Proxy-informed Bayesian transfer learning with unknown sources

Sabina J. Sloman[1] Julien Martinelli*[2] Samuel Kaski[1,2,3]

[1]Department of Computer Science, University of Manchester, Manchester, UK
[2]Department of Computer Science, Aalto University, Espoo, Finland
[3]ELLIS Institute Finland, Helsinki, Finland

Abstract

Generalization outside the scope of one's training data requires leveraging prior knowledge about the effects that transfer, and the effects that don't, between different data sources. Transfer learning is a framework for specifying and refining this knowledge about sets of source (training) and target (prediction) data. A challenging open problem is addressing the empirical phenomenon of negative transfer, whereby the transfer learner performs worse on the target data after taking the source data into account than before. We first introduce a Bayesian perspective on negative transfer, and then a method to address it. The key insight from our formulation is that negative transfer can stem from misspecified prior information about non-transferable causes of the *source* data. Our proposed method, proxy-informed robust method for probabilistic transfer learning (PROMPT), does not require prior knowledge of the source data (the data sources may be "unknown"). PROMPT is thus applicable when differences between tasks are unobserved, such as in the presence of latent confounders. Moreover, the learner need not have access to observations in the target task (may not have the ability to "fine-tune"), and instead makes use of proxy (indirect) information. Our theoretical results show that the threat of negative transfer does not depend on the informativeness of the proxy information, highlighting the usefulness of PROMPT in cases where only noisy indirect information, such as human feedback, is available.

The paradigm of transfer learning takes, often sparse, data from a set of *source* tasks and uses them to predict outcomes in a different but related *target* task. Consider the task of predicting the effectiveness of a treatment for a new patient on the basis of observational data. Inevitably, the measured effects of the treatment in the source data are affected by a myriad of unobserved confounders, such as the quality of treatment in a given clinical setting or the patient's adherence to a treatment regimen. Prediction in this setting requires learning both *shared* parameters (the treatment effect) and *task* parameters (the quality of treatment in *this* patient's local clinic; *this* patient's adherence). Failure to account for the presence of task-specific effects can result in imprecise or inaccurate predictions in the target task. Bayesian learning is a natural paradigm to apply in settings where data is sparse and prior information is available. When this prior information is reliable, the Bayesian transfer learner can leverage the source data to make accurate and calibrated predictions when encountering new target tasks.

In practice, however, the Bayesian transfer learner often experiences *negative transfer*, performing worse in the target task after taking the source data into account than before. Understanding the conditions under which negative transfer occurs, and how to address it, is a challenging open problem (Suder et al., 2023).

Our first contribution is to provide a precise and Bayesian characterization of the phenomenon of negative transfer. Our formulation treats the Bayesian transfer learner's objective as a special case of inference in the presence of nuisance parameters, and applies results from this more general class of problems to elucidate the conditions under which negative transfer occurs. We show that negative transfer can arise when prior information about source task parameters is unavailable or mistaken. This result implies that alleviating the threat of negative transfer requires removing the learner's reliance on, possibly mistaken, prior information about source task parameters.

Our second contribution is to propose a method, proxy-informed robust method for probabilistic transfer learning (PROMPT), that allows the learner to form a posterior pre-

*Work done while at Inserm Bordeaux Population Health, Vaccine Research Institute, Université de Bordeaux, Inria Bordeaux Sud-ouest, France.

dictive distribution in the target task without such prior information. Our third contribution is to use our formulation of the Bayesian transfer learner's objective to provide theoretical guarantees on PROMPT's ability to alleviate the threat of negative transfer.

PROMPT operates in a setting that differs from, and is in some ways more general than, settings in existing literature on probabilistic approaches to transfer and meta-learning (Grant et al., 2018; Yoon et al., 2018; Gordon et al., 2019; Patacchiola et al., 2020). We here discuss some differences between our setting and probabilistic meta-learning, Bayesian meta-learning, and proxy methods for multi-source domain adaptation. We discuss these and other related works in more detail in Appendix A.

Difference #1: PROMPT can cope with unknown sources. Probabilistic meta-learning (Gordon et al., 2019), Bayesian meta-learning (Grant et al., 2018; Yoon et al., 2018; Patacchiola et al., 2020), and multi-domain adaptation (Tsai et al., 2024) assume the availability of some prior knowledge about the source data, such as the number of distinct tasks represented in the source data and which data points correspond to the same task. This assumption would be violated in our motivating example: Each patient's outcomes are influenced by confounders whose values are unknown, and the learner cannot know which outcomes are influenced by the same latent confounder values. Probabilistic meta-learning (Gordon et al., 2019) also requires that the target task arises from the same distribution as the source tasks. PROMPT requires neither the availability of prior information about the source tasks nor that the target task resembles the source tasks.

Difference #2: PROMPT relies on proxy information instead of fine-tuning. Bayesian meta-learning requires that the learner has access to data from the target task in order to fine-tune their estimates (Grant et al., 2018; Yoon et al., 2018; Patacchiola et al., 2020). PROMPT relies on *proxy* (indirect) information about the target task. Examples of proxy information are human feedback (e.g., to prompts such as "What is the quality of treatment in the target hospital?") and instrumental variables (e.g., hospital funding as an instrument for quality of care). This leads to connections between our work and the paradigm of proximal causal learning (Kuroki and Pearl, 2014; Tchetgen et al., 2024; Alabdulmohsin et al., 2023; Tsai et al., 2024); see Appendix A for a more detailed discussion. Unlike other proxy methods for multi-domain adaptation (Tsai et al., 2024), we distinguish between shared and task parameters and require additional techniques for estimation of the shared parameter. PROMPT does assume that the proxy information does not depend on the shared parameters.

Our theoretical results show that, surprisingly, PROMPT's success in eliminating negative transfer does not depend on the quality of the proxy information, making PROMPT particularly useful when proxy information is weak or unreliable. The extent of negative transfer depends instead on the quality of a pre-specified *relevance function*. We describe approaches to defining the relevance function in a purely source data-dependent way, and demonstrate application of these approaches in two synthetic examples and on a dataset of smoking behavior.

1 PRELIMINARIES

Notation. Vectors and matrices are denoted by bold lowercase letters: $\mathbf{a}_{i,j}$ is the entry in the i^{th} row and j^{th} column of $\mathbf{a}$. Sets are denoted by calligraphic font ($\mathscr{A}$), and $\mathscr{A}_i$ is the i^{th} element of $\mathscr{A}$. We use $\mathbf{a}_{\mathscr{I}}$, where $\mathscr{I}$ is a set, to denote the subvector formed by selecting the elements of $\mathbf{a}$ at the indices in $\mathscr{I}$. Random variables are denoted by bold capital letters ($\mathbf{A}$), and the notation for probability distributions is subscripted by the corresponding random variable ($P_{\mathbf{A}}$). For instance, $\mathbf{A}$ is the random variable with domain $\mathscr{A}$ and probability distribution $P_{\mathbf{A}}$.

Bayesian transfer learning is a general framework for leveraging data from source tasks to make predictions in a somewhat unrelated target task (Suder et al., 2023). We consider a standard setting where tasks are characterized by both shared and task parameters. The learner has available to them source data $\mathbf{d} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ composed of stochastic observations $\mathbf{d}_i \in \mathscr{D}$.[1] We write the random variable characterizing the source data as $\mathbf{D}$.

Each observation is generated in the context of a particular, possibly non-unique, *task*. In the setting of *unknown sources*, the learner need not be aware of which observations are generated in the context of the same, or similar, tasks. Below, we formalize this as a potential difference between the dependency structures characterizing the source data-generating process, on one hand, and the learner's model of the source data, on the other.

The probability of each observation $\mathbf{d}_i$ depends both on **shared parameters** $\boldsymbol{\theta} \in \mathscr{T}$, which are the same for each task, and **task parameters** $\boldsymbol{\psi}_i \in \mathscr{S}$, which differ between tasks. Given a $(\boldsymbol{\theta}, \boldsymbol{\psi}_i)$, the learner can evaluate $p(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i)$. As is typical in such formulations, we assume a single, data-generating value of each of the shared and source task parameters, which we denote $\boldsymbol{\theta}^\star$ and $\boldsymbol{\psi}^\star$, respectively.[2]

Definition 1.1 (Task)**.** *The i^{th} task specifies the distribution generating the i^{th} data point. It depends on the value of a*

[1] The source data matrix can equivalently be written $\mathbf{d}_{(1:n)}$ to make explicit that it is composed of all n past observations. When referring to the source data matrix, we omit the subscript $(1:n)$, i.e., write $\mathbf{d} \equiv \mathbf{d}_{(1:n)}$.

[2] As with the source data $\mathbf{d}$, the source task parameters can equivalently be written $\boldsymbol{\psi}_{(1:n)}$. When referring to the source task parameters, we write $\boldsymbol{\psi} \equiv \boldsymbol{\psi}_{(1:n)}$.

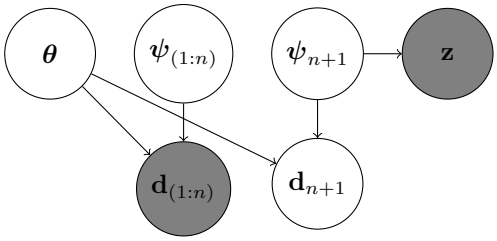


Figure 1: Assumed dependencies between shared parameter $\boldsymbol{\theta}$, task parameters $\boldsymbol{\psi}$, source data $\mathbf{d} \equiv \mathbf{d}_{(1:n)}$, target data $\mathbf{d}_{n+1}$, and proxy information $\mathbf{z}$.

shared parameter $\boldsymbol{\theta}^\star$ *and* ***task parameter*** $\boldsymbol{\psi}_i^\star$*. The value* $\boldsymbol{\theta}^\star$ *is assumed to be shared across all tasks, and so the task is equivalently identified by the value of* $\boldsymbol{\psi}_i^\star$*.*

At deployment, the learner encounters an $(n+1)^{\text{th}}$ task which will induce an observation $\mathbf{d}_{n+1}$. Their goal is to predict $\mathbf{d}_{n+1}$ on the basis of $\mathbf{d}$, which requires identification of the **target data-generating process**, i.e., of the shared parameter $\boldsymbol{\theta}^\star$ and task parameter $\boldsymbol{\psi}_{n+1}^\star$.

The setting is visualized in Figure 1. Throughout, we implicitly depend on the following assumption:

Assumption 1.2. *All dependencies in Figure 1 are present in the data-generating process. The following dependencies are not present in the data-generating process:*

(a) If $i \neq j$*,* $\mathbf{d}_i$ *does not depend on* $\boldsymbol{\psi}_j$ *except possibly through* $\boldsymbol{\psi}_i$*.*

(b) If $i \neq j$*,* $\mathbf{d}_i$ *does not depend on* $\mathbf{d}_j$ *except through* $\boldsymbol{\theta}$ *and possibly through* $\boldsymbol{\psi}_i$*.*

(c) $\boldsymbol{\psi}_{n+1}$ *does not depend on* $\boldsymbol{\theta}$*.*

(d) Proxy information $\mathbf{z}$ *does not depend on* $\boldsymbol{\theta}$ *(see Section 3.1).*

In the setting of **unknown sources**, the learner does not have knowledge of the potential presence of the dependencies in grey text. We thus derive all quantities available to the learner (e.g., the likelihood of the source data given below) to reflect this absence of knowledge, i.e., omitting these potential dependencies. This results in a potential difference between the dependency structures characterizing the data-generating process and the learner's model of the data.

The Bayesian transfer learner assigns to values $(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})$ a prior distribution, and so treats the parameters as random variables $\boldsymbol{\Theta}$ and $\boldsymbol{\Psi}_{n+1}$ with distribution $P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}$. For a possible value of the target data-generating process $(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})$, the likelihood L of the source data $\mathbf{d}$ is

$$L(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}) \equiv p(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}) = L(\mathbf{d}, \boldsymbol{\theta}) \tag{1}$$

because the target task parameter is not known by the learner to affect any of the source data (Assumption 1.2(a)).

The probability of $(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})$ under the posterior $P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}|\mathbf{d}}$ is

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}|\mathbf{d}) &= \frac{L(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})\, p(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})}{\mathbb{E}_{\boldsymbol{\theta}', \boldsymbol{\psi}'_{n+1} \sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}} \left[L(\mathbf{d}, \boldsymbol{\theta}', \boldsymbol{\psi}'_{n+1}) \right]} \\ &= \left(\frac{L(\mathbf{d}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\mathbb{E}_{\boldsymbol{\theta}' \sim P_{\boldsymbol{\Theta}}} \left[L(\mathbf{d}, \boldsymbol{\theta}') \right]} \right) p(\boldsymbol{\psi}_{n+1}) \end{aligned} \tag{2}$$

where the second line follows from Assumption 1.2(a,c).

