Variational Hierarchical Mixtures for Probabilistic Learning of Inverse Dynamics

Hany Abdulsamad, Peter Nickl, Pascal Klink, and Jan Peters, *Fellow, IEEE.*

Abstract—Well-calibrated probabilistic regression models are a crucial learning component in robotics applications as datasets grow rapidly and tasks become more complex. Unfortunately, classical regression models are usually either probabilistic kernel machines with a flexible structure that does not scale gracefully with data or deterministic and vastly scalable automata, albeit with a restrictive parametric form and poor regularization. In this paper, we consider a probabilistic hierarchical modeling paradigm that combines the benefits of both worlds to deliver computationally efficient representations with inherent complexity regularization. The presented approaches are probabilistic interpretations of local regression techniques that approximate nonlinear functions through a set of local linear or polynomial units. Importantly, we rely on principles from Bayesian nonparametrics to formulate flexible models that adapt their complexity to the data and can potentially encompass an infinite number of components. We derive two efficient variational inference techniques to learn these representations and highlight the advantages of hierarchical infinite local regression models, such as dealing with non-smooth functions, mitigating catastrophic forgetting, and enabling parameter sharing and fast predictions. Finally, we validate this approach on large inverse dynamics datasets and test the learned models in real-world control scenarios.

Index Terms—Hierarchical Local Regression, Generative Models, Dirichlet Process Mixtures, Inverse Dynamics Control

✦

1 INTRODUCTION

PRINCIPLED data-driven, adaptive, and incremental learning is desirable in domains in which datasets are dynamic and accumulate slowly over time. For example, robots must build models of their dynamics and the environment as they interact with the world. Moreover, these models must be computationally efficient during both learning and evaluation. In the case of general-purpose robots, these models must incorporate different modalities of continuous and discrete stochastic random variables and possibly incorporate heteroscedastic noise [1], [2]. Predominant and successful regression techniques, such as Gaussian process regression (GPR) [3], artificial neural networks (ANNs) [4], and local regression (LR) [5], have a mixed set of properties that are useful in different scenarios.

Gaussian process regression offers a principled Bayesian treatment that enables continual and incremental learning. Nonetheless, the *vanilla* formulation of GPR [3] suffered from many drawbacks that have been gradually addressed by recent research. Some of these drawbacks are the functional smoothness assumption [6], [7], [8], limitations when scaling to large datasets [9], [10], [11], [12], [13], [14] and difficulties modeling heteroscedasticity [15], [16], [17].

On the other hand, artificial neural networks have proven themselves as very powerful easy-to-train universal approximators. They are, however, still susceptible to over-parameterization [18] and catastrophic forgetting [19]. Moreover, despite major progress on the front of Bayesian neural networks (BNNs) [20], [21], [22], [23], [24], [25], [26], new evidence suggests that issues regarding the accuracy of uncertainty quantification still need to be tackled [27], [28].

Finally, local regression methods have had great success in the domain of robotics and control [29], [30], [31], [32], because of their flexibility, ability to model hard nonlinearities, and to incorporate new data online. More generally, local regression is a family of generative mixture of experts (MoE) techniques that take a basis-function approach to model the input density and automatically induce local model responsibilities [33], [34], [35], [36], see Figure 1. In contrast, discriminative MoEs rely on an explicitly input-conditioned gating to choose the local expert [37], [38], [39].

Two categories of LR exist [40], *lazy* learners that maintain all seen data points in memory [29], [31], and *memoryless* learners that compress data by constructing basis functions in the input space and fitting and storing locally parameterized regression models [35]. Prominent examples of the latter include receptive field weighted regression (RFWR) [30] and locally weighted projection regression (LWPR) [32]. However, these methods are often difficult to tune as they possess many hyperparameters.

A limited attempt at a Bayesian treatment of LR is made in [41] by constructing local nonparametric kernels and placing gamma priors on the kernel widths to alleviate the need to tune the basis functions. This approach leads to a localized GP formulation that needs to retain the training data in memory, again leading to the computational issues of vanilla GPR. Local Gaussian regression (LGR) is a further Bayesian generalization of LR [42]. The authors treat the local models in a Bayesian framework and couple them via the loss function that reinforces global coordination. Nonetheless, both approaches rely on heuristics and thresholds for adding and pruning local models and fall short of formulating a generative model over input and output.

