# Asymmetric Transfer Learning with Deep Gaussian Processes

Melih Kandemir
melih.kandemir@iwr.uni-heidelberg.de
Heidelberg University, HCI/IWR

**Abstract**

We introduce a novel Gaussian process based Bayesian model for asymmetric transfer learning. We adopt a two-layer feed-forward deep Gaussian process as the task learner of source and target domains. The first layer projects the data onto a separate non-linear manifold for each task. We perform knowledge transfer by projecting the target data also onto the source domain and linearly combining its representations on the source and target domain manifolds. Our approach achieves the state-of-the-art in a benchmark real-world image categorization task, and improves on it in cross-tissue tumor detection from histopathology tissue slide images.

## 1 Introduction

Gaussian processes (GPs) [20] attract wide interest as generic supervised learners. This is not only due to them being effective kernel methods, but also to their probabilistic nature. They can be plugged into a larger probabilistic model of a particular purpose as a component. Furthermore, unlike non-probabilistic discriminative models, they take into account the variance of the predicted data points, which is known to boost up prediction performance [22]. This probabilistic nature and the predictive variance has also been used to develop simple, effective, and theoretically well-grounded active learning models [14]. GPs also allow a principled framework for learning kernel hyperparameters, for which a grid search has to be performed in the support vector machine (SVM) [26] framework.

In this paper, we benefit from the probabilistic nature of GPs, and show how Deep GPs [4] can be used as design components to easily build an effective transfer learning model. We adopt a two-layer feed-forward deep GP model as the task learner. The first-layer GP non-linearly projects the instances onto a latent intermediary representation. This latent representation is then fed into the second-layer GP as input. The knowledge transfer takes place asymmetrically (i.e. from source task to target task only) by projecting the target instances onto the latent source manifold, and this representation is linearly combined with the representation of the target instances on their native manifold. The resultant combination is then fed into the second-layer target GP, which maps it to the output labels. Figure 1 illustrates the idea.

We evaluate our approach on two applications. The first is a real-world image categorization benchmark, where the domains are different image data sets collected by cameras of different resolutions and for different purposes. The second is tumor detection in histopathology tissue slide images [10]. We treat each of the two tissue types, breast and esophagus, as different domains. Our model reaches state-of-the-art prediction performance in the first application, and improves it in the second. The source code of our model is publicly available[1].

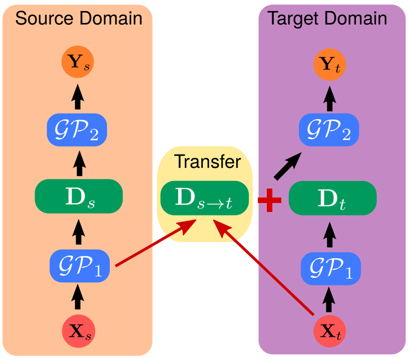

Figure 1: The proposed idea for knowledge transfer from a source deep GP to a target deep GP. We project the target data set $\mathbf{X}_t$ onto the latent source domain space $\mathbf{D}_{s\to t}$ by the first-layer GP ($\mathcal{GP}_1$) of the source task. We then combine the outcome with the latent representation of $\mathbf{X}_t$ on the target domain $\mathbf{D}_t$. Finally, we feed the resultant representation into the second-layer GP ($\mathcal{GP}_2$).

## 2 Prior Art

Transfer learning approaches can be dichotomized into two as *symmetric* and *asymmetric* [15]. In the symmetric approach, identical knowledge transfer mechanisms are established between source and target tasks. A successful representative of this approach is Duan et al. [6], which projects both source and target tasks onto a shared space, augment the shared representations with native features, as originally proposed by Daumé et al. [5], and learn a model on this new representation. Gönen et al. [7] construct task-specific latent representations by taking multiple draws from a Relevance Vector Machine (RVM) [24]. These representations are then linearly mapped to the output by a unified model.

There exist GP based models for symmetric transfer learning. The pioneer work on this line has been by Bonilla et al. [2], which decomposes the covariance matrix of GP into task-specific and shared components. Lázaro-Gredilla et al. [17] learn a weighted combination of multiple kernels for each task, and transfer knowledge by placing a common spike-and-slab prior over the kernel combination weights. Nguyen et al. [19] learn several task-specific and task-independent sparse GPs for each task, and combine them for a joint latent decision margin.

In the asymmetric transfer approach, a transfer mechanism is constructed only from the source task to the target task. Dai et al. [3] make an analogy with transfer learning

[1] `https://github.com/melihkandemir/atldgp`

and text translation, and call it *translated learning*. They propose a Markov chain that translates the classes of the source domain to a target domain. Hoffman et al. [12] project only the target space onto the source space, where both tasks are then learned by a unified model. Leen et al. [18] follow a late fusion strategy, and combine the latent decision margins of multiple GPs operating on the source tasks with the latent decision margin of the target task.

# 3 Our Contribution

Our approach differs from the existing approaches in the following aspects:

- We use two-layer Deep GPs as base learners, and propose to use the output of the first layer for domain adaptation.
- We assign separate learners for source and target tasks, but bind these learners together by linearly combining the projection of the target data onto both source and target manifolds.
- The resultant model both projects input data onto a latent manifold (first-layer GP) and maps this projection to the output space (second-layer GP) *non-linearly*.

# 4 Notation

We denote a vector of variables by boldface lower case letters $\mathbf{v}$, and a matrix of variables by boldface upper case letters $\mathbf{M}$. The $i$th row and $j$th column of a matrix and $i$th element of a vector are given by $[\mathbf{M}](ij)$ and $[\mathbf{v}](i)$, respectively. The operands $[\mathbf{M}_1; \mathbf{M}_2]$, and $[\mathbf{M}_1, \mathbf{M}_2]$ denote row-wise and column-wise concatenation of two matrices or vectors, respectively. We denote $n$th row of a matrix $\mathbf{M}$ by $\mathbf{m}_n$, and its $c$th column by $\mathbf{m}^c$. $\mathbf{I}$ is the identity matrix of the size that is determined by the given context. $\mathbb{E}_{p(\cdot)}[f(\cdot)]$ is the expectation of function $f(\cdot)$ with respect to density $p(\cdot)$, and $\mathbb{H}[p(\cdot)]$ is the Shannon entropy of the density $p(\cdot)$. We use $\mathbb{E}_p[x]$ as a short hand notation for $\mathbb{E}_{p(x)}[x]$. The operator $diag(\mathbf{M})$ returns a vector containing the diagonal elements of matrix $\mathbf{M}$. Lastly, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate normal density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