As we show in Section 2, computing the likelihood in a classic way (described below) can lead to negative transfer. After describing the classic Bayesian learner's approach, we introduce a generic method to "robustify" the likelihood. PROMPT leverages this robustified method in its estimation of the predictive posterior.

Classic Bayesian inference additionally requires a prior over the source task parameters $P_{\boldsymbol{\Psi}}$. The posterior then marginalizes across this prior as follows:

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}|\mathbf{d}) &= \left(\frac{L(\mathbf{d}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\mathbb{E}_{\boldsymbol{\theta}' \sim P_{\boldsymbol{\Theta}}} \left[L(\mathbf{d}, \boldsymbol{\theta}') \right]} \right) p(\boldsymbol{\psi}_{n+1}) \\ &= \left(\frac{\mathbb{E}_{\boldsymbol{\psi} \sim P_{\boldsymbol{\Psi}}} \left[L(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi}) \right]\, p(\boldsymbol{\theta})}{\mathbb{E}_{\boldsymbol{\theta}', \boldsymbol{\psi} \sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}}} \left[L(\mathbf{d}, \boldsymbol{\theta}', \boldsymbol{\psi}) \right]} \right) p(\boldsymbol{\psi}_{n+1}) \\ &= p(\boldsymbol{\theta}|\mathbf{d})\, p(\boldsymbol{\psi}_{n+1}) \end{aligned} \tag{3}$$

The prior $P_{\boldsymbol{\Psi}}$ encodes the learner's knowledge about the joint distribution of source task parameters. In the setting of unknown sources, the learner does not have knowledge of the dependencies between source task parameters (such as which observations arise from the same task, i.e., for which (i, j) it is the case that $\boldsymbol{\psi}_i^\star = \boldsymbol{\psi}_j^\star$) and/or their prior may misrepresent the probability of encountering data generated under a given source task parameter value.

Likelihood weighting is a technique whereby the learner specifies a vector of weights $\boldsymbol{\eta}$ that determines the contribution of each observation to the overall weighted likelihood (Grünwald, 2011). When some $\boldsymbol{\eta}_i > \boldsymbol{\eta}_j$, it can be seen as increasing the influence of the i^{th} data point relative to the j^{th} data point.

2 A BAYESIAN PERSPECTIVE ON NEGATIVE TRANSFER

Negative transfer refers to the phenomenon that learning from source data can hurt performance in the target task (Wang et al., 2019). Here, we give a formal statement of the Bayesian transfer learner's objective which will allow us to make a precise and interpretable statement about when negative transfer will occur.

The Bayesian transfer learner's goal is to identify the target data-generating process. Since the effects of the task

parameters will not transfer, the source data can help the Bayesian learner identify the target data-generating process only insofar as it identifies the shared parameter. This objective has a natural information-theoretic interpretation, given in Definition 2.1: The **information gain**, or degree to which a Bayesian learner has "gained information" about $\boldsymbol{\theta}$, is the expected log ratio of the posterior to prior odds of $\boldsymbol{\theta}$. Information gain measures are applied in contexts like experimental design (Rainforth et al., 2024) and model selection (Oladyshkin and Nowak, 2019).

Because we are interested in the learner's information gains under the true data-generating process, we define the information gain as an expectation across the true distribution of source data. To reduce notational clutter, we use $\mathbf{D}^\star$ to refer to the random variable $\mathbf{D}|\boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star$, which follows the distribution of source data under the true data-generating parameters $(\boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star)$ (which are unavailable to the learner).

Definition 2.1 (Information gained by the classic Bayesian learner IG^c)**.** *The information gained by the classic Bayesian transfer learner about the shared parameter $\boldsymbol{\theta}^\star$ is*

$$\mathrm{IG}^c\left(\boldsymbol{\theta}^\star\right) \equiv \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(\frac{p(\boldsymbol{\theta}^\star|\mathbf{d})}{p(\boldsymbol{\theta}^\star)}\right)\right].$$

If $\mathrm{IG}^c\left(\boldsymbol{\theta}^\star\right) > 0$, the learner has successfully transferred information about $\boldsymbol{\theta}^\star$ from the source to target data ($p(\boldsymbol{\theta}^\star|\mathbf{d}) > p(\boldsymbol{\theta}^\star)$, i.e., they prefer $\boldsymbol{\theta}^\star$ after viewing data). Otherwise, they are worse off than before ($p(\boldsymbol{\theta}^\star|\mathbf{d}) \leq p(\boldsymbol{\theta}^\star)$, i.e., they preferred $\boldsymbol{\theta}^\star$ before viewing data). We define positive and negative transfer as:

Definition 2.2 (Positive and negative transfer)**.** *The classic Bayesian learner experiences* positive transfer *when* $\mathrm{IG}^c\left(\boldsymbol{\theta}^\star\right) > 0$*; otherwise, they experience* negative transfer.

We now provide a result showing that the threat of negative transfer is affected by the reliability of the prior over source task parameters $P_{\boldsymbol{\Psi}}$. The key quantity is a measure of likelihood misspecification:

Definition 2.3 (Misspecification of the classic likelihood Δ^c)**.** *The degree to which the classic likelihood is misspecified is $\Delta^c \equiv \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}^\star}\right)$ where D_{KL} is the Kullback-Leibler divergence measure.*

In the presence of negative transfer, likelihood misspecification increases with the misspecification of the prior over source task parameters (Sloman et al. 2024 Theorem 4.11). To see this, recall from Equation (3) that the density for $P_{\mathbf{D}|\boldsymbol{\theta}^\star}$ marginalizes across $P_{\boldsymbol{\Psi}}$.

Theorem 2.4 shows that Δ^c is responsible for negative transfer. The proof, adapted from Sloman et al. (2024), is given in Appendix B.2. It relies on the following assumption:

Assumption B.3 (informal)**.** *The likelihood $L\left(\mathbf{d}, \boldsymbol{\theta}\right)$ is "smooth enough" in a neighborhood of $\boldsymbol{\theta}^\star$. The formal condition is given in Appendix B.2.*

Theorem 2.4 (Negative transfer with a classic likelihood (modified from Sloman et al. 2024 Theorem 4.5))**.** *Under Assumption B.3 in Appendix B.2,*

$$\mathrm{IG}^c\left(\boldsymbol{\theta}^\star\right) \leq A\left(B - \Delta^c\right) \tag{4}$$

where A and B are constants that do not depend on Δ^c.

Because of its effect on Δ^c, the prior over source task parameters affects the risk of negative transfer. To remove the Bayesian transfer learner's dependence on this prior information, we introduce proxy-informed robust method for probabilistic transfer learning (PROMPT).

3 PROMPT

The Bayesian transfer learner faces two challenges: To make accurate predictions in the target task, they must (1) gain information about the target task parameter $\boldsymbol{\psi}^\star_{n+1}$, which to their knowledge does not depend on the source data, and (2) avoid negative transfer, which as discussed in Section 2 arises from a misspecified source task parameter prior $P_{\boldsymbol{\Psi}}$. Our proposed proxy-informed robust method for probabilistic transfer learning (PROMPT) has three steps: First, to address challenge (1), *proxy information* is used to form a posterior over the target task parameter $\boldsymbol{\psi}_{n+1}$. Then, to address challenge (2), a *relevance function* is used to construct a weighted likelihood for $\boldsymbol{\theta}$ that does not depend on any prior source task information. Finally, the learner combines their posterior over $\boldsymbol{\psi}_{n+1}$ and weighted likelihood for $\boldsymbol{\theta}$ to form a robust posterior over the target data-generating process. The entire procedure is summarized in Algorithm 1.

The computational overhead required by PROMPT is comparable to that required by existing implementations of Bayesian inference. As shown in Algorithm 1 Line 2–Line 9 and discussed in Section 3.2, the reweighting step (step 2) is performed in at most T iterations, where T is the user-supplied number of iterations for refinement of the relevance function. Once this step is performed, computation of the posterior is similar to other Bayesian inference methods.[3] PROMPT thus increases computational overhead by at most the constant factor T.

3.1 STEP 1: LEARNING TASK PARAMETERS VIA PROXIES

In step 1, the learner addresses the challenge of learning the target task parameter $\boldsymbol{\psi}^\star_{n+1}$. We refer to information

[3] As can be seen in the code we provide for the example predicting smoking behavior discussed in Section 5.1, applying the relevance function requires minimal modifications to existing posterior update methods.

Algorithm 1 Proxy-informed RObust Method for Probabilistic Transfer learning (PROMPT)

Input: Source data $\mathbf{d}$, proxy information $\mathbf{z}$, prior $P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}$, relevance function $\mathcal{R}$, and number of iterations for refinement of the relevance function T

Output: R-weighted posterior predictive $P^{\mathcal{R}}_{\mathbf{D}_{n+1}|\mathbf{d},\mathbf{z}}$

Compute $P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}$ (Equation (5)) ▷ Step 1
if $\mathcal{R}$ depends on $P_{\boldsymbol{\Theta}}$ **then** ▷ Refinement of $\mathcal{R}$
  $\widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}} \leftarrow P_{\boldsymbol{\Theta}}$
  for $t \in 1:T$ **do**
    Evaluate $\mathcal{R}$ using $\widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}}$
    Compute $L^{\mathcal{R}}$ (Equation (6))
    $\widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}} \leftarrow P^{\mathcal{R}}_{\boldsymbol{\Theta}|\mathbf{d},\mathbf{z}}$
  end for
end if
Evaluate $\mathcal{R}$ using $\widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}}$
Compute $L^{\mathcal{R}}$ (Equation (6)) ▷ Step 2
Compute $P^{\mathcal{R}}_{\mathbf{D}_{n+1}|\mathbf{d},\mathbf{z}}$ (Definition 3.3) ▷ Step 3

the learner has about the value of $\boldsymbol{\psi}^{\star}_{n+1}$ and which does not depend on $\boldsymbol{\theta}^{\star}$ (Figure 1) as **proxy information**. We denote the proxy information $\mathbf{z} \in \mathscr{Z}$. To leverage the proxy information to learn $\boldsymbol{\psi}^{\star}_{n+1}$, the learner specifies a model for the likelihood of proxy information $\mathbf{z}$ given $\boldsymbol{\psi}_{n+1}$, i.e., can compute $p(\mathbf{z}|\boldsymbol{\psi}_{n+1})$.[4] Combined with the prior $P_{\boldsymbol{\Psi}_{n+1}}$, this induces a distribution over $\mathbf{z}$. We denote the corresponding random variable $\mathbf{Z}$.

The posterior probability of a value $\boldsymbol{\psi}_{n+1}$ is

$$p\left(\boldsymbol{\psi}_{n+1}|\mathbf{z}\right) = \frac{p\left(\mathbf{z}|\boldsymbol{\psi}_{n+1}\right)\ p\left(\boldsymbol{\psi}_{n+1}\right)}{\mathbb{E}_{\boldsymbol{\psi}'_{n+1}}\left[p\left(\mathbf{z}|\boldsymbol{\psi}'_{n+1}\right)\right]}. \tag{5}$$

3.2 STEP 2: LEARNING SHARED PARAMETERS VIA LIKELIHOOD WEIGHTING

In step 2, the learner addresses the challenge of avoiding negative transfer (learning the shared parameter $\boldsymbol{\theta}^{\star}$ without depending on a source task parameter prior).

Estimation of the target data-generating process requires estimating a joint distribution over both the shared and target task parameters $\left(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)$. The challenge arises because the learner requires a model for $p\left(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)$. As we discussed in Section 1, using the classic likelihood of a value $\boldsymbol{\theta}$ requires marginalizing over possibly mistaken prior information about the source task parameters.

In an ideal world, when computing $p\left(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)$ the learner would intervene on the source data and set $\boldsymbol{\psi}^{\star}_1 = \ldots = \boldsymbol{\psi}^{\star}_n = \boldsymbol{\psi}_{n+1}$. While this is infeasible, the learner can perform a *pseudo-intervention*: They can manipulate the source data to resemble the consequences of such an intervention. Using likelihood weighting techniques, the learner can reweight the data in order to assign higher weight to observations that are *relevant* to the consequences of $\boldsymbol{\psi}_{n+1}$. The probability of observing $\mathbf{d}_i$ if the i^{th} task parameter had been "set" to $\boldsymbol{\psi}_{n+1}$ is denoted $p\left(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$. The probability of observing all source data in the task characterized by $\boldsymbol{\psi}_{n+1}$ is denoted $p\left(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)$.

[4] In the absence of substantial prior knowledge about how the proxy information is generated, this model may be extremely expressive or even non-parametric.