Following this introduction, we believe that local regression with a proper generative Bayesian treatment can

• *H. Abdulsamad is with the Department of Electrical Engineering and Automation at Aalto University, Finland. P. Nickl is with the RIKEN Center for Advanced Intelligence Project, Japan. P. Klink, and J. Peters are with the Department of Computer Science at the Technical University of Darmstadt, Germany.*
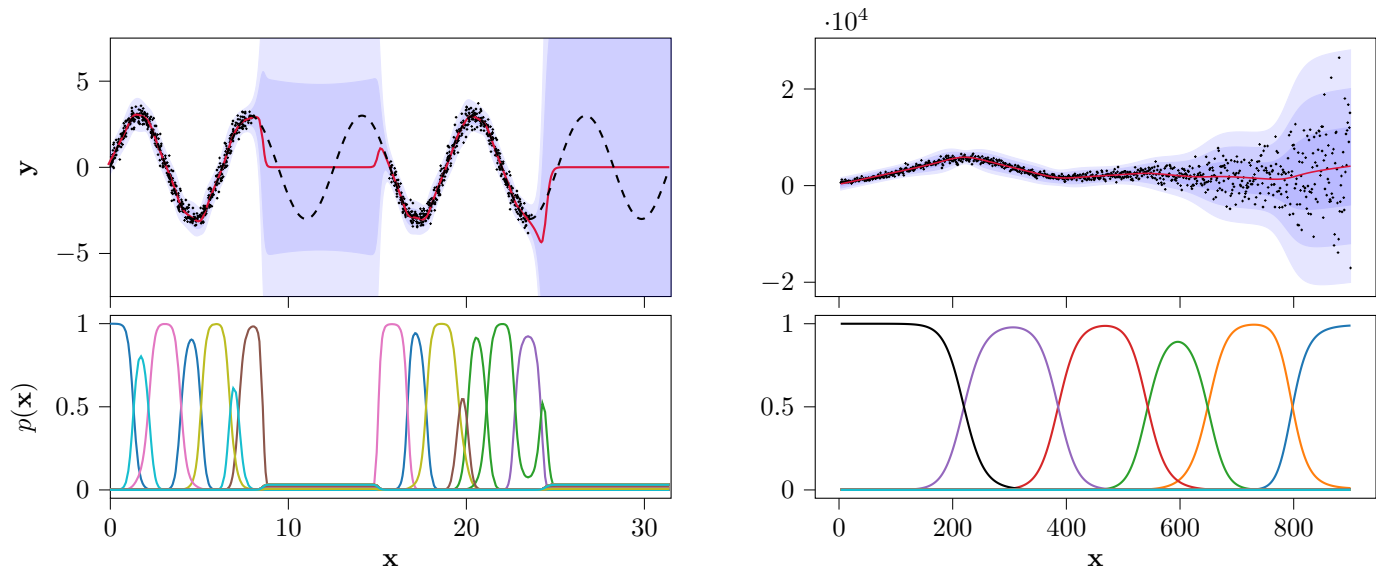
Fig. 1: Left, gap data learned with infinite local regression (ILR). The top plot depicts the mean prediction (red) on the training data (dots) and the true mean function (dashed). The blue area represents the predictive uncertainty of two standard deviations. The predictive uncertainty in areas lacking training data is large, and the mean prediction falls back to the prior. Right, the CMB dataset learned by ILR. The top figure depicts the mean prediction (red) with a predictive uncertainty of three standard deviations (shaded blue). ILR captures the heteroscedastic spread of the data with a handful of local regression models. The bottom plots show the activation of the models over the input space.

potentially serve as a powerful general-purpose function approximator. Moreover, as a probabilistic and efficient representation, it can drive many low-level applications in control and robotics that favor fast predictions and do not require deep representations.

