Informed Meta-Learning

Katarzyna Kobalczyk [1] **Mihaela van der Schaar** [1]

Abstract

In noisy and low-data regimes prevalent in real-world applications, a key challenge of machine learning lies in effectively incorporating inductive biases that promote data efficiency and robustness. Meta-learning and informed ML stand out as two approaches for incorporating prior knowledge into ML pipelines. While the former relies on a purely data-driven source of priors, the latter is guided by prior domain knowledge. In this paper, we formalise a hybrid paradigm, *informed meta-learning*, facilitating the incorporation of priors from unstructured knowledge representations, such as natural language; thus, unlocking complementarity in cross-task knowledge sharing of humans and machines. We establish the foundational components of informed meta-learning and present a concrete instantiation of this framework—the Informed Neural Process. Through a series of experiments, we demonstrate the potential benefits of informed meta-learning in improving data efficiency, robustness to observational noise and task distribution shifts.

1. Introduction

Designing machine learning models that generalise well requires finding a balance between two properties of a model: its support and inductive biases that guide the learning algorithm towards specific solutions (Wilson & Izmailov, 2020). Currently, the field favors models with large support, such as deep neural networks—universal approximators capable of fitting any continuous function (Hornik et al., 1989). However, these universal architectures lack task-specific inductive biases, demanding vast amounts of training data (Welling, 2019). In practical scenarios, where collecting a substantial training dataset is often impractical, human expertise becomes indispensable. Drawing from past experiences and contextual knowledge beyond observable data, experts may offer helpful insights about inductive biases suitable for the task at hand.

Conventionally, equipping ML models with prior knowledge about the learning task has been a manual process executed by practitioners based on their own knowledge and intuition or in collaboration with domain experts. The subfield of **informed machine learning** focuses specifically on designing models with explicit inductive biases derived from formal knowledge representations (von Rueden et al., 2023b). Such formal representations may take various forms including mathematical expressions (Karpatne et al., 2017; Qian et al., 2021), simulation results (Rai et al., 2019; Shen et al., 2020), knowledge graphs (Choi et al., 2017; Zhang et al., 2019), logic rules (Yang et al., 2023; Richardson & Domingos, 2006), or spatial invariances (Wu et al., 2018; Bogatskiy et al., 2020).

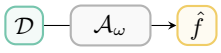
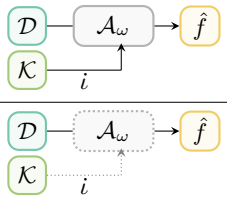
Despite many successes, informed ML methods are limited by researchers' abilities to comprehend and formalise expert knowledge and design a learning method accordingly. Hand-crafted bias specification methods offer control and explainability but demand extensive engineering and communication efforts between domain experts and ML practitioners. While some inductive biases can be easily encoded (e.g. with convolutions or custom loss functions), preferences over competing hypothesis can often be challenging to formalise and manually integrate into ML methods; with the integration step often forming the core contribution of ML papers (Goyal & Bengio, 2020).

Meta-learning is an alternative approach to inductive bias specification imitating human systematic generalisation based on previously solved tasks. This paradigm involves learning from a distribution of related tasks, aka environment, allowing the learner to acquire prior knowledge suited for solving new tasks. Yet, defining task relatedness lacks clarity and often depends on the particular algorithm and the surrounding context beyond the observable data. While there exist theoretical guarantees of successful knowledge transfer and generalisation (Baxter, 2000; Guan & Lu, 2022), they rely on the assumption that training and test tasks come from the same distribution, which is hard to meet in practice. If the environment changes, model performance often drops significantly (Chen et al., 2019; Li et al., 2019).

[1]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK. Correspondence to: Katarzyna Kobalczyk <knk25@cam.ac.uk>.

Accepted by the MHFAIA workshop of ICML 2024, Vienna, Austria. Copyright 2024 by the author(s).

Table 1. Comparison of ML paradigms. Solid lines in the overview diagrams represent fixed components of the ML pipeline. Dashed lines represent meta-learned components. Different learning approaches are distinguished by the following factors: (1) Inductive biases of a model depend on a formal representation of knowledge. (2) Inductive biases of the learning algorithm are learned from a pre-defined task distribution. (3) Inductive biases can be controlled by the domain expert knowledge post model training.

Method		Overview	Prototypes	(1)	(2)	(3)
Conv.	Sup.		NN, BNN, GP	✗	✗	✗
Conv.	Meta		ProtoNet, MAML, NP	✗	✓	✗
Informed	Sup.		PINN, BN, G-CNN	✓	✗	✗
Informed	Meta		INP (Ours)	✓	✓	✓

Recognising the functional flexibility of meta-learning methods in acquiring inductive biases of various forms, we propose a new learning paradigm, *informed meta-learning*, in which the knowledge integration process is meta-learned by training on a distribution of learning tasks and their respective knowledge representations. The complementarity of informed ML and meta-learning is twofold:

- ▶ In contrast to conventional informed ML, the process of knowledge integration is not fixed but meta-learned based on the previously observed tasks and their corresponding knowledge representations, enabling the integration of prior knowledge represented with varying levels of formalism, including natural language.
- ▶ Formal knowledge representations condition the task and thus inform about task similarity, mitigating the adverse effects of task distribution shifts and heterogeneity.

Contributions: This paper introduces the paradigm of informed meta-learning, a novel approach to inductive bias specification utilising human knowledge in various forms, including unstructured natural language. We present the Informed Neural Process as a proof of concept, empirically demonstrating the advantages of informed meta-learning in enhancing data efficiency and robustness to observational noise, task distribution shifts, and task heterogeneity. While previous studies have integrated informed ML with meta-learning in specific domains like physics (Belbute-Peres et al., 2021; Qin et al., 2022; Huang et al., 2022), our work for the first time formalises the generalised paradigm, remaining agnostic to knowledge representation formats. Our unique contributions lies in proposing to leverage meta-learning for a seamless integration of human knowledge in its native formats into machine learning algorithms.

2. Formalising Informed Meta-Learning

We begin by formally describing two supervised learning approaches: meta-learning and informed ML, before introducing the concept of informed meta-learning. Table 1 provides a condensed overview of the methodological distinctions between these learning paradigms. Refer to Appendix A.3 for a further comparisons with related work.

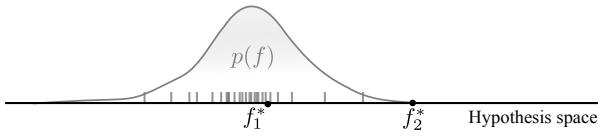
In a supervised setup, we are given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ and aim to fit a predictive function $\hat{f} : \mathcal{X} \to \mathcal{Y}$, or in a probabilistic setting, find a posterior distribution over functions $p(f \mid \mathcal{D})$. The learning algorithm, $\mathcal{A}$, is a mapping from the observation space to the hypothesis space, $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{F}, \mathcal{D} \mapsto \hat{f}$ (stochastic learners can be treated by assuming a distribution-valued $\mathcal{A}$). In this view, $\mathcal{A}$ is seen as all steps leading to the final hypothesis $\hat{f}$, including i.a. data pre-processing, model architecture, loss function, and regularisation.

Meta-learning. The efficacy of $\mathcal{A}$ relies on the "how to learn" aspect—the inductive biases induced by all of its components. Conventionally, these are fixed and pre-specified. Meta-learning is an alternative approach, in which parts of the learning algorithm are learned by maximising the expected performance of $\mathcal{A}$ on a distribution of related tasks $p(\mathcal{T})$. Each task $\mathcal{T}$ is defined by a data distribution on $\mathcal{X} \times \mathcal{Y}$, from which a task dataset $\mathcal{D}$ is sampled. We let ω denote the learnable component of the algorithm and the dependence of $\mathcal{A}$ on ω by a subscript $\mathcal{A}_\omega$. To find a good choice of ω, we assume access to a collection of meta-training tasks, $\{\mathcal{T}_j\}_{j \in \mathcal{J}}$, $\mathcal{T}_j \sim p(\mathcal{T})$ and frame "learning how to learn" as an optimisation problem (Hospedales et al., 2022):

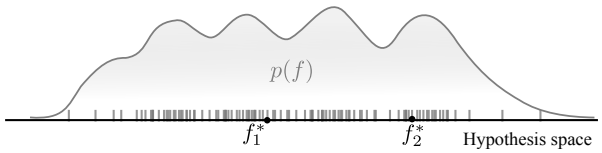
$$\omega^* = \arg\max_{\omega} p\left(\omega \mid \{\mathcal{T}_j : j \in \mathcal{J}\}\right). \tag{1}$$

ω^* is commonly referred to as the meta-knowledge or cross-task knowledge. Once chosen, ω^* remains the same for solving any task $\mathcal{T} \sim p(\mathcal{T})$ and solutions are obtained with the learner $\mathcal{A}_{\omega^*}$, as $\hat{f} = \mathcal{A}_{\omega^*}(\mathcal{D})$. Popular examples of meta-learning methods include MAML (Finn et al., 2017), where ω is the initialisation of a neural network or prototypical networks (ProtoNet (Sung et al., 2018)) where ω stands for an embedding function shared among all tasks. Importantly, the meta-learned representation ω is mostly data-driven and dependent on the selection of the meta-training tasks $\{\mathcal{T}_j\}_{j \in \mathcal{J}}$. Consequently, the influence of in-domain expert knowledge on model's inductive biases remains minimal.

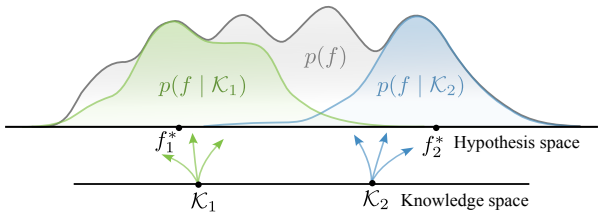
Informed Machine Learning. In contrast, in informed ML, the learning algorithm is explicitly dependent on expert knowledge given by a formal representation, which we denote as $\mathcal{K}$. Alongside the dataset $\mathcal{D}$, knowledge $\mathcal{K}$ creates an additional and independent source of information that is explicitly integrated into the learning algorithm. Unlike



(a) Concentrated meta distribution provides strong inductive biases, as long as the task of interest belongs to the same environment as the training tasks.



(b) Heterogeneous meta distribution supports a wider range of tasks, at the cost of weaker inductive biases.



(c) Informed meta-learning conditions the meta-distribution on prior knowledge $\mathcal{K}$. In result, the meta-distribution concentrates around the regions of the hypothesis space surrounding the true solution, providing strong inductive biases.

Figure 1. Probabilistic perspective on (informed) meta-learning (see appx. A.4).

empirical data, which may be noisy, **knowledge, is a form of true and already validated information. Its meaning with respect to the learning task is a priori known by the modeller.** The correctness of knowledge and its relevance provide grounding for the ML method, often leading to an improved performance over purely data-driven models. Let ω denote the part of the learning algorithm in which knowledge $\mathcal{K}$ is integrated, then we may represent this explicit dependence of $\mathcal{A}$ on $\mathcal{K}$ via a $\mathcal{A}_\omega$, where $\omega = i(\mathcal{K})$. Here i is a loosely defined map from prior knowledge $\mathcal{K}$ to a part of the learning algorithm. The knowledge integration process, i, is conventionally performed by the machine learning practitioner. Representations of $\mathcal{K}$ and ω manifest in various forms. In physics-informed neural networks (PINNs) (Raissi et al., 2019) knowledge is represented by algebraic equations and integrated with the means of an additional regularisation term; in Bayesian Networks (BNs) (Constantinou et al., 2016) knowledge is represented by probabilistic relations between random variables, constraining the hypothesis space; knowledge represented by invariances may define new model architectures like, e.g. group-equivariant CNNs (G-CNNs) (Cohen & Welling, 2016).

Informed meta-learning. Both approaches of meta-learning and informed ML aim to integrate a form of prior knowledge into the learning algorithm. In meta-learning, this prior knowledge is defined by a collection of meta-training tasks $\{\mathcal{T}_j\}_{j\in\mathcal{J}}$. In informed ML, it is given by a formal representation $\mathcal{K}$. Informed meta-learning takes a hybrid approach in which the process of prior knowledge integration, i, is meta-learned based on a collection of training tasks $\{\mathcal{T}_j\}_{j\in\mathcal{J}}$ and their corresponding knowledge representations $\{\mathcal{K}_j\}_{j\in\mathcal{J}}$. As previously, let ω represent the part of the learning algorithm in which prior knowledge is integrated, then informed meta-learning reduces to:

$$i^* = \arg\max_i p(i \mid \{\mathcal{T}_j, \mathcal{K}_j : j \in \mathcal{J}\}), \tag{2}$$

where the solution to a learning task $\mathcal{T}$ and its corresponding knowledge representation $\mathcal{K}$ is obtained via $\mathcal{A}_{\omega^*}$ with $\omega^* = i^*(\mathcal{K})$. Precise formulations of the objective in (2) depend both on ω and i. Section 3 introduces one particular instantiation of informed meta-learning, based on a probabilistic approach to meta-learning. Below, we elaborate on our motivation behind this particular choice.

Perhaps the most popular meta-learning framework is that of gradient-based optimisation (Finn et al., 2017) with the goal of finding an optimal weight initialisation ($\omega = \theta_0$) for a model $h(\cdot\,;\theta)$ with parameters $\theta \in \Theta$. This involves estimating θ_0 based on the training tasks, where the solution for each task is obtained by a few steps of gradient descent with respect a loss function $\mathcal{L}$ evaluated on the task-specific dataset $\mathcal{D}$; i.e., $\mathcal{A}_\omega(\mathcal{D}) = h(\cdot\,;\omega - \alpha\nabla\mathcal{L}(\mathcal{D}))$. An alternative approach to meta-learning is that of inductive bias learning, as originally formalised by Baxter (1997). This method aims to choose a suitable prior p_θ over the parameter space Θ from a predefined collection of priors using hierarchical Bayesian inference. While seemingly distinct, gradient-based meta-learning can be recast under the hierarchical Bayesian framework (Grant et al., 2018). We thus view the probabilistic approach as more general.