	Requires	Negative transfer is due to
Classic	Source task prior $P_{\boldsymbol{\Psi}}$	Misspecified $P_{\boldsymbol{\Psi}}$
R-weighted	Relevance function $\mathcal{R}$	Low-fidelity $\mathcal{R}$

Table 1: Key differences between classic and r-weighted Bayesian learning.

Formally, the relevance of an observation is:

Definition 3.1 (Relevance $\mathcal{R}_i(\boldsymbol{\psi}_{n+1})$)**.** *The relevance of the i^{th} observation to $\boldsymbol{\psi}_{n+1}$ is computed by a **relevance function** $\mathcal{R}$ which is positively correlated with $p\left(\mathbf{d}_i|\boldsymbol{\theta}^{\star}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$ in expectation with respect to $P_{\mathbf{D}^{\star},\boldsymbol{\Psi}_{n+1}}$.*

Unlike the classic Bayesian transfer learner who uses the likelihood expression in Equation (1) to construct their posterior, PROMPT uses the **relevance- (r-)weighted likelihood** of each observation:

$$L^{\mathcal{R}}(\mathbf{d}_i, \boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}) \equiv p(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1})^{\mathcal{R}_i(\boldsymbol{\psi}_{n+1})}. \tag{6}$$

Table 1 summarizes the differences between r-weighted and classic Bayesian inference. The key idea of r-weighting is to substitute the requirement for accurate prior knowledge of the source task parameters with a requirement for a suitably-specified relevance function (i.e., the ability to anticipate the consequences of a pseudo-intervention on the source data). At first glance, this may appear to substitute a requirement for one form of prior knowledge with another. However, as we discuss below, specifying a suitable relevance function often does not require knowledge beyond that which is already encoded in the learner's model.

Defining the relevance function. Definition 3.1 requires that $\mathcal{R}_i(\boldsymbol{\psi}_{n+1})$ positively correlate with $p\left(\mathbf{d}_i|\boldsymbol{\theta}^{\star}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$. In Section 4, we provide a result showing that the *fidelity* of the relevance function — the strength of this correlation — affects the extent of the threat of negative transfer. However, computing $p\left(\mathbf{d}_i|\boldsymbol{\theta}^{\star}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$ exactly would require access to $\boldsymbol{\theta}^{\star}$, which the learner does not have.

Given their ignorance of $\boldsymbol{\theta}^\star$, one approach the learner could take would be to construct the relevance function in a way that depends only on their prior $P_{\boldsymbol{\Theta}}$, for instance, as

$$\mathcal{R}_i(\boldsymbol{\psi}_{n+1}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[p\left(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right]. \quad (7)$$

While Equation (7) will likely not exactly recover the consequences of the pseudo-intervention $\boldsymbol{\psi}_i = \boldsymbol{\psi}^\star_{n+1}$, in many cases of practical interest it will tend to correlate with $p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$.[5]

To correct for potential bias in $P_{\boldsymbol{\Theta}}$, we propose a procedure to iteratively refine the relevance function, outlined in Line 2–Line 9 of Algorithm 1. Notice that the source data, which the learner *does* have access to, depend on $\boldsymbol{\theta}^\star$, and so the learner can leverage these data to, for instance, refine the distribution across which the expectation in Equation (7) is taken. We propose the learner first evaluate the relevance function using $P_{\boldsymbol{\Theta}}$, then substitute $P_{\boldsymbol{\Theta}}$ in the definition of the relevance function with the resulting relevance-weighted posterior (Definition 3.2 in Section 3.3), reevaluate the relevance function, and repeat this process for a pre-specified number of iterations. In Section 5, we detail application of this iterative procedure in the context of two synthetic examples. While we observe that this procedure is effective in the context of these examples, an important direction for future work is establishing the conditions under which it converges, i.e., the conditions under which a relevance function satisfying Definition 3.1 is available to the learner.

3.3 STEP 3: COMPUTING THE R-WEIGHTED POSTERIOR PREDICTIVE DISTRIBUTION

We can now define the **relevance- (r-)weighted posterior** and **r-weighted posterior predictive distribution**.

Definition 3.2 (Relevance- (r-)weighted posterior distribution $P^{\mathcal{R}}_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}|\mathbf{d},\mathbf{z}}$)**.** *The r-weighted posterior distribution $P^{\mathcal{R}}_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}|\mathbf{d},\mathbf{z}}$ is the distribution with density*

$$p^{\mathcal{R}}(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}|\mathbf{d}, \mathbf{z}) = \frac{L^{\mathcal{R}}(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1})\, p\left(\mathbf{z}|\boldsymbol{\psi}_{n+1}\right)\, p\left(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)}{\mathbb{E}_{\boldsymbol{\theta}', \boldsymbol{\psi}'_{n+1}\sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}}\left[L^{\mathcal{R}}(\mathbf{d}, \boldsymbol{\theta}', \boldsymbol{\psi} = \boldsymbol{\psi}'_{n+1})\, p\left(\mathbf{z}|\boldsymbol{\psi}'_{n+1}\right)\right]}.$$

Definition 3.3 (Relevance- (r-)weighted posterior predictive distribution $P^{\mathcal{R}}_{\mathbf{D}_{n+1}|\mathbf{d},\mathbf{z}}$)**.** *The r-weighted posterior predictive distribution $P^{\mathcal{R}}_{\mathbf{D}_{n+1}|\mathbf{d},\mathbf{z}}$ is the distribution with density*

$$p^{\mathcal{R}}(\mathbf{d}_{n+1}|\mathbf{d}, \mathbf{z}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{\psi}_{n+1}\sim P^{\mathcal{R}}_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}|\mathbf{d},\mathbf{z}}}\left[p(\mathbf{d}_{n+1}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1})\right].$$

4 THEORETICAL RESULTS

In Section 2, we introduced a formal framework for assessing the threat of negative transfer. In Section 3, we introduced a framework for Bayesian transfer learning that uses a pre-specified relevance function to r-weight the likelihood. Our goal here is to assess whether r-weighting can effectively reduce the threat of negative transfer, and if so, the conditions under which this is the case.

To assess the threat of negative transfer to the r-weighted Bayesian transfer learner, we introduce an information gain measure analogous to Definition 2.1, but that measures the degree to which the r-weighted posterior favors $\boldsymbol{\theta}^\star$ with respect to the prior:[6]

Definition 4.1 (Information gained by the r-weighted Bayesian learner $\mathrm{IG}^{\mathcal{R}}$)**.** *The information gained by the r-weighted Bayesian transfer learner about the shared parameter $\boldsymbol{\theta}^\star$ is*

$$\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^\star\right) \equiv \mathop{\mathbb{E}}_{\mathbf{d},\mathbf{z}\sim P_{\mathbf{D}^\star,\mathbf{Z}}}\left[\log\left(\frac{p^{\mathcal{R}}(\boldsymbol{\theta}^\star|\mathbf{d}, \mathbf{z})}{p(\boldsymbol{\theta}^\star)}\right)\right].$$

Analogously to Definition 2.2, we say that the r-weighted Bayesian transfer learner experiences negative transfer when $\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^\star\right) \leq 0$.

Below, we provide two results that together show that the relevance function controls the threat of negative transfer. Theorem 4.4 shows that the threat of negative transfer to the r-weighted Bayesian transfer learner depends on misspecification of the r-weighted likelihood, where the misspecification can be interpreted as the degree to which the relevance function corrects for a mismatch between the source and possible target tasks. Proposition 4.5 decomposes this measure of misspecification, showing that it is a negative function of the *fidelity* of the relevance function. The proofs of all results are deferred to Appendix B.

Misspecification of the r-weighted likelihood is:

Definition 4.2 (Misspecification of the r-weighted likelihood $\Delta^{\mathcal{R}}$)**.** *The degree to which the r-weighted likelihood is misspecified is $\Delta^{\mathcal{R}} \equiv \mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right]$ where $P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}}$ is the distribution of data resulting from viewing $\mathcal{R}_i(\boldsymbol{\psi}_{n+1})$ replicates of each $\mathbf{d}_i$.*

In the r-weighted case, the misspecification stems from the failure of the pseudo-replication to correct for a mismatch in the source tasks (the consequences of $\boldsymbol{\psi}^\star$) and possible target tasks (the consequences of possible values $\boldsymbol{\psi}_{n+1}$).

Theorem 4.4 gives a result analogous to Theorem 2.4 for the r-weighted case. It depends on the following assumptions:

Assumption 4.3 ($L^{\mathcal{R}}$ is bounded)**.** *The r-weighted likelihood $L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)$ is bounded from both below and above: $\exists a, b \in \mathbb{R}^+$ such that $\forall \mathbf{d} \in \mathscr{D}, \boldsymbol{\theta} \in \mathscr{T}, \boldsymbol{\psi}_{n+1} \in \mathscr{S}$, $a \leq L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right) \leq b$.*

[5] See Appendix B.4 for discussion of a counterexample.

[6] See discussion in Appendix B.3 for interpretation of $P_{\mathbf{Z}}$.

Assumption B.8 (informal)**.** *The proxy is sufficiently informative in the sense that the "variability" of $\mathbf{\Psi}_{n+1}|\mathbf{z}$ is smaller than the "variability" of $\mathbf{\Psi}_{n+1}$ by a "large enough" margin. The formal condition is given in Appendix B.3.*

Assumption B.9 (informal)**.** *The r-weighted likelihood $L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)$ is "smooth enough" in a neighborhood of $\boldsymbol{\theta}^\star$ **and** the estimated relevances are not "too large". The formal condition is given in Appendix B.3.*

Theorem 4.4 (Negative transfer with an r-weighted likelihood)**.** *Under Assumption 4.3 and Assumptions B.8 and B.9 in Appendix B.3,*

$$\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^\star\right) \leq A\left(C - \Delta^{\mathcal{R}}\right)$$

where A and C are constants that do not depend on $\Delta^{\mathcal{R}}$.

Proposition 4.5 analyzes the effect of $\mathcal{R}$ on $\Delta^{\mathcal{R}}$. The role of $\mathcal{R}$ in mitigating negative transfer depends on the *fidelity* of the relevance function:

Definition B.11 (informal)**.** *$\rho^{\mathcal{R}}$ is a measure of the fidelity of the relevance function, i.e., the extent of the correlation of $\mathcal{R}_i(\boldsymbol{\psi}_{n+1})$ with $p(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1})$ in expectation with respect to $P_{\mathbf{D}^\star, \mathbf{\Psi}_{n+1}}$. The formal definition is given in Appendix B.4.*

Proposition 4.5 (Negative transfer is reduced by high-fidelity relevance functions)**.** *$\Delta^{\mathcal{R}}$ is a negative function of $\rho^{\mathcal{R}}$. In particular,*

$$\Delta^{\mathcal{R}} = \mathbb{E}\left[\mathrm{ESS}\left(\mathbf{d}, \boldsymbol{\psi}_{n+1}\right) \mathrm{DIS}\left(\mathbf{d}, \boldsymbol{\psi}_{n+1}\right)\right] - n\rho^{\mathcal{R}} + D$$

where $\mathrm{ESS}\left(\mathbf{d}, \boldsymbol{\psi}_{n+1}\right) \equiv \sum_{i=1}^{n} \mathcal{R}_i(\boldsymbol{\psi}_{n+1})$ is the effective sample size induced by the relevance function $\mathcal{R}$ evaluated on the sample $\mathbf{d}$ and task parameter $\boldsymbol{\psi}_{n+1}$, $\mathrm{DIS}\left(\mathbf{d}, \boldsymbol{\psi}_{n+1}\right) \equiv -\log\left(p\left(\mathbf{d}|\boldsymbol{\theta}^\star, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right)$ captures the dissimilarity of the source data to the target task characterized by $\boldsymbol{\psi}_{n+1}$, the expectation is taken with respect to $P_{\mathbf{D}^\star, \mathbf{\Psi}_{n+1}}$, and the constant D does not depend on $\mathcal{R}$.

Remark 4.6 (Weakly informative proxies mitigate negative transfer)**.** *Proposition 4.5 shows that $\Delta^{\mathcal{R}}$ does not depend on the accuracy of the learner's inferences about $\boldsymbol{\psi}^\star_{n+1}$, i.e., on the informativeness of the proxy information. Informative proxies facilitate targeted inference insofar as they facilitate estimation of the target task parameter, but do not improve the r-weighted learner's ability to recover the shared parameter from the source data. PROMPT's provable advantage over classic Bayesian inference does depend on the availability of some proxy information only to satisfy Assumption B.8, required in the proof of Theorem 4.4: If the available proxy information is not somewhat informative, the magnitude of $\Delta^{\mathcal{R}}$ does not necessarily imply the degree of the threat of negative transfer.*

5 EXAMPLES

We here demonstrate application of PROMPT in two synthetic settings and on one real-world dataset. Additional details of all examples are provided in Appendix C.

