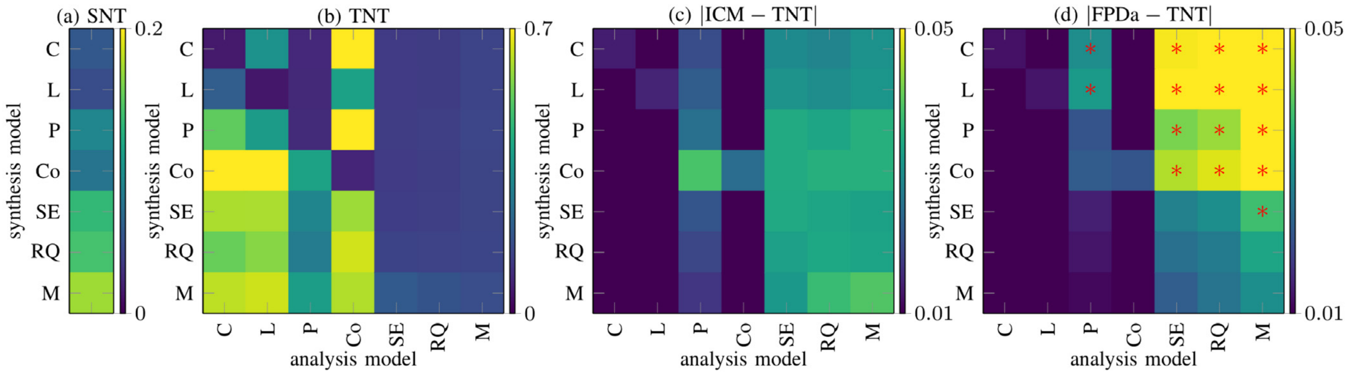


Fig. 3. Synthetic data experiments. (a) The MAE (21) of the SNT algorithm utilizing the analysis model which is equivalent to the source part of the ICM synthesis model. (b) The MAE of the TNT algorithm for all combinations of its analysis model kernel and the ICM synthesis model kernel (Table 2). In (a) and (b), *lower* is better. (c) and (d): The *absolute* value of differential MAE of the ICM algorithm and the FPDa algorithm, respectively, for all combinations of analysis model kernel and ICM synthesis model kernel (the absolute differential is with respect to the MAE of the TNT algorithm in (b), for ease of comparison of the positive transfer attained (always) by the respective algorithms). In (c) and (d), *higher* is better. The results are averaged over 1000 independent simulation runs. The asterisk $*$ highlights the cases where the FPDa algorithm achieves greater positive transfer than the ICM algorithm, indicating only the cases with statistical significance (p-value) $p < 0.05$, as computed via the unequal variance t-test [60].

The ICM algorithm delimits the optimal performance for all σ_S^2, since it adopts the ICM synthesis model (18) as its analysis model (i.e. misspecification is completely eliminated). Our FPDa algorithm (Algorithm 1) very closely follows the performance of the ICM algorithm (Fig. 2), despite (i) adopting incomplete modelling, and (ii) processing source statistics rather than the raw source output data, y_S; i.e. FPDa is a robust BTL algorithm. This improves on our previous FPDb algorithm, which achieves positive transfer only for $\sigma_S^2 < \sigma_T^2$ (high-quality source knowledge) but negative transfer for $\sigma_S^2 > \sigma_T^2$ (low-quality source knowledge); i.e. FPDb is non-robust.

Fig. 2 depicts the MAE for fixed target weight $a_T = 1$, and varying source weight, a_S (Fig. 2(c) and (d)). Recall that we eliminate misspecification between the ICM synthesis model (18) and the analysis models of the various algorithms in Table 1 by letting them have perfect knowledge of θ. The FPDa algorithm again achieves close tracking of the ICM algorithm for all the parameter settings. We note that the critical point where the performance curves converge occurs when $\sigma_S^2 = \sigma_T^2 = 1$. For this setting, we explore the influence of a_S in Fig. 2(c), holding $a_T = 1$. The main feature here is the symmetry of all the performance curves: around $a_S = 0$ for FPDa, ICM and SNT; around $a_S = 1$ for FPDb; and (trivially) everywhere for TNT. In all cases, this is due to the fact that a_S quadratically enters the correlation structure, B (20), of the global target modeller. When $\sigma_S^2 \neq \sigma_T^2$ (e.g. the case in Fig. 2(d) where $\sigma_S^2 \ll \sigma_T^2$), the main point to note is that FPDb cannot benefit from the improved positive transfer available from this more precise source. FPDa does benefit from this source precision because it, uniquely, exploits the target's correlation structure, B.

In summary, Fig. 2 illustrates the fact that the ICM algorithm – with exactly matched synthesis and analysis models – delimits the best predictive performance that can be attained by the FPDa algorithm. That said, misspecification of the ICM analysis model – as will occur with probability one in real-data settings – will undermine its performance, and provide an opportunity for FPDa to outperform it. This is so, since FPDa allows the source and target to elicit their models independently (Fig. 1(c)). We demonstrate this crucial model robustness feature of our BTL algorithm in the next section.

4.2. Transfer of the source's (local) analysis model (Fig. 3)

In our BTL framework (i.e. FPDa), the source modeller independently chooses a different model from the global target modeller to drive its source learning task. The source distribution, F_S (14), therefore encodes not only the information from the local source data, $\mathbf{y}_S$, but also the source model structure. In this section, we demonstrate the features of this approach under conditions of mismatch (i.e. disagreement) between the source's and target's model of $\mathbf{y}_S$. Specifically, we will now design an experiment where the source modeller more closely captures the synthesis model than the global target modeller does; i.e. the source exhibits *local expertise*, as is often the case in practice.

Settings: The purpose of this study is to explore the impact of kernel structure mismatch between the global target's analysis model and the ICM synthesis model. In all these cases, we arrange for the source analysis model to match the synthesis model, achieving local source expertise (above). The structures of the synthesis and analysis models are varied only via the kernel function, $k_{\theta_u}(\cdot,\cdot)$ (19), and its parameters, θ_u, for the seven cases in Table 2. The specific choices of these parameters in our current study are as follows: $(\sigma^2, \gamma, d, l^2, \alpha) \equiv (1.0, 1.0, 3, 0.2, 1.0)$. Recall that – in the previous simulation study (Fig. 2) – the remaining parameters, $(\sigma_S, \sigma_T, a_S, a_T)$, influence the transfer learning properties of the ICM and FPDa algorithms. We choose the settings, $\sigma_S = \sigma_T = 1$ and $a_S = a_T = 1$, where all the transfer algorithms benefit equally from the source data, and so the only differential benefit between the algorithms will be in respect of the quality of the analysis model adopted by the source in computing its transferred knowledge.