Abstracting away model parameterisation details[†], the goal of meta-learning is then to learn a suitable prior over the hypothesis space, $p(f)$, based on the training tasks $\{\mathcal{T}_j\}_{j\in\mathcal{J}}$. In this view, the meta-learned ω represents the learned prior and the stochastic learner $\mathcal{A}_\omega$ maps a task-specific dataset $\mathcal{D}$ to the posterior $p(f \mid \mathcal{D})$. Figures 1(a), 1(b) illustrate the idea of learning a prior via sampling a fixed number of tasks from the environment. It also highlights the trade-offs between well-concentrated and heterogeneous meta distributions.

In the context of informed meta-learning, probabilistic approaches may prove particularly advantageous. On one hand, such methods enable the sampling of multiple solutions, spanning a region of the hypothesis space, $\mathcal{F}$, instead of returning a single MLE estimate. On the other hand, we posit that expert knowledge $\mathcal{K}$ is often conceptual in nature and thus corresponds to entire regions of the hypothesis space rather than precise solutions $\hat{f} \in \mathcal{F}$. For instance, if $\mathcal{K}$ requires that the fitted function is linear, this specification

corresponds to a subset $\mathcal{F}_{\text{linear}} \subseteq \mathcal{F}$. This observation motivates our focus on probabilistic methods while establishing a first instantiation of an informed meta-learner. Given a collection of training tasks and knowledge representations, we will aim to find a suitable prior $p(f)$, and learn to condition it on various forms of knowledge (see Fig. 1(c)).

3. Informed Neural Processes

The family of Neural Processes (Garnelo et al., 2018a;b) is one particular example of probabilistic, amortised meta-learners, forming the foundation for our informed meta-learner. We choose NPs as they reduce the cost of learning to a feed-forward operation, eliminating the need for expensive gradient-based optimisation. NPs offer functional flexibility, being suited to both regression and classification tasks. Moreover, the fact that NPs model a distribution over functions, instead of returning a single, MLE prediction enables us to measure the reduction in uncertainty about solutions given observed data, and in the informed meta-learning scenario, reduction of uncertainty given expert knowledge.

Setup. Let $\mathcal{T}$ represent a learning task consisting of a context $\mathcal{D}_C = \{(x_i, y_i)\}_{i=1}^{n}$ and target $\mathcal{D}_T = \{(x_i, y_i)\}_{i=n+1}^{m}$ data sets, aka training and validation sets. We assume that data are generated according to the following process. Let $p(f)$ be a probability distribution over functions f, formally known as a stochastic process, then for $f \sim p(f)$, set $y_i = f(x_i) + \epsilon_i$, where ϵ_i stands for the observational noise. Given a collection of training tasks, $\{\mathcal{T}_j\}_{j\in\mathcal{J}}$, and their corresponding knowledge representations, $\{\mathcal{K}_j\}_{j\in\mathcal{J}}$, our goal is to train a model that makes predictions on unlabelled target data points, given a small sample of context points for any new task generated according to $p(f)$.

Neural Process. NPs model the distribution over functions f through a fixed dimensional latent variable z sampled from a variational distribution q. That is, each sample $z \sim q$ corresponds to one realisation of the stochastic process. NPs model the predictive posterior distribution as:

$$p(y \mid x, \mathcal{D}_C) := p(y \mid x, r_C) := \int p(y \mid x, z) q(z \mid r_C) dz. \tag{3}$$

The variable r_C is an aggregation of all observation in $\mathcal{D}_C$, $r_C = \frac{1}{|C|}\sum_{i\in C} h(x_i, y_i)$. The variational distribution $q(z \mid r)$ is taken as Normal, $q(z \mid r) = \mathcal{N}(z; \mu_z, \sigma_z)$, with $(\mu_z, \sigma_z) = r$. In the case of regression, we will assume normal observational noise, i.e. $p(y \mid x, z) = \mathcal{N}(y; \mu_y, \sigma_y)$, with $(\mu_y, \sigma_y) = g(x, z)$, where g is a decoder network. In this view, the meta-knowledge, ω, can be represented with the tuple $\omega = (g, h)$. During meta-training, NPs estimate the prior distribution over all functions, $p(f)$, and the conditionals $p(f \mid \mathcal{D}_C)$. Learning a single task corresponds to computing the posterior $p(f \mid \mathcal{D}_C)$, which is obtained with a single forward pass through networks h and g. The parameters of these two networks are estimated by episodic training over a distribution of tasks.

[†]Let Θ be the parameter space of a model, with solutions defined by $f = h(\cdot\,; \theta)$ for a specific choice of $\theta \in \Theta$. If p_θ is a prior distribution over Θ, then the distribution over f can be defined as follows. Let $g(\theta)(\cdot) := h(\cdot\,; \theta)$, then $f = g(\theta) \sim g_*(p_\theta)$, where $g_*(p_\theta)$ is the push-forward of p_θ defined by the measurable function $g : \Theta \to \mathcal{F}$, a mapping from parameters to functions that is assumed to be measurable.

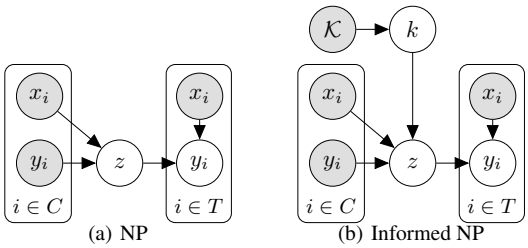


Figure 2. Graphical models. Comparison of NPs with INPs. Dark grey nodes represent the observables.

Informed Neural Process. As discussed in section 2, external knowledge about a given learning task should allow for concentrating the mass of $p(f)$ around the region of functions coherent with that knowledge. To achieve this, we condition the variational distribution q on $\mathcal{K}$ and model the predictive distribution as:

$$p(y \mid x, \mathcal{D}_C, \mathcal{K}) := \int p(y \mid x, z) q(z \mid \mathcal{D}_C, \mathcal{K}) dz. \tag{4}$$

From the implementation point of view, similarly to NPs, INPs are also constructed with two networks: g and h. However, in INPs, the outputs of h that parameterise the variational distribution, q, are dependent on expert knowledge $\mathcal{K}$. Connecting the concepts from section 2, we have that $\omega = i(\mathcal{K}) = (g, h(\cdot\,; \mathcal{K}))$. In our implementation of INPs, the fusion of knowledge with data is realised with $h(\cdot\,; \mathcal{K}) = a(h_1(\cdot), h_2(\mathcal{K}))$, where h_1 and h_2 represent data and knowledge encoding networks, respectively, and a is an aggregation operator. For precise implementation details refer to appx. A.1. If we let $r_C = \frac{1}{|C|}\sum_{i\in C} h_1(x_i, y_i)$, $k = h_2(\mathcal{K})$ and $r'_C = a(r_C, k)$, INPs model (4) as:

$$p(y \mid x, \mathcal{D}_C, \mathcal{K}) := p(y \mid x, r_C, k) \tag{5}$$

$$= p(y \mid x, r'_C) = \int p(y \mid x, z) q(z \mid r'_C) dz. \tag{6}$$

As in (Kim et al., 2019), if no data has been observed, we can set the global data representation, r_C, to a zero vector, approximating the prior distribution of f under expert knowledge, $p(f \mid \mathcal{K})$. Similarly, the conditioning on expert knowledge can also be omitted by setting $k = 0$, resulting in a purely data-driven, uninformed prediction $p(f \mid \mathcal{D}_C)$.

Training. INPs are trained in an episodic fashion over a distribution of learning task $\mathcal{T}_j$ and their associated knowledge

representations $\mathcal{K}_j$. To train and evaluate an INP model we sample training, validation and testing collections of tasks. Each task, (omitting the dependence on j for clarity), consists of a labeled context dataset $\mathcal{D}_C$. and the target dataset $\mathcal{D}_T$. The labels of the target dataset are the goal of each prediction task. Denoting by r_C and r_T the context and target data representations and by k the knowledge embedding vector of a single task, parameters of the model are learned by maximising the expectation of ELBO over all training tasks,

$$\log p(y_T|x_T, r_C, k) \geq \mathbb{E}_{q(z|r_T,k)} \left[\log p(y_T \mid x_T, z)\right] \quad (7)$$
$$-D_{\text{KL}} \left(q(z \mid r_T, k) \mid\mid q(z \mid r_C, k)\right). \quad (8)$$

At each training iteration, the number of context and target data points are chosen randomly. We also randomly mask knowledge representations by setting $k = 0$. This allows for the possibility of knowledge being missing at test time. Further details on the derivation and estimation of the ELBO loss can be found in appx. A.2.

4. Experiments

The experimental section is divided into two parts. First, we anchor the key ideas of informed meta-learning on illustrative experiments with synthetic data, where knowledge representations are well-structured and there exists an analytic, closed-form expression linking knowledge with the true data generating process (DGP). This serves to illustrate the potential benefits of informed ML in terms of data efficiency, uncertainty reduction, and robustness, and how these can be measured. In the second part, we showcase possible applications on real-world data where the underlying DGP is unknown and knowledge may be loosely formatted, particularly, presented in natural language. Full experimental details are presented in appx. A.2.

4.1. Part I: Illustrative experiments

4.1.1. Data efficiency and task distribution shift

Setup 1.: For each task, context, and target data points are sampled according to the following process. A function f is sampled from the family of sinusoidal functions with a linear trend and bias, $f(x) = ax + \sin(bx) + c$, for some randomly sampled values of the parameters a, b, c. We introduce a Gaussian observational noise, s.t. $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 0.2)$. The parameters a, b, c are randomly sampled according to: $a \sim U[-1, 1]$, $b \sim U[0, 6]$, $c \sim U[-1, 1]$. We let $\mathcal{K}$ encode the value of two, one or none ($\mathcal{K} = \varnothing$) of the parameters a, b, or c. The number of context points n ranges uniformly between 0 and 10; the number of targets is set to $m = 100$. This setup simulates a scenario, in which $\mathcal{K}$ contains partial, incomplete information about f. By training over distribution of tasks $\mathcal{T}$, we expect the model to learn how to put a strong prior on the function's slope, level of oscillations and bias.

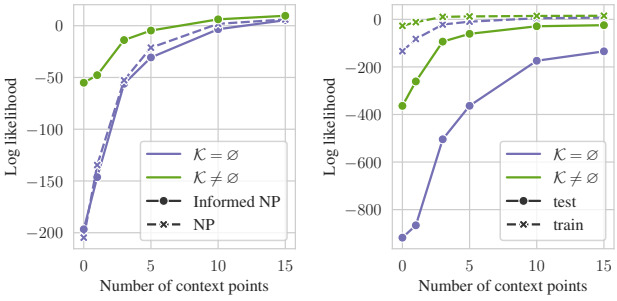


Figure 3. Average log likelihood vs. number of context points. Left: Comparison of plain NPs and INPs. Knowledge integration enhances data efficiency. Right: Performance under distribution shift between meta-training and testing tasks. Knowledge integration reduces the performance gap between training and testing tasks.

Fig. 3 (left) shows the estimated log-likelihood on the test tasks against the number of context data points for both the original NP model and the INP. Results for INP are shown with knowledge presented at test time ($\mathcal{K} \neq \varnothing$) and when it is omitted ($\mathcal{K} = \varnothing$). We observe that informing our model significantly improves predictions. As the number of context points decreases, the performance gap between raw and informed predictions increases. Moreover, under $\mathcal{K} = \varnothing$, our implementation of INPs performs on par with vanilla NPs. Thus, the ability to condition the prior on expert knowledge is not at the cost of reduced performance of purely data-driven predictions.

To summarise the impact of knowledge on the predictive performance of INPs we compute the relative ΔAUC score defined as the integral of the "Δ-likelihood against n" (von Rueden et al., 2023a), where "Δ-likelihood" is defined as: $p(\mathcal{D}_T|\mathcal{D}_C, \mathcal{K})$ - $p(\mathcal{D}_T|\mathcal{D}_C)$. We report relative values with respect to the AUC of the uninformed predictions.Fig. 4 shows the estimated ΔAUC depending on which of the parameters a, b, or c have their values revealed at test time. Intuitively, exposing more information about f should provide the model with stronger priors; thus, simplifying the learning problem. As expected, when $|\mathcal{K}| = 2$ the performance gains are larger than when $|\mathcal{K}| = 1$. Figure A.2 in the appendix shows qualitatively the impact of knowledge on predictions and its integration with observed data; knowledge provides information about the global behavior of sampled functions while individual data points anchor the predictions in the x-y plane.

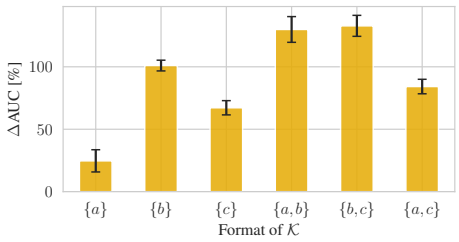


Figure 4. Average relative improvement of informed predictions vs. uninformed predictions by knowledge format.

> **Setup 2.:** Performance of meta-learners often drops drastically in the presence of a distribution shift between training and testing tasks (Chen et al., 2019). In this experiment, we simulate a distribution shift of this type. Keeping everything else equal as in setup 1., for the training tasks, we sample $b \sim \mathcal{N}(2, 1)$, and for testing tasks we let $b \sim \mathcal{N}(3, 1)$. We let $\mathcal{K}$ encode the true value of b.

Fig. 3 (right) shows how the performance gap between training and testing tasks is significantly reduced upon informing the model about the true value of b, which is the source of the distribution shift between training and testing tasks.

> **Take-aways:** The first experiment illustrates how a successful integration of oracle knowledge about the learning task, impacts data efficiency. In cases when such knowledge is not available, we show that INP does not fall short of the purely data-driven NP. We also demonstrate how data efficiency gains depend on the type of information carried by $\mathcal{K}$, and how this can be measured quantitatively. Finally, we illustrate how oracle knowledge about the given learning task may prove useful in mitigating the adverse effects of distribution mismatch between training and testing tasks.