# 5 The Model

Let $\mathbf{X}_i$ be the $N_i \times D$ dimensional data matrix for task $i$ with $N_i$ instances of $D$ dimensions in its rows, and $\mathbf{Y}_i$ be the $N_i \times C$ matrix having the corresponding real valued outputs. We consider the transfer learning setup, where we have one source task $\{\mathbf{X}_s, \mathbf{Y}_s\}$ for which we have a sufficient number of labeled instances, and one target task $\{\mathbf{X}_t, \mathbf{Y}_t\}$ for which we have a scarce data regime. Our goal is to learn a joint model, which transfers knowledge from the source task to the target task. To this end, we propose a Bayesian model with the generative process

$$
\begin{aligned}
p(\mathbf{Y}|\mathbf{F}) &= \prod_{i\in\{s,t\}} \prod_{c=1}^{C} \mathcal{N}(\mathbf{y}_i^c|\mathbf{f}_i^c, \beta^{-1}\mathbf{I}),\\
p(\mathbf{F}|\mathbf{D}) &= \prod_{i\in\{s,t\}} \prod_{c=1}^{C} \mathcal{N}(\mathbf{f}_i^c|\mathbf{0}, \mathbf{K}_{\mathbf{D}_i\mathbf{D}_i}),\\
p(\mathbf{D}_t|\mathbf{B}_t, \mathbf{B}_{s\to t}, \pi) &= \prod_{n=1}^{N_t} \mathcal{N}(\mathbf{d}_n^t|\pi\mathbf{b}_n^{s\to t} + (1-\pi)\mathbf{b}_n^t, \alpha^{-1}\mathbf{I}),\\
p(\pi) &= Beta(\pi|e, f),\\
p(\mathbf{B}_t|\mathbf{X}_t) &= \prod_{r=1}^{R} \mathcal{N}(\mathbf{b}_t^r|\mathbf{0}, \mathbf{K}_{\mathbf{X}_t\mathbf{X}_t}),\\
p(\mathbf{D}_s|\mathbf{B}_s) &= \prod_{r=1}^{R} \mathcal{N}(\mathbf{d}_s^r|\mathbf{b}_s^r, \lambda^{-1}\mathbf{I}),\\
p([\mathbf{B}_s;\mathbf{B}_{s\to t}]|\mathbf{X}_s, \mathbf{X}_t) &= \prod_{r=1}^{R} \mathcal{N}([\mathbf{b}_s^r;\mathbf{b}_s^t]|\mathbf{0}, \mathbf{K}_{[\mathbf{X}_s;\mathbf{X}_t][\mathbf{X}_s;\mathbf{X}_t]}),
\end{aligned}
$$

where $\mathbf{Y} = \{\mathbf{Y}_s, \mathbf{Y}_t\}$, $\mathbf{F} = \{\mathbf{F}_s, \mathbf{F}_t\}$ are decision margins, $\mathbf{D} = \{\mathbf{D}_s, \mathbf{D}_t\}$ are latent representations of input data, $\mathbf{K}_{\mathbf{XX}}$ is a Gram matrix with $[\mathbf{K}_{\mathbf{X}}](ij) = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\alpha$, $\beta$, and $\lambda$ are precisions of normal densities. Here, $\mathbf{B}_i = [\mathbf{b}_i^1, \cdots, \mathbf{b}_i^R]$ is the representation of the task $i$ instances on the latent non-linear manifold in its native latent space. This latent mapping is performed by a first-layer GP. For the target task, we additionally have $\mathbf{B}_{s\to t}$, which is the projection of the target instances onto the latent source space. The two representations of the target instances, one in the source and one in the target space are blended into one single representation $\mathbf{D}_t = [\mathbf{d}_t^1, \cdots, \mathbf{d}_t^R]$ by weighted averaging. The mixture weight $\pi$ follows a Beta distribution hyperparameterized by $e$ and $f$. For both tasks, the second layer GP takes $\mathbf{D}_i$ as input and maps it to the output labels $\mathbf{Y}_i$. The parts of the process responsible for the knowledge transfer are shown in blue. We call this model *Asymmetric Transfer Learning with Deep Gaussian Processes*, and abbreviate it as **ATL-DGP**.

## 5.1 The sparse Gaussian process prior

While being effective non-linear models, GPs have the disadvantage of requiring the inversion of the $N \times N$ covariance matrix at every iteration to learn the kernel hyperparameters, $N$ being the sample size. A workaround would be to approximate this matrix with another lower rank matrix. We also adopt this solution, and use *Fully Independent Training Conditional (FITC)* approximation by Snelson et al. [23], due to its eligibility to variational inference [1]. Successful applications of variational inference to FITC approximation include Deep GPs [4], GPs for large data masses [11], and the Bayesian Gaussian process latent variable model (GPLVM) [25].

A FITC approximated sparse GP assumes a small set of data points, called *inducing points*, as given, and predicts the data from the inducing points, which results in

$$
\mathcal{SGP}(\mathbf{f}, \mathbf{u}|\mathbf{X}, \mathbf{Z}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}})\mathcal{N}(\mathbf{f}|\mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{u}, diag(\mathbf{K}_{\mathbf{XX}} - \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{K}_{\mathbf{ZX}})),
$$

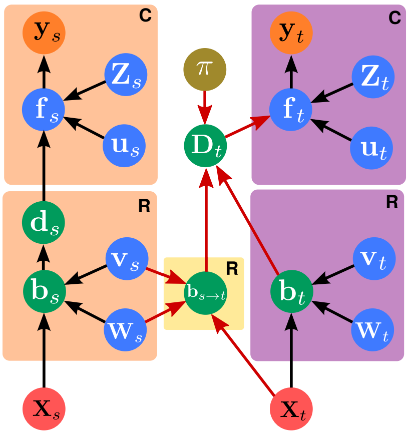

Figure 2: The plate diagram of the proposed model. For clarity, we use the same coloring conventions as Figure 1.

where $\mathbf{f}$ is the vector of decision margins for all data points, and $\mathbf{Z} = [\mathbf{z}_1; \cdots; \mathbf{z}_P]$ has the inducing points in its rows. We call $\mathbf{u}$ the *inducing output* vector, since it corresponds to the output labels of the inducing points in the GP predictive mean formula. The inducing points can be chosen by subsampling the data set, or can be treated as model parameters, hence *pseudo inputs*, and learned from data.