Performance: Fig. 3 shows the MAE for all pairwise combinations of the seven kernel functions in Table 2. The first kernel function is adopted in the ICM synthesis model, and the second in the analysis model of the TNT, ICM, and FPDa algorithm, respectively, of Table 1 (i.e. there are $7 \times 7 = 49$ kernel combinations, *in toto*, used in assessing the performance of each of these three algorithms). Since the SNT algorithm is arranged to capture perfectly the source part of the ICM synthesis model, therefore it requires only a single column in Fig. 3(a). We do not present the FPDb algorithm, since – with the currently adopted experimental settings – it reduces to the FPDa algorithm (Fig. 2(c)). The TNT algorithm delineates the baseline MAE performance of no transfer (as it did in Section 4.1), allowing us to assess the extent of positive transfer for the various kernel settings in ICM and FPDa. Note that both algorithms stay in the positive transfer regime for all combinations of the synthesis and analysis models, thus benefiting from the source knowledge even in the misspecification cases. For this reason, we have illustrated the absolute value of differential MAE in Fig. 3(c) and (d) (i.e.

higher values are better in these two sub-figures). Furthermore, the red asterisks indicate those kernel combinations for which our FPDa algorithm outperforms the ICM algorithm. Fig. 3(c) and (d) reveal that the ICM algorithm slightly outperforms the FPDa algorithm on all diagonal entries, i.e. where there is no mismatch between the synthesis and analysis models. This is consistent with our finding in Section 4.1, namely that the ICM algorithm achieves the optimal performance in this (unrealistic) case of perfect analysis–synthesis model matching. Conversely, the FPDa algorithm demonstrates improved robustness to kernel structure misspecification when compared to the ICM algorithm, particularly when FPDa adopts a more complex analysis model – i.e. SE, QR, M (Table 2) – and the synthesis model is simple, i.e. C, L, P, CO. This can be seen from the upper-right yellow quadrant in Fig. 3(d). These more complex kernel functions for analysis are, indeed, a common choice for real data, where it is often hard to choose a kernel function based on a mere visual inspection only. The remaining analysis–synthesis combinations – especially those under the diagonal in Fig. 3(d) – yield FPDa predictive performances in the target that are almost indistinguishable from those of the ICM algorithm.

5. Real-data experiments

The Swiss Jura geological dataset[3] records the concentrations of seven heavy metals – measured in units of parts-per-million (ppm) – at 359 (spatial) locations in a 14.5 km^2 region. This dataset has been widely studied in the context of multitask learning (see, for example, [5]). In our study, the objective is to predict concentrations of the less easily detected metal, cadmium (Cd), based on knowledge of the more easily detected metal, nickel (Ni). In order to demonstrate our BTL approach (Algorithm 1) in this context, the source task learns Ni concentrations and transfers its probabilistic output-data predictor to the target learning task of Cd concentration prediction. Note that we perform transfer with only this Ni source task, whereas, in [5], zinc (Zn) concentrations were also processed at the source.

Alignment with the FPD framework: The SNT algorithm processes the Ni concentrations measured at all $n_{\mathrm{S}} = 359$ locations (i.e. source data, $\mathbf{y}_{\mathrm{S}}$) and uses these to compute the Ni output-data predictor (14). The FPDa algorithm processes $n_{\mathrm{T}} = 259$ of the measured Cd concentrations (i.e. target data, $\mathbf{y}_{\mathrm{T}}$), along with the transferred Bayesian predictive moments, $\bar{\mathbf{m}}_{\mathrm{S}}$ and $\bar{\mathbf{R}}_{\mathrm{S}}$, of the source Ni output-data predictor (14), in order to improve the prediction of the Cd concentrations at the $n^* = 100$ hold-out locations, $\mathbf{x}^* \equiv (x_i^*)_{i=1}^{n^*}$, with spatial coordinates $x_i^* \equiv (x_{x,i}^*, x_{y,i}^*)$. The aim is to compute $\mathbf{m}_{\mathrm{T}}^{\circ *} \equiv m_{\mathrm{T}}^{\circ}(\mathbf{x}^*)$ and $\mathbf{K}_{\mathrm{TT}}^{\circ *} \equiv k_{\mathrm{TT}}^{\circ}(\mathbf{x}^*, \mathbf{x}^*)$, where m_{T}° and k_{TT}° are the lower sub-vector and lower-right sub-matrix of (AL1a) and (AL1b), respectively. In the Ni source task (SNT), the MAE for the GP predictive mean (13a) was computed across a wide range of candidate kernel functions, weighting the inputs, $\mathbf{x}_{\mathrm{S}}$, via automatic relevance determination (ARD) [58]. The optimal choice was the rational quadratic kernel (Table 2). In the isolated Cd target task (also with ARD), the Matérn kernel proved to be optimal, and was adopted for all the target learning algorithms (Table 1).

Parameter learning: The notion of a synthesis model (of the type in (18)) does not, of course, arise in this real-data context. Instead, we adopt maximum likelihood (ML) estimation [58] in order to learn the parameters of the analysis models underlying the various algorithms in Table 1. For this purpose, we use the iterative L-BFGS-B algorithm [61] – widely available as a library tool – to compute a local optimum of the log-marginal likelihood surface. The maximum number of iterations is limited to 20,000. We initialize the parameters (Table 2) of the rational quadratic kernel with $(\sigma_u^2, l_u^2, \alpha_u) \equiv (1.0, 1.0, 1.0)$ and the parameters of the Matérn kernel with $(\sigma_u^2, l_u^2) \equiv (1.0, 1.0)$, where the value of the length-scale, l_u^2, now applies to both input dimensions of x in accordance with the ARD mechanism. The coefficients, a_{S} and a_{T}, parameterizing the coregionalization matrix, B (20), are initialized as $(a_{\mathrm{S}}, a_{\mathrm{T}}) = (0.1, 0.1)$. For this Jura dataset, we found that the adopted ML procedure was insensitive to perturbations of these initial parameters, as recorded in Table 3.

Table 3
Real-data experiments. The MAE (21) of predicted Cadmium concentrations for the TNT, ICM, FPDa, and FPDb algorithms in the Jura dataset. The results are averaged over 10 runs with different initialization of the estimated unknown parameters, presenting the mean and standard deviation. The results are statistically significant with a p-value, $p < 0.05$, via the Mann–Whitney U-test [62].

Algorithm	MAE
TNT	5.9273e−01 ± 0.0001e−01
ICM	5.2808e−01 ± 0.0003e−01
FPDa	4.9966e−01 ± 0.0005e−01
FPDb	1.9510e−01 ± 0.0003e+01

Performance: To assess quantitatively the prediction performance of the algorithms in Table 1, we compute the MAE (21) across the $n^* = 100$ hold-out locations where we wish to predict Cadmium (Table 3). We find that the FPDa algorithm delivers positive transfer (i.e. it has a lower MAE than the baseline TNT algorithm), and, importantly, it outperforms the ICM algorithm. This improves on our former FPDb algorithm which suffers negative transfer because – as explained in Section 4 – it is not equipped to adopt and learn the target's (mineral) inter-task correlation via B (20). Our FPDa performance is close to that reported in [5], despite the fact that we transfer predictive source knowledge only from Ni concentrations, whereas Zn concentrations are also processed in [5], as already noted above. Returning to our own FPDa algorithm and the alternatives in Table 1, we illustrate our findings in Fig. 4. In the top row, we display the spatial maps of the marginal predictive mean concentration of Ni (SNT) and Cd (TNT, ICM, FPDa), equipping these with uncertainty represented by the corresponding marginal predictive standard deviation maps (bottom row). These have been computed on a uniform 50 × 50 spatial grid of the input, $x^* \equiv (x_x^*, x_y^*)$. In this sense, the current study can be characterized as an image reconstruction task, driven by non-uniformly sampled and noisy measurements (i.e. pixel values), with registration between the source and target images. The Matérn kernel provides the required prior regularization that induces spatial smoothness in the (Cd) reconstruction [63,64]. Note that the FPDb algorithm is not illustrated in Fig. 4, since it performs poorly (Table 3).

Visual inspection of the TNT, ICM and FPDa columns of Fig. 4 reveals that our algorithm (FPDa) localizes the Cd deposits more sharply than either the ICM algorithm, or (unsurprisingly) the TNT algorithm, which does not benefit from any Ni source learning. In particular, note the horizontal blurring of the Cd predictive mean map for ICM (third column, top), an artefact that is avoided in FPDa (fourth column, top). This localization of the FPDa prediction supports the exploration of Cd deposits in a focussed area around $x^* = (3.8, 4.6)$ km^2. In contrast, the ICM predictive mean map does not sufficiently resolve the x_x coordinate to support an exploration decision in this area.