4.1.2. Knowledge and Uncertainty Reduction

NPs, chosen as the foundation for our informed meta-learner, possess a key feature: the capability to sample from the solution space, instead of providing a single point estimate. This enables us to measure the decrease in model uncertainty when incorporating expert knowledge. Our focus is primarily on measuring epistemic uncertainty—the uncertainty stemming from a lack of knowledge about the true relationship between model inputs and outputs, rather than the inherent randomness of the process.

We compute the predictive uncertainty as the conditional entropy, $\mathbb{H}[y^* \mid x^*, \mathcal{K}]$, at a specific location $x^* \in \mathcal{X}$ and knowledge $\mathcal{K}$. Predictive uncertainty is measured in the observation space and therefore amounts for the uncertainty

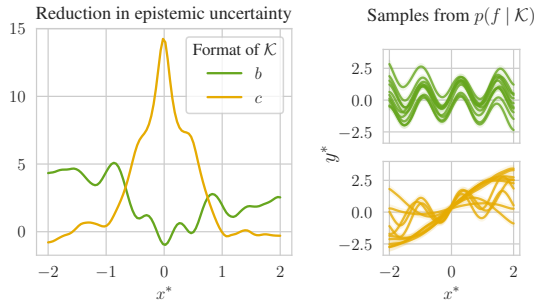


Figure 5. Reduction in epistemic uncertainty across the entire input range. $\mathcal{K}$ represents the exact values of parameters b or c (in this case $b = 2.5$ and $c = -0.5$). Revealing the value of the parameter c, is equivalent to proving the information about the value that f takes at the origin. This is represented by a spike in the uncertainty reduction at $x^* = 0$. In contrast, revealing the value of b provides information about global characteristics of f, rather than local; no significant spikes in uncertainty reduction are observed.

associated with observational noise. However, we can decompose it as a sum:

$$\underbrace{\mathbb{I}(y^*, f \mid x^*, \mathcal{K})}_{\text{epistemic}} + \underbrace{\mathbb{E}_{f \sim p(f \mid \mathcal{K})}[\mathbb{H}[y^* \mid x^*, f]]}_{\text{aleatoric}}$$

and approximate the predictive and aleatoric uncertainties with MC samples. The epistemic uncertainty is then obtained as the difference of the two quantities (see appx. A.5 for more details). With the same setup as in section 4.1.1, we look at the reduction in epistemic uncertainty given a knowledge, i.e. $\mathbb{I}(y^*, f \mid x^*) - \mathbb{I}(y^*, f \mid x^*, \mathcal{K})$. We compute this metric for all values $x^* \in [-2, 2]$. By averaging over the entire input range we obtain a single-valued metric of the impact of $\mathcal{K}$ on the uncertainty about f. Figures 5 and 6 summarize our observations.

> **Take-away:** This example captures how different information represented by $\mathcal{K}$ can impact the inductive biases of a model. By introducing knowledge, the distribution of a priori likely functions concentrates (c.f. Figure 1(c)), leading to reduction in model uncertainty. The method of INPs enables us to quantify the magnitude of this effect.

4.2. Part II: Real data and loosely formatted knowledge

For illustrative purposes, representations of knowledge in the previous part were highly structured and with a well-defined relationship to the underlying DGP. Nautraly, in such scenarios we would resolve to direct knowledge integration methods (e.g. Bayesian linear regression). However, the advantages of informed meta-learning become evident when: a) the functions to be learned lack a known, closed-form expression; b) knowledge about the learning

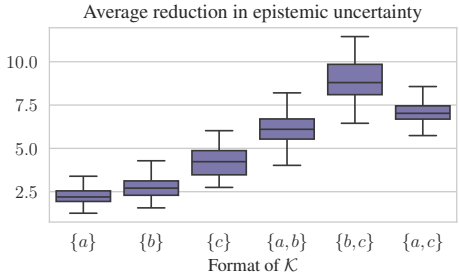


Figure 6. Reduction in epistemic uncertainty averaged across the input range. Knowledge that represents two values of the parameters, results in a greater reduction in uncertainty than knowledge about just a single parameter. Due to the strong, local effect of c in determining the value of f at the origin, knowledge formats including c have a larger impact on uncertainty reduction.

task is loosely formatted, making manual integration of prior knowledge a significant challenge.

4.2.1. Informed Weather Predictions

Setup: We use the sub-hourly temperature dataset from the U.S. Climate Reference Network, representing values of the air temperature measured at regular 5-minute intervals. For each task, target observations are uniformly sampled from a 24h time range. Context data points are selected by sub-sampling at most 10, chronologically first samples. This setup enables us to assess extrapolation. We perform independent experiments with two formats of knowledge :

A: For each task, knowledge $\mathcal{K}$ is a vector encoding two values: the minimum temperature and the maximum temperature on the day.

B: For each task, knowledge $\mathcal{K}$ is a synthetically generated "weather forecast" presented in natural language. We generate these with GPT-4 (OpenAI et al., 2023) prompted to write two sentences mimicking a weather forecast, based on values from the ground truth temperature measurements.

Table 2. Relative performance gap (%) between informed and uninformed predictions. Numbers in brackets represent the standard errors of the estimates based on 110 testing tasks.

	$n=0$	$n=1$	$n=3$	$n=5$	$n=10$	Δ AUC
A	57.7 (1.4)	21.2 (1.0)	18.8 (1.0)	15.0 (0.8)	2.9 (0.4)	23.0 (0.8)
B	48.1 (2.1)	15.6 (1.2)	14.8 (1.0)	11.1 (0.8)	2.1 (0.4)	17.0 (0.7)

Fig. 7 shows representative examples of the daily temperature paths from test tasks alongside purely data-driven and informed predictions. NPs capture the general trend of the temperature rising during the day, and then falling down towards the night, but unsurprisingly, fail to accurately extrapolate beyond the observed regions. This is due to a high level of heterogeneity present in the collection of meta-

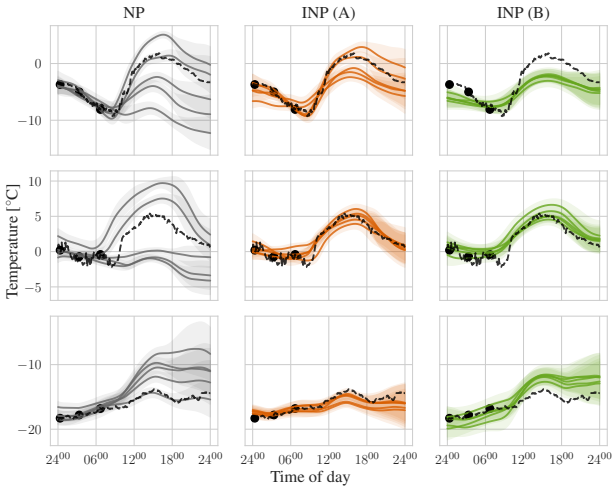


Figure 7. Sampled evolution paths of the temperature, given 3 context measurements. NP: raw predictions of the plain, uninformed neural process. INP (A): predictions with the informed neural process, given oracle knowledge about the minimum and maximum temperature on the day. INP (B): predictions with the informed neural process, given oracle knowledge about the temperature, presented in a text format (available in the Appendix A.2).

training tasks, which is reflected in the high variability of the sampled functions outside of the observed data range. In terms of the informed predictions, we observe that the information contained in $\mathcal{K}$ enables guided extrapolation beyond the observed range of values and reduces the variance of the sampled functions. Table 2 compares the performance gap between informed and uninformed predictions. Notably, knowledge enables sensible, 0-shot predictions with an average improvement in log-likelihood of 57.7% and 48.1% for setups A and B, respectively. We also note that the representation of knowledge, as presented in setup A should, in the theoretically optimal case, impose hard constraints on the maximum and minimum values of the function's range. However, given that INP is only a neural approximation of these constraints, the resulting curves may exceed the specified range as opposed to strictly adhering to it; as it could be possible with a custom-designed model that explicitly incorporates such constraints into its optimisation objective.

Take-away: In practical scenarios, predictive functions are difficult to model with closed-form mathematical expressions, making the process of external knowledge integration a challenging task. The benefit of neural, meta- approaches, is their functional flexibility. In particular, NPs can learn non-trivial 'kernels' from the collection of training tasks directly. INPs take this a step further, enabling the incorporation of non-trivially representable information about the underlying function into the model.

4.2.2. Informed Image Classification

Setup: We apply INPs to few-shot classification on the CUB-200-2011 dataset (Wah et al., 2011). We use 100 bird categories for training, 50 for validation, and 50 for testing and follow the standard N-way, k-shot classification setup. We adjust the INP architecture to suit the image classification task, employing CLIP vision and text encoders (Fu et al., 2022) (details in appx. A.2.3). We perform independent experiments with three formats of knowledge:

A: Knowledge represents features of a given bird class, e.g. wing span, feather color. Class-level attributes are obtained by averaging the attribute vectors associated with each image from the dataset. Class-level attribute vectors are stacked together to obtain $N \times 312$ tensors.

B: Knowledge represents class-level textual descriptions of the N classes obtained by averaging sentence embedding of individual image captions belonging to the given class. We use human-generated captions as collected in (Reed et al., 2016) and embed them with CLIP. Per-class averaged text embeddings are then stacked to form a $N \times 512$ tensor.

C: Here knowledge represents a set of N individual descriptions of each class. We generate these with GPT-4 based on the captions from B (see appx. A.2.3 for examples).

Table 3 shows results for 5-way and 10-way classification. Across all settings, we observe higher classification accuracy when additional knowledge is utilised. This trend holds for 1, 3, 5, and 10-shot tasks, with the performance gap widening as the number of shots decreases. The information about characteristic elements of each class contained in $\mathcal{K}$ proves sufficient for relatively good zero-shot prediction performance. While the zero-shot performance for setup C is lower than that of setups A or B, it is nevertheless significantly higher than the accuracy of random guessing.

Take-away: INPs align the representations of images and knowledge about class-specific features to construct latent representations that contain the essential multi-modal information. This alignment facilitates robust generalisation to new, previously unseen classes, enabling both zero-shot classification and improved few-shot classification accuracy.

Table 3. Accuracy (%) on N-way, k-shot classification tasks for the CUB-200-2011 dataset. Numbers in brackets represent the standard errors of the estimates based on 60 tasks per each setting. Individual tasks are constructed with only previously unseen bird categories.

N	k	NP	INP (A)	INP (B)	INP (C)
5	0	–	87.5 (0.7)	87.4 (0.5)	50.3 (0.6)
	1	82.2 (0.6)	88.1 (0.6)	89.1 (0.5)	85.1 (0.5)
	3	87.0 (0.5)	88.4 (0.6)	89.3 (0.5)	88.3 (0.5)
	5	88.1 (0.5)	88.5 (0.6)	89.6 (0.5)	88.9 (0.4)
	10	88.5 (0.5)	88.5 (0.6)	89.6 (0.5)	89.0 (0.4)
10	0	–	81.1 (0.4)	78.5 (0.4)	33.7 (0.3)
	1	73.3 (0.4)	82.2 (0.4)	81.9 (0.4)	77.0 (0.5)
	3	79.8 (0.4)	82.7 (0.4)	82.7 (0.4)	81.8 (0.4)
	5	81.5 (0.4)	82.8 (0.4)	82.8 (0.4)	83.0 (0.4)
	10	82.6 (0.4)	82.7 (0.4)	83.0 (0.4)	84.1 (0.4)

5. Discussion

Limitations. Our work primarily focuses on key principles of informed meta-learning; the introduced class of INP models serves as a foundational proof of concept within this paradigm. As discussed by the previous works on NPs (Garnelo et al., 2018a;b; Kim et al., 2019) these models tend to underfit observed data and may not easily scale to high dimensional problems. We also note that good performance of INPs relies on the model accurately interpreting the true meaning of knowledge. This is, however, contingent on the number of available training tasks, lack of spurious correlations, and knowledge complexity (see Appendix A.6.1-A.6.3). Future work should explore new informed meta-learning architectures improving learning efficiency, disentanglement of latent knowledge representations, and generalisation.

Broader impact. In this paper, we have laid the groundwork for a novel ML framework of inductive bias specification based on expert knowledge expressible with varying degrees of formalism. Importantly, this paradigm facilitates the incorporation of knowledge articulated in natural language—a primary medium of communication between domain experts and ML practitioners. While it is important to ensure that the integrated information is accurate and free from harmful or unfair biases, we believe that informed meta-learning, supported by increasingly expressive language models, can offer significant benefits in applications where knowledge is conveyed verbally or in writing, such as in textbooks or scientific articles. This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here

References

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. Label-Embedding for Image Classification. *CoRR*, abs/1503.08677, 2015. URL http://arxiv.org/abs/1503.08677. arXiv: 1503.08677.

Al-Halah, Z., Tapaswi, M., and Stiefelhagen, R. Recovering the Missing Link: Predicting Class-Attribute Associations for Unsupervised Zero-Shot Learning. *CoRR*, abs/1610.04787, 2016. URL http://arxiv.org/abs/1610.04787. arXiv: 1610.04787.

Annervaz, K. M., Chowdhury, S. B. R., and Dukkipati, A. Learning beyond datasets: Knowledge Graph Augmented Neural Networks for Natural language Processing. *CoRR*, abs/1802.05930, 2018. URL http://arxiv.org/abs/1802.05930. arXiv: 1802.05930.

Baxter, J. A Bayesian/Information Theoretic Model of Learning to Learn via Multiple Task Sampling. *Machine Learning*, 28(1):7–39, July 1997. ISSN 1573-0565. doi: 10.1023/A:1007327622663. URL https://doi.org/10.1023/A:1007327622663.

Baxter, J. A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*, 12:149–198, March 2000. ISSN 1076-9757. doi: 10.1613/jair.731. URL http://dx.doi.org/10.1613/jair.731. Publisher: AI Access Foundation.