### 5.2 Asymmetric transfer learning by deep sparse Gaussian processes

The posterior density of the model given above for **ATL-DGP** is not tractable, hence approximate inference is needed. The typical Laplace approximation would not be practical, since the model has much more latent variables than the standard GP, which would involve taking the inverse Hessian of a much larger matrix. It can easily be seen that multiple dependencies between variables are non-conjugate, such as the normal distributed $\mathbf{D}_t$, which serves as an input to a GP in the next stage, hence is passed through a kernel function to construct a covariance matrix. Due to these non-conjugacies, Gibbs sampling is also not applicable in its naive form. Most Metropolis-like samplers also suffer from large number of covariates. Hence, we approximate the posterior by variational inference. We convert the full GPs to sparse GPs, and attain

$$\begin{aligned}
p(\mathbf{Y}|\mathbf{F}) &= \textstyle\prod_{i\in\{s,t\}} \prod_{c=1}^{C} \mathcal{N}(\mathbf{y}_i^c|\mathbf{f}_i^c, \beta^{-1}\mathbf{I}) \\
p(\mathbf{F}, \mathbf{U}|\mathbf{D}, \mathbf{Z}) &= \textstyle\prod_{i\in\{s,t\}} \prod_{c=1}^{C} \mathcal{SGP}(\mathbf{f}_i^c, \mathbf{u}_i^c|\mathbf{D}_i, \mathbf{Z}_i^c) \\
p(\mathbf{B}, \mathbf{V}|\mathbf{X}, \mathbf{W}) &= \textstyle\prod_{i\in\{s,t\}} \prod_{r=1}^{R} \mathcal{SGP}(\mathbf{b}_i^r, \mathbf{v}_i^r|\mathbf{X}_i, \mathbf{W}_i^r) \\
p(\mathbf{D}_t|\mathbf{B}_t, \mathbf{B}_{s\to t}, \pi) &= \textstyle\prod_{n=1}^{N_t} \mathcal{N}(\mathbf{d}_n^t|\pi \mathbf{b}_n^{s\to t} + (1-\pi)\mathbf{b}_n^t, \alpha^{-1}\mathbf{I}) \\
p(\mathbf{B}_{s\to t}|\mathbf{X}_t, \mathbf{W}_s, \mathbf{V}_s) &= \textstyle\prod_{r=1}^{R} \mathcal{N}(\mathbf{b}_{s\to t}^r|\mathbf{K}_{\mathbf{X}_t\mathbf{W}_s}\mathbf{K}_{\mathbf{W}_s\mathbf{W}_s}^{-1}\mathbf{v}_s^r, \\
&\quad diag(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} - \mathbf{K}_{\mathbf{X}_t\mathbf{W}_s}\mathbf{K}_{\mathbf{W}_s\mathbf{W}_s}^{-1}\mathbf{K}_{\mathbf{W}_s\mathbf{X}_t})) \\
p(\pi) &= Beta(\pi|e, f) \\
p(\mathbf{D}_s|\mathbf{B}_s) &= \textstyle\prod_{r=1}^{R} \mathcal{N}(\mathbf{d}_s^r|\mathbf{b}_s^r, \lambda^{-1}\mathbf{I}).
\end{aligned}$$

Here, $\mathbf{W}_i^r$ and $\mathbf{Z}_i^c$ are inducing pseudo data sets, and $\mathbf{v}_i^c$ and $\mathbf{u}_i^c$ are the inducing outputs for the first and second layer GPs, respectively. We highlight the densities responsible for the knowledge transfer in blue. The dependencies of the model variables are shown in the plate diagram in Figure 2.

For binary outputs, we add

$$\begin{aligned} p(\mathbf{t}_i^c|\mathbf{y}_i^c) &= \prod_{n=1}^{N_i} Bernoulli\Big([\mathbf{t}_i^c](n)\Big|\Phi([\mathbf{t}_i^c](n))\Big), \\ &= \prod_{n=1}^{N} \Phi\Big([\mathbf{y}_i^c](n)\Big)^{[\mathbf{t}_i^c](n)}\Big(1-\Phi\Big([\mathbf{y}_i^c](n)\Big)\Big)^{1-[\mathbf{t}_i^c](n)} \end{aligned}$$

to each output dimension of each task. Here, $\Phi(s) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}s^2}$ is the probit link function and $\mathbf{t}_i^c$ is the vector of output classes $[\mathbf{t}_i^c](n) \in \{0, 1\}$. For multiclass classification, each output dimension can be assigned to one class, and 1-of-K coding can be used. The latent representation layer binds these output tasks to each other, hence they are learned jointly.

**Variational inference.** Using Jensen's inequality, we obtain a lower bound for the log-marginal density

$$\log\ p(\mathbf{Y}|\mathbf{Z}, \mathbf{W}, \mathbf{X}) \geq \mathcal{L} = \mathbb{E}_Q[\log\ p(\mathbf{Y}, \mathbf{F}, \mathbf{U}, \mathbf{V}, \mathbf{D}, \mathbf{B}, \pi|\mathbf{Z}, \mathbf{W}, \mathbf{X})]$$

where $\mathcal{L}$ is the variational lower bound. We define the variational density as

$$Q = p(\mathbf{B}_{s\to t}|\mathbf{X}_t, \mathbf{W}_s, \mathbf{V}_s)q(\mathbf{D}_i)q(\pi) \prod_{i\in\{s,t\}} Q_i,$$

where

$$Q_i = p(\mathbf{F}_i|\mathbf{U}_i, \mathbf{D}_i, \mathbf{Z}_i)q(\mathbf{U}_i)p(\mathbf{B}_i|\mathbf{X}_i, \mathbf{W}_i, \mathbf{V}_i)q(\mathbf{V}_i),$$

and

$$\begin{aligned} q(\mathbf{U}) &= \prod_{i\in\{s,t\}} q(\mathbf{U}_i) = \prod_{i\in\{s,t\}} \prod_{c=1}^{C} \mathcal{N}(\mathbf{u}_i^c|\mathbf{m}_i^c, \mathbf{S}_i^c), \\ q(\mathbf{D}_t) &= \prod_{n=1}^{N_t} \mathcal{N}(\mathbf{d}_n^t|\mathbf{h}_n^t, \gamma_t^{-1}\mathbf{I}), \quad q(\pi) = Beta(\pi|g, h) \\ q(\mathbf{D}_s) &= \prod_{r=1}^{R} \mathcal{N}(\mathbf{d}_s^r|\mathbf{K}_{\mathbf{X}_s\mathbf{X}_u^r}\mathbf{K}_{\mathbf{X}_u^r\mathbf{X}_u^r}^{-1}\mathbf{e}_s^r, \gamma_s^{-1}\mathbf{I}), \\ q(\mathbf{V}) &= \prod_{i\in\{s,t\}} q(\mathbf{V}_i) = \prod_{i\in\{s,t\}} \prod_{r=1}^{R} \mathcal{N}(\mathbf{v}_i^r|\mathbf{e}_i^r, \mathbf{G}_i^r), \end{aligned}$$

where $\mathbf{X}_u^r$ is a randomly chosen subset of data points from the source data set. The essence of the above factorization is that the factors corresponding to a GP predictive density, hence involving latent variables assigned to each data point, are cancelled out by keeping them identical in $Q$. The inducing points $\mathbf{Z}$ and $\mathbf{W}$ can also be learned to maximize $\mathcal{L}$, as suggested by Titsias et al. [25].