Taken together, these examples demonstrate that PROMPT can significantly reduce the threat of negative transfer, and that its effectiveness in doing so is robust to unreliable and misleading proxy information. In all examples, we defined the relevance function in a purely source data-dependent way, illustrating the availability of effective relevance functions in settings of practical interest.

5.1 TREATMENT EFFECT ESTIMATION

To continue with our motivating example, we first demonstrate application of PROMPT to treatment effect estimation using similar modeling paradigms to those used in clinical prediction tasks (Gunn-Sandell et al., 2023). We first apply PROMPT in a synthetic setting that allows us to manipulate factors like the risk of negative transfer. We then apply PROMPT to a real-world dataset of smoking behavior.

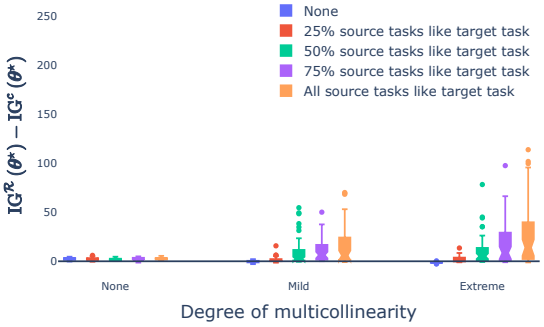
In both cases, we consider the treatment effect to be transferable, i.e., the parameter corresponding to the size of the influence of the treatment on outcomes is shared across tasks. Here, negative transfer refers to a situation where learning from the source data causes the learner to believe that the treatment has an effect opposite to its true effect (e.g., a negative rather than positive treatment effect).

Linear regression. The synthetic data in this example are generated according to the model

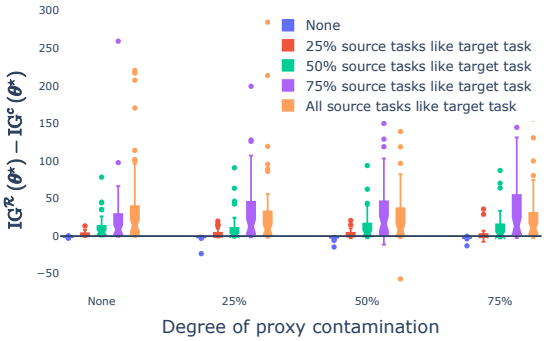
$$\mathbf{y}_i|\mathbf{x}_i \sim \mathcal{N}\left(\boldsymbol{\theta}^\star \mathbf{x}_{i,1} + \boldsymbol{\psi}_i^\star \mathbf{x}_{i,2}, 1\right)$$

where the shared parameter $\boldsymbol{\theta}^\star$ represents the effect of a synthetic treatment $\mathbf{x}_{i,1}$ and $\boldsymbol{\psi}_i^\star$ represents the effect of a synthetic confounder (e.g., quality of care) $\mathbf{x}_{i,2}$. We computed the relevances as $\mathcal{R}_i(\boldsymbol{\psi}_{n+1}) \propto \mathbb{E}_{\boldsymbol{\theta} \sim \widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}}}\left[p\left(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right]$ where $\widehat{P^{\mathcal{R}}_{\boldsymbol{\Theta}}}$ was formed using the iterative procedure described in Section 3.2.

Alleviating the risk of negative transfer: Figure 2a shows how $\mathrm{IG}^{\mathcal{R}}$ compares with IG^{c} as a function of the risk of negative transfer and the representativeness of the target task in the distribution of source tasks. To induce the risk of negative transfer, we manipulated the degree of multicollinearity between $\mathbf{x}_{(\cdot,1)}$ and $\mathbf{x}_{(\cdot,2)}$: More multicollinearity makes $\boldsymbol{\theta}^\star$ and $\boldsymbol{\psi}^\star_{n+1}$ harder to separately identify, so we interpret this as a higher risk of negative transfer. We also varied the distribution of source tasks. When $p\%$ of tasks resemble the target task, $1 - p\%$ of tasks are set to a value that is well-represented by $P_{\boldsymbol{\Psi}}$. In this sense, our results are a somewhat conservative test of PROMPT.



(a) Alleviating the risk of negative transfer. In all cases, proxy information is uncontaminated.



(b) Robustness to noisy proxy information. In all cases, there is an extreme degree of multicollinearity (i.e., large threat of negative transfer).

Figure 2: Advantage of the r-weighted learner in the linear regression setting. Each box includes results from 50 simulations and shows the interquartile region (boxes) and outliers (points) of $\mathrm{IG}^{\mathcal{R}}(\boldsymbol{\theta}^\star) - \mathrm{IG}^{c}(\boldsymbol{\theta}^\star)$. In each simulation, $\boldsymbol{\psi}^\star_{n+1}$, $\mathbf{d}$, and $\mathbf{z}$ are randomly regenerated.

When there is no multicollinearity, the classic learner is not at risk of negative transfer, and performs on par with the learner with an r-weighted likelihood. When all source tasks are well-represented in the learner's prior (blue box), the classic learner's prior is well-specified and they perform on par with the r-weighted learner. When there is a risk of negative transfer, $\mathrm{IG}^{\mathcal{R}}$ is generally higher, especially when many source tasks resemble the target task (and the classic learner's prior $P_{\boldsymbol{\Psi}}$ is more misspecified).

Robustness to noisy proxy information: The synthetic proxy information represents feedback from a domain expert. While domain experts may not be able to articulate precise knowledge of the target task, they can often provide intuitive assessments (Kahneman and Klein, 2009) such as the degree to which an outcome is representative of a given situation (Tversky and Kahneman, 1974). Our synthetic proxy represents a domain expert who is presented with a hypothetical outcome and asked the degree to which it is representative of the target task on a scale of 0–7. To assess the robustness of PROMPT to noisy proxies, we contaminated a percentage of these synthetic judgments. The percentage of contaminated proxy values is unknown to the learner, who always models the proxy information as completely uncontaminated.

In line with our result in Proposition 4.5, Figure 2b shows that noisy proxy information does not affect the relative advantage of the r-weighted learner: Regardless of the degree of proxy contamination, the r-weighted learner tends to outperform the classic learner.

Predicting smoking behavior. We also applied PROMPT to predict smoking behavior in a dataset from Hasselblad (1998) provided by the R package `netmeta` (Balduzzi et al., 2023), which consists of data from 24 studies on the number of patients who stopped smoking after receiving one of four treatments.[7] Each study includes data from patients who received some but not all treatments. We considered each study a separate task. Each observation is indexed by study and treatment (so $\mathbf{y}_i$ is the number of patients who stopped smoking after receiving a given treatment in a given study and $\boldsymbol{\psi}_i = \boldsymbol{\psi}_j$ if i and j index data from different treatments administered as part of the same study). We modeled the data as

$$\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\psi}_i \sim \text{Binomial}\left(\text{sigmoid}\left(\boldsymbol{\theta}\mathbf{x}_{i,(1:4)}^\top + \boldsymbol{\psi}_i\right), N_i\right)$$

where $\mathbf{x}_{i,(1:4)}$ are indicators of the treatment received and N_i is the number of patients who received the indicated treatment in the indicated study. We considered 24 different partitions of the data into source and target data, with each partition treating data from one study as target data and data from the remaining 23 studies as source data. We defined the relevance function as

$$\mathcal{R}_i(\boldsymbol{\psi}_{n+1}) = \text{sigmoid}\left(n\frac{p\left(\mathbf{d}_i|\boldsymbol{\theta}=0, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)}{p\left(\mathbf{d}|\boldsymbol{\theta}=0, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)}\right).$$

Figure 3 shows the relative performance of the r-weighted and classic Bayesian transfer learners as a function of the informativeness of the proxy information. Unlike in our synthetic example, here we do not have access to the true value $\boldsymbol{\theta}^\star$ and so cannot directly compute $\mathrm{IG}^{\mathcal{R}}(\boldsymbol{\theta}^\star)$ and $\mathrm{IG}^{c}(\boldsymbol{\theta}^\star)$. Instead, we assess how well the two methods can predict the outcome in the target task. The classic learner here has the advantage of prior source information in the

[7] This example was inspired by the example detailed in Holzhauer and Bean (2025). The code used the package `brms` (Bürkner, 2017) and Stan modeling language (Stan Development Team, 2024).

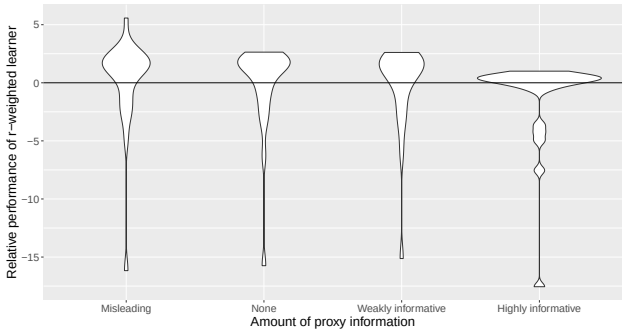


Figure 3: Advantage of the r-weighted learner in the dataset of smoking behavior. Each plot shows the distribution of values of $\log\left(\frac{p^{\mathcal{R}}(\mathbf{d}_{n+1}|\mathbf{d},\mathbf{z})}{p(\mathbf{d}_{n+1}|\mathbf{d},\mathbf{z})}\right)$ across 24 partitions of source/target data.

form of knowledge of which observations belong to the same task. We also do not anticipate a substantial threat of negative transfer here. Nevertheless, the r-weighted learner outperforms the classic learner in the majority of cases.

Robustness to noisy proxy information: The synthetic proxy information represents an imprecise estimate of the value of $\boldsymbol{\psi}^{\star}_{n+1}$. Highly informative proxy information refers to more precise estimates than weakly informative proxy information. To assess the robustness of PROMPT to misleading proxies, we added a bias to some synthetic estimates. While the learner is aware of the degree of precision of an estimate, they are unaware of the potential presence of bias.

Proposition 4.5 showed that the extent of negative transfer for the r-weighted learner does not depend on the informativeness of the proxy information. Figure 3 shows that the r-weighted learner's advantage appears to actually decrease with the informativeness of the proxy information. Further inspection revealed that, in this example, the r-weighted learner's performance is not sensitive to the amount of proxy information, and the difference reflects the classic learner's higher performance in the presence of more informative proxy information.[8] Further understanding this insensitivity to proxy information, as well as the nature of the tasks that lead the r-weighted learner to perform much worse than the classic learner, is a direction for future investigation.

5.2 GAUSSIAN PROCESS REGRESSION

We next demonstrate application of PROMPT in a Gaussian Process (GP) regression setting with a composite kernel. Additional details and results are given in Appendix C.3.

[8]This difference between the r-weighted and classic learners' sensitivity to proxy information is largely accounted for by an effective difference in model structure; as we describe in Appendix C.2, the r-weighted learner learns a model with a single intercept rather than a separate linear effect for each study.

Data were generated according to the model

$$\mathbf{y}_i|\mathbf{x}_i \sim \mathcal{GP}\left(0, \mathbf{k}(\mathbf{x}_i, \mathbf{x})\right)$$

where $\mathbf{k}(\mathbf{x}_i, \mathbf{x}) = \mathrm{RBF}_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{x}) + \mathrm{RBF}_{\boldsymbol{\psi}_i}(\mathbf{x}_i, \mathbf{x})$ and RBF_l is the radial basis function with lengthscale l.[9] This setting poses a risk of negative transfer because the shared and task parameters act in combination to determine the smoothness of the sampled functions (Sloman et al., 2024). We used the same methods to generate proxy information and specify the relevance function as for the linear regression example, but varied the number of iterations used for refinement of the relevance function. Figure 4 in Appendix C.3 shows how $\mathrm{IG}^{\mathcal{R}}$ compares with IG^{c} as a function of each of multiple simulation parameters. In all cases, the r-weighted learner tends to identify the value of the shared parameter as well as, and usually more successfully than, the classic learner.

Robustness to noisy proxy information: Figure 4f in Appendix C.3 shows how $\mathrm{IG}^{\mathcal{R}}$ compares with IG^{c} as a function of the amount of proxy contamination. In line with our result in Proposition 4.5, the r-weighted learner outperforms the classic Bayesian learner even in the presence of substantial proxy contamination (although their advantage is greatest in the absence of proxy contamination).