Belbute-Peres, F. d. A., Chen, Y.-f., and Sha, F. HyperPINN: Learning parameterized differential equations with physics-informed hypernetworks, 2021. _eprint: 2111.01008.

Bogatskiy, A., Anderson, B., Offermann, J., Roussi, M., Miller, D., and Kondor, R. Lorentz Group Equivariant Neural Network for Particle Physics. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 992–1002. PMLR, July 2020. URL https://proceedings.mlr.press/v119/bogatskiy20a.html.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A Closer Look at Few-shot Classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkxLXnAcFQ.

Choi, E., Bahadori, M. T., Song, L., Stewart, W. F., and Sun, J. GRAM: Graph-Based Attention Model for Healthcare Representation Learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pp. 787–795, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098126. URL https://doi.org/10.1145/3097983.3098126. eventplace: Halifax, NS, Canada.

Choi, K., Cundy, C., Srivastava, S., and Ermon, S. LMPriors: Pre-Trained Language Models as Task-Specific Priors. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. URL https://openreview.net/forum?id=U2MnmJ7Sa4.

Cohen, T. and Welling, M. Group Equivariant Convolutional Networks. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, New York, USA, June 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Constantinou, A. C., Fenton, N., and Neil, M. Integrating expert knowledge with data in Bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. *Expert Systems with Applications*, 56:197–208, 2016. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.02.050. URL https://www.sciencedirect.com/science/article/pii/S0957417416300896.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1184–1193. PMLR, July 2018. URL https://proceedings.mlr.press/v80/depeweg18a.html.

Ding, C. and Tao, D. Robust Face Recognition via Multimodal Deep Face Representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, 2015. doi: 10.1109/TMM.2015.2477042.

Elhoseiny, M., Elgammal, A. M., and Saleh, B. Write a Classifier: Predicting Visual Classifiers from Unstructured Text. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2539–2553, 2017. doi: 10.1109/TPAMI.2016.2643667. URL https://doi.org/10.1109/TPAMI.2016.2643667.

Finn, C., Abbeel, P., and Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of*

the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, August 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Fu, J., Xu, S., Liu, H., Liu, Y., Xie, N., Wang, C.-C., Liu, J., Sun, Y., and Wang, B. CMA-CLIP: Cross-Modality Attention Clip for Text-Image Classification. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 2846–2850, 2022. doi: 10.1109/ICIP46576.2022.9897323.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. M. A. Conditional Neural Processes. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, July 2018a. URL https://proceedings.mlr.press/v80/garnelo18a.html.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. Neural Processes, 2018b. _eprint: 1807.01622.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative Adversarial Networks. *CoRR*, abs/1406.2661, 2014. URL http://arxiv.org/abs/1406.2661. arXiv: 1406.2661.

Goyal, A. and Bengio, Y. Inductive Biases for Deep Learning of Higher-Level Cognition. *CoRR*, abs/2011.15091, 2020. URL https://arxiv.org/abs/2011.15091. arXiv: 2011.15091.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting Gradient-Based Meta-Learning as Hierarchical Bayes, 2018. _eprint: 1801.08930.

Guan, J. and Lu, Z. Task Relatedness-Based Generalization Bounds for Meta Learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=A3HHaEdqAJL.

Hendrycks, D. and Gimpel, K. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415. arXiv: 1606.08415.

Ho, J. and Salimans, T. Classifier-Free Diffusion Guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/ARXIV.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598. arXiv: 2207.12598.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239. arXiv: 2006.11239.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. Meta-Learning in Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5149–5169, 2022. doi: 10.1109/TPAMI.2021.3079209. URL https://doi.org/10.1109/TPAMI.2021.3079209.

Huang, X., Ye, Z., Liu, H., Shi, B., Wang, Z., Yang, K., Li, Y., Weng, B., Wang, M., Chu, H., Yu, F., Hua, B., Chen, L., and Dong, B. Meta-Auto-Decoder for Solving Parametric Partial Differential Equations, 2022. _eprint: 2111.08823.

Iwata, T. and Kumagai, A. Sharing Knowledge for Meta-learning with Feature Descriptions. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=kyY4w4IgtM8.

Kaddour, J., Sæmundsson, S., and Deisenroth, M. P. Probabilistic Active Meta-Learning, 2020. _eprint: 2007.08949.

Karpatne, A., Watkins, W., Read, J., and Kumar, V. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2, 2017.

Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., and Teh, Y. W. Attentive Neural Processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkE6PjC9KX.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Li, B. Z., Chen, W., Sharma, P., and Andreas, J. LaMPP: Language Models as Probabilistic Priors for Perception and Action, 2023. _eprint: 2302.02801.

Li, Y., Yang, Y., Zhou, W., and Hospedales, T. Feature-Critic Networks for Heterogeneous Domain Generalization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3915–3924. PMLR, June 2019. URL https://proceedings.mlr.press/v97/li19l.html.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784. arXiv: 1411.1784.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal Deep Learning. In Getoor, L. and Scheffer, T. (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 689–696. Omnipress, 2011. URL https://icml.cc/2011/papers/399_icmlpaper.pdf.

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, , Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Kondraciuk, , Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Peres, F. d. A. B., Petrov, M., Pinto, H. P. d. O., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C. J., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. GPT-4 Technical Report, 2023. _eprint: 2303.08774.

Paz-Argaman, T., Atzmon, Y., Chechik, G., and Tsarfaty, R. ZEST: Zero-shot Learning from Text Descriptions using Textual Similarity and Visual Summarization. *CoRR*, abs/2010.03276, 2020. URL https://arxiv.org/abs/2010.03276. arXiv: 2010.03276.

Peng, J., Wang, X., and Shang, X. Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-Seq data. *BMC Bioinform.*, 20-S(8):284:1–284:12, 2019. doi: 10.1186/S12859-019-2769-6. URL https://doi.org/10.1186/s12859-019-2769-6.

Perez, E., Strub, F., Vries, H. d., Dumoulin, V., and Courville, A. C. FiLM: Visual Reasoning with a General Conditioning Layer. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th in-*

novative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 3942–3951. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11671. URL https://doi.org/10.1609/aaai.v32i1.11671.

Qian, Z., Zame, W. R., Fleuren, L. M., Elbers, P., and Schaar, M. v. d. Integrating Expert ODEs into Neural ODEs: Pharmacology and Disease Progression. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=tDqef76wFaO.

Qin, T., Beatson, A., Oktay, D., McGreivy, N., and Adams, R. P. Meta-PDE: Learning to Solve PDEs Quickly Without a Mesh, 2022. _eprint: 2211.01604.

Rai, A., Antonova, R., Meier, F., and Atkeson, C. G. Using Simulation to Improve Sample-Efficiency of Bayesian Optimization for Bipedal Robots. *J. Mach. Learn. Res.*, 20(1):1844–1867, January 2019. ISSN 1532-4435. Publisher: JMLR.org.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR*, abs/2204.06125, 2022. doi: 10.48550/ARXIV.2204.06125. URL https://doi.org/10.48550/arXiv.2204.06125. arXiv: 2204.06125.

Reed, S. E., Akata, Z., Schiele, B., and Lee, H. Learning Deep Representations of Fine-grained Visual Descriptions. *CoRR*, abs/1605.05395, 2016. URL http://arxiv.org/abs/1605.05395. arXiv: 1605.05395.

Richardson, M. and Domingos, P. Markov logic networks. *Machine Learning*, 62(1):107–136, February 2006. ISSN 1573-0565. doi: 10.1007/s10994-006-5833-1. URL https://doi.org/10.1007/s10994-006-5833-1.

Schönfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., and Akata, Z. Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8247–8255. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00844. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Schonfeld_Generalized_Zero-_and_Few-Shot_Learning_via_Aligned_Variational_Autoencoders_CVPR_2019_paper.html.

Shen, H., Jiang, Y., Li, T., Cheng, Q., Zeng, C., and Zhang, L. Deep learning-based air temperature mapping by fusing remote sensing, station, simulation and socioeconomic data. *Remote Sensing of Environment*, 240:111692, 2020. Publisher: Elsevier.

Shi, X., Mueller, J., Erickson, N., Erickson, N., Li, M., Smola, A., and Smola, A. Benchmarking Multimodal AutoML for Tabular Data with Text Fields. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021a. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/9bf31c7ff062936a96d3c8bd1f8f2ff3-Paper-round2.pdf.

Shi, X., Mueller, J., Erickson, N., Li, M., and Smola, A. Multimodal AutoML on Structured Tables with Text Fields. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021b. URL https://openreview.net/forum?id=OHAIVOOl7Vl.

Sohn, K., Lee, H., and Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 3483–3491, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html.

Song, Y. and Ermon, S. Generative Modeling by Estimating Gradients of the Data Distribution. *CoRR*, abs/1907.05600, 2019. URL http://arxiv.org/abs/1907.05600. arXiv: 1907.05600.

Srivastava, N. and Salakhutdinov, R. Multimodal Learning with Deep Boltzmann Machines. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems*

2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pp. 2231–2239, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to Compare: Relation Network for Few-Shot Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 1199–1208. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00131. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Sung_Learning_to_Compare_CVPR_2018_paper.html.

Tsai, Y.-H. H. and Salakhutdinov, R. Improving One-Shot Learning through Fusing Side Information. *CoRR*, abs/1710.08347, 2017. URL http://arxiv.org/abs/1710.08347. arXiv: 1710.08347.

Tseng, G., Kerner, H., and Rolnick, D. TIML: Task-Informed Meta-Learning for Agriculture, 2022. _eprint: 2202.02124.

von Rueden, L., Garcke, J., and Bauckhage, C. How Does Knowledge Injection Help in Informed Machine Learning? In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2023a. doi: 10.1109/IJCNN54540.2023.10191994.

von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., and Schuecker, J. Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2023b. doi: 10.1109/TKDE.2021.3079836.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*. California Institute of Technology, July 2011.

Welling, M. Do we still need models or just more data and compute. *University of Amsterdam, April*, 20, 2019.

Wilson, A. G. and Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., and Lin, H.-T. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/322f62469c5e3c7dc3e58f5a4d1ea399-Abstract.html.

Wu, J.-L., Xiao, H., and Paterson, E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids*, 3(7):074602, July 2018. doi: 10.1103/PhysRevFluids.3.074602. URL https://link.aps.org/doi/10.1103/PhysRevFluids.3.074602. Publisher: American Physical Society.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Broeck, G. V. d. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. *CoRR*, abs/1711.11157, 2017. URL http://arxiv.org/abs/1711.11157. arXiv: 1711.11157.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2048–2057. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/xuc15.html.

Yang, Y., Xiong, S., Payani, A., Shareghi, E., and Fekri, F. Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation, 2023. _eprint: 2305.15541.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. Deep Sets. In Guyon, I., Luxburg, U. v., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3391–3401, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://aclanthology.org/P19-1139.

A. Appendix

A.1. Architectural and training details for INPs

A.1.1. Model Architecture

The architecture of INPs consists of the following key components:

- A **data encoder**, $h_1 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ that takes in pairs (x_i, y_i) and produces an order-invariant representation $r = \sum_i h_1(x_i, y_i)$.
- A **knowledge encoder**, h_2, a map from the knowledge representation space to the latent space $\mathbb{R}^d$ that takes in the knowledge inputs $\mathcal{K}$ and extracts a latent knowledge vector $k = h_2(\mathcal{K})$.
- An **aggregator**, a, that combines the data representation, r, and the latent knowledge representation, k, into one representation, $r' = a(r, k)$, that parameterizes the latent distribution q. We take $q(z \mid r') = \mathcal{N}(z; \mu_z, \sigma_z)$, where $(\mu_z, \sigma_z) = r'$.
- A **conditional decoder**, g, that takes in samples of the global latent variable $z \sim q(z \mid r')$ and the new target location x^* to output the predictions parameterised by $p(y^* \mid x^*, z) = \mathcal{N}(y; \mu_y, \sigma_y)$, where $(\mu_y, \sigma_y) = r'$

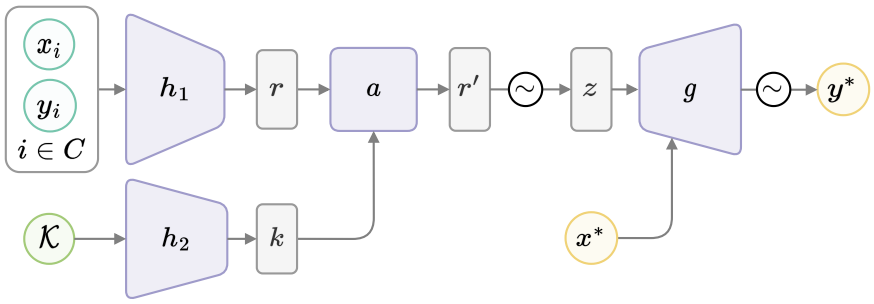


Figure A.1. Overview of the INP model architecture.

In all experiments any MLP is implemented with the GELU non-linearity (Hendrycks & Gimpel, 2016). We experiment with different forms of aggregation, a:

1. **sum & MLP**: $a(r, k) = \text{MLP}(r + k)$,
2. **concat & MLP**: $a(r, k) = \text{MLP}([r||k])$,
3. **MLP & FiLM**: $a(r, k) = \text{FiLM}(k)\,[\text{MLP}(r)]$. We use the idea of modulation parameters introduced by (Perez et al., 2018). Here a is an MLP whose parameters are modulated with a modulated with the outputs of h_2.

We find that in most cases, the first, least complex option performs the best.