6. Discussion

We will now assess our FPD-optimal BTL framework (i.e. Algorithm 1 (FPDa)) in the context of the conventional Bayesian

[3] https://sites.google.com/site/goovaertspierre/

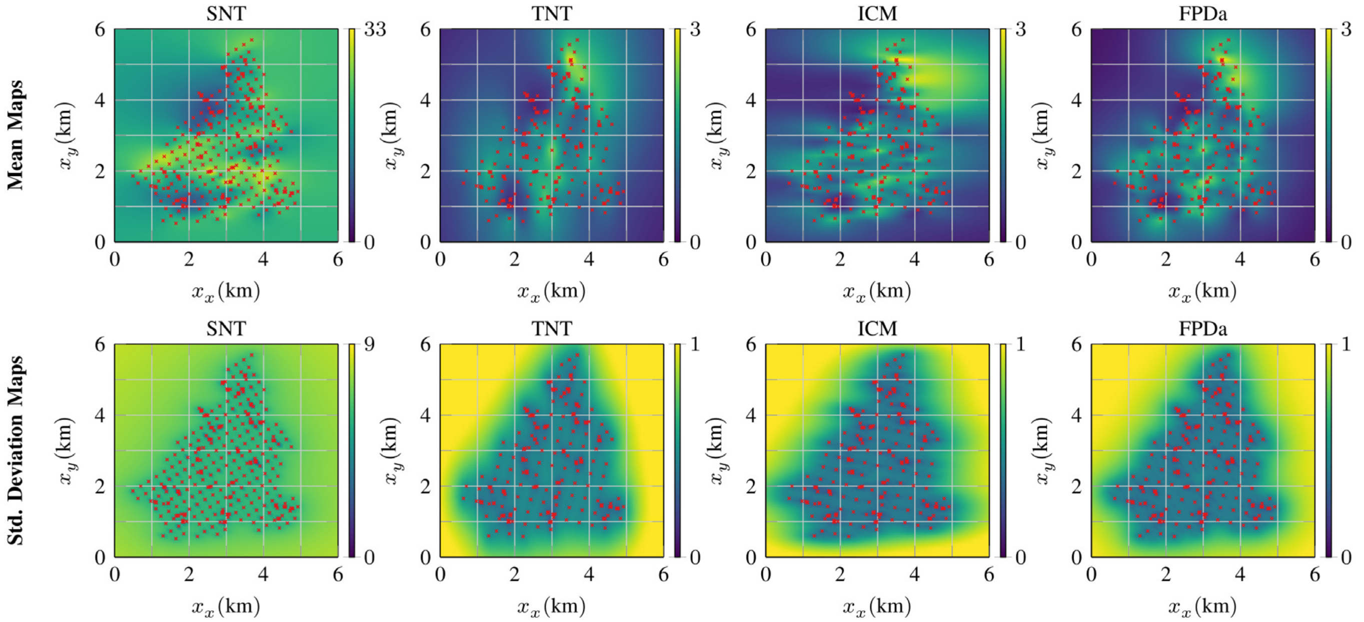


Fig. 4. Real-data experiments. Marginal concentration predictions of Ni (ppm) via the SNT algorithm, and of Cd via the TNT, ICM, and FPDa algorithms (Table 1) in a region of the Swiss Jura. **Top**: the marginal predictive mean map at super-sampled spatial coordinates, $x^* \equiv (x_x^*, x_y^*)$; **bottom**: the associated marginal predictive standard deviation maps. The red crosses are the locations of observed Ni and Cd concentrations used for training (359 for Ni in the source task (first column), and 259 for Cd in the target task (remaining columns)).

multitask learning framework (via the ICM algorithm). We will focus on a number of specific themes that emerge from the evidence in Sections 4 and 5.

Transfer of moments: In the previously considered Gaussian settings [39,40,44,46] of our FPD-optimal BTL framework, the second-order moments of the output-data predictor, F_S, were not successfully transferred, leading to negative transfer. This is also true of FPDb in this paper (Fig. 2(a) and (b) and Table 3). In all those papers, the order of arguments in the KLD was the reverse of the one proposed in this paper (11), a reversal which is essential for the robust transfer property of our FPDa algorithm. This reversal has ensured the successful transfer of the second-order moment, $\bar{\mathbf{R}}_\mathsf{S}$, of (14), even in the current Gaussian setting, yielding the block-diagonal covariance structure (AL1). Equipping the FPDa algorithm with $\bar{\mathbf{R}}_\mathsf{S}$ delivers robust knowledge transfer, as demonstrated in the following remark.

Remark 2. Consider (AL1) for $q = p = \mathsf{T}$ (i.e. the target after transfer) in the limit where the source knowledge becomes uniform (and, therefore, non-informative [65]). In this case,

$$m_\mathsf{T}^\mathsf{o}(x) \rightarrow \bar{m}(x),$$
$$k_\mathsf{TT}^\mathsf{o}(x, x') \rightarrow \bar{k}(x, x'),$$

being the moments of the isolated TNT learning task. This rejection of non-informative source knowledge by the target in our FPDa algorithm is what we mean by robust transfer.

The role of $\bar{\mathbf{R}}_\mathsf{S}$ in effecting robust transfer – as well as in strengthening the positive transfer above threshold which we observed in Fig. 2(a) and (b) – will be studied more formally in a forthcoming paper.

Weighting of the transferred knowledge: Recall that the matrix, $\mathbf{K}$ (16) – which is instantiated in the ICM case in (19) and (20) – expresses the target's interaction with the source. Inspecting (AL1) and (AL2), we also see that $\mathbf{K}$ controls the weighting mechanism in the FPDa algorithm, tuning the influence of the transferred knowledge, F_S, on the optimal target conditional, $\mathsf{M}^\mathsf{o}(\mathbf{f}_\mathsf{T}|\mathsf{F}_\mathsf{S}, \mathbf{y}_\mathsf{T}, \mathbf{K}, \cdot)$ (11). This is seen in the fact that its target moments, $\mathbf{m}_\mathsf{T}^\mathsf{o}$ and $\mathbf{K}_\mathsf{TT}^\mathsf{o}$, are functions of $\mathbf{K}$. The study of this behaviour can be formalized, by adopting similar limit arguments as in Remark 2. Furthermore, the optimization of $\mathbf{K}$ – which is a hyperparameter of $\mathsf{M}^\mathsf{o}(\mathbf{f}_\mathsf{T}|\mathsf{F}_\mathsf{S}, \mathbf{y}_\mathsf{T}, \mathbf{K}, \cdot)$ (see (17)) – via maximum likelihood estimation (Section 5) ensures that this weighting is data-driven (i.e. adaptive). This is also an advance beyond our previous proposals for FPD-optimal BTL [39,40,44], in which no transfer weighting mechanism was induced. This progress has been achieved only because the target modeller is an extended modeller of the interacting source–target tasks, unlike the situation in our previous work where the target models only its local task. Technical details relating to this inducing of a data-driven transfer weighting will be developed further in future work.

Robustness to analysis–synthesis model mismatch: In simulations contexts – such as in Section 4 – the data are simulated from the chosen *synthesis model*, while each assessed algorithm is consistent with an *analysis model*. This provides an opportunity to assess the robustness of each algorithm to model misspecification, as studied in Section 4.2. We have already noted that the ICM algorithm achieved a performance optimum across all the studied algorithms in Section 4 since its (ICM) analysis model equals the (ICM) synthesis model used to generate all data in that study. Nevertheless, the FPDa algorithm performs almost as well (Fig. 2), despite the fact that it does not rely on complete instantiation of the synthesis model. Indeed, as explained in Section 2, FPD-optimal BTL does not require completion of the interaction model between the source and target at all, and has therefore proved to be more robust to ignorance of the synthesis model than the ICM algorithm has. In real-data applications – such as in Section 5 – there is, of course, no synthesis model. The benefit of our incompletely modelled approach in terms of robustness has been demonstrated: FPDa can provide a closer fit to the data than ICM, as presented in Fig. 4 and Table 3. The SNT task involves a second independent local modeller (Fig. 1(c)) which constructs its source output-data predictor, F_S (14), using the same source data that the ICM algorithm processes. The source's local modelling expertise (Section 4.2) has enriched the knowledge transfer, and provided a supplementary learning resource which is not available to ICM.