For the target task, which is expected to have a small sample size, we use a factor density $q(\mathbf{D}_t)$, which consists of fully parameterized multivariate normal densities per latent dimension, identically to Damianou et al. [4] (see Equation 11). As for the source task, which desirably has a much larger sample size, we prefer a sparser parameterization. We assume $q(\mathbf{d}_s^r)$ to follow a normal distribution whose mean is a kernel linear regressor that maps $\mathbf{X}_u^r$ to the latent manifold $\mathbf{d}_s^r$. The variational inducing output parameters $\mathbf{e}_i^r$ are assumed to be pseudo outputs of $\mathbf{X}_u^r$. This fixes the number of variational parameters required per latent dimension to the number of inducing points. This way, overparameterization of the model is avoided, and it is protected against overfitting.

**The variational lower bound.** The variational lower bound can be decomposed into three terms: $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_t + \mathcal{L}_{asy}$, where $\mathcal{L}_s$ and $\mathcal{L}_t$ include terms identical for both tasks, and $\mathcal{L}_{asy}$ is the sum of the terms asymmetric across tasks, which reads as

$$\mathcal{L}_{asy} = \mathbb{E}_{Q_{asy}}[\log p(\mathbf{D}_t|\mathbf{B}_t, \mathbf{B}_{s\rightarrow t}, \pi)] + \mathbb{E}_q[\log p(\mathbf{D}_s|\mathbf{B}_s)] + \mathbb{E}_q[\log p(\pi)] + \mathbb{H}[q(\pi)],$$

where $Q_{asy} = q(\mathbf{D}_t) \times p(\mathbf{B}_{s\rightarrow t}|\mathbf{X}_t, \mathbf{W}_s, \mathbf{V}_s) \times q(\mathbf{V}_s) \times p(\mathbf{B}_t|\mathbf{X}_t, \mathbf{W}_t, \mathbf{V}_t) \times q(\mathbf{V}_t) \times q(\pi)$. Taking the expectations with respect to the variational densities, we get

$$\begin{aligned}
\mathcal{L}_{asy} = {} & \alpha \sum_{n=1}^{N_t} \mathbb{E}_q[\mathbf{d}_n^t]^T \Big( \mathbb{E}_p[\mathbf{b}_n^{s\rightarrow t}]\mathbb{E}_q[\pi] + \mathbb{E}_p[\mathbf{b}_n^t](1 - \mathbb{E}_q[\pi]) \Big) \\
& - \frac{\alpha}{2} \sum_{n=1}^{N} \mathbb{E}_q[(\mathbf{d}_n^t)^T \mathbf{d}_n^t] - \frac{\alpha}{2} \sum_{n=1}^{N_t} \mathbb{E}_p[(\mathbf{b}_n^{s\rightarrow t})^T \mathbf{b}_n^{s\rightarrow t}]\mathbb{E}_q[\pi^2] \\
& - \alpha \sum_{n=1}^{N_t} \mathbb{E}_p[\mathbf{b}_n^{s\rightarrow t}]^T \mathbb{E}[\mathbf{b}_n^t] \Big( \mathbb{E}_q[\pi] - \mathbb{E}_q[\pi^2] \Big) \\
& - \frac{\alpha}{2} \sum_{n=1}^{N_t} \mathbb{E}_p[(\mathbf{b}_n^t)^T \mathbf{b}_n^t] \Big( 1 - 2\mathbb{E}_q[\pi] + \mathbb{E}_q[\pi^2] \Big) + \mathbb{H}[q(\pi)] \\
& + \sum_{r=1}^{R} \Big( \lambda \mathbb{E}_p[\mathbf{b}_s^r]^T \mathbb{E}_q[\mathbf{d}_s^r] - \frac{\lambda}{2} \mathbb{E}_p[(\mathbf{b}_s^r)^T \mathbf{b}_s^r] - \frac{\lambda}{2} \mathbb{E}_q[(\mathbf{d}_s^r)^T \mathbf{d}_s^r] \Big) \\
& + (e-1)\mathbb{E}_q[\log \pi] + (f-1)\mathbb{E}_q[\log(1-\pi)].
\end{aligned}$$

Here, $\mathbb{E}_q[\pi] = \frac{g}{h}$ and $\mathbb{E}_q[\pi^2] = \left(\frac{g}{h}\right)^2 + \frac{gh}{(g+h)^2(g+h+1)}$, and $\mathbb{H}[q(\pi)] = \log B(g,h) - (g-1)\psi(g) - (h-1)\psi(h) + (g+h+2)\psi(g+h)$ are given by the standard identities of the Beta distribution, where $\psi(\cdot)$ is the digamma function. The derivative of the Beta function $B(\cdot,\cdot)$ can simply be approximated by finite difference.

The task-specific lower bound term is

$$\begin{aligned}
\mathcal{L}_i = {} & \sum_{r=1}^{R} \left( \mathbb{E}_{q(\mathbf{v}_i^r)}[\log p(\mathbf{v}_i^r|\mathbf{X}_i)] + \mathbb{H}[q(\mathbf{v}_i^r)] \right) + \mathbb{H}[q(\mathbf{D}_i)] \\
& + \sum_{c=1}^{C} \left( \mathbb{E}_{q(\mathbf{u}_i^c)}[\log p(\mathbf{u}_i^c|\mathbf{Z}_i^c)] + \mathbb{E}_{Q_i}[\log p(\mathbf{y}_i^c|\mathbf{f}_i^c)] + \mathbb{H}[q(\mathbf{u}_i^c)] \right).
\end{aligned}$$