A.1.2. Training

INPs are trained in an episodic fashion over a distribution of learning tasks consisting of context and target datasets, and associated knowledge representations. Denoting by r_C and r_T the context and target data representations and by k the knowledge embedding vector of a single task, we derive the evidence lower bound via:

$$p(y_T \mid x_T, r_C, k) = \int p(y_T | x_T, z) q(z \mid r_C, k) dz \tag{9}$$

$$= \int p(y_T \mid x_T, z) \frac{q(z \mid r_C, k)}{q(z \mid r_T, k)} q(z \mid r_T, k) dz \tag{10}$$

$$= \mathbb{E}_{q(z|r_T,k)} \left[p(y_T \mid x_T, z) \frac{q(z \mid r_C, k)}{q(z \mid r_T, k)} \right] \tag{11}$$

And therefore, by Jensen we obtain:

$$\log p(y_T \mid x_T, r_C, k) \geq \mathbb{E}_{q(z|r_T,k)} \left[\log p(y_T \mid x_T, z) \right] - D_{\text{KL}} \left(q(z \mid r_T, k) \,||\, q(z \mid r_C, k) \right) \tag{12}$$

The parameters of the model are learned by maximising the ELBO in (12) for randomly sampled batches of tasks. During training, we use one sample of $q(z \mid r_T, k)$ to form a MC estimate of the ELBO. For evaluation, we use 32 samples. Additionally, during training, we randomly mask knowledge by setting $k = 0$, the frequency of masking is a hyperparameter of the model.

A.2. Experimental details

Throughout all experiments we use the Adam optimizer (Kingma & Ba, 2015). During training, we use validation-based early stopping. All experiments were run on a machine with an AMD Epyc Milan 7713 CPU, 120GB RAM, and using a single NVIDIA A6000 Ada Generation GPU accelerator with 48GB VRAM.

A.2.1. 1-D SINUSOIDAL REGRESSION (SECTION 4.1)

For each task, context and target data points are sampled according to the following process. A function f is sampled from the family of sinusoidal functions with trend and bias, $f(x) = ax + \sin(bx) + c$. We also introduce a Gaussian observational noise, s.t. $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 0.2)$. The parameters a, b, c are randomly sampled according to: $a \sim U[-1, 1]$, $b \sim U[0, 6]$, $c \sim U[-1, 1]$. For each task, the context and target points are uniformly sampled from the range $[-2, 2]$. The number of context points n ranges uniformly between 0 and 10; the number of targets, $m = 100$. We let $\mathcal{K}$ to encode the value of two, one, or none ($\mathcal{K} = \varnothing$) of the parameters a, b, or c.

The data encoder, h_1, is implemented as a 3-layer MLP. The knowledge encoder, h_2, is implemented with the DeepSet architecture (Zaheer et al., 2017), made of two 2-layer MLPs. Each element of the set is represented by a one-hot encoding of the parameter type with its value appended at the end. The decoder is a 4-layer MLP. We set the hidden dimension, $d = 128$ and use the sum & MLP method for the aggregator, a. We use a learning rate of 1e-3 and set the batch size to 64. During training, knowledge is masked at rate 0.3.

In section 4.1 we use this setup to demonstrate and discuss the impact of expert knowledge on enhanced data-efficiency, reduction in uncertainty, and robustness to distribution shifts. Fig. A.2 shows sample predictions under 0, 1, or 3 observed data points and different formats of knowledge $\mathcal{K}$.

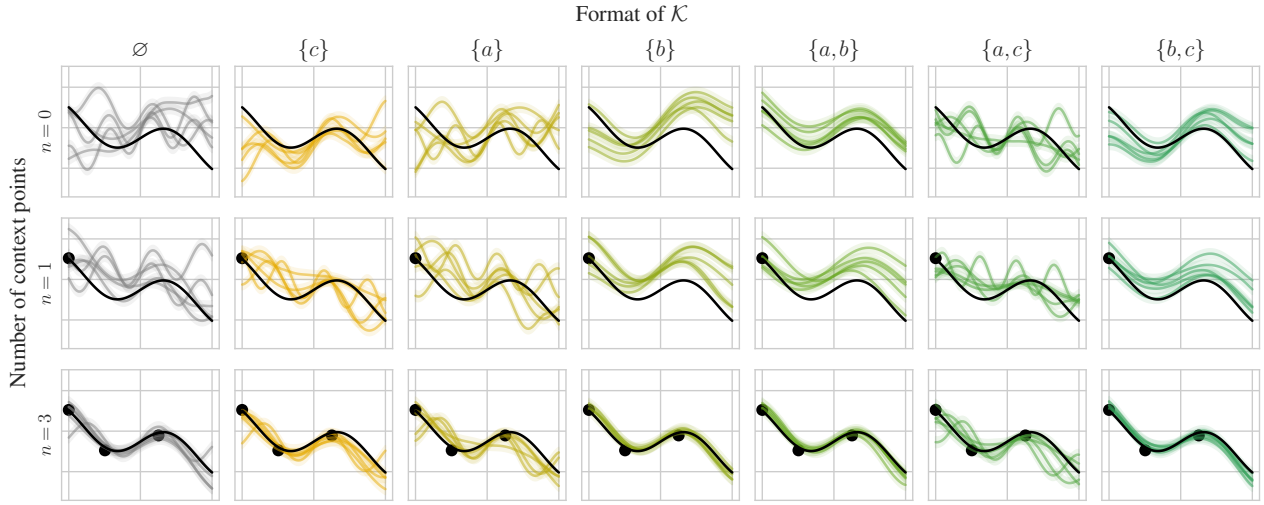


Figure A.2. Sample predictions under varying formats of knowledge. Knowledge about the value of the slope or frequency of oscillations provides global information about the overall shape of the function. Observing additional data points anchors the curves in the xy-coordinate system. Based on a qualitative investigation we conclude that the INP successfully learned how to integrate prior knowledge with observed data points.

A.2.2. Informed Weather Predictions (section 4.2.1)

We use the sub-hourly temperature dataset from the U.S. Climate Reference Network (USCRN)*. The data contains values of the air temperature measured at regular 5-minute intervals. For each task, the context and target datasets consist of measurements from one day. Training, validation, and testing collections of tasks are created by randomly selecting 507, 108, and 110 days, respectively, between the years 2021 and 2022 in Aleknagik, Alaska. For each task, the target dataset consists of all 288 measurements in the 24h range. Context observations are sampled by first uniformly sampling 10 data points and then selecting the chronologically first n observations where $n \sim U[0, 10]$. We perform independent experiments with two formats of knowledge:

A: For each task, knowledge $\mathcal{K}$ is a vector encoding two values: the minimum temperature and the maximum temperature on the day. In this setup, the knowledge encoder, h_2, is a simple 2-layer MLP .

B: For each task, knowledge $\mathcal{K}$ is a synthetically generated "weather forecast" presented in a natural language format. For illustrative purposes, these weather descriptions were generated with GPT-4 (OpenAI et al., 2023). In total, 726 descriptions, one per day were generated. The prompt used contains instructions to generate 2 sentences mimicking a weather forecast, based on 48 values sampled at 30-minute intervals from the ground truth temperature values. We use the following prompt:

> **System:** `You are given a vector of values representing the temperature for the next 24h at 30-minute intervals, starting at 12 am. Your task is to present the weather forecast according to these values. Keep it to max 2 sentences. Use descriptive words to refer to the times of the day, e.g. morning, afternoon, evening.`
>
> **User:** `<<Temperature values>>`

In this setup, the knowledge encoder h_2 is implemented with a RoBERTa language model (Liu et al., 2020) with all weights frozen except for the layer norm weights, which are tuned during the end-to-end training. The latent knowledge representation k is obtained as a pooled sentence embedding. Here, we use the last hidden state of the CLS token.

For both setups A and B, the data encoder h_1 is implemented as a 3-layer MLP and the decoder g as a 4-layer MLP. We used the MLP & FiLM aggregator a. We set the hidden dimension, $d = 128$. We use a learning rate of 1e-3 and set the batch size to 64. The knowledge representation is randomly masked at a rate 0.3 by setting $k = 0$. Vanilla NPs are known to underfit context observations and underestimate the variance, which became apparent with this more complex and noisy dataset. To mitigate this issue, in this experiment, we have employed multi-head cross-attention during the encoding of the data representation, r, as proposed by (Kim et al., 2019). Precisely, $r = \sum_{i=1}^{n} \text{Att}_i h_1(x_i, y_i)$, where $\text{Att} = \text{MultiHead}(Q, K, V)$, with Q being a matrix of target inputs, K a matrix of context inputs and V a matrix consisting of individual data representations $r_i = h_1(x_i, y_i)$. We use 4 attention heads.

See the main body of the paper for a discussion of the results. Figure A.3, shows sample tasks and their corresponding GPT-4 generated weather descriptions.

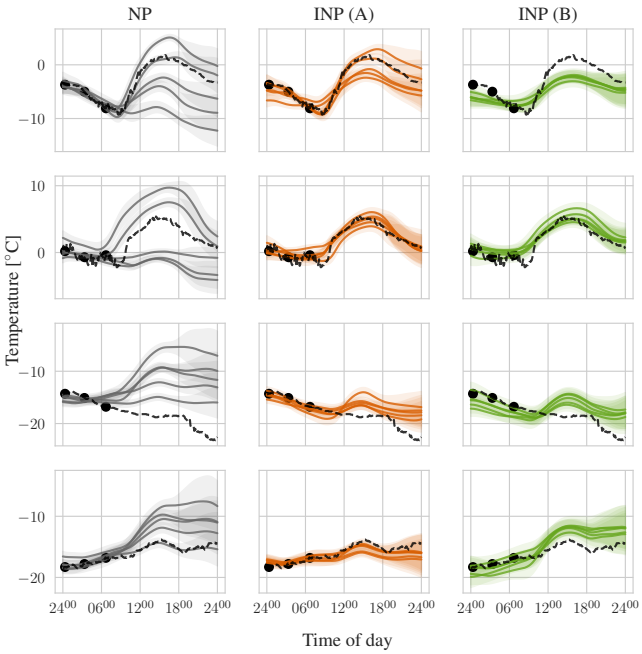
A.2.3. Few-shot and zero-shot image classification with CUB-200-2011 (section 4.2.2)

We apply our model to zero and few-shot classification using the CUB-200-2011 dataset (Wah et al., 2011). It contains 11,788 images of 200 subcategories belonging to birds. Following Akata et al. (2015), we use 100 bird categories for training, 50 for validation, and 50 for testing. We generate the labels for N-way classification tasks by choosing N random classes at each training step and arbitrarily assigning the labels $0, \ldots, N-1$ to each. For each task, the number of shots k, i.e. the number of example images per class ranges uniformly between 0 and 10. The target set consists of 20 images per class. We perform independent experiments with three formats of knowledge:

A: Knowledge $\mathcal{K}$ represents attributes characteristic for a given class, e.g. wing span, feather color, shape of the beak. This is obtained by a class-wide average of the binary attribute vectors from the original dataset associated with each image. Knowledge representations, $\mathcal{K}$, are constructed by stacking all N class attribute vectors into a $N \times 312$ tensor. In this setup, the knowledge encoder, h_2, is a simple 2-layer MLP.

B: Knowledge $\mathcal{K}$ represents the average per class, natural language descriptions of the N classes. These are obtained by

*https://www.ncei.noaa.gov/access/crn/qcdatasets.html



The night will start off cold with temperatures falling to -8.9°C by late morning, and then gradually rise to a high of 1.6°C in the late afternoon. Temperatures will start to drop again in the evening, reaching -3.1°C by midnight.

The night will start off chilly with temperatures around 0.5°C, but it will drop to -1.7°C by early morning. The day will gradually warm up, reaching a high of 5.1°C in the afternoon before cooling off to 1.0°C by midnight.

The night will start off cold with temperatures falling to -16.8°C by dawn, and the day will continue to get colder, reaching a chilly -23.0°C by midnight. Afternoon temperatures will hover around -18.5°C, so bundle up if you're heading out.

The night will be bitterly cold with temperatures around -18 degrees, gradually increasing to -14 degrees by late afternoon. The temperature will slightly drop again to -15 degrees in the evening, warming up a bit to -14 degrees at midnight.

Figure A.3. Extended Figure 7 with GPT-4 generated "weather forecasts" for setup B.

averaging sentence embedding of individual image captions belonging to the given class. We use human-generated captions as collected in (Reed et al., 2016) and encode them using CLIP embeddings (Fu et al., 2022). Averaged per class text embeddings are then stacked to form a $N \times d^{model}$, where $d^{model} = 512$. In this setup, the knowledge encoder, h_2 is a 2-layer MLP.

C: We use GPT-4 to generate individual descriptions of each class based on the human-generated image captions. We present 5 randomly sampled image captions pertaining to one class and prompt GPT-4 to generate short descriptions of features characteristic of the given bird breed.To generate the class descriptions, we use the following prompt format:

> **System:** `You are given 5 descriptions of a bird breed. Based on this information generate one comprehensive description of the bird breed. Keep it short and informative.`
>
> **User:** `<<List of 5 randomly sampled image captions>>`

In this setup, the knowledge encoder, h_2 is the CLIP text encoder. The embeddings of class descriptions are obtained as the average of all outputs from the last layer of CLIP. After stacking them together in a $N \times d^{model}$ tensor, they are passed through a linear projection layer.

For all setups, A, B, and C, the data encoder, h_1 is implemented with a frozen CLIP vision model, followed by a linear projection layer. Following the approach of Garnelo et al. (2018a), we only aggregate over inputs of the same class. The aggregated class-specific representations are then concatenated to form the final representation of size $N \times d$. We set $d = 512$. We use the sum & 2-layer MLP aggregation a. We modify the decoder, g to return the logits of the categorical distribution. For a N-way task with class labels $c_1, \ldots, c_N$, we define $p(y^* \mid x^*, z)$ as:

$$p(y^* = c_j \mid x^*, z) = \frac{\exp(-w_j^T x^*)}{\sum_{j'} \exp(-w_j'^T x^*)}, \quad [w_1, \ldots, w_N] = \text{MLP}(z), z \in \mathbb{R}^{N \times d},$$

where $x*$ is a CLIP image embedding from the target set. In our experiments, we use the Hugging Face implementation of the CLIP ViT-B/32 model (https://huggingface.co/openai/clip-vit-base-patch32). We use a learning rate or 1e-4, batch size of 32 and knowledge is randomly masked at rate 0.5. For setups A and B, the INP model is trained end-to-end. For setups C,

the weights of the INP model from the trained weights of the already trained, plain NP, and all model components, including the CLIP text encoder, are fine-tuned. As opposed to setup B, in setup C fine-tuning of the CLIP text encoder was necessary to ensure alignment between the class-wide descriptions and image representations. Empirically, the two-stage training resulted in improved convergence.