Computational load of our BTL algorithm: The principal computational cost of all the algorithms in Table 1 resides in inverting the linear systems involved in the standard Gaussian (conditional) data update (13). This cost scales cubically with the number of data in the source and target tasks, respectively, i.e. $\mathcal{O}(n_q^3)$, $q \in (\mathsf{S}, \mathsf{T})$ [66]. Owing to the sequential nature of the knowledge processing in FPDb (i.e. source processing of raw data $\mathbf{y}_\mathsf{S}$, *then* target processing of the transferred source Bayesian predictor, F_S, along with the target raw data, $\mathbf{y}_\mathsf{T}$), the net computational load of FPDb is $\mathcal{O}(n_\mathsf{S}^3) + \mathcal{O}(n_\mathsf{T}^3)$. In contrast, our new BTL algorithm (FPDa) shares the property with ICM that knowledge flow can be bidirectional. This leads to the matrix augmentation form of the inverse systems in (AL1) and, correspondingly, a more computationally expensive – $\mathcal{O}((n_\mathsf{S}+n_\mathsf{T})^3)$ – algorithm. There is a rich literature on efficient inversions for GP regression, exploiting possible reductions (notably sparsity) in the covariance matrix, $\mathbf{K}$ (16). These same reductions have been applied in multitask GP learning [66], such as in ICM. Since our new FPDa BTL algorithm shares the same augmented matrix structure (AL1) as ICM, it, too, can benefit from these reductions.

Source knowledge compression: Conventional multitask learning – such as in Fig. 1(b) – requires the communication of all n_S of the source's extended (training) data, $(\mathbf{x}_\mathsf{S}, \mathbf{y}_\mathsf{S})$, to the processor (either centrally or in some distributed manner), i.e. the transferred message size (quantified as the number of scalars) is $n_\mathsf{S}(n_{x_\mathsf{S}} + 1)$. Conversely, our FPDa algorithm (Fig. 1(c)) is a truly *Bayesian* transfer learning algorithm, whose communication overhead resides only in the transfer of the source's sufficient statistics, $(\bar{\mathbf{m}}_\mathsf{S}, \bar{\mathbf{R}}_\mathsf{S})$, of its output-data predictor at predictive inputs, $\mathbf{x}_\mathsf{S}^\star$, i.e. $\mathsf{F}_\mathsf{S}(\mathbf{y}_\mathsf{S}^\star|\mathbf{y}_\mathsf{S}, \mathbf{x}_\mathsf{S}, \mathbf{x}_\mathsf{S}^\star, \theta_\mathsf{S}) = \mathcal{N}(\bar{\mathbf{m}}_\mathsf{S}, \bar{\mathbf{R}}_\mathsf{S})$ (14). Therefore, the transferred knowledge is fully encoded by these source sufficient statistics without the requirement to transfer the raw extended source data, $(\mathbf{y}_\mathsf{S}, \mathbf{x}_\mathsf{S}, \mathbf{x}_\mathsf{S}^\star)$. The message size for our BTL algorithm is therefore $n_\mathsf{S}^\star(1+\frac{n_\mathsf{S}^\star+1}{2})$. It follows that the condition which must hold for compression to be achieved in our BTL algorithm versus conventional raw data transfer is[4] $n_\mathsf{S}^\star < 2n_{x_\mathsf{S}} - 1$, i.e. the number of predictive points must be lower than two times the dimension of each input (minus one). This condition expresses the objective of avoiding dilution of source knowledge in the predictor [67]. Note also that the source sufficient statistics are functions of the source GP kernel structure, k_S (Remark 1). This can be highly expressive – with many degrees of freedom – in cases of local source expertise. We have provided a preliminary study of the benefit to the target of a distinct source covariance structure (Section 4.2, and see also the third discussion theme above). Indeed, far more expressive covariance structures in k_S – involving weighted combinations of canonical kernel functions (Table 2) – can also be adopted by the source [6]. In all cases, our Bayesian knowledge transfer compresses this resource into dimension-invariant sufficient statistics for subsequent processing by the target task.

We now summarize key findings in respect of our BTL framework (and its incomplete modelling of interaction between the independent source and target modellers (Fig. 1(c))) versus conventional multitask learning (and its complete and unitary modelling of all source–target variables (Fig. 1(b))).

- Our FPD-optimal BTL algorithm (Fig. 1(c)) is truly a *transfer* learning algorithm – as opposed to a multitask learning algorithm – because it does not require complete modelling of source–target interactions. Among the consequences we discovered in this paper are (i) the possibility to avoid problems of model misspecification, and (ii) the opportunity to transfer potentially compressed source sufficient statistics of the data-predictive distribution, F_S, instead of the raw data themselves.
- The source and target modellers (Fig. 1(c)) are independent, and, indeed, our BTL algorithm is truly a multiple model algorithm, in contrast to the single global modeller of multitask learning (Fig. 1(b)). We have shown that a direct consequence is the ability of our algorithm to transfer source knowledge enriched by an expressive local kernel structure.
- Our BTL algorithm delegates the processing of local data to the local (source) task before the resulting stochastic message, F_S, is transferred for subsequent processing by the target task (along with its local (target) data). This opens up the possibility for distributed and parallel processing in a way that is not intrinsic to multitask learning.
- In our formulation of BTL (Fig. 1(c)), we have exploited the opportunity to transfer knowledge to a target task that is, itself, a global modeller. The paper has presented evidence to show that this extension of the target ensures that unreliable source knowledge is rejected (i.e. robust transfer is achieved (Remark 2)), and our positive transfer above threshold is as good as that of conventional multitask learning, such as ICM.

7. Conclusion

A vulnerability of conventional multitask learning arises from the fact that there must exist a global modeller of all the tasks in the system. This imposes a requirement on the global modeller to capture task interactions accurately, and performance is undermined otherwise. Furthermore, it is intrinsic to that framework that the global modeller must process raw data from the remote sources. This potentially incurs a communication overhead and forces the target to process remote source data about which it may lack expertise. Against this, the complete nature of multitask modelling ensures its optimal performance in the (admittedly unlikely) case where task interactions are accurately modelled.

The FPD-optimal Bayesian transfer learning (BTL) framework developed and tested in this paper has achieved important progress beyond the conventional state-of-the-art above. Its key advance is that it does not require elicitation of a model of dependence between the interacting tasks (a defining aspect of BTL, in our opinion, called *incomplete modelling*), and so our approach avoids the misspecification that inevitably arises in conventional multitask learning. Instead, it chooses the conditional target distribution – from among the uncountable cases consistent with the partial model and with the knowledge constraints – in a decision-theoretically optimal manner. A number of important practical benefits flow from this. Firstly, the local source processes its local data – exploiting its local modelling expertise – into a Bayesian (i.e. probabilistic) predictor. Secondly, only the sufficient statistics of this predictor need be transferred to the target. Thirdly, the target *then* sequentially has the opportunity to process this source knowledge along with its local target data. Fourthly, our framework allows this target to be a global modeller of both tasks, ensuring that optimal positive transfer is preserved in the case specified at the end of the previous paragraph.