6 DISCUSSION

We presented a Bayesian perspective on negative transfer, from which we showed that negative transfer can arise from misspecified prior source information. Based on this insight, we developed PROMPT, a novel framework for Bayesian transfer learning which alleviates the learner's dependence on prior source information. The framework of PROMPT can accommodate a variety of relevance functions and forms of proxy information. PROMPT's provable advantage depends on the fidelity of the specified relevance function. In Section 5, we provided concrete examples of possible relevance functions and demonstrated PROMPT's robustness to noisy and misleading proxies in a variety of settings. We found that in practice we were able to specify relevance functions of sufficiently high fidelity to reduce negative transfer. Ultimately, however, there may exist situations where such a relevance function is unavailable (for example, if $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ interact such that the direction of the gradient of predictions with respect to $\boldsymbol{\psi}$ depends on $\boldsymbol{\theta}$). As Table 1 shows, in such cases, the practitioner must make a choice about whether they can more confidently specify the prior over source task parameters or the relevance function. The development of a more systematic framework for defining the relevance function is a promising avenue for future work. Many transfer learning applications leverage high-dimensional, non-linear datasets (Suder et al., 2023) and future work should in particular look to the development of a scalable framework for applying and evaluating PROMPT in such contexts.

[9]The kernel was renormalized to have an amplitude of 1.

Acknowledgements

The authors thank Ayush Bharti and Sammie Katt for helpful feedback on an initial draft, and several anonymous reviewers for helpful comments. This work was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI and decisions 358958, 359567. SJS and SK were supported by the UKRI Turing AI World-Leading Researcher Fellowship, [EP/W002973/1]. This work used the Computational Shared Facility at The University of Manchester.

Bibliography

Ibrahim Alabdulmohsin, Nicole Chiou, Alexander D'Amour, Arthur Gretton, Sanmi Koyejo, Matt J. Kusner, Stephen R. Pfohl, Olawale Salaudeen, Jessica Schrouff, and Katherine Tsai. Adapting to latent subgroup shifts via concepts and proxies. In *Proceedings of the* 26^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, 2023.

Sara Balduzzi, Gerta Rücker, Adriani Nikolakopoulou, Theodoros Papakonstantinou, Georgia Salanti, Orestis Efthimiou, and Guido Schwarzer. netmeta: An R package for network meta-analysis using frequentist methods. *Journal of Statistical Software*, 106(2), 2023.

Paul-Christian Bürkner. brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software*, 80(1), 2017.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction. In *The Seventh International Conference on Learning Representations (ICLR 2019)*, 2019.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *The Sixth International Conference on Learning Representations (ICLR 2018)*, 2018.

Peter Grünwald. Safe learning: bridging the gap between bayes, mdl and statistical learning theory via empirical convexity. In *24th Annual Conference on Learning Theory*, 2011.

Lauren B. Gunn-Sandell, Edward J. Bedrick, Jacob L. Hutchins, Aaron A. Berg, Alexander M. Kaizer, and Nichole E. Carlson. A practical guide to adopting bayesian analyses in clinical research. *Journal of Clinical and Translational Studies*, 8, 2023.

Vic Hasselblad. Meta-analysis of multitreatment studies. *Medical Decision Making*, 18, 1998.

Björn Holzhauer and Andrew Bean. Network meta-analysis, 2025. Accessed via https://opensource.nibr.com/bamdd/src/02j_network_meta_analysis.html.

Daniel Kahneman and Gary Klein. Conditions for intuitive expertise: A failure to disagree. *American Psychologist*, 64(6), 2009.

Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 2014.

Sergey Oladyshkin and Wolfgang Nowak. The connection between bayesian inference and information theory for model selection, information gain and experimental design. *Entropy*, 21, 2019.

Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, Michael O'Boyle, and Amos Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1), 2024.

Sabina J. Sloman, Ayush Bharti, Julien Martinelli, and Samuel Kaski. Bayesian active learning in the presence of nuisance parameters. In *Proceedings of the* 40^{th} *Conference on Uncertainty in Artificial Intelligence (UAI 2024)*, 2024.

Stan Development Team. Stan reference manual, version 2.36, 2024. Accessed via https://mc-stan.org.

Piotr M. Suder, Jason Xu, and David B. Dunson. Bayesian transfer learning, 2023. Accessed via https://arxiv.org/abs/2312.13484.

Eric J. Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An Introduction to Proximal Causal Inference. *Statistical Science*, 2024.

Katherine Tsai, Stephen R Pfohl, Olawale Salaudeen, Nicole Chiou, Matt Kusner, Alexander D'Amour, Sanmi Koyejo, and Arthur Gretton. Proxy methods for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185, 1974.

Zirui Wang, Zihang Dai, Barnabas Poczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018.

Proxy-informed Bayesian transfer learning with unknown sources (Supplementary Material)

Sabina J. Sloman[1] Julien Martinelli[*2] Samuel Kaski[1,2,3]

[1]Department of Computer Science, University of Manchester, Manchester, UK
[2]Department of Computer Science, Aalto University, Espoo, Finland
[3]ELLIS Institute Finland, Helsinki, Finland

The appendix is organized as follows:

- In Appendix A, we discuss related works in more detail.
- In Appendix B, we provide proofs of all our mathematical results.
- In Appendix C, we provide further details of the examples described in Section 5. Appendix C.3 additionally provides the results of the GP regression example.

A RELATED WORK

Likelihood weighting has been applied for purposes that include potential model misspecification (Grünwald, 2011; Miller and Dunson, 2019; Dewaskar et al., 2025), potential conflation of transferable and task-specific effects (Ibrahim and Chen, 2000; Ibrahim et al., 2011, 2014; Suder et al., 2023), model selection (Ibrahim et al., 2014), and increased efficiency of MCMC samplers (Schuster and Klebanov, 2021).

Probabilistic meta-learning (Gordon et al., 2019) is a paradigm in which a meta-learner simultaneously learns a transferable parameter value and a distribution over task parameter values. Unlike PROMPT, this framework assumes the data sources are known in the sense that each data point can be indexed by its task. This distinction also sets us apart from other Bayesian meta-learning approaches (Grant et al., 2018; Yoon et al., 2018; Patacchiola et al., 2020). Moreover, the aim of probabilistic meta-learning is to learn a distribution over task parameters. When the target task will arise from the same distribution as the source tasks, probabilistic meta-learning facilitates good performance on average across tasks. However, the goal of PROMPT is to provide a posterior predictive distribution tailored to a target task that may not arise from the same distribution as the source tasks.

Using domain similarity for domain adaptation. Many existing theoretical bounds for domain adaptation rely on the similarity between source and target tasks (Redko et al., 2019). Some approaches to domain adaptation use similarity of covariates (inputs) in the target and source tasks to weight source data during training (Plank and van Noord, 2011; Ponomareva and Thelwall, 2012; Remus, 2012; Ruder and Plank, 2017) or importance sampling techniques (Quiñonero-Candela et al., 2009). While this can be effective in cases of pure covariate shift (a change in the distribution of inputs), our formulation allows for differences in the map between covariates and outcomes that cannot be detected on the basis of covariate information alone.

Proximal causal learning is a paradigm that uses proxy information to learn causal effects (Kuroki and Pearl, 2014; Tchetgen et al., 2024; Alabdulmohsin et al., 2023; Tsai et al., 2024). Our setting is similar to the multi-domain adaptation setting of Tsai et al. (2024). We differ in that (i) we assume data sources are unknown, while they assume data can be indexed by its task, and (ii) we assume the presence of both shared and task parameters, while they do not distinguish

[*]Work done while at Inserm Bordeaux Population Health, Vaccine Research Institute, Université de Bordeaux, Inria Bordeaux Sud-ouest, France.

between these. While our method for estimating the task parameter also leverages proxy methods, we differ in our usage of reweighting methods to estimate the shared parameter, which facilitates robust estimation without requiring additional proxy information.

As shown in Figure 1, our formulation is stated in terms of the dependencies between shared parameters, task parameters, and observations, and so our work shares conceptual connections with the more general paradigm of causal inference. For instance, conceptualizing r-weighting as a pseudo-intervention requires conceptualizing the task parameters as a cause of the observations. We do not however require that either the shared or task parameters parameterize the causal effect of one observable variable on another; these parameters can represent any unobservable factor influencing the data.

Human-in-the-loop learning. In many applications, domain experts are a viable source of proxy information, and so our work can be tied to human-in-the-loop machine learning (Wu et al., 2022). Like us, some human-in-the-loop methods leverage expert feedback in a Bayesian framework. For example, Nahal et al. (2024) use expert feedback for learning in out-of-distribution settings, while Sundin et al. (2018) query experts about the relevance of a given feature for outcome prediction.

B MATHEMATICAL DETAILS

B.1 DEFINITIONS

- $\mathrm{H}(P)$ is the entropy of distribution P with density p:

$$\mathrm{H}(P) = -\mathop{\mathbb{E}}_{\mathbf{x}\sim P}[\log(p(\mathbf{x}))]$$

- $\mathrm{H}(P \mid\mid Q)$ is the cross-entropy from distribution P to distribution Q with density q:

$$\mathrm{H}(P \mid\mid Q) = -\mathop{\mathbb{E}}_{\mathbf{x}\sim P}[\log(q(\mathbf{x}))]$$

- $\mathrm{D}_{\mathrm{KL}}(P \mid\mid Q)$ is the Kullback-Leibler divergence from distribution P with density p, to distribution Q with density q:

$$\mathrm{D}_{\mathrm{KL}}(P \mid\mid Q) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\right]$$

B.2 PROOF OF THEOREM 2.4

The information gain achieved by the classic Bayesian learner (Definition 2.1) can be written as:

$$\begin{aligned}
\mathrm{IG}^{c}(\boldsymbol{\theta}^{\star}) &= \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^{\star}}}\left[\log\left(\frac{p(\boldsymbol{\theta}^{\star}|\mathbf{d})}{p(\boldsymbol{\theta}^{\star})}\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^{\star}}}\left[\log\left(\frac{\frac{L(\mathbf{d},\boldsymbol{\theta}^{\star})\,p(\boldsymbol{\theta}^{\star})}{\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}[L(\mathbf{d},\boldsymbol{\theta})]}}{p(\boldsymbol{\theta}^{\star})}\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^{\star}}}\left[\log\left(\frac{L(\mathbf{d},\boldsymbol{\theta}^{\star})}{\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}[L(\mathbf{d},\boldsymbol{\theta})]}\right)\right]
\end{aligned}$$

The proof follows the proof of Proposition 4.1 and Theorem 4.5 of Sloman et al. (2024). It depends on the following definitions:

Definition B.1 (ϵ-neighborhood of $\boldsymbol{\theta}$ $N_{\epsilon}(\boldsymbol{\theta})$ (Definition 4.2 of Sloman et al. 2024))**.** $N_{\epsilon}(\boldsymbol{\theta}) \equiv \{\boldsymbol{\theta}' \in \mathscr{T} \mid d(\boldsymbol{\theta},\boldsymbol{\theta}') < \epsilon\}$, *where d is a suitable distance measure, is the ϵ-neighborhood of $\boldsymbol{\theta}$.*

Definition B.2 ($P^{\mathscr{A}}_{\boldsymbol{\Theta}}$ (modification of Definition 4.3 of Sloman et al. 2024))**.** $P^{\mathscr{A}}_{\boldsymbol{\Theta}}$ *refers to the distribution of $\boldsymbol{\Theta}$ obtained by restricting the support of the learner's prior to the set $\mathscr{A}$, under which*

$$p^{\mathscr{A}}(\boldsymbol{\theta}) \equiv \frac{p(\boldsymbol{\theta})}{\int_{\mathscr{A}} p(\boldsymbol{\theta})\,d\boldsymbol{\theta}}$$

for any $\boldsymbol{\theta} \in \mathscr{A}$.

Assumption B.3 (Smoothness in parameter space (Assumption 4.4 of Sloman et al. (2024))). *There exists some $\epsilon > 0$ such that*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right] \\
&\geq \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\left(\int_{N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(L(\mathbf{d},\boldsymbol{\theta}^\star)\right) + \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right]
\end{aligned}$$

where $\left(\int_{N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)$ and $\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)$ are the probability that a value $\boldsymbol{\theta}$ is inside and outside the ϵ-neighborhood of $\boldsymbol{\theta}^\star$, respectively.

Remark B.4. *Assumption B.3 holds when $\boldsymbol{\Theta}$ is a discrete random variable (in which case the ϵ-neighborhood of $\boldsymbol{\theta}^\star$ can be defined as $\{\boldsymbol{\theta}^\star\}$ and to exclude all other parameter values). When $\boldsymbol{\Theta}$ is a continuous random variable, Assumption B.3 is essentially a smoothness condition: For likelihoods that are sufficiently smooth around $\boldsymbol{\theta}^\star$, we can expect it to hold for $\epsilon \to 0$. To see this, notice that Jensen's inequality implies that*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right] \\
&\geq \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\left(\int_{N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right) + \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right].
\end{aligned}$$

Assumption B.3 holds when $\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{N_\epsilon(\boldsymbol{\theta}^\star)}}\left[p(\mathbf{d}|\boldsymbol{\theta})\right] \approx p(\mathbf{d}|\boldsymbol{\theta}^\star)$ and the approximation is tight enough that it does not close the Jensen gap.