We introduce two probabilistic graphical mixture models for local regression in the upcoming sections. The first, infinite local regression (ILR) [43], is a generative formulation that relies on the paradigm of Bayesian nonparametrics (BNP) [44] to automatically grow the mixture size based on observed data. This technique ultimately results in a general formulation of related methods that alleviates the need for any heuristic considerations. However, despite the effectiveness of ILR, and like other local regression techniques that rely on locally linear or polynomial approximations, it maintains a one-to-one correspondence between the activations and local regression units. This effect limits the model's capacity to share parameters across the input space and often forces the generation of duplicate components, needlessly increasing the overall number of parameters. To address this limitation, we introduce hierarchical infinite local regression (HILR), a multi-level development of ILR that enables multi-modal activations of the same regression component, giving the model a structure that allows sharing of regression parameters across repeating local patterns in the data. This architecture increases the flexibility of the representation and contributes towards its compression.

For learning these models, we derive two general variational Bayes (VB) schemes [45] that efficiently infer the posterior parameters and overcome the need for computationally heavy sampling methods. We benchmark the models on a range of tasks highlighting their strengths, such as dealing with heteroscedasticity, non-continuous functions, and multi-modal activation. Additionally, we test on large real-world high-dimensional datasets to benchmark learn-

ing the inverse dynamics of robotic manipulators. Most importantly, we deploy an instance of ILR to perform inverse dynamics control on a real Barrett-WAM manipulator.

Alternative Bayesian extensions of the generative mixture of experts exist in the literature. A prominent area of research focused on Gaussian processes mixtures, in which local components are modeled by separate GPs [46], [47]. Given that a single Gaussian process is an excellent approximator of nonlinear trends, the motivation for constructing such experts is not to improve the quality of approximation but rather to reduce the computational complexity and memory requirements of GPs by associating every component only with a slice of the overall data, thus limiting the negative effects of cubic scaling. While this motivation is understandable from a computational standpoint, it is often unclear how to decompose a nonlinear function into a mixture of *nonlinear* experts, especially when each expert has the representational flexibility of a GP. In contrast, models like ILR and HILR rely on locally linear experts that offer a natural unit of local approximation.

Finally, the infinite mixtures used in ILR and HILR to regularize model complexity are rooted in Bayesian nonparametrics. Here we reference influential work on Markov chain Monte Carlo (MCMC) techniques for Bayesian nonparametric density estimation [48], [49], [50], [51], which developed the first seeds of Bayesian inference for Dirichlet processes [52], [53]. These concepts inspired comparable infinite mixture regression models. However, these attempts exclusively relied on expensive Gibbs sampling algorithms [54], [55], [56], [57]. Instead, we focus on developing efficient deterministic variational inference (VI) algorithms that dramatically improve the practical aspects of training and deploying Bayesian finite and infinite mixture models [58], [59]. For a detailed discussion on the scalability of different paradigms of Bayesian inference, we refer the reader to [60].

This paper is organized as follows. In Section 2, we review Bayesian linear regression, finite and infinite mixture models, and variational inference. Section 3 presents the infinite local regression (ILR) model and a variational Bayes inference scheme [43]. In Section 4, we introduce the hierarchical infinite local regression (HILR), an extension of ILR, that incorporates multi-modal activations and parameter sharing. Finally, in Section 5, we test ILR and HILR on a set of simulated and real-world benchmarks.

2 PRELIMINARIES

This section introduces related concepts, such as Bayesian linear regression, Bayesian mixture models, Dirichlet processes, and variational inference.

2.1 Bayesian Linear Regression

We start by discussing the Bayesian treatment of a single component of a Bayesian local regression model, namely Bayesian linear regression [61]. The conditional data likelihood takes a feature vector $\mathbf{x} \in \mathbb{R}^m$ as a random input variable and returns a random response variable $\mathbf{y} \in \mathbb{R}^d$ according to a linear mapping $\mathbf{A} : \mathbb{R}^m \to \mathbb{R}^d$, a bias vector $\mathbf{c}$, and additive zero-mean noise with a precision matrix $\mathbf{V}$

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{c} + \mathbf{e}, \quad \mathbf{e} \sim \mathrm{N}(\mathbf{0}, \mathbf{V}).$$