For the empirical results and short discussion, refer to the main body of the paper. In Table A.1 we present sample human-generated captions (used in setup B) and their corresponding GPT-generated class descriptions (used in setup C).

Table A.1. Example images, image captions and GPT-generated class descriptions.

Sample Images	Sample image captions	GPT-generated class description
	1. *A large bird with a white belly, black and white wings with a long beak.* 2. *This bird is white and grey in color with a curved beak, and black eye rings.* 3. *A large bird with a white belly and face, black back and wings, and peach bill.* 4. *Bird has gray body feathers, white breast feather, and long beak* 5. *A medium sised bird with black wings, and a bill that curves downwards*	*This bird breed is a medium to large size, characterised by its grey body feathers, contrasting white belly and face, black back and wings, distinctive black eye rings, and a long, downward-curving peach bill.*
	1. *This big bird has a sharp beak and has black covering its body.* 2. *An all black bird with a distinct thick, rounded bill.* 3. *This entirely black bird has long and wide rectrices relative to the size of its body.* 4. *A black bird with a long tail and large beak.* 5. *This black bird has sparse plumage and a thick brown beak.*	*This bird breed is large and entirely black with sparse plumage, characterised by its thick brown beak, long tail, and wide rectrices relative to its body size.*
	1. *This goofy looking bird sports webbed feet and a bright orange bill, with piercing white eyes and a dull coat of gray.* 2. *A black bird with a small, orange beak and a inverted feather curl at the base of the beak.* 3. *A black body, white eye with stripe next to it, and an orange bill are on this bird.* 4. *This black bird has a orange bill with hair coming out of it, small pupils, and a white line across its face.* 5. *This bird has wings that are black and has an orange bill*	*This bird breed is characterised by its black body, webbed feet, a bright orange bill with an inverted feather curl at the base, piercing white eyes with a distinctive stripe, and a dull grey coat.*
	1. *This is a black bird with a white spotted belly and a white eye.* 2. *This bird is black with white and has a very short beak.* 3. *This bird has wings that are black and white and has a small bill* 4. *This small bird is white with black spots, a white neck, and black around its eyes.* 5. *This is a short stocky bird with webbed feet, it is mostly white with black wings and black speckles throughout.*	*This bird breed features a black body with a white and black spotted underbelly, a white and grey speckled chest, a black crown, bright white eyes with very small pupils, and a short, pointed, black and orange bill.*

A.3. Extended Related Work

Conditional generative models. The goal of deep generative models (DGMs) is to learn a neural approximation of the distribution of the data $p(x)$ over a space $\mathcal{X}$, most commonly the space of images. Popular DGMs include, VAEs (Kingma & Welling, 2014), GANs (Goodfellow et al., 2014), and diffusion models (Ho et al., 2020; Song & Ermon, 2019). Their conditional versions, e.g. CVAEs (Sohn et al., 2015), CGANs (Mirza & Osindero, 2014), and conditional diffusion models (Ho & Salimans, 2022; Ramesh et al., 2022) model the conditional distribution $p(x \mid c)$, where c is an additional conditioning variable, e.g. a class label or a text sequence. A similar analogy can be drawn between NPs and INPs which, as meta-learners, bring the idea of (conditional) generative modeling to the space of hypothesis $f \in \mathcal{F}$. The goal of NPs is to model the prior distribution over functions $p(f)$; as well as the posterior predictive distribution $p(f \mid \mathcal{D}_C)$. Similarly to CVAEs and CGANs, INPs introduce an additional conditioning variable–the expert knowledge, and model the conditional distribution $p(f \mid \mathcal{K})$, guiding the prior over the space of functions, such that the informed predictions, dictated by $p(f \mid D_C, \mathcal{K})$ are concentrated around the region of functions agreeing with both the observed dataset $\mathcal{D}_C$ and the expert knowledge $\mathcal{K}$.

Multimodal deep learning. broadly refers to deep learning methods that can process and relate information from multiple modalities simultaneously, such as image, audio, and text. Our framework assumes that the datasets $\mathcal{D}$ and knowledge representations $\mathcal{K}$ may belong to two different data modalities (e.g. $\mathcal{D}$ contains input-output pairs for 1-D regression and $\mathcal{K}$ contains a natural language description of the expected shape of the regression curve). This places informed meta-learning in the area of multimodal methods. What makes it distinct is that standard multimodal strategies (e.g. Ngiam et al. (2011); Srivastava & Salakhutdinov (2012); Ding & Tao (2015); Shi et al. (2021b;a)) consider finding a predictive function f, where $\mathcal{X}$ is a multimodal input space $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_M$, with each $\mathcal{X}_j, j \in [M]$ corresponding to a different data modalitly. In informed meta-learning, the learned functions f are typically unimodal, but the learning algorithm to fit each function is conditioned on the knowledge representation $\mathcal{K}$, belonging to a different modality.

Natural language priors. LLMs trained on vast text corpora can be seen as databases of knowledge about the world. Recent studies of Choi et al. (2022) and Li et al. (2023) explore utilising LLMs as sources of prior knowledge in a non-meta setting. With carefully designed prompts, a LLM outputs a prior distribution over the space of outcomes, which, when combined with downstream ML models, leads to "informed" predictions. This strategy has been shown to be successful in many tasks where semantic meta-data is available, including feature selection, reinforcement learning, causal discovery, and image segmentation. However, using LLMs as a source of knowledge may raise ethical concerns, especially when querying a LLM about sensitive attributes, potentially propagating harmful biases from their pre-training text corpora. In contrast to these methods, our framework does not rely on a language model as a source of common-sense knowledge. Instead, LLMs are merely used to generate sentence embeddings of the human expert's knowledge presented in natural language.

Zero-shot and few-shot learning. As presented in the experimental section, informed meta-learning enables sensible zero-shot predictions guided by expert knowledge. For instance, in multi-class image classification, $\mathcal{K}$ may contain a list of characteristic attributes of each class or class-wide descriptions in natural language. Seemingly similar ideas of utilising side information about each class for zero-shot learning have been explored in works of Al-Halah et al. (2016); Elhoseiny et al. (2017); Paz-Argaman et al. (2020). In contrast to these methods, informed meta-learning does not focus on zero-shot learning only, but on the process of integrating external knowledge (e.g. knowledge about what are the characteristic features of each class) with observed few-shot or zero-shot ($\mathcal{D}_C = \varnothing$) data sample. In the image classification domain, the idea of combining sample images with zero-shot attribute information was considered by Tsai & Salakhutdinov (2017) in application to one-shot learning, and extended by Schönfeld et al. (2019) to few-shot learning. None of these works, however, consider the meta-learning setup of N-way, k-shot classification and require that the class attribute information is always present at training and test time, as it implicitly defines class labels. In our setup, the role of class information contained in $\mathcal{K}$ lies in enhancing model performance by emphasising which visual features are most distinctive for a given class, enabling zero-shot classification as a byproduct. Contrary to Schönfeld et al. (2019), the additional information contained in $\mathcal{K}$ is not necessary for few-shot predictions on previously unseen classes.

Domain knowledge infusion for neural networks. Informed ML for deep learning aims to develop methods for explicit knowledge integration into neural representations. This can be achieved by, for instance, designing specialised layers or complete model architectures (Peng et al., 2019; Wu et al., 2018; Bogatskiy et al., 2020), introducing additional regularisation terms (Karpatne et al., 2017; Xu et al., 2017), or equipping the model with the ability to query external information (Annervaz et al., 2018; Xu et al., 2015). In contrast to informed meta-learning, these methods only tackle a single-dataset setting. In informed meta-learning the external information is not queried per each input, but per each learning task and pertains to all inputs and outputs within a task, often informing about global properties of the learned functions.

Meta-learning for PINNs. The idea of combining meta-learning with informed machine learning methods has previously been employed in PINNs. For instance, (Huang et al., 2022; Qian et al., 2021; Belbute-Peres et al., 2021) use meta-learning to learn how to solve a differential equation given a specific parameterisation. In this context, the parameterisation of the equation serves as the knowledge representation for a particular task. These approaches leverage meta-learning to solve tasks represented as pairs $(\mathcal{D}, \mathcal{K})$ more efficiently than training independent models for each data-equation pair. Crucially, however, it is possible, albeit less efficient, to train an independent PINN for each individual task. In our setup, incorporating the information contained in $\mathcal{K}$ directly and explicitly into a learning algorithm is challenging. By employing meta-learning across a distribution of tasks, an informed meta-learner learns how to incorporate this information as a prior distribution over the space of possible solutions. This approach enables domain experts to inject their prior knowledge, represented for example in natural language, into future tasks.

Meta-learning with meta-data. There exist works that apply meta-learning to hierarchical datasets with task data and their corresponding metadata, such as feature names (Iwata & Kumagai, 2022), task-specific parameters (Kaddour et al., 2020), or labels (Tseng et al., 2022). The metadata of each task is used, in addition to the empirical data, to quantify the similarities between different tasks. While seemingly similar to our setup, motivation of these works is distinct from ours. For instance, (Kaddour et al., 2020) extends ideas from active learning to meta-learning, where the algorithm can choose which task to learn next by considering the metadata of candidate tasks and their similarity to previously solved tasks. The method of (Iwata & Kumagai, 2022) uses feature descriptions, and (Tseng et al., 2022) uses multi-class agricultural labels to infer similarities between different learning tasks, thereby improving generalization to new tasks. The underlying assumptions of these methods are that datasets with similar metadata share similar relationships between the variables of the data. We want to stress the difference between metadata and knowledge. Metadata need not contain explicit knowledge about the underlying DGP of a task. For instance, the fact that a tabular dataset contains features named "age" and "salary" does not express knowledge about how age relates to salary. This relationship would need to be discovered by the meta-learning algorithm, and even if successfully learned, it would remain unknown to the human user due to the black-box nature of modern meta-learning approaches. The key motivation of this paper is to enable human experts to inject their prior knowledge about the functional relationships between the variables of a task via meaningful and intuitive knowledge representations—specifically, sentences in natural language that explicitly articulate (partial) knowledge about the underlying DGP of a task.

A.4. Probabilistic perspective on informed meta-learning

From knowledge to inductive biases. In conventional, single-task learning settings, a machine learning practitioner, based on their own understanding of the subject, or insights of a domain expert elicits a ML model equipped with inductive biases tailored to the specific task. This process can be seen as assigning a prior probability to the space of all data generating processes, or alternatively the space of all functions[†] mapping inputs to outputs, $\mathscr{F}$. We will denote this prior as $p(f)$. The model's support, or hypothesis space, is the subset of $\mathscr{F}$ where the prior probability, $p(f)$, is non-zero, $\mathcal{F} = \{f \in \mathscr{F} : p(f) > 0\}$. Inductive biases are then defined as the relative, prior probabilities of possible solutions. Let f^* denote the true state of nature according to which our observed dataset $\mathcal{D}$ has been generated. The model is well specified if f^* is assigned a non-zero prior probability, i.e. it belongs to the support of the model. If f^* is within the support, the model's posterior $p(f \mid \mathcal{D})$ will concentrate around f^* as the model is updated with an increasing number of observations. The effectiveness of the inductive biases depends on how much mass is a priori assigned to f^* relative to all other solutions supported by the model, dictating the convergence rate of $p(f \mid \mathcal{D})$ to f^*. A perfect inductive bias is one that completely solves the task, essentially represented by a delta distribution centered at f^*.

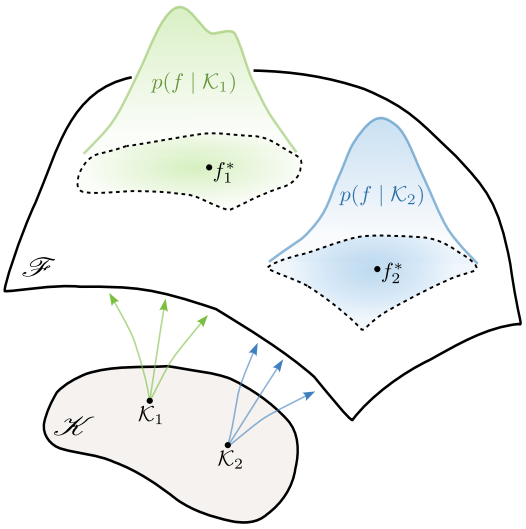


Figure A.4. Machine learning practitioners map expert knowledge $\mathcal{K} \in \mathscr{K}$ to a prior probability over functions $f \in \mathscr{F}$. The regions of $\mathscr{F}$ assigned non-zero probability defines the hypothesis space of a model. Knowledge about different learning tasks is related to distinct regions of $\mathscr{F}$.

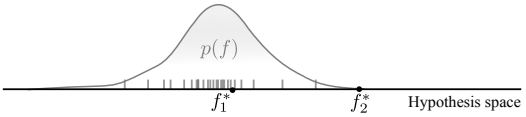
What human machine learning researchers and engineers are skilled at is 1) understanding the context of the given learning problem; 2) translating this acquired knowledge to the function space $\mathscr{F}$ and defining adequate inductive biases. If we denote the relevant knowledge for a specific machine learning problem as $\mathcal{K}$, the actions of a machine learning practitioner can be abstractly viewed as a mapping from $\mathcal{K}$ to the prior probability distribution over $\mathscr{F}$, represented as $p(f)$. To emphasise the dependency on $\mathcal{K}$, we shall write $p(f \mid \mathcal{K})$, and the knowledge integration process as a map $\mathcal{K} \mapsto p(f \mid \mathcal{K})$. For instance, if based on their knowledge, a domain expert demands that the fitted function should be linear ($\mathcal{K}_1 =$ "linear"), the machine learning practitioner, by choosing a linear regression model assigns a non-zero probability to all linear functions and 0 otherwise. When faced with a different learning problem, and new knowledge concerning it, e.g. $\mathcal{K}_2 =$ "non-linear", a new model, perhaps a polynomial regression model or a neural network, with different support and inductive biases will be defined through $\mathcal{K}_2 \mapsto p(f \mid \mathcal{K}_2)$.