Taking $P_{\mathbf{D}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}$ to be the source data distribution conditioned on the event that the shared parameter is not in the ϵ-neighborhood of $\boldsymbol{\theta}^\star$, we obtain

$$\begin{aligned}
\mathrm{IG}^c\left(\boldsymbol{\theta}^\star\right) &= \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(L\left(\mathbf{d},\boldsymbol{\theta}^\star\right)\right) - \log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L\left(\mathbf{d},\boldsymbol{\theta}\right)\right]\right)\right] \\
&\leq \mathbb{E}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(L(\mathbf{d},\boldsymbol{\theta}^\star)\right) - \left(\int_{N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(L(\mathbf{d},\boldsymbol{\theta}^\star)\right) - \right. \\
&\qquad \left. \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right] \qquad \text{(Assumption B.3)} \\
&= \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\left(\log\left(L(\mathbf{d},\boldsymbol{\theta}^\star)\right) - \log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L(\mathbf{d},\boldsymbol{\theta})\right]\right)\right)\right] \\
&= \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\left(\mathrm{H}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}\right) - \mathrm{H}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}^\star}\right)\right) \\
&= \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\left(\mathrm{H}\left(P_{\mathbf{D}^\star}\right) + \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}\right) - \mathrm{H}\left(P_{\mathbf{D}^\star}\right) - \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}^\star}\right)\right) \\
&= \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)\left(\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}\right) - \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}^\star}\right)\right) \qquad (8)
\end{aligned}$$

as stated in the theorem for $A = \left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)$ and $B = \mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star} \,||\, P_{\mathbf{D}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}\right)$.

B.3 PROOF OF THEOREM 4.4

The r-weighted information gain (Definition 4.1) can be written as:

$$\begin{aligned}
\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^{\star}\right) &= \mathop{\mathbb{E}}_{\mathbf{d},\mathbf{z}\sim P_{\mathbf{D}^{\star},\mathbf{Z}}}\left[\log\left(\frac{p^{\mathcal{R}}(\boldsymbol{\theta}^{\star}|\mathbf{d},\mathbf{z})}{p(\boldsymbol{\theta}^{\star})}\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{d},\mathbf{z}\sim P_{\mathbf{D}^{\star},\mathbf{Z}}}\left[\log\left(\frac{\frac{\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\,p(\boldsymbol{\theta}^{\star})}{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi}'_{n+1}\sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}'_{n+1}\right)\right]}}{p\left(\boldsymbol{\theta}^{\star}\right)}\right)\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{d},\mathbf{z}\sim P_{\mathbf{D}^{\star},\mathbf{Z}}}\left[\log\left(\frac{\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]}{\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi}'_{n+1}\sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}'_{n+1}\right)\right]}\right)\right]
\end{aligned} \tag{9}$$

Remark B.5. *Notice that* $\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^{\star}\right)$ *is defined as an expectation over* $P_{\mathbf{Z}}$ *as well as* $P_{\mathbf{D}^{\star}}$*. This, and all other quantities in our analysis which include expectations over* $\mathbf{Z}$*, can be interpreted as marginalizing across the learner's subjective uncertainty about the proxy information they will receive. We could have defined* $\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^{\star}\right)$ *as an expectation across a "true" distribution of proxy information, with a corresponding interpretation as the extent to which the learner can expect to gain information upon encountering a given distribution generating* both *source data and proxy information. Although such an extension of the current analysis would in some sense be technically more complete, we opt to simplify our analysis and define the expectation over proxy information with respect to the learner's subjective uncertainty. Both the learner using a classic likelihood and the learner using an r-weighted likelihood use the same prior over* $\mathbf{Z}$ *in estimation of* $\boldsymbol{\psi}^{\star}_{n+1}$*, and so the incorrectness of the prior over proxy information is less important than the incorrectness of the prior over source task parameters in understanding the relative advantage of r-weighting.*

The proof of Theorem 4.4 uses the following lemma:

Lemma B.6. *Define* $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}\right) \equiv \log\left(\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right) - \mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right)\right]$ *and* $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}}\right) \equiv \log\left(\mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right) - \mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\log\left(\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right)\right]$*. Under Assumption 4.3 and Assumption B.8 (stated formally in the proof of the lemma),* $\mathbb{E}_{\mathbf{d}\sim P_{\mathbf{D}^{\star}}}\left[\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}}\right)\right] \geq \mathbb{E}_{\mathbf{d},\mathbf{z}\sim P_{\mathbf{D}^{\star},\mathbf{Z}}}\left[\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}\right)\right]$.

Proof of Lemma B.6. The lemma leverages a result known as Hölder's defect (Steele, 2004; Becker, 2012):

Theorem B.7 (Hölder's defect (restated from Steele 2004[1]))**.** *If* $f : [a,b] \to \mathbb{R}$ *is twice differentiable and if we have the bounds*

$$0 \leq m \leq f''(x) \leq M \text{ for all } x \in [a,b]\,,$$

then for a distribution P *over* $[a,b]$*, there exists a real value* $\mu \in [m,M]$ *for which one has the formula*

$$\underbrace{\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] - f\left(\mathop{\mathbb{E}}_{x\sim P}\left[x\right]\right)}_{\mathcal{J}(-f;P)} = \frac{1}{2}\mu \mathrm{Var}_{x\sim P}\left[x\right]$$

for $\mathrm{Var}_{x\sim P}\left[x\right] \equiv \mathbb{E}_{x\sim P}\left[\left(x - \mathbb{E}_{x\sim P}\left[x\right]\right)^2\right]$.

Our goal is to use Hölder's defect to relate $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}\right)$ and $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}}\right)$ to $\mathrm{Var}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]$ and $\mathrm{Var}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right]$, respectively. We first verify that the conditions required for Hölder's defect formula to apply are met. For both applications of the result, f is the negative of the log function. In application to $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}\right)$, f takes as input values of $L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^{\star},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)$. In application to $\mathcal{J}\left(\log; P_{\boldsymbol{\Psi}_{n+1}}\right)$, f takes as input values of $\mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]$.

[1] Steele (2004) states the result in terms of discrete sums; we here modified the statement of the result so it can be interpreted for continuous random variables.

- $f : [a, b] \to \mathbb{R}$: Assumption 4.3 ensures that inputs in both cases are bounded from both below and above.
- f is twice differentiable: The second derivative of f evaluated at x is $f''(x) = \frac{1}{x^2}$.
- $0 \leq m \leq f''(x) \leq M$: Assumption 4.3 ensures this for $m = \frac{1}{b^2}$ and $M = \frac{1}{a^2}$.

Hölder's defect then implies the following:

$$\mathcal{J}\left(\log ; P_{\boldsymbol{\Psi_{n+1}}|\mathbf{z}}\right) = \frac{1}{2}\mu_1\left(\mathbf{d}, \mathbf{z}\right) \mathrm{Var}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^{\star}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right] \tag{10}$$

for a scalar μ_1 that depends on $\mathbf{d}$ and $\mathbf{z}$, and

$$\mathcal{J}\left(\log ; P_{\boldsymbol{\Psi_{n+1}}}\right) = \frac{1}{2}\mu_2\left(\mathbf{d}\right) \mathrm{Var}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right]\right] \tag{11}$$

for a scalar μ_2 that depends on $\mathbf{d}$.

We can now formally state Assumption B.8:

Assumption B.8 (Sufficiently informative proxy)**.** *The proxy is sufficiently informative in the sense that the following condition holds on the relative variances of $\boldsymbol{\Psi_{n+1}}|\mathbf{z}$ and $\boldsymbol{\Psi_{n+1}}$:*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{d} \sim P_{\mathbf{D}^{\star}}}\left[\mu_2\left(\mathbf{d}\right) \mathrm{Var}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right]\right]\right] \geq \\
&\mathop{\mathbb{E}}_{\mathbf{d}, \mathbf{z} \sim P_{\mathbf{D}^{\star}, \mathbf{Z}}}\left[\mu_1\left(\mathbf{d}, \mathbf{z}\right) \mathrm{Var}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^{\star}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right]\right].
\end{aligned}$$

Direct substitution of the condition in Assumption B.8 into Equations (10) and (11) completes the proof. □

In addition to Lemma B.6, the proof of Theorem 4.4 uses the following assumption, which is a variant of Assumption B.3 for the r-weighted case:

Assumption B.9 (Smoothness in parameter space)**.** *There exists some $\epsilon > 0$ such that*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{d}, \boldsymbol{\psi}_{n+1} \sim P_{\mathbf{D}^{\star}, \boldsymbol{\Psi_{n+1}}}}\left[\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1})\right]\right)\right] \geq \\
&\mathbb{E}_{\mathbf{d}, \boldsymbol{\psi}_{n+1} \sim P_{\mathbf{D}^{\star}, \boldsymbol{\Psi_{n+1}}}}\left[\left(\int_{N_{\epsilon}(\boldsymbol{\theta}^{\star})} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right) \log\left(L^{\mathcal{R}}(\mathbf{d}, \boldsymbol{\theta}^{\star}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1})\right)\right. \\
&\qquad\left. + \left(\int_{\mathscr{T} \setminus N_{\epsilon}(\boldsymbol{\theta}^{\star})} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right) \log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim P_{\boldsymbol{\Theta}}^{\mathscr{T} \setminus N_{\epsilon}(\boldsymbol{\theta}^{\star})}}\left[L^{\mathcal{R}}(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1})\right]\right)\right]
\end{aligned}$$

where $\left(\int_{N_{\epsilon}(\boldsymbol{\theta}^{\star})} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)$ and $\left(\int_{\mathscr{T} \setminus N_{\epsilon}(\boldsymbol{\theta}^{\star})} p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right)$ are the probability that a value $\boldsymbol{\theta}$ is inside and outside the ϵ-neighborhood of $\boldsymbol{\theta}^{\star}$, respectively.

Remark B.10. *In addition to the smoothness condition on the likelihood imposed by Assumption B.3, Assumption B.9 additionally imposes what is essentially a ceiling on the outputs of the relevance function. Weights < 1 "flatten", or smooth out, the likelihood function; weights > 1 "sharpen" it, and may cause violation of Assumption B.9 even in cases where Assumption B.3 is met. The relevance functions used in our examples (Section 5) output weights ≤ 1.*

We obtain

$$\begin{aligned}
\mathrm{IG}^{\mathcal{R}}\left(\boldsymbol{\theta}^{\star}\right) &= \mathop{\mathbb{E}}_{\mathbf{d}, \mathbf{z} \sim P_{\mathbf{D}^{\star}, \mathbf{Z}}}\left[\log\left(\frac{\mathbb{E}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^{\star}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right]}{\mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{\psi}'_{n+1} \sim P_{\boldsymbol{\Theta}, \boldsymbol{\Psi}_{n+1}}}\left[L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi}'_{n+1}\right)\right]}\right)\right] \\
\text{(Lemma B.6)} \quad &\leq \mathop{\mathbb{E}}_{\mathbf{d}, \mathbf{z} \sim P_{\mathbf{D}^{\star}, \mathbf{Z}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\psi}_{n+1} \sim P_{\boldsymbol{\Psi}_{n+1}|\mathbf{z}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^{\star}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right)\right]\right]
\end{aligned}$$

$$
\begin{aligned}
&- \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right)\right]\right] \\
=&\mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right)\right]\right] \\
&- \mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right)\right]\right] \\
=&\mathop{\mathbb{E}}_{\mathbf{d},\boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star,\boldsymbol{\Psi}_{n+1}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right)-\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right]\right)\right] \\
(\text{Assumption B.9}) \leq&\mathbb{E}_{\mathbf{d},\boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star,\boldsymbol{\Psi}_{n+1}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right)-\left(\int_{N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\log\left(L^{\mathcal{R}}(\mathbf{d},\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1})\right)-\right. \\
&\left.\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L^{\mathcal{R}}(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1})\right]\right)\right] \\
=&\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathbb{E}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\left(\log\left(L^{\mathcal{R}}\left(\mathbf{d},\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}\right)\right)\right.\right.\right. \\
&\left.\left.\left.-\log\left(\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}^{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}}\left[L^{\mathcal{R}}(\mathbf{d},\boldsymbol{\theta},\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1})\right]\right)\right)\right]\right] \\
=&\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathrm{H}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star),\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right. \\
&\left.-\,\mathrm{H}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right] \\
=&\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathrm{H}\left(P_{\mathbf{D}^\star}\right)+\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star),\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right. \\
&\left.-\mathrm{H}\left(P_{\mathbf{D}^\star}\right)-\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right] \\
=&\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star),\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right. \\
&\left.-\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}^\star,\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right] \qquad (12)
\end{aligned}
$$

as stated in the theorem for $A=\left(\int_{\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star)}p(\boldsymbol{\theta})\,d\boldsymbol{\theta}\right)$ and $C=\mathbb{E}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathrm{D}_{\mathrm{KL}}\left(P_{\mathbf{D}^\star}\,||\,P_{\mathbf{D}^{\mathcal{R}(\boldsymbol{\psi}_{n+1})}|\boldsymbol{\theta}\in\mathscr{T}\setminus N_\epsilon(\boldsymbol{\theta}^\star),\boldsymbol{\psi}=\boldsymbol{\psi}_{n+1}}\right)\right]$.