For a fully Bayesian treatment, we consider all parameters of this model to be random variables on which we place proper conjugate priors. In this case, we place matrix-normal (MN) and normal-Wishart (NW) priors on the matrix $\mathbf{A}$, the bias coefficient $\mathbf{c}$, and precision matrix $\mathbf{V}$

$$p(\mathbf{A}, \mathbf{c}, \mathbf{V}) = \mathrm{MN}(\mathbf{A} \,|\, \mathbf{M}, \mathbf{V}, \mathbf{K})\,\mathrm{N}(\mathbf{c} \,|\, \boldsymbol{\theta}, \rho\mathbf{V})\,\mathrm{W}(\mathbf{V} \,|\, \boldsymbol{\Phi}, \eta),$$

where $\mathbf{M}$, the mean of $\mathbf{A}$, is a $d \times m$ matrix and $\mathbf{V}$ and $\mathbf{K}$ are $d \times d$ and $m \times m$ that serve as row and column precision matrices of $\mathbf{A}$, respectively. The mean $\boldsymbol{\theta}$ is an m-dimensional vector, and the scalar ρ modulates the amplitude of the precision. Finally, the parameters of the Wishart distribution are the $d \times d$ positive definite scale matrix $\boldsymbol{\Phi}$ and the degrees of freedom η. Due to the conjugate nature of the priors, the joint posteriors $p(\mathbf{A}, \mathbf{c}, \mathbf{V} \,|\, \mathcal{D})$ are matrix-normal and normal-Wishart distributions, conditioned on the data of N independent and identically distributed data pairs $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$.

2.2 Bayesian Finite Mixture Models

Gaussian mixture models (GMMs) are hierarchical latent variable models with universal approximation capabilities for arbitrary continuous densities. This insight is of central interest when connected to density estimation for local regression models, which are themselves universal nonlinear function approximators [5]. A finite K-component Gaussian mixture of a random variable $\mathbf{x}$ is a weighted linear combination of densities

$$p(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(\mathbf{z} = k \,|\, \boldsymbol{\pi})p(\mathbf{x} \,|\, \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \,\mathrm{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k),$$

with K unique mean vectors $\boldsymbol{\mu}_k$ and precision matrices $\boldsymbol{\Lambda}_k$. The latent quantity $\mathbf{z}$ is a one-hot random variable distributed according to a categorical distribution $p(\mathbf{z}) =$ $\mathrm{Cat}(\boldsymbol{\pi})$, governed by the weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ that satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

The Bayesian extension of this model [58] introduces a conjugate normal-Wishart prior on the means and precision matrices $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathrm{NW}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ contains the hyperparameters. Furthermore, a conjugate Dirichlet prior, with a concentration parameter α, is placed on the mixing weights $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. This Bayesian perspective has proven effective in regularizing the shortcomings of GMMs by allowing superfluous components to fall back onto their priors instead of severely over-fitting to small clusters. This effect can be understood as sparsification bias over K [62], [63].

2.3 Dirichlet Process and Stick-Breaking

Scaling the finite mixtures in Section 2.2 to infinite components requires placing a nonparametric Dirichlet process (DP) prior on the parameters of the mixture. A DP is a distribution over probability measures G. We write $\mathrm{G} \sim \mathrm{DP}(\alpha, \mathrm{H})$, where α is the concentration parameter and H is the base measure [64], [65]. Intuitively, a Dirichlet process is a distribution over distributions, meaning each draw G is itself a distribution. The base distribution H is the mean of the DP, and the concentration parameter α can be interpreted as an inverse variance. The larger α is, the smaller the variance, and the process concentrates more of its mass around the mean distribution H.

We will rely on the stick-breaking construction [66] of a DP as an algorithmic realization. Stick-breaking delivers an infinite sequence of mixture weights π_k of an infinite mixture model from the stochastic process

$$\pi_k = s_k \prod_{l=1}^{k-1} (1 - s_l), \quad s_k \sim \mathrm{Beta}(1, \alpha).$$

This process is sometimes denoted as $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha)$ [64]. The stick-breaking procedure describes how the random variables s_k, representing stick lengths, are drawn from a beta distribution and combined to obtain the mixture weights π_k. If the concentration parameter α increases, the magnitude of the mixing weights π_k decreases on average, and the number of possible active components increases.