The success of our algorithm hinges on its ability to transfer all moments of the source predictor, an advance beyond our earlier variants of BTL, and achieved via careful specification of the KLD objective (11). The source covariance matrix, $\bar{\mathbf{R}}_\mathsf{S}$, has been vital in knowledge-driving the weighting attached to the transferred predictor by the target. This has obviated the need for hierarchical relaxation of the target model which was necessary in our previous work in order to learn this weight, and which incurred relatively expensive variational approximation [41,45].

The FPD-optimal BTL design of the target's source-conditional learning (11) can be interpreted as a binary operator, closed within the class of probability distributions. It operates (non-commutatively) on the transferred source output-data predictor, F_S, and on the target's posterior distribution, $\mathsf{F}(\mathbf{f}_\mathsf{S}, \mathbf{f}_\mathsf{T}|\mathbf{y}_\mathsf{T})$,

[4] Here, we quote the implied inequality for the special case, $n_\mathsf{S}^\star \equiv n_\mathsf{S}$.

yielding the optimal-knowledge-conditional target posterior inference, $\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}})$. This closure of our BTL operator within the class of probability distributions recommends it in a wide range of continual learning tasks [63,68]. For instance, in incompletely modelled networks, the requirement for all BTL knowledge objects to be distributions ensures that local computational resources onboard local nodes are exploited in processing local data into these local distributions. Furthermore, target nodes can recursively act as source nodes in a continual learning process, so that probabilistic decision-making is distributed across a network, subject to specification of the network architecture. Such applications of our work in Bayesian networks – and, indeed, in deep learning contexts [69] – can provide a rich opportunity for our Bayesian transfer learning paradigm to address key technology challenges at this time.

CRediT authorship contribution statement

Milan Papež: Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Anthony Quinn:** Conceptualization, Methodology, Funding acquisition, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Preliminaries

Lemma 1. *Let us consider the following factorized, jointly Gaussian model:*

$$\mathsf{F}(\mathbf{y}|\mathbf{f}) \equiv \mathcal{N}(\mathbf{y}; \mathbf{f}, \boldsymbol{\Sigma}),$$
$$\mathsf{F}(\mathbf{f}) \equiv \mathcal{N}(\mathbf{f}; \mathbf{m}, \mathbf{K}).$$

Then, the joint distribution is

$$\mathsf{F}(\mathbf{f}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\mathbf{m}\\ \mathbf{m}\end{bmatrix}, \begin{bmatrix}\mathbf{K} & \mathbf{K}\\ \mathbf{K} & \mathbf{K}+\boldsymbol{\Sigma}\end{bmatrix}\right). \tag{A.1}$$

Proof. The proof follows from standard analysis. □

Lemma 2. *Let us consider a slight generalization of the Gaussian distribution in (A.1):*

$$\mathsf{F}(\mathbf{f}, \mathbf{y}) \equiv \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\bar{\mathbf{f}}\\ \bar{\mathbf{y}}\end{bmatrix}, \begin{bmatrix}\mathbf{K} & \mathbf{K}\\ \mathbf{K} & \mathbf{K}+\boldsymbol{\Sigma}\end{bmatrix}\right).$$

Then, the conditional and marginal densities are

$$\mathsf{F}(\mathbf{f}|\mathbf{y}) = \mathcal{N}(\mathbf{f}; \bar{\mathbf{m}}, \bar{\mathbf{K}}),$$
$$\mathsf{F}(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \bar{\mathbf{y}}, \mathbf{K}+\boldsymbol{\Sigma}),$$

where

$$\bar{\mathbf{m}} = \bar{\mathbf{f}} + \mathbf{K}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}(\mathbf{y}-\bar{\mathbf{y}}),$$
$$\bar{\mathbf{K}} = \mathbf{K} - \mathbf{K}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\mathbf{K}.$$

Proof. The proof follows from standard analysis. □

Appendix B. Proof of Proposition 1

By substituting the constrained unknown and ideal models, (5) and (7) respectively, into (3), we can write

$$\begin{aligned}
&\mathcal{D}(\mathsf{M}\|\mathsf{M}_{\mathsf{I}})\\
&= \int \mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta)\mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)\\
&\quad \times \log \frac{\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta)\mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)}{\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{y}_{\mathsf{S}})\mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathbf{y}_{\mathsf{T}}, \theta)} d\mathbf{y}_{\mathsf{S}}^{\star} d\mathbf{f}_{\mathsf{S}} d\mathbf{f}_{\mathsf{T}}\\
&= \int \mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta) \log\Bigg[\frac{\mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)}{\mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathbf{y}_{\mathsf{T}}, \theta)}\\
&\quad + \int \mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta) \log \frac{\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta)}{\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{y}_{\mathsf{S}})} d\mathbf{y}_{\mathsf{S}}^{\star}\Bigg] d\mathbf{f}_{\mathsf{S}} d\mathbf{f}_{\mathsf{T}}\\
&= \int \mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)\\
&\quad \times \log \frac{\mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)}{\mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathbf{y}_{\mathsf{T}}, \theta) \exp\{-\mathcal{D}(\mathsf{F}\|\mathsf{F}_{\mathsf{S}})\}} d\mathbf{f}_{\mathsf{S}} d\mathbf{f}_{\mathsf{T}}\\
&\quad + \log c_{\mathsf{M}^{o}} - \log c_{\mathsf{M}^{o}}\\
&= \int \mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta) \log\left[\frac{\mathsf{M}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)}{\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta)}\right] d\mathbf{f}_{\mathsf{S}} d\mathbf{f}_{\mathsf{T}}\\
&\quad - \log c_{\mathsf{M}^{o}},
\end{aligned}$$

where the ($\mathbf{f}_{\mathsf{S}}$-conditioned) KLD from $\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta)$ to $\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{y}_{\mathsf{S}})$ is given by

$$\mathcal{D}(\mathsf{F}\|\mathsf{F}_{\mathsf{S}}) = \int \mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta) \log \frac{\mathsf{F}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{f}_{\mathsf{S}}, \theta)}{\mathsf{F}_{\mathsf{S}}(\mathbf{y}_{\mathsf{S}}^{\star}|\mathbf{y}_{\mathsf{S}})} d\mathbf{y}_{\mathsf{S}}^{\star},$$

and the normalizing constant of $\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\cdot)$ is

$$c_{\mathsf{M}^{o}} = \int \mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathbf{y}_{\mathsf{T}}, \theta) \exp\{-\mathcal{D}(\mathsf{F}\|\mathsf{F}_{\mathsf{S}})\} d\mathbf{f}_{\mathsf{S}} d\mathbf{f}_{\mathsf{T}}.$$

These rearrangements reveal the marginal distribution (11) of the FPD-optimal model (10). □

Appendix C. Proof of Proposition 2

To simplify the subsequent rearrangements, let us rewrite (11) as follows:

$$\begin{aligned}
&\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta) \propto\\
&\qquad \mathsf{F}(\mathbf{y}_{\mathsf{T}}|\mathbf{f}_{\mathsf{T}}, \theta)\mathsf{F}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\theta) \exp\{-\mathcal{D}(\mathsf{F}\|\mathsf{F}_{\mathsf{S}})\}.
\end{aligned} \tag{C.1}$$

After adopting (14) and (15a), the exponential term in (C.1) results in

$$\exp\{-\mathcal{D}(\mathsf{F}\|\mathsf{F}_{\mathsf{S}})\} \propto \mathcal{N}(\bar{\mathbf{m}}_{\mathsf{S}}; \mathbf{f}_{\mathsf{S}}, \bar{\mathbf{R}}_{\mathsf{S}}). \tag{C.2}$$