B.4 PROOF OF PROPOSITION 5.5

The proof depends on the following definition:

Definition B.11 (Fidelity of the relevance function $\rho^{\mathcal{R}}$)**.** *$\rho^{\mathcal{R}}$ is a measure of the fidelity of the relevance function. More specifically, it is:*

$$
\begin{aligned}
\rho^{\mathcal{R}}\equiv\mathbb{E}_{\mathbf{d},\boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star,\boldsymbol{\Psi}_{n+1}}}&\left[\frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{R}_i(\boldsymbol{\psi}_{n+1})-\frac{1}{n}\sum_{i=1}^{n}\mathcal{R}_i(\boldsymbol{\psi}_{n+1})\right)\right. \\
&\left.\left(\log\left(p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star,\boldsymbol{\psi}_i=\boldsymbol{\psi}_{n+1}\right)\right)-\frac{1}{n}\sum_{i=1}^{n}\log\left(p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star,\boldsymbol{\psi}_i=\boldsymbol{\psi}_{n+1}\right)\right)\right)\right],
\end{aligned}
$$

i.e., is the covariance of $\mathcal{R}_i(\boldsymbol{\psi}_{n+1})$ *and* $\log\left(p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right)$ *with respect to a uniform distribution over the source data, in expectation over* $P_{\mathbf{D}^\star, \boldsymbol{\Psi}_{n+1}}$.

$\Delta^{\mathcal{R}}$ can be rewritten as

$$
\begin{aligned}
\Delta^{\mathcal{R}} &= \mathop{\mathbb{E}}_{\boldsymbol{\psi}_{n+1}\sim P_{\boldsymbol{\Psi}_{n+1}}}\left[\mathop{\mathbb{E}}_{\mathbf{d}\sim P_{\mathbf{D}^\star}}\left[\log\left(\frac{L\left(\mathbf{d}, \boldsymbol{\theta}^\star, \boldsymbol{\psi}^\star\right)}{L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^\star, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)}\right)\right]\right] \\
&= -\mathop{\mathbb{E}}_{\mathbf{d}, \boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star, \boldsymbol{\Psi}_{n+1}}}\left[\log\left(L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}^\star, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right)\right] - \mathrm{H}\left(P_{\mathbf{D}^\star}\right) \\
&= -\mathop{\mathbb{E}}_{\mathbf{d}, \boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star, \boldsymbol{\Psi}_{n+1}}}\left[\sum_{i=1}^{n}\mathcal{R}_i(\boldsymbol{\psi}_{n+1})\log\left(p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right)\right] - \mathrm{H}\left(P_{\mathbf{D}^\star}\right) \\
&= \mathop{\mathbb{E}}_{\mathbf{d}, \boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star, \boldsymbol{\Psi}_{n+1}}}\left[\left(\sum_{i=1}^{n}\mathcal{R}_i(\boldsymbol{\psi}_{n+1})\right)\left(-\sum_{i=1}^{n}\log\left(p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right)\right)\right] - n\rho^{\mathcal{R}} - \mathrm{H}\left(P_{\mathbf{D}^\star}\right) \\
&= \mathop{\mathbb{E}}_{\mathbf{d}, \boldsymbol{\psi}_{n+1}\sim P_{\mathbf{D}^\star, \boldsymbol{\Psi}_{n+1}}}\left[\left(\sum_{i=1}^{n}\mathcal{R}_i(\boldsymbol{\psi}_{n+1})\right)\left(-\log\left(p\left(\mathbf{d}|\boldsymbol{\theta}^\star, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\right)\right)\right] - n\rho^{\mathcal{R}} - \mathrm{H}\left(P_{\mathbf{D}^\star}\right) \qquad \text{(Assumption 1.2)}
\end{aligned}
$$

as stated in the proposition for $D = -\mathrm{H}\left(P_{\mathbf{D}^\star}\right)$.

Remark B.12. *As discussed in Section 3.2, in practice the learner can often specify a sufficiently high-fidelity relevance function even in the absence of knowledge of* $\boldsymbol{\theta}^\star$*, i.e., a relevance function for which* $\rho^{\mathcal{R}}$ *is sufficiently large. An example relevance function is given in Equation (7). However, this relevance function is not guaranteed to positively correlate with* $p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$*. If the learner is particularly unlucky, this relevance function could have a negative corresponding value of* $\rho^{\mathcal{R}}$*, i.e., increase the relevance of source data points* least *likely under a particular pseudo-intervention. This may occur if* $\boldsymbol{\theta}$ *and* $\boldsymbol{\psi}$ *interact such that the direction of the gradient of predictions with respect to* $\boldsymbol{\psi}$ *depends on* $\boldsymbol{\theta}$*. For example, consider a case in which for all except very few values of* $\boldsymbol{\theta}$*, outcomes increase as a function of* $\boldsymbol{\psi}$*. In the context of our motivating example of treatment effect estimation, this might correspond to a situation where hospital quality generally increases the relative effectiveness of a treatment, unless the treatment effect is very extreme (in which case hospital quality has a larger impact on the effect of a placebo). If the true treatment effect is in fact very extreme, the relevance function shown in Equation (7) would likely negatively correlate with* $p\left(\mathbf{d}_i|\boldsymbol{\theta}^\star, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)$.

C DETAILS OF EXAMPLES

We here report the details of the examples described in Section 5. Appendix C.1 gives details of the linear regression example, Appendix C.2 gives details of the example predicting smoking behavior, and Appendix C.3 gives details of the GP regression example and results showing the relative performance of PROMPT as a function of the values of each of several simulation parameters.

C.1 LINEAR REGRESSION

All simulations were run using only a CPU. In all simulations, the value of the shared parameter $\boldsymbol{\theta}^\star = -1$. The prior $P_{\boldsymbol{\Theta}, \boldsymbol{\Psi}_i} = \mathcal{N}\left([0, 0]^\top, \mathrm{diag}\left([1, 1]\right)\right)$ for all $i \in 1 : n + 1$.

To generate source data, we first specified a particular level of multicollinearity ρ. A higher degree of multicollinearity makes $\boldsymbol{\theta}^\star$ and $\boldsymbol{\psi}^\star_{n+1}$ harder to separately identify, so we interpret this as a higher risk of negative transfer. We varied ρ among 0 (no multicollinearity), 1 (mild multicollinearity), and 2 (extreme multicollinearity).

For a given value ρ we sampled values $\mathbf{x}' \sim \mathcal{N}\left(\rho, .25\right)$, and then constructed values $\mathbf{x}_{(\cdot,1)} \sim \mathcal{N}\left(\mathbf{x}', .25\right)$ and values $\mathbf{x}_{(\cdot,2)} \sim \mathcal{N}\left(-\frac{\rho^2}{\mathbf{x}'}, .25\right)$. We created 100 such data points. Twenty-five of these data points were used to create proxy information (i.e., used to generate values $\mathbf{z}_i$ as described below), and 75 were used as outcome information on the basis of which to estimate the shared parameter.

In the simulations shown in Figure 2a, all proxy values are uncontaminated. In the simulations shown in Figure 2b, $\rho = 2$ always, i.e., all simulations are run in the presence of extreme multicollinearity.

Relevance function. We first computed the relevances as $\mathcal{R}_i(\boldsymbol{\psi}_{n+1}) \propto p\left(\mathbf{d}_i|\boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right) = \mathbb{E}_{\boldsymbol{\theta}\sim P_{\boldsymbol{\Theta}}}\left[p\left(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right]$ where the constant of proportionality was the probability a distribution with the same variance would assign to its mode. Using the calculated relevances, we computed $P^{\mathcal{R}}_{\boldsymbol{\Theta},\boldsymbol{\Psi}_{n+1}|\mathbf{d},\mathbf{z}}$. We then defined $\widehat{P}^{\mathcal{R}}_{\boldsymbol{\Theta}}$ as a Gaussian approximation to samples from the r-weighted posterior $P^{\mathcal{R}}_{\boldsymbol{\Theta}|\mathbf{d},\mathbf{z}}$, recomputed each $\mathcal{R}_i(\boldsymbol{\psi}_{n+1}) \propto \mathbb{E}_{\boldsymbol{\theta}\sim\widehat{P}^{\mathcal{R}}_{\boldsymbol{\Theta}}}\left[p\left(\mathbf{d}_i|\boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)\right]$, and recomputed the r-weighted posterior. In each simulation, we repeated this three times before ultimately defining the relevance function as an expectation across the distribution $\widehat{P}^{\mathcal{R}}_{\boldsymbol{\Theta}}$ obtained at the final iteration.

Proxy information. When $q\%$ of proxy values are contaminated, $1-q\%$ of proxy values are generated as $\mathbf{z} \sim \text{Binomial}\left(7, \tilde{p}(\mathbf{d}'|\boldsymbol{\psi}' = \boldsymbol{\psi}^{\star}_{n+1})\right)$ where $\mathbf{d}'$ are observations used to prompt the synthetic expert for feedback, $\boldsymbol{\psi}'$ are the corresponding task parameters, and $\tilde{p}$ indicates that the probability has been normalized to not exceed 1. The remaining $q\%$ of proxy values are generated as $\mathbf{z} \sim \text{Binomial}\left(7, 1-\tilde{p}(\mathbf{d}'|\boldsymbol{\psi}' = \boldsymbol{\psi}^{\star}_{n+1})\right)$.

C.2 PREDICTING SMOKING BEHAVIOR

All computations were run using only a CPU. The prior for all effects in both the classic and r-weighted models was $\mathcal{N}(0, 3)$.

The classic Bayesian learner estimated the fixed effects model

$$\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\psi} \sim \text{Binomial}\left(\text{sigmoid}\left(\boldsymbol{\theta}\mathbf{x}^{\top}_{i,(1:4)} + \boldsymbol{\psi}\mathbf{x}^{\top}_{i,(5:28)}\right), N_i\right)$$

where $\mathbf{x}_{i,(1:4)}$ are indicators of the treatment received, $\mathbf{x}_{i,(5:28)}$ are study indicators, and N_i is the number of patients who received the indicated treatment in the indicated study. The classic learner's estimate of the study indicator for the target task conditioned on the proxy information, generated as described in the main text, and their estimate of $(\boldsymbol{\theta}, \boldsymbol{\psi})$ used standard Bayesian updating to condition on the source data.

The r-weighted Bayesian learner estimated the model

$$\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\psi}_{n+1} \sim \text{Binomial}\left(\text{sigmoid}\left(\boldsymbol{\theta}\mathbf{x}^{\top}_{i,(1:4)} + \boldsymbol{\psi}_{n+1}\right), N_i\right)$$

The r-weighted learner's estimate of $\left(\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)$ used the following proxy-informed r-weighted likelihood of the 23 source data points $p^{\mathcal{R}}\left(\mathbf{d}, \mathbf{z}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right)$:

$$\begin{aligned} p^{\mathcal{R}}\left(\mathbf{d}, \mathbf{z}|\boldsymbol{\theta}, \boldsymbol{\psi}_{n+1}\right) &= L^{\mathcal{R}}\left(\mathbf{d}, \boldsymbol{\theta}, \boldsymbol{\psi} = \boldsymbol{\psi}_{n+1}\right)\ p\left(\mathbf{z}|\boldsymbol{\psi}_{n+1}\right) \\ &= p\left(\mathbf{z}|\boldsymbol{\psi}_{n+1}\right) \prod_{i=1}^{n} p\left(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}, \boldsymbol{\psi}_i = \boldsymbol{\psi}_{n+1}\right)^{\mathcal{R}_i(\boldsymbol{\psi}_{n+1})}. \end{aligned}$$

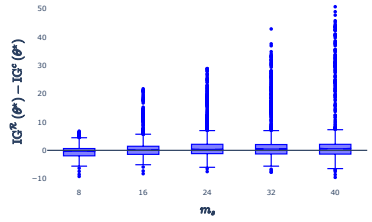
Proxy information. To simulate proxy information, we sampled $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\psi}^{\star}_{n+1}, \sigma\right) + \mathbb{1}_{noisy}\epsilon$ where $\mathbb{1}_{noisy}$ indicates whether proxy contamination is present and $\epsilon \sim \mathcal{N}(0, 3)$ is the bias added to contaminate the proxy information. Since we do not know the true value $\boldsymbol{\psi}^{\star}_{n+1}$, we approximated $\boldsymbol{\psi}^{\star}_{n+1}$ by the mean of the corresponding fixed effect distribution estimated in a model that incorporated data from all 24 studies.