Knowledge and cross-task generalisation. Notably, humans have the powerful ability to generalise between different learning tasks from various domains. This is possible as the information about the learning task that they are initially presented with contains knowledge about the true DGP or functional properties that are **independent** of the task's domain. For example, the requirement for the function mapping inputs to outputs to be increasing is a universal property unaffected by the specific learning problem. The same inductive biases can be applied whether modeling economic growth or the dosage-response relationship in medicine. The word "increasing" conveys **semantically meaningful** information for the machine learning practitioner, expressing a functional property that is domain-agnostic. This transcendence of knowledge combined with semantically meaningful representations is what allows human ML practitioners to generalise across different learning tasks. An ML practitioner, having the ability to construct models conforming to several forms of knowledge, gained from previously solved tasks, can easily construct ML models that are in line with knowledge pertaining to a new learning task.

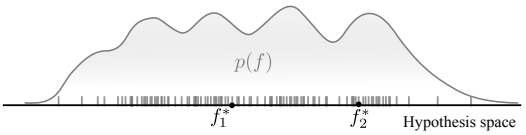
Yet, the process of manually defining inductive biases of a model is constrained by the machine learning engineers' and researchers' abilities to construct models that adhere to different prior specifications. While properties like linearity or translation invariance are relatively easy to encode by e.g. constraining the hypothesis space to linear models or modeling the problem with a convolutional neural network, more fine-grained or less formally defined priors may appear challenging

[†]The two approaches of modeling the data distribution or learning a function from inputs $x \in \mathcal{X}$ to outputs in $y \in \mathcal{Y}$ can be seen as equivalent if we assume that the observed dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is generated according to $y_i = f^*(x_i) + \epsilon_i$, where f^* is a realisation of a stochastic process indexed by $\mathcal{X}$ and each ϵ_i is an independent mean-zero random variable.

or even impossible to encode manually, in which case they often need to be disregarded and models with weak-inductive biases and large support are used instead, necessitating vast amounts of training data to correctly recover f^*.



(a) Concentrated meta distribution, provides strong inductive biases, so long as the task of interest belongs to the same environment as the tasks used during meta-training.



(b) Heterogeneous environments span a wide range of functions, and thus support a wider range of tasks, at the cost of weaker inductive biases.

Figure A.5. Meta-learning under well-concentrated and heterogeneous task distributions.

Meta-learning as inductive bias learning. To address this challenge, meta-learning, viewed through the lens of inductive bias learning (Baxter, 1997), suggests that the inductive bias can indeed be learned. This presupposes the ability to sample from an environment of related problems, which induces a prior probability over our hypothesis space; the meta-learner estimates this prior as $p(f)$. Consider f_1^* as the function defining the dataset for our task of interest. If f_1^* is likely to belong to the same environment as was observed during the estimation of $p(f)$, then $p(f)$ will provide good inductive biases for solving this task, as its mass will be concentrated near f_1^* (see Figure 1(a)). If, however, we are given a problem generated according to a different process, f_2^*, that is "far away" from most of the problems observed during meta-training, the meta-learned distribution, $p(f)$, may offer suboptimal inductive biases compared to assuming a non-informative (e.g. uniform) prior over our hypothesis space, a phenomenon known as negative transfer. To mitigate this issue, one may try to meta-learn across more tasks that would cover a larger region of $\mathscr{F}$. This however would lead to $p(f)$ that is heterogeneous and dispersed, resulting in inductive biases that are not sufficiently strong for learning with just a small, few-shot data sample. Essentially, this presents a trade-off akin to the classic bias-variance trade-off from a meta-perspective.

Informed meta learning as conditional inductive bias learning. This discussion brings us to conclude that neither the conventional approaches nor automatic approaches of inductive bias specification are self-sufficient; yet, each approach brings distinct advantages for enhancing data efficiency. Conventional methods rely on human ML practitioners to establish the mapping $\mathcal{K} \mapsto p(f \mid \mathcal{K})$. In meta-learning, prior knowledge about task relatedness is crucial to ensuring that the meta-learned distribution, $p(f)$, concentrates around the class of problems or a specific problem of interest. On the positive side, conventional methods offer greater explainability, as the inductive biases of a model can be (at least partially) traced back to the prior knowledge $\mathcal{K}$, which is meaningful to humans. Whereas, meta-learning, provides greater functional flexibility, enabling the model to learn non-trivial inductive biases, which otherwise may be impossible to explicitly hard-code into the learning method.

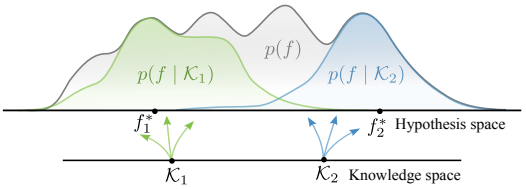


Figure A.6. Informed meta-learning allows to condition the meta-distribution on the expert knowledge $\mathcal{K}$, concentrating the meta-distribution around the regions of our hypothesis space close to the true data generating functions.

Given the above, it is reasonable to consider a method that could bring together the two schools of thought. We want to retain the possibility of learning fine-grained or less formally stringent inductive biases, while at the same time being able to guide the learner to the space of solutions that conform to our prior knowledge. To that end, informed meta-learning proposes that the mapping $\mathcal{K} \mapsto p(f \mid \mathcal{K})$ is meta-learned by training over multiple tasks and their associated knowledge. This process induces a possibly heterogeneous meta distribution $p(f)$, that is conditioned on the expert knowledge $\mathcal{K}$, thus concentrating the mass of $p(f)$ around the region of the hypothesis that conforms to $\mathcal{K}$ (see Figure A.6).

Knowledge vs. (meta-) data. One may argue, that this proposed approach does not significantly differ from meta-learning with meta-data or meta-features, where the meta-distribution could be conditioned on additional meta-data instead of $\mathcal{K}$. While from the learning method point of view, as presented in sections 3 and 4, this is to some extent true, the key difference lies in the assumptions on the kind of information contained in $\mathcal{K}$ versus meta-data or meta-features.

First of all, while the meaning of knowledge is an ongoing debate in philosophy, here we take the scientific perspective and assume that knowledge is a form of information that has already been validated, a justified true belief. This means, that the relationships between knowledge and the underlying DGP of a given task should be a prior known to hold true. In practice this means that if $\mathcal{K}$ is the knowledge about the task for which the true data generating distribution is defined through f^*,

then $p(f \mid \mathcal{K})$ should assign non-zero mass to a region around f^*. On the other hand, meta-data or meta-features are not a priori known to contain relevant information for a learning task. The relationship between meta-data and the learning task would need to be discovered by the learner during meta-training. Unlike knowledge, this relationship is unknown to the human domain expert or ML practitioner and cannot be utilised for successful and explainable generalisation to new, previously unobserved learning tasks. This means that knowledge, in contrast to meta-data or meta-features is **verifiable**. If a human expert, knowing $\mathcal{K}$, is presented with a hypothesis f', they can easily identify whether f' is consistent with $\mathcal{K}$ (in which case $p(f' \mid \mathcal{K}) > 0$) or not ($p(f' \mid \mathcal{K}) = 0$). This is not necessarily true for meta-data or meta-features.

Furthermore, the knowledge space, $\mathscr{K}$, is understandable for humans, while in contrast, the observation space is not, and neither is the hypothesis space of a model, particularly in the context of deep learning. The meta-learned mapping $\mathcal{K} \mapsto p(f \mid \mathcal{K})$ can be seen as a new form of communication between humans and machines. For instance, suppose that $\mathscr{K}$ is generated by a set of the two functional properties $\{\text{"linear"}, \text{"increasing"}\}$ that can be composed together through negation and conjunction, e.g. $\text{"Linear and not increasing"}$. The set of all possible knowledge representations in $\mathscr{K}$ is then aligned with regions of $\mathscr{F}$ adhering to the meaning of each phrase. The set $\{\text{"linear"}, \text{"increasing"}\}$ creates an "alphabet" that together with the operations of negation and conjunction create a "language", that is human-readable. During meta-training, the rules of this language should be learned and associated with the corresponding regions of the hypothesis space of a model. This process ultimately establishes a communication channel between the human expert and a black-box ML model, facilitating meaningful interaction and improved transparency.

Informed meta-learning: future perspectives. The INP method, as introduced in section 3 and the subsequent experiments outlined in section 4, offers just a glimpse into the broader goals of informed meta-learning. Looking ahead, we envision that the knowledge representation space, $\mathscr{K}$, corresponds to the space of human natural language, and the existing semantic relationships between different words as well as their surrounding context can be leveraged for more effective learning of the map $\mathcal{K} \to p(f \mid \mathcal{K})$, requiring only a modest amount of training tasks. As a simple example, suppose that during training, the word "increasing" was successfully mapped to all increasing functions supported by a black-box model of choice. At test time, if the model is confronted with the word "decreasing" (unseen during training), the informed meta-learner, by exploiting the semantic relationship between the two words ("increasing" implying "not decreasing"), should generalize, consequently assigning a zero prior probability to all increasing functions.

A.5. Uncertainty quantification in Informed Neural Processes

One particularly appealing property of Neural Process, which motivated their choice for the basis of our informed meta-learner, is the ability to estimate probabilities over the space of solutions, instead of returning a single point estimate. This allows us to measure the reduction in model uncertainty given prior expert knowledge and/or observed data. We are mostly interested in measuring the epistemic, rather than aleatoric uncertainty.

Aleatoric uncertainty refers to the notion of randomness seen as the variability in the outcomes which is due to inherently random, unpredictable effects. As opposed to this, epistemic uncertainty refers to uncertainty caused by the lack of knowledge about the true relationship between model inputs and outputs. By observing data, or by inserting prior knowledge into the model, the epistemic uncertainty is reduced.

A natural choice for measuring the epistemic uncertainty would be the (conditional) entropy. By comparing $\mathbb{H}[p(f)]$ with $\mathbb{H}[p(f \mid \mathcal{K})]$ or $\mathbb{H}[p(f \mid \mathcal{D}_C)]$ we can measure the impact of prior expert knowledge or observed data on the reduction in the epistemic uncertainty for a single learning task. However, in INPs, we only have access to samples from the variational distribution and since the decoder is implemented as a neural network, evaluating the distribution over functions is not possible directly. Instead, we need to resolve to measure the uncertainty in the observation space. Thus, we are interested in computing

$$\mathbb{H}[y^* \mid x^*, I], \quad I \in \{\mathcal{K}, \mathcal{D}_C, \mathcal{K} \cup \mathcal{D}_C, \varnothing\} \tag{13}$$

at a particular location $x^* \in \mathcal{X}$ in our input space, which can be then, for instance, averaged across uniformly distributed points in $\mathcal{X}$. The quantity in (13) is known as the predictive uncertainty. To approximate (13) for an input x^*, we rely on Monte-Carlo estimation by sampling S functions based on our variational decoder.

$$\begin{aligned}
\mathbb{H}[y^* \mid x^*, I] &= -\int p(y^* \mid x^*, I) \log p(y^* \mid x^*, I) dy^* \\
&= -\int \left(\int p(y^* \mid x^*, f) p(f \mid I) df \right) \log \left(\int p(y^* \mid x^*, f) p(f \mid I) df \right) dy^* \\
&\approx -\int \left(\frac{1}{S} \sum_{s=1}^{S} p(y^* \mid x^*, f^{(s)}) \right) \log \left(\frac{1}{S} \sum_{s=1}^{S} p(y^* \mid x^*, f^{(s)}) \right) dy^*
\end{aligned} \tag{14}$$

For each sample $f^{(s)}$, $p(y^* \mid x^*, f^{(s)})$ has a closed-form expression, as in the case of regression it is modeled with a normal distribution. Thus (14) can be computed by numerically approximating the integral in the last line. Note that, since predictive uncertainty is measured in the observation space, it also encompasses the uncertainty associated with the observational noise. Depeweg et al. (2018) suggest that (13) can be decomposed as:

$$\mathbb{H}[y^* \mid x^*, I] = \underbrace{\mathbb{I}(y^*, f \mid x^*, I)}_{\text{epistemic}} + \underbrace{\mathbb{E}_{f \sim p(f \mid I)}[\mathbb{H}[y^* \mid x^*, f]]}_{\text{aleatoric}} \tag{15}$$

The second part, $\mathbb{E}_{f \sim p(f \mid I)}[\mathbb{H}[y^* \mid x^*, f]]$, is the average entropy when the predictive function is known, thus can be interpreted as the aleatoric uncertainty. If we model $p(y \mid x, f)$ with a normal distribution, $\mathbb{H}[y^* \mid x^*, f]$ has a closed-form expression, $\frac{1}{2}\log(\sigma(x^*)2\pi e)$, where $\sigma^2(x)$ is the variance at location x^*.

The first part, $\mathbb{I}(y^*, f \mid x^*, I)$, representing the information gain can be interpreted as the epistemic uncertainty of interest. This quantity can be computed as the difference of $\mathbb{H}[y^* \mid x^*, I]$ and $\mathbb{E}_{f \sim p(f \mid I)}[\mathbb{H}[y^* \mid x^*, f]]$, where both quantities are easy to estimate, as discussed above.

A.6. Additional experiments

A.6.1. Model performance vs. number of training tasks

Setup: We follow the same setup as in the illustrative experiment from section 4.1.1. We create multiple training collection of tasks with a varying number of total training tasks, $N^{train} \in \{25, 50, 75, 100, 200, 1000\}$, with the upper limit being the number of tasks used in the original experiment 4.1.1. For each training collection of tasks, we train independent INP and NP models. The INP models receive information about two, one or none of the parameters a, b or c via knowledge representations, $\mathcal{K}$. All models are validated and tested on the same collection of validation /

testing tasks.