Substituting (15b), (15c), and (C.2) into (C.1), we obtain

$$\begin{aligned}
&\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta) \propto\\
&\qquad \mathcal{N}(\mathbf{y}_{\mathsf{T}}; \mathbf{f}_{\mathsf{T}}, \sigma_{\mathsf{T}}^{2} I_{n_{\mathsf{T}}})\mathcal{N}(\bar{\mathbf{m}}_{\mathsf{S}}; \mathbf{f}_{\mathsf{S}}, \bar{\mathbf{R}}_{\mathsf{S}})\\
&\qquad\qquad \mathcal{N}\left(\begin{bmatrix}\mathbf{f}_{\mathsf{S}}\\ \mathbf{f}_{\mathsf{T}}\end{bmatrix}; \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathsf{SS}} & \mathbf{K}_{\mathsf{ST}}\\ \mathbf{K}_{\mathsf{TS}} & \mathbf{K}_{\mathsf{TT}}\end{bmatrix}\right).
\end{aligned} \tag{C.3}$$

To use Lemma 1 with (C.3), we adopt the following notational assignments:

$$\begin{aligned}
&\mathbf{y} \equiv \begin{bmatrix}\bar{\mathbf{m}}_{\mathsf{S}}\\ \mathbf{y}_{\mathsf{T}}\end{bmatrix}, \quad \mathbf{f} \equiv \begin{bmatrix}\mathbf{f}_{\mathsf{S}}\\ \mathbf{f}_{\mathsf{T}}\end{bmatrix}, \quad \mathbf{m} \equiv \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix},\\
&\boldsymbol{\Sigma} \equiv \begin{bmatrix}\bar{\mathbf{R}}_{\mathsf{S}} & \mathbf{O}\\ \mathbf{O} & \sigma_{\mathsf{T}}^{2} I_{n_{\mathsf{T}}}\end{bmatrix}, \qquad \mathbf{K} \equiv \begin{bmatrix}\mathbf{K}_{\mathsf{SS}} & \mathbf{K}_{\mathsf{ST}}\\ \mathbf{K}_{\mathsf{TS}} & \mathbf{K}_{\mathsf{TT}}\end{bmatrix},
\end{aligned} \tag{C.4}$$

which allows us to write

$$\mathsf{M}^{o}(\mathbf{f}_{\mathsf{S}}, \mathbf{f}_{\mathsf{T}}|\mathsf{F}_{\mathsf{S}}, \mathbf{y}_{\mathsf{T}}, \theta) \propto \tag{C.5}$$

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{f}_\mathsf{S}\\ \mathbf{f}_\mathsf{T}\\ \bar{\mathbf{m}}_\mathsf{S}\\ \mathbf{y}_\mathsf{T}\end{bmatrix};\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix},\begin{bmatrix}\mathbf{K}_\mathsf{SS} & \mathbf{K}_\mathsf{ST} & \mathbf{K}_\mathsf{SS} & \mathbf{K}_\mathsf{ST}\\ \mathbf{K}_\mathsf{TS} & \mathbf{K}_\mathsf{TT} & \mathbf{K}_\mathsf{TS} & \mathbf{K}_\mathsf{TT}\\ \mathbf{K}_\mathsf{SS} & \mathbf{K}_\mathsf{ST} & \mathbf{K}_\mathsf{SS}+\bar{\mathbf{R}}_\mathsf{S} & \mathbf{K}_\mathsf{ST}\\ \mathbf{K}_\mathsf{TS} & \mathbf{K}_\mathsf{TT} & \mathbf{K}_\mathsf{TS} & \mathbf{K}_\mathsf{TT}+\sigma_\mathsf{T}^2 I_{n_\mathsf{T}}\end{bmatrix}\right).$$

Consequently, applying Lemma 2 to (C.5) – with the same blocks as in (C.4) –we obtain

$$\mathsf{M}^\mathsf{o}(\mathbf{f}_\mathsf{S},\mathbf{f}_\mathsf{T}|\mathsf{F}_\mathsf{S},\mathbf{y}_\mathsf{T},\theta)=\mathcal{N}(\mathbf{m}^\mathsf{o},\mathbf{K}^\mathsf{o}),$$

where

$$\mathbf{m}^\mathsf{o}\equiv\begin{bmatrix}\mathbf{m}_\mathsf{S}^\mathsf{o}\\ \mathbf{m}_\mathsf{T}^\mathsf{o}\end{bmatrix}\equiv\begin{bmatrix}m_\mathsf{S}^\mathsf{o}(\mathbf{x}_\mathsf{S})\\ m_\mathsf{T}^\mathsf{o}(\mathbf{x}_\mathsf{T})\end{bmatrix},$$

$$\mathbf{K}^\mathsf{o}\equiv\begin{bmatrix}\mathbf{K}_\mathsf{SS}^\mathsf{o} & \mathbf{K}_\mathsf{ST}^\mathsf{o}\\ \mathbf{K}_\mathsf{TS}^\mathsf{o} & \mathbf{K}_\mathsf{TT}^\mathsf{o}\end{bmatrix}\equiv\begin{bmatrix}k_\mathsf{SS}^\mathsf{o}(\mathbf{x}_\mathsf{S},\mathbf{x}_\mathsf{S}) & k_\mathsf{ST}^\mathsf{o}(\mathbf{x}_\mathsf{S},\mathbf{x}_\mathsf{T})\\ k_\mathsf{TS}^\mathsf{o}(\mathbf{x}_\mathsf{T},\mathbf{x}_\mathsf{S}) & k_\mathsf{TT}^\mathsf{o}(\mathbf{x}_\mathsf{T},\mathbf{x}_\mathsf{T})\end{bmatrix},$$

which is subblock-wise computed, $q,p\in(\mathsf{S},\mathsf{T})$, as

$$m_q^\mathsf{o}(\mathbf{x}_q)=\mathbf{k}_q(\mathbf{K}+\mathrm{blkdiag}(\bar{\mathbf{R}}_\mathsf{S},\sigma_\mathsf{T}^2 I_n))^{-1}\begin{bmatrix}\bar{\mathbf{m}}_\mathsf{S}\\ \mathbf{y}_\mathsf{T}\end{bmatrix},$$

$$k_{qp}^\mathsf{o}(\mathbf{x}_q,\mathbf{x}_p)=k_{qp}(\mathbf{x}_q,\mathbf{x}_p)\\ \qquad -\mathbf{k}_q(\mathbf{K}+\mathrm{blkdiag}(\bar{\mathbf{R}}_\mathsf{S},\sigma_\mathsf{T}^2 I_n))^{-1}\mathbf{k}_p,$$

and

$$\mathbf{k}_q\equiv\begin{bmatrix}k_{q\mathsf{S}}(\mathbf{x}_q,\mathbf{x}_\mathsf{S}) & k_{q\mathsf{T}}(\mathbf{x}_q,\mathbf{x}_\mathsf{T})\end{bmatrix},\qquad \mathbf{k}_p\equiv\begin{bmatrix}k_{\mathsf{S}p}(\mathbf{x}_\mathsf{S},\mathbf{x}_p)\\ k_{\mathsf{T}p}(\mathbf{x}_\mathsf{T},\mathbf{x}_p)\end{bmatrix}.$$