When proxy information is *weakly informative*, $\sigma = 3$ and $\mathbb{1}_{noisy} = 0$. When proxy information is *highly informative*, $\sigma = .1$ and $\mathbb{1}_{noisy} = 0$. When proxy information is *misleading*, $\sigma = 3$ and $\mathbb{1}_{noisy} = 1$. The value of $\mathbb{1}_{noisy}$ is unknown to the learner, who always models the proxy information as completely uncontaminated (i.e., as if $\mathbb{1}_{noisy} = 0$).

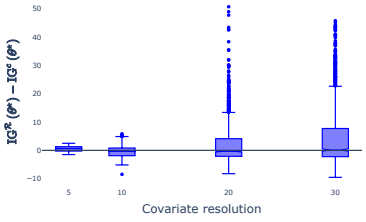
C.3 GAUSSIAN PROCESS REGRESSION

Each simulation was run on a single Nvidia A100 GPU.[2] The priors were $P_{\boldsymbol{\Theta}} = \text{Lognormal}(1, 1)$ and $P_{\boldsymbol{\Psi}_i} = \text{Gamma}(3, .8)$ for all $i \in 1 : n+1$.

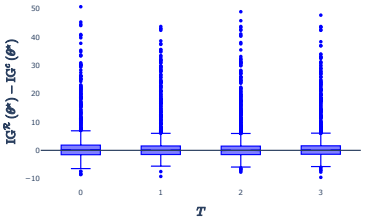
[2]The set of simulations run under 36 sets of simulation parameters (.5% of all sets of simulation parameters) did not complete successfully. For an additional 22 sets of simulation parameters (.3% of the total number of all sets of simulation parameters), all simulations encountered runtime errors.



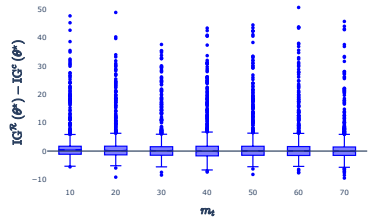
(a) Amount of source data. The advantage is more pronounced when more source data and proxy information is available. See interpretation in the text.



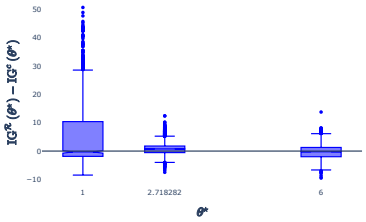
(b) Covariate resolution. The advantage is more pronounced for higher covariate resolutions. See interpretation in the text.



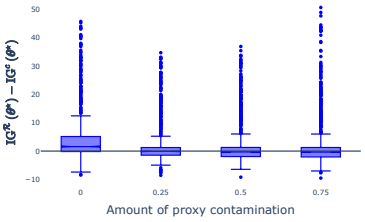
(c) Number of iterations for refinement of the relevance function T.



(d) Amount of target information.



(e) Value of $\boldsymbol{\theta}^\star$. The advantage is more pronounced for lower values of $\boldsymbol{\theta}^\star$. See interpretation in the text.



(f) Amount of proxy contamination. See interpretation in the text of Section 5.2.

Figure 4: Advantage of learning with an r-weighted likelihood in the GP regression setting as a function of the simulation parameter indicated in the subfigure caption. Each box in the plot shows the interquartile region (boxes) and outliers (points), across all values of all other simulation parameters, of the mean of $\mathrm{IG}^{\mathcal{R}}(\boldsymbol{\theta}^\star) - \mathrm{IG}^{c}(\boldsymbol{\theta}^\star)$ across 50 simulations.

For each simulation, we generated 80 trajectories drawn from a GP of the form given in the main text. From these 80 trajectories, trajectories $1 : m_t$ were generated from the target task, where the *amount of target information* m_t was a variable simulation parameter (see below). Trajectories $(m_t + 1) : 80$ were generated under a task parameter sampled at random from the learner's prior. Since these trajectories comprise most of the source data (see discussion of the effect of m_t below), the learner's prior is in most cases relatively well-specified. In this sense, the results in Figure 4 are a somewhat conservative test of PROMPT.

Trajectories $1 : m_s$ were then used to create synthetic proxy information, while trajectories $(80 - m_s) : 80$ were used for estimation of the target parameter (i.e., as source data), where the *amount of source data* m_s was a variable simulation parameter (see below). Proxy information was generated in the same way as for the linear regression example (see Appendix C.1).

The relevance function was computed using the same method described in Appendix C.1, with the exception that the number of iterations T used for refinement of the relevance function was a variable simulation parameter.

After observing that results were affected by the value of some simulation parameters, we varied these parameters across simulations. We varied the following simulation parameters:

- **Amount of source data** m_s**:** This parameter, which took values in $\{8, 16, 24, 32, 40\}$, controlled the number of trajectories in the source data. A distinct set of the same number of trajectories was used to create synthetic proxy information. Notice that the number of trajectories in the source data always equals the number of trajectories used to create proxy information, i.e., when more source data is available more proxy information is also available.
- **Covariate resolution:** Trajectories were evaluated on a grid of evenly-spaced values $\mathbf{x}$ ranging from 0 to 1. This parameter, which took values in $\{5, 10, 20, 30\}$, controlled the resolution and size of that grid.
- **Number of iterations for refinement of the relevance function** T**:** This parameter, which took values in $\{0, 1, 2, 3\}$,

controlled the number of iterations used for refinement of the relevance function.

- **Amount of target information m_t:** This parameter, which took values in $\{10, 20, 30, 40, 50, 60, 70\}$, controlled the number of trajectories generated by the target task. Notice that observations from the target task are almost exclusively used to create synthetic proxy information (the exception is when $m_t > 80 - m_s$, in which case the source data contains $m_t + m_s - 80$ trajectories from the target task). This reflects that the learner uses proxy information *instead of* direct observations from the target task (i.e., instead of fine-tuning in the target task). The amount of target information m_t to a certain extent admits a parallel interpretation as the number of trajectories a learner with both *knowledge of the data sources* and *the ability to fine-tune* (neither of which are available to the learners in our setting) would have to adapt to the target task, i.e., as the cost of operating in the setting of unknown data sources.
- **Value of $\boldsymbol{\theta}^\star$:** We set $\boldsymbol{\theta}^\star$ to either 1 (left tail of $P_{\boldsymbol{\Theta}}$), e (mode of $P_{\boldsymbol{\Theta}}$), or 6 (right tail of $P_{\boldsymbol{\Theta}}$).
- **Amount of proxy contamination:** We generated and contaminated synthetic proxy information in the same way as described for the linear regression example in Section 5.1. This parameter, which took values in $\{0, .25, .5, .75\}$, controlled the fraction of proxy values which were contaminated.

Figure 4 shows how the relative performance of PROMPT depends on the value of each of the simulation parameters listed above.

Figure 4a shows that PROMPT's advantage is more pronounced when more source data and proxy information are available. In other words, when the source data is sparse, the classic learner performs on par with the r-weighted learner. We speculate that this result reflects that in cases of source data sparsity both methods gain equally little information about $\boldsymbol{\theta}^\star$. Figure 4b shows a similar effect of the informativeness of the source data: PROMPT's advantage is more pronounced for higher covariate resolutions. When the covariate resolution is low, the covariates are relatively far apart and so all observations will be relatively uncorrelated regardless of the value of the shared and task parameters. In these cases, observations provide little information about the smoothness of the underlying function. Like in cases of source data sparsity, we speculate that this result reflects that in cases of disparate observations both methods gain equally little information about $\boldsymbol{\theta}^\star$.

Figure 4e shows that PROMPT's advantage is more pronounced for smaller values of $\boldsymbol{\theta}^\star$. This may be because of the effect of the value of $\boldsymbol{\theta}^\star$ on the threat of negative transfer: Values of $\boldsymbol{\psi}_i^\star$ tend to be large (the distribution from which source task parameters are drawn is right-skewed), and the learner partially attributes the effect of a larger bandwidth in the task-specific component of the kernel to the shared component of the kernel. When the bandwidth of the shared component of the kernel is small, the result is negative transfer. In this sense, Figure 4e corroborates the result shown in Figure 2a that r-weighting is especially effective in the presence of the threat of negative transfer.

Bibliography

Ibrahim Alabdulmohsin, Nicole Chiou, Alexander D'Amour, Arthur Gretton, Sanmi Koyejo, Matt J. Kusner, Stephen R. Pfohl, Olawale Salaudeen, Jessica Schrouff, and Katherine Tsai. Adapting to latent subgroup shifts via concepts and proxies. In *Proceedings of the* 26^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, 2023.

Robert A. Becker. The variance drain and jensen's inequality, 2012. Accessed via https://caepr.indiana.edu/RePEc/inu/caeprp/caepr2012-004.pdf.

Miheer Dewaskar, Christopher Tosh, Jeremias Knoblauch, and David B. Dunson and. Robustifying likelihoods by optimistically re-weighting data. *Journal of the American Statistical Association*, 2025.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction. In *The Seventh International Conference on Learning Representations (ICLR 2019)*, 2019.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *The Sixth International Conference on Learning Representations (ICLR 2018)*, 2018.

Peter Grünwald. Safe learning: bridging the gap between bayes, mdl and statistical learning theory via empirical convexity. In *24th Annual Conference on Learning Theory*, 2011.

Joseph G. Ibrahim and Ming-Hui Chen. Power prior distributions for regression models. *Statistical Science*, 15(1), 2000.

Joseph G. Ibrahim, Ming-Hui Chen, and Debajyoti Sinha. On optimality properties of the power prior. *Journal of the American Statistical Association*, 98, 2011.

Joseph G. Ibrahim, Ming-Hui Chen, Yeongjin Gwon, and Fang Chen. The power prior: theory and applications. *Statistics in Medicine*, 34, 2014.

Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 2014.

Jeffrey W. Miller and David B. Dunson. Robust bayesian inference via coarsening. *Journal of the American Statistical Association*, 114(527), 2019.

Yasmine Nahal, Janosch Menke, Julien Martinelli, Markus Heinonen, Mikhail Kabeshov, Jon Paul Janet, Eva Nittinger, Ola Engkvist, and Samuel Kaski. Human-in-the-loop active learning for goal-oriented molecule generation. *Journal of Cheminformatics*, 16, 2024.

Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, Michael O'Boyle, and Amos Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Barbara Plank and Gertjan van Noord. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011.

Natalia Ponomareva and Mike Thelwall. Biographies or blenders: Which resource is best for cross-domain sentiment analysis? In *Computational Linguistics and Intelligent Text Processing (CICLing 2012)*, 2012.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. *Advances in Domain Adaptation Theory*. Elsevier, 2019.

Robert Remus. Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis. In *IEEE 12th International Conference on Data Mining Workshops*, 2012.

Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Ingmar Schuster and Ilja Klebanov. Markov chain importance sampling—a highly efficient estimator for MCMC. *Journal of Computational and Graphical Statistics*, 30, 2021.

Sabina J. Sloman, Ayush Bharti, Julien Martinelli, and Samuel Kaski. Bayesian active learning in the presence of nuisance parameters. In *Proceedings of the* 40^{th} *Conference on Uncertainty in Artificial Intelligence (UAI 2024)*, 2024.

J. Michael Steele. *The Cauchy-Schwarz Master Class*. Cambridge University Press, 2004.

Piotr M. Suder, Jason Xu, and David B. Dunson. Bayesian transfer learning, 2023. Accessed via https://arxiv.org/abs/2312.13484.

Iiris Sundin, Tomi Peltola, Luana Micallef, Homayun Afrabandpey, Marta Soare, Muntasir Mamun Majumder, Pedram Daee, Chen He, Baris Serim, Aki Havulinna, Caroline Heckman, Giulio Jacucci, Pekka Marttinen, and Samuel Kaski. Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge. *Bioinformatics*, 2018.

Eric J. Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An Introduction to Proximal Causal Inference. *Statistical Science*, 2024.

Katherine Tsai, Stephen R Pfohl, Olawale Salaudeen, Nicole Chiou, Matt Kusner, Alexander D'Amour, Sanmi Koyejo, and Arthur Gretton. Proxy methods for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135, 2022.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018.