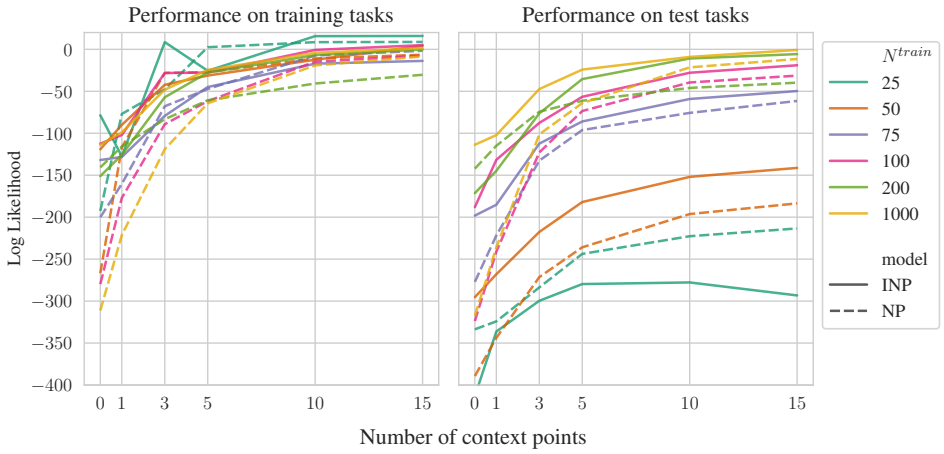


Figure A.7. Log likelihood of target data vs. number of context data points (higher is better). Comparison across varying number of all training tasks, N^{train}. Left - model performance on training tasks, Right - model performance on test tasks.

Figure A.7 shows the performance of all models on training (left) and testing (right) tasks. We observe that: 1) Both for the NP and INP models as the number of training tasks decreases the performance gap between training and testing tasks increases. However, we note that this performance gap is already at a (subjectively) reasonable level with only as few as 75 training tasks. 2) For all INP models trained with $N^{train} \geq 50$ tasks we also observe that the additional knowledge presented for each task improves the performance over the plain, uninformed NP. However, when the number of training tasks is too small, here $N^{train} = 25$, we observe a "knowledge overfitting" effect. With insufficient number of training tasks our model is unable to appropriately capture the relationship between knowledge and empirical data, and thus fails to generalise to new, previously unseen tasks and their corresponding, also previously unseen, knowledge representations.

Take-away: We tested the robustness of the INP model to the reduction in the number of training tasks. We showed that in the experimental setup of section 4.1.1, adding external knowledge continues to deliver noticeable performance gains over the uninformed NP when dropping from 1000 to as few as 50 training tasks. We also noted that with too few training tasks, the INP may fail to generalize. To prevent this effect from occurring in real-world deployment, we advise testing the model on held-out validation tasks and comparing its performance against an uninformed baseline, monitoring the knowledge overfitting effect.

A.6.2. Model performance and knowledge complexity

Setup: To assess the impact of knowledge complexity on the efficacy of learning the relationship between knowledge and the model hypothesis space, we again follow the same setup as in section 4.1.1. We create multiple training collection of tasks with a varying number of total training tasks, $N^{train} \in \{25, 50, 75, 100, 200, 1000\}$, with the upper limit being the number of tasks used in the original experiment 4.1.1. All models are validated and tested on the same collection of validation / testing tasks. For each setting of N^{train} we train an uninformed NP and 3 independent INP models with different knowledge representations used during training:

- $\mathcal{M}_{abc}$ is a model where for each task its corresponding knowledge encodes, at random, one of the three parameters, a, b, or c;
- $\mathcal{M}_{ab}$ is a model where for each task its corresponding knowledge encodes, at random, one of the two parameters: a or b (the value of c is never revealed);
- $\mathcal{M}_{b}$ is a model where for each task its corresponding knowledge encodes the value of a (the values of parameters a and c are never revealed).

Knowledge representations are constructed by one-hot encoding the type of the revealed parameter with its value

appended at the end. We note that for the INP models $\mathcal{M}_b$, $\mathcal{M}_{ab}$, $\mathcal{M}_{abc}$, the complexity of knowledge representations gradually increases; the knowledge space is 1, 2 and 3 dimensional, respectively. We hypothesise that as the complexity of the knowledge space grows, more training tasks are needed to effectively learn the mapping from knowledge representations to prior distributions over functions. Given the same number of training tasks, the INP model $\mathcal{M}_{ab}$ needs to learn how to disentangle the information about the function's oscillations (parameter b) from the information about the function's slope (parameter a). Model $\mathcal{M}_{abc}$ additionally needs to discover the meaning of knowledge about the intercept (parameter c). Therefore, we expect that, given *the same number of context points* and *the same information contained in* $\mathcal{K}$, the relative performance gains of the INP models $\mathcal{M}_b$, $\mathcal{M}_{ab}$, $\mathcal{M}_{abc}$ over the uniformed NP model should decrease as the complexity of knowledge space increases.

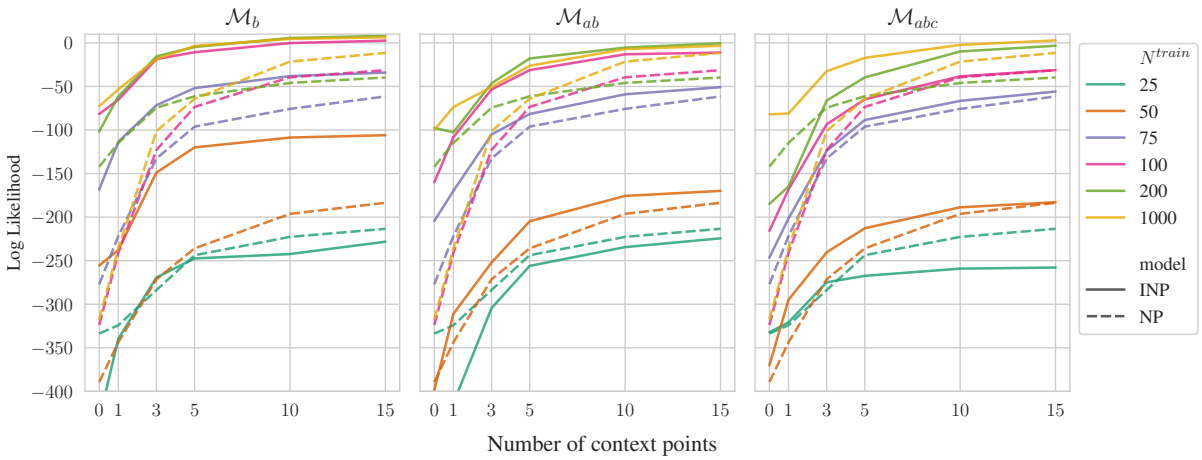


Figure A.8. LL of target data vs. number of context points (higher is better). Comparison across a varying number of tasks used for training N^{train}. Complexity of knowledge space grows from left to right. All INP models are presented with the same knowledge about each task–the value of the parameter b.

Figure A.8 shows the log-likelihood of the target data evaluted on 500 testing tasks. For every INP model at test time we reveal the same information via knowledge representations—the value of the parameter b. Firstly, we observe the same two effects as in experiment A.6.1. With more training tasks, model performance improves. 2) An insufficient number of training tasks may lead to the "knowledge overfitting" effect; here at $N^{train} = 25$ the INP performs worse than the NP. Secondly, we look at the performance gap between the INP and the NP (the gap between solid and dashed lines). We observe that as the complexity of the knowledge space grows (left to right) the performance gap between the INP and the NP decreases. This is summarised through the ΔAUC metric, presented in the Table A.2. From Figure A.8 we can also conclude that the more complex the the knowledge space is the more training tasks are needed to effectively train an INP model. For instance, performance of the INP model $\mathcal{M}_b$ trained with $N^{train} = 75$ tasks is comparable to the performance of the INP $\mathcal{M}_{abc}$ trained wtih $N^{train} = 100$ tasks.

Table A.2. Average relative performance improvement (%) between informed and uninformed predictions. The performance gains become smaller as the complexity of the knowledge space grows (top to bottom). N^{train} is the number of training tasks used. INP performs better than the NP for all settings of $N^{train} \geq 50$ indicating effective transfer between knowledge representations and functional priors. INP overfits with not enough training tasks, here at $N^{train} = 25$.

N^{train} model	1000	500	200	100	75	50	25
$\mathcal{M}_b$	81.74	83.51	64.52	79.1	44.78	39.71	-8.52
$\mathcal{M}_{ab}$	65.3	67.45	43.46	54.66	22.41	6.67	-18.67
$\mathcal{M}_{abc}$	71.76	57.51	2.02	26.46	9.48	8.05	-5.6

Take-away: The above experiment confirms our hypothesis about the complexity of the information conveyed in knowledge representations and the hardness of learning the mapping between knowledge representations and the

model hypothesis space. As the complexity of knowledge increases, more training tasks are needed to effectively learn the relationship between knowledge representations and the functional priors.

A.6.3. CORRELATION IN TRAINING DATA AND KNOWLEDGE DISENTANGLEMENT

Setup: For each task, context and target data points are sampled according to a similar process as in the experiments from section 4.1. A function f is sampled from the family of sinusoidal functions with a linear trend, $f(x) = ax + \sin(bx)$. As previously, we also introduce a Gaussian observational noise, s.t. $y_i = f(x_i) + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 0.2)$. In this experiment. we simulate a scenario in which the training data exhibits a potentially spurious correlation. We sample the parameters a and b from a multivariate Guassian,

$$\begin{bmatrix} a \\ b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 & \sigma \\ \sigma & 2 \end{bmatrix} \right)$$

We create 6 training and validation collection of tasks, one for each value of the covariance between a and b, $\sigma \in \{0.0, 0.3, 0.6, 0.9, 1.2, 1.4\}$. We then train 6 independent INP and NP models. For the INP models we let $\mathcal{K}$ encode the value of one of the two parameters a or b. The number of context points n ranges uniformly between 0 and 10; the number of targets is set to $m = 100$. The testing collection of tasks is created by sampling functions where a and b are *independent* (i.e. $\sigma = 0.0$). This setup aims to test the robustness of the INP model to spurious correlations in the training data. We want to investigate whether the INP model is able disentangle the meanings of parameters a and b.

Table A.3. Average log-likelihood on test tasks vs. correlation in training data (higher is better). ρ - the correlation coefficient between random parameters a and b. n - number of context data points per task. Model results for which the log likelihood is higher by a statistically significant margin highlighted in bold. Values in brackets stand standard errors estimated with bootstrap.

ρ	**0.00**		**0.21**		**0.42**	
model	**INP**	**NP**	**INP**	**NP**	**INP**	**NP**
$n = 0$	**-139.1 (10.2)**	-209.2 (8.3)	**-174.4 (13.7)**	-196.5 (7.7)	-266.1 (19.6)	**-221.7 (9.2)**
$n = 1$	**-99.0 (10.6)**	-102.1 (6.5)	**-73.1 (6.2)**	-120.4 (5.4)	**-95.0 (9.4)**	-108.6 (4.4)
$n = 4$	**-16.9 (1.7)**	-30.9 (2.6)	**-34.8 (3.9)**	-41.0 (3.4)	**-29.2 (3.8)**	-38.2 (3.4)
$n = 5$	-12.1 (2.0)	-11.9 (2.0)	-15.7 (1.8)	-17.8 (2.6)	**-12.4 (2.5)**	-21.7 (2.9)
$n = 10$	1.3 (1.1)	1.8 (1.4)	-0.5 (1.2)	2.1 (0.9)	**0.2 (0.9)**	-4.5 (2.0)
$n = 15$	3.5 (0.7)	5.4 (1.4)	0.8 (0.8)	2.6 (1.6)	**3.1 (0.6)**	-2.2 (2.1)

ρ	**0.64**		**0.85**		**0.99**	
model	**INP**	**NP**	**INP**	**NP**	**INP**	**NP**
$n = 0$	-356.4 (22.2)	**-214.0 (9.2)**	-795.6 (38.9)	**-321.8 (14.8)**	-1367.3 (63.6)	**-410.7 (15.2)**
$n = 1$	**-108.1 (7.1)**	-160.3 (7.4)	-234.4 (9.3)	**-200.0 (9.9)**	-830.9 (40.9)	**-527.3 (19.2)**
$n = 3$	**-26.2 (2.2)**	-64.6 (4.4)	-149.0 (6.6)	**-118.6 (6.3)**	-551.5 (32.5)	**-360.1 (10.6)**
$n = 5$	**-18.3 (2.0)**	-30.3 (3.0)	-101.4 (4.9)	**-94.0 (5.1)**	-404.2 (13.2)	**-319.9 (9.0)**
$n = 10$	**-6.2 (1.2)**	-18.2 (2.6)	-81.2 (4.4)	**-70.6 (4.5)**	-342.7 (11.1)	**-324.9 (9.3)**
$n = 15$	**-4.3 (1.2)**	-11.6 (2.3)	-74.1 (4.3)	**-66.2 (4.3)**	-332.2 (11.1)	**-313.0 (8.9)**

Results presented in table A.3 show that when the correlation between the parameter a and b increases, the test-time performance of both the INP and NP models downgrades. This is due to the train-test distribution shift. Moreover, when the correlation is moderate ($\rho \leq 0.64$), the INP model outperforms or mathces the performance of the NP. We note, however, that for $\rho >= 0.42$, the zero-shot predictions ($n = 0$) are better for the uninformed model than the INP. This is also true for all values of n at higher correlation levels ($\rho \geq 0.85$). We hypothesise that this is because the INP has overfitted to the correlation between the parameters a and b. In the training dataset, revealing the information about the value of one parameter gives information about the value of the other, unrevealed parameter. INP exploits this dependency.

Take-away: INPs learn the meaning of knowledge based on its relationship with the empirical data. If this relationship changes at test time, good performance of the INP can no longer be guaranteed. This characteristic may be especially dangerous when there are spurious correlations in the dataset. The INP is prone to overfitting to these correlations, "misunderstaning" the true meaning of knowledge, and thus failing to generalize to new knowledge representations and their corresponding tasks, where the spurious correlations are no longer present.