Now, extending both $\mathbf{x}_q$ and $\mathbf{x}_p$ with x leads to

$$\mathsf{M}^\mathsf{o}(\mathbf{f}_\mathsf{S},\mathbf{f}_\mathsf{T},f_\mathsf{S},f_\mathsf{T}|\mathsf{F}_\mathsf{S},\mathbf{y}_\mathsf{T},\theta,x)=\mathcal{N}(\mathbf{m}_x^\mathsf{o},\mathbf{K}_x^\mathsf{o}),$$

where

$$\mathbf{m}_x^\mathsf{o}\equiv\begin{bmatrix}m_\mathsf{S}^\mathsf{o}(\mathbf{x}_\mathsf{S})\\ m_\mathsf{T}^\mathsf{o}(\mathbf{x}_\mathsf{T})\\ m_\mathsf{S}^\mathsf{o}(x)\\ m_\mathsf{T}^\mathsf{o}(x)\end{bmatrix},$$

$$\mathbf{K}_x^\mathsf{o}\equiv\begin{bmatrix}k_\mathsf{SS}^\mathsf{o}(\mathbf{x}_\mathsf{S},\mathbf{x}_\mathsf{S}) & k_\mathsf{ST}^\mathsf{o}(\mathbf{x}_\mathsf{S},\mathbf{x}_\mathsf{T}) & k_\mathsf{SS}^\mathsf{o}(\mathbf{x}_\mathsf{S},x') & k_\mathsf{ST}^\mathsf{o}(\mathbf{x}_\mathsf{S},x')\\ k_\mathsf{TS}^\mathsf{o}(\mathbf{x}_\mathsf{T},\mathbf{x}_\mathsf{S}) & k_\mathsf{TT}^\mathsf{o}(\mathbf{x}_\mathsf{T},\mathbf{x}_\mathsf{T}) & k_\mathsf{TS}^\mathsf{o}(\mathbf{x}_\mathsf{T},x') & k_\mathsf{TT}^\mathsf{o}(\mathbf{x}_\mathsf{T},x')\\ k_\mathsf{SS}^\mathsf{o}(x,\mathbf{x}_\mathsf{S}) & k_\mathsf{ST}^\mathsf{o}(x,\mathbf{x}_\mathsf{T}) & k_\mathsf{SS}^\mathsf{o}(x,x') & k_\mathsf{ST}^\mathsf{o}(x,x')\\ k_\mathsf{TS}^\mathsf{o}(x,\mathbf{x}_\mathsf{S}) & k_\mathsf{TT}^\mathsf{o}(x,\mathbf{x}_\mathsf{T}) & k_\mathsf{TS}^\mathsf{o}(x,x') & k_\mathsf{TT}^\mathsf{o}(x,x')\end{bmatrix},$$

which – after marginalizing w.r.t. $(\mathbf{f}_\mathsf{S},\mathbf{f}_\mathsf{T})$ – yields the result (17). □

References

[1] Y. Zhang, Q. Yang, A survey on multi-task learning, 2018, arXiv preprint arXiv:1707.08114.

[2] Y.W. Teh, M. Seeger, M.I. Jordan, Semiparametric latent factor models, in: 10th Conference on Artificial Intelligence and Statistics, AISTATS, 2005.

[3] E.V. Bonilla, K.M. Chai, C. Williams, Multi-task Gaussian process prediction, in: Advances in Neural Information Processing Systems, 2008, pp. 153–160, http://dx.doi.org/10.5555/2981562.2981582.

[4] T.V. Nguyen, E.V. Bonilla, Collaborative multi-output Gaussian processes, in: 38th Conference on Uncertainty in Artificial Intelligence, UAI, AUAI Press, 2014, pp. 643–652, http://dx.doi.org/10.5555/3020751.3020818.

[5] M.A. Álvarez, N.D. Lawrence, Computationally efficient convolved multiple output Gaussian processes, J. Mach. Learn. Res. 12 (2011) 1459–1500, http://dx.doi.org/10.5555/1953048.2021048.

[6] G. Parra, F. Tobar, Spectral mixture kernels for multi-output Gaussian processes, in: Advances in Neural Information Processing Systems, 2017, pp. 6681–6690, http://dx.doi.org/10.5555/3295222.3295413.

[7] P. Moreno-Muñoz, A. Artés-Rodríguez, M. Álvarez, Heterogeneous multi-output Gaussian process prediction, in: Advances in Neural Information Processing Systems, 2018, pp. 6711–6720.

[8] P. Moreno-Muñoz, A. Artés, M.A. Álvarez, Modular Gaussian processes for transfer learning, in: Advances in Neural Information Processing Systems, 2021.

[9] K. Chen, T. van Laarhoven, P. Groot, J. Chen, E. Marchiori, Multi-output convolution spectral mixture for Gaussian processes, IEEE Trans. Neural Netw. Learn. Syst. 31 (7) (2019) 2255–2266, http://dx.doi.org/10.1109/TNNLS.2019.2946082.

[10] M.A. Álvarez, W. Ward, C. Guarnizo, Non-linear process convolutions for multi-output Gaussian processes, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1969–1977.

[11] M. Van der Wilk, C.E. Rasmussen, J. Hensman, Convolutional Gaussian processes, in: Advances in Neural Information Processing Systems, 2017, pp. 2849–2858.

[12] K. Blomqvist, S. Kaski, M. Heinonen, Deep convolutional Gaussian processes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 582–597, http://dx.doi.org/10.1007/978-3-030-46147-8_35.

[13] V. Dutordoir, M. Wilk, A. Artemev, J. Hensman, Bayesian image classification with deep convolutional Gaussian processes, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 1529–1539.

[14] A.G. Wilson, D.A. Knowles, Z. Ghahramani, Gaussian process regression networks, in: 29th International Conference on Machine Learning, ICML, Omnipress, 2012, pp. 1139–1146, http://dx.doi.org/10.5555/3042573.3042720.

[15] T.V. Nguyen, E.V. Bonilla, Efficient variational inference for Gaussian process regression networks, in: 16th Conference on Artificial Intelligence and Statistics, AISTATS, 2013, pp. 472–480.

[16] A. Damianou, N. Lawrence, Deep Gaussian processes, in: Artificial Intelligence and Statistics, 2013, pp. 207–215.

[17] M. Kandemir, Asymmetric transfer learning with deep Gaussian processes, in: 32nd International Conference on Machine Learning, ICML, 2015, pp. 730–738, http://dx.doi.org/10.5555/3045118.3045197.

[18] A. Boustati, R.S. Savage, Multi-task learning in deep Gaussian processes with multi-kernel layers, 2019, arXiv preprint arXiv:1905.12407.

[19] A.G. Wilson, Z. Hu, R. Salakhutdinov, E.P. Xing, Deep kernel learning, in: Artificial Intelligence and Statistics, 2016, pp. 370–378.

[20] J.M. Bernardo, A.F.M. Smith, Bayesian Theory, Wiley, 1994, http://dx.doi.org/10.1002/9780470316870.

[21] A. Lazaric, M. Ghavamzadeh, Bayesian multi-task reinforcement learning, in: 27th International Conference on Machine Learning, ICML, Omnipress, 2010, pp. 599–606, http://dx.doi.org/10.5555/3104322.3104399.

[22] K. Swersky, J. Snoek, R.P. Adams, Multi-task Bayesian optimization, in: Advances in Neural Information Processing Systems, 2013, pp. 2004–2012, http://dx.doi.org/10.5555/2999792.2999836.

[23] G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, J. Gomez-Dans, A survey on Gaussian processes for Earth-observation data analysis: A comprehensive investigation, IEEE Geosci. Remote Sens. Mag. 4 (2) (2016) 58–78, http://dx.doi.org/10.1109/MGRS.2015.2510084.

[24] Y. Zhang, D.-Y. Yeung, Multi-task warped Gaussian process for personalized age estimation, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2622–2629, http://dx.doi.org/10.1109/CVPR.2010.5539975.

[25] M. Ghassemi, M.A.F. Pimentel, T. Naumann, T. Brennan, D.A. Clifton, P. Szolovits, M. Feng, A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data, in: AAAI 29th Conference on Artificial Intelligence, 2015, http://dx.doi.org/10.5555/2887007.2887070.

[26] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, 2010, pp. 242–264, http://dx.doi.org/10.4018/978-1-60566-766-9.ch011.

[27] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.

[28] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, Knowl.-Based Syst. 80 (2015) 14–23, http://dx.doi.org/10.1016/j.knosys.2015.01.010.