This representation of DPs can be used to replace the priors placed on the finite Gaussian mixture model [59]. In such a setting, the base H is a normal-Wishart distribution, and the sampled measure $\mathrm{G} \sim \mathrm{DP}(\alpha, \mathrm{NW})$ is a draw of an unbounded number of parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathrm{NW}(\boldsymbol{\lambda})$ for an infinite number of clusters, associated with an infinite number of weights π_k generated by the stick-breaking process. These draws from a Dirichlet process are discrete with probability one, which leads to the clustering effect of the DP. Eventually, the same parameters will be sampled over and over, forcing the associated data points to cluster.

2.4 Variational Inference of Structured Models

MCMC [67], and VI [68] have become the two main approaches for (approximate) probabilistic inference in graphical models. While MCMC constructs a stochastic sampling process that converges to the posterior, VI formulates the inference task as a deterministic optimization problem. Despite its reliance on a coarser functional posterior approximation, VI is often preferred in settings with large number

of parameters. Moreover, the deterministic nature of VI circumvents the issue of label switching in Monte-Carlo-based posterior inference of mixture models.

In a nutshell, in variational inference, a typically intractable posterior is approximated by a tractable functional distribution $q(\boldsymbol{\beta})$ that minimizes the Kullback-Leibler divergence (KL) to true posterior $p(\boldsymbol{\beta}\,|\,\mathcal{D})$

$$q^*(\boldsymbol{\beta}) = \arg\min \quad \mathrm{KL}(q(\boldsymbol{\beta})\,||\,p(\boldsymbol{\beta}\,|\,\mathcal{D})),$$

where $\mathcal{D}$ is observed data and the vector $\boldsymbol{\beta}$ subsumes both the parameters $\boldsymbol{\theta}$ and hidden indicators $\mathbf{z}$ in structured models. Note that the KL in the above objective is mode-seeking, meaning it will lock on one mode of a possibly multi-modal posterior. In general, the posterior $p(\boldsymbol{\beta}\,|\,\mathcal{D})$ is unknown, because the normalizer $p(\mathcal{D})$ is not tractable. In consequence, the KL cannot be minimized directly but rather optimized via a related objective that is equal up to the constant term equivalent to the evidence $p(\mathcal{D})$

$$\begin{aligned}
\mathrm{KL}(q\,||\,p) &= \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\beta}\,|\,\mathcal{D})\right] \\
&= \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log p(\mathcal{D},\boldsymbol{\beta})\right] + p(\mathcal{D}) \\
&= \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log p(\mathcal{D},\boldsymbol{\beta})\right] + \mathrm{const.}
\end{aligned}$$

This modified objective is denoted as the negative evidence lower bound (ELBO) and can be reformulated to take the traditional maximization objective of VI algorithms

$$\begin{aligned}
\mathrm{ELBO}(q) &= \mathbb{E}_q\left[\log p(\mathcal{D},\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] \\
&= \mathbb{E}_q\left[\log p(\mathcal{D}\,|\,\boldsymbol{\beta})\right] + \mathbb{E}_q\left[\log p(\boldsymbol{\beta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\beta})\right] \\
&= \mathbb{E}_q\left[\log p(\mathcal{D}\,|\,\boldsymbol{\beta})\right] - \mathrm{KL}(q(\boldsymbol{\beta})\,||\,p(\boldsymbol{\beta})).
\end{aligned}$$

The choice of the approximate posterior distribution $q(\boldsymbol{\beta})$ is open. In this paper, we focus on variational Bayes methods that rely on the (structured) mean-field assumption as a general recipe for maximizing the ELBO [45], [69]. This approximation requires that the posterior factorizes over the set of the hidden variables $q(\boldsymbol{\beta}) = \prod_{i=1}^{M} q_i(\beta_i)$. It is emphasized that no other assumptions are made about $q(\boldsymbol{\beta})$. The resulting posterior will be determined solely by the assumed likelihood and priors.