Extending the terms and taking the expectations, we get

$$\begin{aligned}\mathcal{L}_i = \sum_{c=1}^{C} \Bigg( & \beta(\mathbf{y}_i^c)^T \mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}]^T \mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1} \mathbf{m}_i^c + \frac{N}{2}\log\beta \\
& - \frac{\beta}{2} tr\Big\{ \mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1} \mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}^T] \mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1} (\mathbf{m}_i^c(\mathbf{m}_i^c)^T + \mathbf{S}_i^c) \Big\} \\
& + \frac{\beta}{2} tr\Big\{ \mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1} \mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}^T] \Big\} + \frac{1}{2}\log|\mathbf{S}_i^c| - \frac{1}{2}\log|\mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}| \\
& - \frac{1}{2}(\mathbf{m}_i^c)^T \mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1} \mathbf{m}_i^c - \frac{\beta}{2} tr\{\mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{D}_i\mathbf{D}_i}]\} - \frac{1}{2} tr(\mathbf{K}_{\mathbf{Z}_i^c\mathbf{Z}_i^c}^{-1}\mathbf{S}_i^c) - \frac{\beta}{2}(\mathbf{y}_i^c)^T\mathbf{y}_i^c \Bigg) \\
& + \frac{1}{2}\sum_{r}^{R}\log|\mathbf{G}_i^r| - \frac{N_i R}{2}\log\gamma_i - \frac{1}{2}\sum_{r}^{R} tr\Big\{ \mathbf{K}_{\mathbf{W}_i^r\mathbf{W}_i^r}^{-1} \Big( \mathbf{e}_i^r(\mathbf{e}_i^r)^T + \mathbf{G}_i^r \Big) \Big\}.\end{aligned}$$

We can calculate $\mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}]$, $\mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{D}_i\mathbf{D}_i}]$, and $\mathbb{E}_{q(\mathbf{D}_i)}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}\mathbf{K}_{\mathbf{Z}_i^c\mathbf{D}_i}^T]$ from the expectation of a kernel response $k(\mathbf{z},\mathbf{x})$ with respect to the normal density $p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ for some static $\mathbf{z}$ and random $\mathbf{x}$. For Gaussian kernel functions $k(\mathbf{z},\mathbf{x}) = \exp\{-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T\mathbf{J}^{-1}(\mathbf{z}-\mathbf{x})\}$, such as RBF ($\mathbf{J} = \gamma\mathbf{I}$), this integral is analytically tractable, as discussed in Titsias et al. [25].

For classification, we add the Bernoulli-Probit likelihood, and marginalize out $\mathbf{y}_i^c$, as suggested by Hensman et al. [11]. After taking this integral, the lower bound becomes

$$\begin{aligned}\mathcal{L}_i^{clsf} = \mathcal{L}_i^{-Y} &+ \sum_{c=1}^{C}\sum_{n}^{N_i}\mathbf{t}_i^c[n]\log\sqrt{\frac{2\pi}{\beta}} + \sum_{c=1}^{C}\sum_{n}^{N_i}\mathbf{t}_i^c[n]\frac{\beta}{2}\Big((\mathbf{m}_i^c)^T\mathbf{K}_{\mathbf{D}_i\mathbf{D}_i}^{-1}\mathbb{E}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{d}_n^i}]\Big)^2 \\
&\sum_{c=1}^{C}\sum_{n}^{N_i}(2\mathbf{t}_i^c[n]-1)\log\Phi\left(\frac{(\mathbf{m}_i^c)^T\mathbf{K}_{\mathbf{K}_{\mathbf{D}_i\mathbf{D}_i}}^{-1}\mathbb{E}[\mathbf{K}_{\mathbf{Z}_i^c\mathbf{d}_n^i}]}{\sqrt{\beta^{-1}+1}}\right),\end{aligned}$$

where $\mathcal{L}_i^{-Y}$ is $\mathcal{L}_i$ with terms including $\mathbf{y}_i^c$ discarded.

The joint lower bound $\mathcal{L}$ could be maximized by a gradient-based approach using its derivatives with respect to the variational parameters:

$$\{\forall i,c,r : \mathbf{m}_i^c, \mathbf{S}_i^c, \mathbf{e}_i^r, \mathbf{G}_i^r, \mathbf{Z}_i^c, \mathbf{W}_i^c\} \cup \{\forall n : \mathbf{h}_n^t\} \cup \{g, h\}.$$

**Generalization to multiple source tasks.** **ATL-DGP** can easily be generalized to the case where multiple source tasks interact with the target task. For this, it suffices to replace the cross-domain projection density with

$$\begin{aligned}p(\mathbf{D}_t|\mathbf{B}_t,\mathbf{B}_{s\to t},\pi_1,\cdots,\pi_{k+1}) &= \prod_{n=1}^{N_t}\mathcal{N}\left(\mathbf{d}_n^t\,\middle|\,\sum_{k=1}^{K}\pi_k\mathbf{b}_n^{s_k\to t} + \pi_{K+1}\mathbf{b}_n^t, \alpha^{-1}\mathbf{I}\right), \\
p(\pi_1,\cdots,\pi_{k+1}|a_1,\cdots,a_{K+1}) &= Dir(\pi_1,\cdots,\pi_{k+1}|a_1,\cdots,a_{K+1}),\end{aligned}$$

where $Dir(\cdots|\cdots)$ is a Dirichlet density. The related lower bound term remains as a function consisting only of tractable expectations of the same shape as in the single source task case.

**The symmetric architecture alternative.** As a straightforward alternative to our approach, a symmetric transfer across two deep GPs can easily be made by projecting the instances of both tasks onto each other's manifold, and augmenting their native latent representations by these cross-task projections. In other words, we can add $p(\mathbf{B}_{t \to s}|\mathbf{X}_s, \mathbf{W}_t, \mathbf{v}_t)$ to the above generative process, and replace the densities of $\mathbf{D}_i$'s with $p(\mathbf{D}_t|\mathbf{B}_t, \mathbf{B}_{s \to t}) = \prod_{r=1}^{R^*} \mathcal{N}(\mathbf{d}_t^r|[\mathbf{B}_t, \mathbf{B}_{s \to t}]_r, \lambda^{-1}\mathbf{I})$, and $p(\mathbf{D}_s|\mathbf{B}_s, \mathbf{B}_{t \to s}) = \prod_{r=1}^{R^*} \mathcal{N}(\mathbf{d}_s^r|[\mathbf{B}_s, \mathbf{B}_{t \to s}]_r, \lambda^{-1}\mathbf{I})$, where $R^*$ is the sum of the number of task-specific and shared latent dimensions, and $[\mathbf{B}_t, \mathbf{B}_{s \to t}]_r$ and $[\mathbf{B}_s, \mathbf{B}_{t \to s}]_r$ are $r$th columns of the matrices in brackets. We call this architecture *Symmetric Transfer Learning with Deep Gaussian Processes*, and abbreviate it as **STL-DGP**.