[29] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 9, http://dx.doi.org/10.1186/s40537-016-0043-6.

[30] L. Zhang, X. Gao, Transfer adaptation learning: A decade survey, 2020, arXiv preprint arXiv:1903.04687.

[31] M. Wang, W. Deng, Deep visual domain adaptation: A survey, Neurocomputing 312 (2018) 135–153, http://dx.doi.org/10.1016/j.neucom.2018.05.083.

[32] S. Niu, Y. Liu, J. Wang, H. Song, A decade survey of transfer learning (2010–2020), IEEE Trans. Artif. Intell. 1 (2) (2020) 151–166, http://dx.doi.org/10.1109/TAI.2021.3054609.

[33] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2021) 43–76, http://dx.doi.org/10.1109/JPROC.2020.3004555.

[34] G. Wilson, D.J. Cook, A survey of unsupervised deep domain adaptation, ACM Trans. Intell. Syst. Technol. (TIST) 11 (5) (2020) 1–46, http://dx.doi.org/10.1145/3400066.

[35] M. Crawshaw, Multi-task learning with deep neural networks: A survey, 2020, arXiv preprint arXiv:2009.09796.

[36] S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J.E. Gonzalez, A.L. Sangiovanni-Vincentelli, S.A. Seshia, K. Keutzer, A review of single-source deep unsupervised visual domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2022) 473–493, http://dx.doi.org/10.1109/TNNLS.2020.3028503.

[37] W. Zhang, L. Deng, L. Zhang, D. Wu, A survey on negative transfer, 2021, arXiv preprint arXiv:2009.00909.

[38] P. Xu, Z. Deng, J. Wang, Q. Zhang, K.-S. Choi, S. Wang, Transfer representation learning with TSK fuzzy system, IEEE Trans. Fuzzy Syst. 29 (3) (2021) 649–663, http://dx.doi.org/10.1109/TFUZZ.2019.2958299.

[39] C. Foley, A. Quinn, Fully probabilistic design for knowledge transfer in a pair of Kalman filters, IEEE Signal Process. Lett. 25 (4) (2018) 487–490, http://dx.doi.org/10.1109/LSP.2017.2776223.

[40] M. Papež, A. Quinn, Dynamic Bayesian knowledge transfer between a pair of Kalman filters, in: IEEE 28th International Workshop on Machine Learning for Signal Processing, MLSP, IEEE, 2018, http://dx.doi.org/10.1109/MLSP.2018.8517020.

[41] M. Papež, A. Quinn, Robust Bayesian transfer learning between Kalman filters, in: IEEE 29th International Workshop on Machine Learning for Signal Processing, MLSP, IEEE, 2019, http://dx.doi.org/10.1109/MLSP.2019.8918783.

[42] L. Jirsa, L.K. Pavelková, A. Quinn, Knowledge transfer in a pair of uniformly modelled Bayesian filters, in: 2019 16th International Conference on Informatics in Control, Automation and Robotics, ICINCO, IEEE, 2019, http://dx.doi.org/10.5220/0007854104990506.

[43] L.K. Pavelková, L. Jirsa, A. Quinn, Fully probabilistic design for knowledge fusion between Bayesian filters under uniform disturbances, Knowledge-Based Systems 238 (2022) 107879, http://dx.doi.org/10.1016/j.knosys.2021.107879.

[44] M. Papež, A. Quinn, Bayesian transfer learning between Gaussian process regression tasks, in: IEEE 19th International Symposium on Signal Processing and Information Technology, ISSPIT, IEEE, 2019, http://dx.doi.org/10.1109/ISSPIT47144.2019.9001885.

[45] M. Papež, A. Quinn, Bayesian transfer learning between student-*t* filters, Signal Process. 175 (2020) 107624, http://dx.doi.org/10.1016/j.sigpro.2020.107624.

[46] M. Papež, A. Quinn, Hierarchical Bayesian transfer learning between a pair of Kalman filters, in: 32nd Irish Signals and Systems Conference, ISSC, IEEE, 2021, http://dx.doi.org/10.1109/ISSC52156.2021.9467863.

[47] M. Kárný, T. Kroupa, Axiomatisation of fully probabilistic design, Inform. Sci. 186 (1) (2012) 105–113, http://dx.doi.org/10.1016/j.sysconle.2020.104719.

[48] A. Quinn, M. Kárný, T.V. Guy, Fully probabilistic design of hierarchical Bayesian models, Inform. Sci. 369 (2016) 532–547, http://dx.doi.org/10.1016/j.ins.2016.07.035.

[49] J. Shore, R. Johnson, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, IEEE Trans. Inform. Theory 26 (1) (1980) 26–37, http://dx.doi.org/10.1109/TIT.1980.1056144.

[50] J. Kracík, M. Kárný, Merging of data knowledge in Bayesian estimation, in: Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics, INSTICC, Barcelona, 2005, pp. 229–232.

[51] M. Kárný, J. Andrýsek, A. Bodini, T.V. Guy, J. Kracík, F. Ruggeri, How to exploit external model of data for parameter estimation? Internat. J. Adapt. Control Signal Process. 20 (1) (2006) 41–50, http://dx.doi.org/10.1002/acs.886.

[52] S. Azizi, A. Quinn, Hierarchical fully probabilistic design for deliberator-based merging in multiple participant systems, IEEE Trans. Syst. Man Cybern. A 48 (4) (2016) 565–573, http://dx.doi.org/10.1109/TSMC.2016.2608662.

[53] M. Kárný, Towards fully probabilistic control design, Automatica 32 (12) (1996) 1719–1722, http://dx.doi.org/10.1016/S0005-1098(96)80009-4.

[54] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86, http://dx.doi.org/10.1214/aoms/1177729694.

[55] A. Quinn, M. Kárný, T.V. Guy, Optimal design of priors constrained by external predictors, Internat. J. Approx. Reason. 84 (2017) 150–158, http://dx.doi.org/10.1016/j.ijar.2017.02.001.

[56] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006, http://dx.doi.org/10.1117/1.2819119.

[57] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[58] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[59] M.A. Álvarez, L. Rosasco, N.D. Lawrence, Kernels for vector-valued functions: A review, Found. Trends Mach. Learn. 4 (3) (2012) 195–266, http://dx.doi.org/10.1561/2200000036.

[60] E.L. Lehmann, J.P. Romano, G. Casella, Testing Statistical Hypotheses, vol. 3, Springer, 2005.

[61] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560, http://dx.doi.org/10.1145/279232.279236.

[62] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60, http://dx.doi.org/10.1214/aoms/1177730491.

[63] M.B. Ring, Continual Learning in Reinforcement Environments (Ph.D. thesis), University of Texas at Austin, 1994.

[64] J.-F. Ton, S. Flaxman, D. Sejdinovic, S. Bhatt, Spatial mapping with Gaussian processes and nonstationary Fourier features, Spatial Stat. 28 (2018) 59–78, http://dx.doi.org/10.1016/j.spasta.2018.02.002.

[65] H. Jeffreys, Theory of Probability, Clarendon Press, 1961.

[66] H. Liu, Y.-S. Ong, X. Shen, J. Cai, When Gaussian process meets big data: A review of scalable GPs, IEEE Trans. Neural Netw. Learn. Syst. (2020) http://dx.doi.org/10.1109/TNNLS.2019.2957109.

[67] M.J. Wainwright, High-Dimensional Statistics: A Non-Asymptotic Viewpoint, vol. 48, Cambridge University Press, 2019, http://dx.doi.org/10.1017/9781108627771.

[68] C.V. Nguyen, Y. Li, T.D. Bui, R.E. Turner, Variational continual learning, in: International Conference on Learning Representations, 2018, http://dx.doi.org/10.17863/CAM.35471.

[69] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.