Specifically, we follow the scheme of variational Bayes expectation-maximization (VBEM) as a probabilistic generalization of expectation-maximization (EM). This approach constitutes a coordinate ascent scheme that iteratively optimizes the ELBO for individual factors of the approximate posterior $q(\boldsymbol{\beta})$ while holding the others constant

$$\ln q_j(\beta_j) = \mathbb{E}_{q_{i\neq j}}\left[\log p(\mathcal{D},\boldsymbol{\beta})\right] + \mathrm{const.}$$

A more practical version of this optimization can be achieved by using stochastic variational inference (SVI), a batched stochastic gradient ascent approach. In the case of the conjugate exponential family, SVI not only facilitates scalability over large datasets but also resembles a natural gradient ascent algorithm on the ELBO with favorable convergence properties [70].

3 THE INFINITE MIXTURE OF LOCAL LINEAR REGRESSION MODEL

Using the previously presented concepts of Bayesian linear regression, Bayesian mixture models, and Dirichlet processes, we now construct the Bayesian infinite local regression (ILR) model.

Our approach to tackling the regression problem is a Bayesian joint density estimation technique over the input $\mathbf{x}$ and target $\mathbf{y}$, both conditioned on a latent discrete label $\mathbf{z}$. To avoid the need for a fixed number of components over $\mathbf{z}$, we assume a generative model driven by a stick-breaking process with a concentration parameter α and a base distribution H, which is a product of a normal-Wishart, a Matrix-normal, and Wishart distribution.

The data generation proceeds as depicted in Figure 2. The stick-breaking process is sampled to generate the categorical weights $\boldsymbol{\pi}$, mixture activations $\{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{k=1}^{\infty}$, and regression parameters $\{\mathbf{A}_k, \mathbf{c}_k, \mathbf{V}_k\}_{k=1}^{\infty}$

$$\boldsymbol{\pi}(\mathbf{s}) \sim \mathrm{GEM}(\alpha),$$
$$\boldsymbol{\Lambda}_k \sim \mathrm{W}(\boldsymbol{\Psi},\nu), \ \boldsymbol{\mu}_k \sim \mathrm{N}(\mathbf{m},\kappa\boldsymbol{\Lambda}_k),$$
$$\mathbf{V}_k \sim \mathrm{W}(\boldsymbol{\Phi},\eta), \ \mathbf{A}_k \sim \mathrm{MN}(\mathbf{M},\mathbf{K},\mathbf{V}_k), \ \mathbf{c}_k \sim \mathrm{N}(\boldsymbol{\theta},\rho\mathbf{V}_k),$$

and those parameters are used to generate the one-hot labels $\mathbf{z}_n$ and input-output data pairs $\mathbf{x}_n, \mathbf{y}_n$

$$\mathbf{z}_n \sim \mathrm{Cat}(\boldsymbol{\pi}(\mathbf{s})), \ \mathbf{x}_n \sim \mathrm{N}(\boldsymbol{\mu}_{z_n},\boldsymbol{\Lambda}_{z_n}),$$
$$\mathbf{y}_n \sim \mathrm{N}(\mathbf{A}_{z_n}\mathbf{x}_n + \mathbf{c}_{z_n}, \mathbf{V}_{z_n}).$$

Notice that the densities over the input space, parameterized by $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$, naturally play the role of basis functions or so-called receptive fields as in the receptive field weighted regression (RFWR) [31] and locally weighted projection regression (LWPR) [32] algorithms.

Given these modeling assumptions, the regression objective can be cast as a joint density inference problem of the posterior $p(\mathbf{Z},\mathbf{s},\boldsymbol{\mu},\boldsymbol{\Lambda},\mathbf{A},\mathbf{c},\mathbf{V}\,|\,\mathbf{X},\mathbf{Y})$, where $\mathbf{X} = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, $\mathbf{Y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_N\}$, and $\mathbf{Z} = \{\mathbf{z}_1,\ldots,\mathbf{z}_N\}$. The result of this inference is then leveraged to perform predictions via conditioning and marginalization.

3.1 Variational Bayes for ILR

For inference, we focus on a variational Bayes EM algorithm that alternates between a variational expectation step (E-step) and a maximization step (M-step). Deriving such an algorithm for this model requires pinning down the following definitions of the likelihood, prior, and posterior.