**Prediction.** A nice property of the FITC approximation is that it converts the non-parametric standard GP into a parametric model (i.e. a model that summarizes a data set of arbitrary length by a fixed number of parameters). Hence, it is no longer necessary to store the training set (or the related Gram matrix) for prediction. Instead, it suffices to store the inducing data set and the inducing outputs. The predictive density for data point $(\mathbf{x}^*, y^*)$ and output dimension $c$ is

$$\begin{aligned} p(y_c^*|\mathbf{x}^*, \mathbf{X}_t, \mathbf{y}_t^c, \mathbf{Z}_t^c) = \int \Big( & p(y_c^*|f_c^*) p(f_c^*|\mathbf{D}^*, \mathbf{Z}_t^c, \mathbf{u}_t^c) q(\mathbf{u}_t^c) \\ & p(\mathbf{D}^*|\mathbf{B}^*) p(\mathbf{B}^*|\mathbf{x}^*, \mathbf{W}_t, \mathbf{V}_t) q(\mathbf{V}_t) \Big) df_c^* d\mathbf{u}_t^c d\mathbf{B}^* d\mathbf{D}^* d\mathbf{V}_t. \end{aligned}$$

For a fully Bayesian treatment, all these tractable Gaussian integrals have to be taken. A more practical alternative, which we also preferred in our implementation, is to use the point estimates of the latent variables. For classification, the positive class probability can be approximated by $p(t_c^* = +1|y_c^*) \approx \Phi\Big(\mathbb{E}[p(y_c^*|\mathbf{x}^*, \mathbf{X}_t, \mathbf{y}_t^c, \mathbf{Z}_t^c)]\Big)$.

# 6 Experiments

Table 1: Ten-class object categorization results on the benchmark computer vision database consisting of four data sets, each corresponding to one domain. Our model **ATL-DGP** provides better average classification accuracy than the models in comparison. The results for **GFK**, **MMDT**, and **KBTL** are taken from Gönen et al. [7], Table 1. The highest accuracy of each domain is given in boldface.

| Source→Target | NGP-S | NGP-T | STL-DGP | GFK | MMDT | KBTL | ATL-DGP |
|---|---|---|---|---|---|---|---|
| caltech→amazon | 40.3 ± 2.6 | 52.1 ± 4.7 | 50.4 ± 5.2 | 44.7 ± 0.8 | 49.4 ± 0.8 | 52.9 ± 1.0 | **53.9 ± 3.8** |
| dslr→amazon | 35.5 ± 2.1 | 50.9 ± 4.2 | 48.5 ± 3.5 | 45.7 ± 0.6 | 46.9 ± 1.0 | **51.9 ± 0.9** | 50.8 ± 3.1 |
| webcam→amazon | 37.7 ± 2.8 | 52.5 ± 3.5 | 50.5 ± 3.2 | 44.1 ± 0.4 | 47.7 ± 0.9 | **53.4 ± 0.8** | 51.8 ± 2.9 |
| amazon→caltech | **40.1 ± 1.6** | 35.5 ± 3.8 | 33.9 ± 3.0 | 36.0 ± 0.5 | 36.4 ± 0.8 | 35.9 ± 0.7 | **40.1 ± 2.9** |
| dslr→caltech | 34.1 ± 1.7 | 35.9 ± 3.3 | 33.7 ± 2.7 | 32.9 ± 0.5 | 34.1 ± 0.8 | 35.9 ± 0.6 | **38.8 ± 2.8** |
| webcam→caltech | 33.1 ± 2.5 | 33.1 ± 4.8 | 30.6 ± 3.5 | 31.1 ± 0.6 | 32.2 ± 0.8 | 34.0 ± 0.9 | **37.9 ± 2.9** |
| amazon → dslr | 37.6 ± 3.8 | 57.0 ± 5.8 | 54.2 ± 4.9 | 50.7 ± 0.8 | 56.7 ± 1.3 | 57.6 ± 1.1 | **58.4 ± 5.1** |
| caltech→dslr | 38.5 ± 3.9 | 56.9 ± 6.6 | 57.4 ± 4.6 | 57.7 ± 1.1 | 56.5 ± 0.9 | **58.8 ± 1.1** | 58.5 ± 4.0 |
| webcam→dslr | 62.2 ± 4.7 | 57.6 ± 3.1 | 54.9 ± 5.1 | **70.5 ± 0.7** | 67.0 ± 1.1 | 61.8 ± 1.3 | 66.7 ± 3.8 |
| amazon →webcam | 37.3 ± 4.6 | 67.0 ± 6.0 | 67.5 ± 5.6 | 58.6 ± 1.0 | 64.6 ± 1.2 | **69.8** ± 1.1 | 68.9 ± 5.4 |
| caltech→webcam | 35.2 ± 7.0 | 64.9 ± 8.9 | 64.5 ± 5.6 | 63.7 ± 0.8 | 63.8 ± 1.1 | **68.5** ± 1.2 | 65.9 ± 5.0 |
| dslr →webcam | 70.8 ± 3.4 | 66.4 ± 5.2 | 65.3 ± 4.4 | **76.5 ± 0.5** | 74.1 ± 0.8 | 70.0 ± 1.0 | 74.5 ± 2.6 |
| Average Accuracy | 41.8 ± 3.4 | 52.4 ± 5.3 | 51.0 ± 4.3 | 51.0 ± 0.7 | 52.5 ± 1.0 | 54.2 ± 0.9 | **55.5 ± 3.7** |

We evaluate **ATL-DGP** on one benchmark object categorization application, and one novel cross-tissue tumor detection application. For all sparse GP components, we use 10 inducing points that are initialized to cluster centroids found by k-means, as in Hensman et al. [11]. We set the inducing points of the first layer GPs to instances chosen from the training set at random, and learn them from data for the second layer GPs. This is meant for avoiding overparameterization of the model. We initialize $\mathbf{e}_i^r$ and $\mathbf{m}_i^c$ to their least-squares fit to the predictive mean of the GP they belong:

$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\operatorname{argmin}} \; ||\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u} - \mathbf{y}||_2^2$. We observed that this computationally cheap initialization procedure provided significantly better performance than random initialization for all sparse GP models.

We compare **ATL-DGP** to: i) **NGP-S**: A Deep GP trained only on the source data set, hence performs no domain transfer (i.e., **ATL-DGP** with $\pi = 0$), ii) **NGP-T**: A Deep GP trained only on the target data set, iii) **STL-DGP**: Two Deep GPs that perform symmetric transfer as described in Section 5.2 (10 task-specific and 20 shared manifold dimensions are used), iv) **GFK**: *Geodesic Flow Kernel* [8], v) **MMDT**: *Max-Margin Domain Transforms* [12], vi) **KBTL**: *Kernelized Bayesian Transfer Learning* [7].

We choose **KBTL**, **MMDT**, and **GFK** as the three highest performing models on the benchmark computer vision data base with respect to Table 1 from Gönen et al. [7], and **STL-DGP** as another deep GP based design alternative. **NGP-S** and **NGP-T** are proof-of-concept baselines used to show the occurrence and benefit of cross-domain knowledge transfer.