Complete Data Likelihood. For the general case of multivariate regression with m inputs and d outputs, we assume the following structured joint likelihood over the inputs, outputs, and indicator variables

$$\begin{aligned}
p(\mathbf{X},\mathbf{Y},\mathbf{Z}\,|\,.) &= p(\mathbf{Z})\,p(\mathbf{X}\,|\,\mathbf{Z})\,p(\mathbf{Y}\,|\,\mathbf{X},\mathbf{Z}) \\
&= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{z}_n\,|\,\boldsymbol{\pi}(\mathbf{s})) \\
&\quad \times \prod_{n=1}^{N}\prod_{k=1}^{\infty} \mathrm{N}(\mathbf{x}_n\,|\,\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)^{z_{nk}} \\
&\quad \times \prod_{n=1}^{N}\prod_{k=1}^{\infty} \mathrm{N}(\mathbf{y}_n\,|\,\mathbf{A}_k\mathbf{x}_n + \mathbf{c}_k, \mathbf{V}_k)^{z_{nk}},
\end{aligned}$$

where the dimensions of all quantities follow the notation of Bayesian linear regression.

Infinite Conjugate Prior. We construct a factorized conjugate infinite mixture prior

$$\begin{aligned}
p(\mathbf{s},\boldsymbol{\mu},\boldsymbol{\Lambda},\mathbf{A},\mathbf{c},\mathbf{V}) &= p(\mathbf{s})\,p(\boldsymbol{\mu}|\boldsymbol{\Lambda})\,p(\boldsymbol{\Lambda}) \\
&\quad \times p(\mathbf{A}|\mathbf{V})\,p(\mathbf{c}|\mathbf{V})\,p(\mathbf{V}).
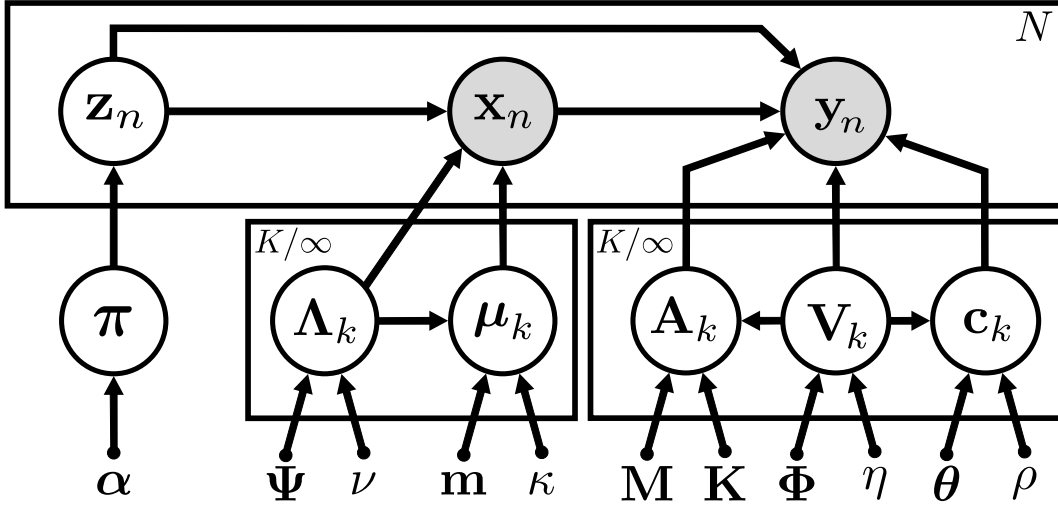\end{aligned}$$

Fig. 2: A unified plate notation for infinite mixtures of Bayesian local regression. Assuming Gaussian and linear Gaussian densities, the basis parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ are sampled from a normal-Wishart distribution, while the regression parameters $(\mathbf{A}_k, \mathbf{c}_k)$ and precision matrices $\mathbf{V}_k$ are sampled from a matrix-Normal-Wishart, for every component k. The latent variables $\mathbf{z}_n$ assign every $\mathbf{x}_n$ and $\mathbf{y}_n$ to a component and are drawn from a categorical distribution parameterized by $\boldsymbol{\pi}$. The mixture weights $\boldsymbol{\pi}$ are generated by a stick-breaking process with a concentration parameter α.