For all sparse GP models, we start the learning rate from $0.001$, take a gradient step if it increases the lower bound, or multiply the learning rate by $0.9$ otherwise. For all models that learn latent data representations, such as all deep GP variants and **KBTL**, we set the latent dimensionality size to 20. For all kernel learners, we used an RBF kernel with isotropic covariance. For GP variants, we also learned the length scale by taking its gradient with respect to the lower bound. For others, we set it to the mean euclidean distance of all instance pairs in the training set, as suggested by Gönen et al. [7]. All other design decisions of competing models are made following the principles suggested in the original papers.

Table 2: Tumor detection accuracies of the models in comparison. Our model **ATL-DGP** transfers useful knowledge across both tissue types and reaches higher performance than the baselines.

| | **Breast → Esophagus** | **Esophagus → Breast** | **Average Accuracy** |
|---|---|---|---|
| **NGP-S** | $63.4 \pm 4.6$ | $59.3 \pm 2.1$ | $61.3 \pm 3.3$ |
| **NGP-T** | $64.7 \pm 5.5$ | $59.4 \pm 4.6$ | $62.0 \pm 5.1$ |
| **STL-DGP** | $64.3 \pm 5.2$ | $55.4 \pm 1.9$ | $58.9 \pm 3.8$ |
| **FMTL** | $58.2 \pm 2.3$ | $59.6 \pm 5.2$ | $56.8 \pm 2.1$ |
| **GFK** | $56.1 \pm 1.9$ | $56.6 \pm 1.3$ | $56.4 \pm 1.6$ |
| **MMDT** | $65.0 \pm 6.4$ | $59.1 \pm 5.4$ | $62.1 \pm 5.9$ |
| **KBTL** | $57.8 \pm 6.4$ | $54.7 \pm 4.9$ | $56.3 \pm 5.7$ |
| **ATL-DGP** | $\mathbf{67.2 \pm 3.9}$ | $\mathbf{61.1 \pm 3.3}$ | $\mathbf{64.2 \pm 3.6}$ |

## 6.1 Benchmark application: Real-world image categorization

We use the benchmark data set constructed by Saenko et al. [21] for domain adaptation experiments, which consists of images of 10 categories (backpack, bicycle, calculator, headphones, keyboard, laptop, monitor, mouse, mug, and projector) taken in four different conditions, corresponding to the following four domains:

- **amazon:** images from `http://www.amazon.com`, which are taken by merchants to sell their products in the online market,

- **caltech:** images chosen from the experimental Caltech 256 data set [9], which is constructed from images collected by web search engines,

- **dslr:** images taken with a high resolution ($4288 \times 2848$) digital SLR camera,
- **webcam:** low resolution ($640 \times 480$) images taken with a webcam.

We use the 800-dimensional SURF-BoW features provided by Gong et al. [8], and the 20 train/test splits provided by Hoffman et al. [12]. Each train split consist of 20 images from the source domain for amazon and eight images for the other three domains, and three images from the target domain. All the remaining points in the target domain are left for the test split.

Prediction accuracies of the models trained and tested on each 12 domain pairs and averaged over 20 splits are compared in Figure 1. **ATL-DGP** provides the highest average accuracy across the domains. The other modeling alternative **STL-DGP** suffers from negative transfer, and performs worse even than the no transfer case **NGP-T**.

## 6.2 Novel application: cross-tissue tumor detection

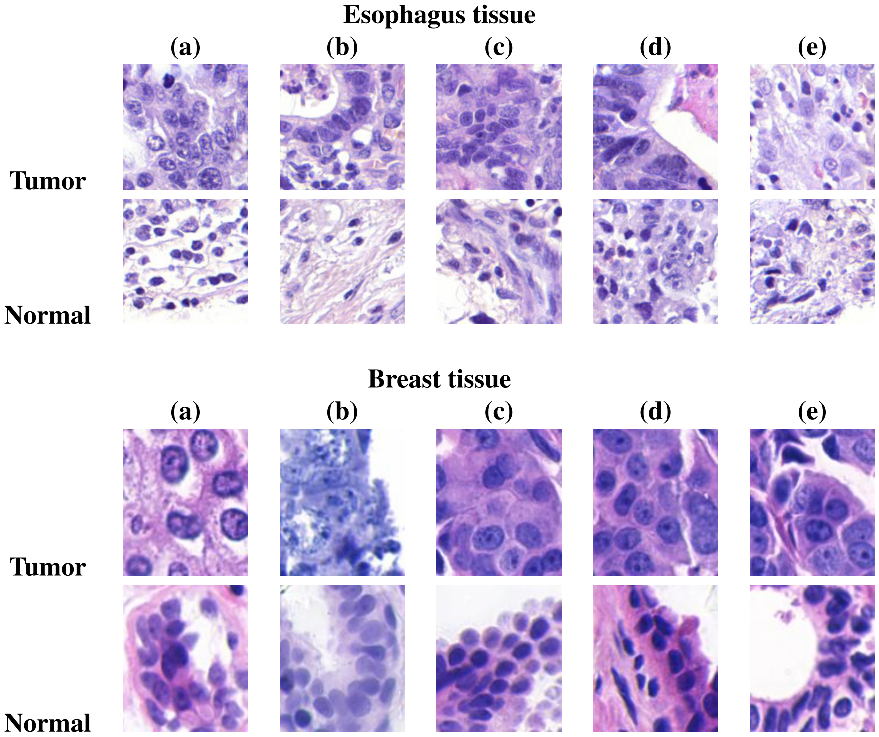

Figure 3: Sample histopathology images taken from two tissue types: Breast and Esophagus. The two image sets are acquired from different tissues, using a different microscope and different magnification levels, resulting in different data distributions.

We perform a feasibility study for domain transfer across histopathology cancer diagnosis data sets, taken from two different tissues: i) breast and ii) esophagus. We study classification of patches taken from histopathology microscopic tissue images stained by hematoxylin & eosin (H & E) into two classes: i) tumor and ii) normal. Visual indicators of cancer are known to differ drastically from one tissue to another. For instance, in breast cancer, the glandular formations of cells get distorted as cancer develops. Contrarily, in Barrett's cancer, which takes place in esophagus, cells may mimic another tissue by forming glands. Other sources of difference in input data distributions are staining conventions, scanner types, and magnification levels that vary from one laboratory to another. Cross-tissue knowledge transfer would be useful in cases when the target tissue is taken from a rare cancer case, such as Barrett's cancer [16]. For such cases, available data from more widespread cancer types, such as breast cancer, could facilitate tumor detection.

Figure 3 shows sample tumor and normal histopathology images from breast and esophagus tissues, where the differences in the image characteristics of the tissues can clearly be seen. While the glandular structures are in the tumor class for esophagus (samples (b) and (d)), they are in the normal class for breast (samples (b) and (e)). Cells in the breast images are much larger than those in esophagus due to higher magnification, and the texture is inclined more towards pink due to higher dose of eosin.

The *esophagus data set* consists of 210 Tissue Micro Array (TMA) images taken from the biopsy samples of 77 Barrett's cancer patients. We split each TMA image into a regular grid of $200 \times 200$ pixel patches, and treat each patch as an instance. Consequently, we have a data set of 14353 instances, 6733 of which are tumors, and 7620 normal cases. The *breast data set*[2] consists of 58 images of $896 \times 768$ pixels taken from 32 benign and 26 malignant breast cancer patients. Splitting each image into an equal-sized $7 \times 7$ grid, we get 2002 instances, 944 of which are tumors, and 1058 normal cases.

We represent all image instances by a 657-dimensional feature vector consisting of their intensity histogram of 26 bins, 7 box counting features for grid sizes from 2 to 8, mean of $20 \times 20$-pixel local binary pattern (LBP) histograms of 58 bins, and mean of 128-dimensional dense SIFT descriptors of 40 pixels step size. We reduce the data dimensionality to 50 using principal component analysis (PCA) to eliminate uninformative features.

For this application, we compare **ATL-DGP** also to Focused Multitask Gaussian Process [18] (**FMTL**), which is an alternative approach for GP based asymmetric transfer learning. We omitted **FMTL** from comparison in the previous 10-class image categorization application, since its available implementation is tailored only for a single target task with binary labels.

We generate a training split by randomly choosing 200 instances from the source domain, and five instances from the target domain per class. We treat the remaining instances of the target domain as the test split. We repeat this procedure 20 times and report the average prediction accuracies in Table 2. In this application, **NGP-S** is more competitive than in the previous benchmark due to the larger difference between the input data distributions of the domains than those of real-world tasks, which makes knowledge transfer more difficult. Yet, **ATL-DGP** consistently improves on both **NGP-S** and **NGP-T**, implying that it successfully performs positive knowledge transfer. On this data set, **ATL-DGP** improves statistically significantly on all baselines for both source-target combinations (paired t-test, $p < 0.05$) with two exceptions: **STL-DGP** and **MMDT** for Breast as source and Esophagus as target.

# 7 Discussion

Experiments above showed that our asymmetric transfer strategy **ATL-DGP** is more effective than the symmetric alternative **STL-DGP**. A possible reason for this could be

[2] http://www.bioimage.ucsb.edu/research/biosegmentation

the fact that the small sample size in the target domain makes its manifold too noisy for the source domain. Hence, such a noisy transfer could harm the overall learning procedure. Another reason could be that transferring knowledge from target to source task does not always leverage transfer in the opposite direction. In such cases, this transfer only consumes part of the expressive power of the model for a task, which is not directly useful for the intended purpose, and causes suboptimal performance.

The outcome of our feasibility study on cross-tissue type tumor detection from histopathology tissue images encourages further investigation of this application on a wider variety of tissue types. Cross-tissue knowledge transfer could be especially useful for the automatic tissue scanners to handle rare cancer types based on their knowledge on widespread cancer types. Obviously, applicability of our proposed model is beyond the limits of computer vision and medical image analysis.

Our model outperforms **FMTL**, as it transfers knowledge across input representations (early fusion), where the information gain is expected to happen in a domain adaptation setup, as opposed to **FMTL**'s late fusion strategy, which is more suitable for multitask learning. **ATL-DGP** outperforms **KBTL** and **MMDT** in cross-tissue tumor detection thanks to the non-linearity of the mapping it applies from the latent representations to the outputs.

As future work, the variational scheme of our model can be improved for handling large data masses using stochastic variational Bayes [13], which allows the model to process the data points one (or few) at a time.

# References

[1] M.J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.

[2] E. Bonilla, K.M. Chai, and C. Williams. Multi-task Gaussian process prediction. In *NIPS*, 2008.

[3] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *NIPS*, 2008.

[4] A.C. Damianou and N.D. Lawrence. Deep Gaussian processes. In *AISTATS*, 2013.

[5] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[6] L. Duan, D. Xu, and I. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *ICML*, 2012.

[7] M. Gönen and A.A. Margolin. Kernelized Bayesian transfer learning. In *AAAI*, 2014.

[8] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[9] G. Griffin, A. Holub, and A.D. Perona. Caltech-256 object category data set. Technical report, 7654, California Institute of Technology, 2007.

[10] M.N. Gürcan, L.E. Boucheron, A. Can, A. Madabhushi, Nasir M. Rajpoot, and B. Yener. Histopathological image analysis: A review. *Biomedical Engineering, IEEE Reviews in*, 2:147–171, 2009.

[11] J. Hensman, N. Fusi, and N.D. Lawrence. Gaussian processes for big data. In *UAI*, 2013.

[12] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. In *ICLR*, 2013.

[13] M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[14] N. Houlsby, F. Huszar, Z. Ghahramani, and J.M. Hernández-Lobato. Collaborative Gaussian processes for preference learning. In *NIPS*, 2012.

[15] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[16] r. Langer, S. Rauser, M. Feith, J.M. Nährig, A. Feuchtinger, H. Friess, H. Höfler, and A. Walch. Assessment of ErbB2 (Her2) in oesophageal adenocarcinomas: summary of a revised immunohistochemical evaluation system, bright field double in situ hybridisation and fluorescence in situ hybridisation. *Modern Pathology*, 24(7):908–916, 2011.

[17] M. Lázaro-Gredilla and M.K. Titsias. Spike and slab variational inference for multi-task and multiple kernel learning. In *NIPS*, 2011.

[18] G. Leen, J. Peltonen, and S. Kaski. Focused multi-task learning using Gaussian processes. In *Machine Learning and Knowledge Discovery in Databases*, pages 310–325. 2011.

[19] V.T. Nguyen and E. Bonilla. Collaborative multi-output Gaussian processes. In *UAI*, 2014.

[20] C.E. Rasmussen and C.I. Williams. Gaussian processes for machine learning. 2006.

[21] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*. 2010.

[22] M. Seeger. Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations. *PhD Thesis*, 2003.

[23] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS*, 2006.

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

[25] M.K. Titsias and N.D. Lawrence. Bayesian Gaussian process latent variable model. In *AISTATS*, 2010.

[26] V. Vapnik. *Statistical learning theory*. Wiley New York, 1998.