This prior assumes that the cluster means $\boldsymbol{\mu}_k$ and precision matrices $\boldsymbol{\Lambda}_k$ are sampled from normal-Wishart distributions

$$p(\boldsymbol{\mu} \,|\, \boldsymbol{\Lambda})\, p(\boldsymbol{\Lambda}) = \prod_{k=1}^{\infty} \mathrm{N}(\boldsymbol{\mu}_k \,|\, \mathbf{m}_0, \kappa_0 \boldsymbol{\Lambda}_k)\, \mathrm{W}(\boldsymbol{\Lambda}_k \,|\, \boldsymbol{\Psi}_0, \nu_0),$$

while matrix-normal-Wishart and normal-Wishart priors are placed on the regression coefficients $(\mathbf{A}_k, \mathbf{c}_k)$ and the precision matrices $\mathbf{V}_k$

$$\begin{aligned} p(\mathbf{A} \,|\, \mathbf{V})p(\mathbf{c} \,|\, \mathbf{V})p(\mathbf{V}) = \prod_{k=1}^{\infty} & \mathrm{MN}(\mathbf{A}_k \,|\, \mathbf{M}_0, \mathbf{K}_0, \mathbf{V}_k) \\ & \times \mathrm{N}(\mathbf{c}_k \,|\, \boldsymbol{\theta}_0, \rho_0 \mathbf{V}_k)\, \mathrm{W}(\mathbf{V}_k \,|\, \boldsymbol{\Phi}_0, \eta_0). \end{aligned}$$

The parameters π_k are generated by a stick-breaking process $\pi_k(\mathbf{s}) = s_k \prod_{l=1}^{k-1}(1 - s_l)$, where the stick lengths $\mathbf{s}$ are independently beta distributed

$$p(\mathbf{s}) = \prod_{k=1}^{\infty} \mathrm{Beta}(s_k \,|\, 1, \alpha_0).$$

Truncated Mean-Field Posterior. We rely on a structured mean-field approximation of the posterior [69] that factorizes between the labels $q(\mathbf{Z})$ and the remaining parameters $q(\mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V})$, thus automatically leading to the decomposition $p(. \,|\, \mathcal{D}) \approx q(\mathbf{Z})\, q(\mathbf{s})\, q(\boldsymbol{\mu}, \boldsymbol{\Lambda})\, q(\mathbf{A}, \mathbf{c}, \mathbf{V})$. Further, we follow [59] by allowing a truncation of the posterior while maintaining an infinite prior, so that $q(s_K = 1) = 1$, implying that $\pi_k = 0$ for $k > K$

$$\begin{aligned} p(. \,|\, \mathcal{D}) \approx \prod_{n=1}^{N} & \mathrm{Cat}(\mathbf{z}_n \,|\, \mathbf{r}_n) \prod_{k=1}^{K-1} \mathrm{Beta}(s_k \,|\, \gamma_k, \alpha_k) \\ & \times \prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_k \,|\, \mathbf{m}_k, \kappa_k \boldsymbol{\Lambda}_k)\, \mathrm{W}(\boldsymbol{\Lambda}_k \,|\, \boldsymbol{\Psi}_k, \nu_k) \\ & \times \prod_{k=1}^{K} \mathrm{MN}(\mathbf{A}_k \,|\, \mathbf{M}_k, \mathbf{K}_k, \mathbf{V}_k) \\ & \times \mathrm{N}(\mathbf{c}_k \,|\, \boldsymbol{\theta}_k, \rho_k \mathbf{V}_k)\, \mathrm{W}(\mathbf{V}_k \,|\, \boldsymbol{\Phi}_k, \eta_k), \end{aligned}$$

where $\mathbf{r}_n$ are the expected responsibilities of the mixture. During evaluation, the truncation threshold K is chosen to be very high and is seldom reached.

Variational Expectation Step. In the E-step, the responsibilities are computed by following the recipe of VBEM in Section 2.4. The responsibilities are the variational parameters of the posterior categorical

$$\begin{aligned} \log q(\mathbf{Z}) &= \mathbb{E}_{q(\mathbf{s})}\left[\log p(\mathbf{Z} \,|\, \boldsymbol{\pi}(\mathbf{s}))\right] + \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})}\left[\log p(\mathbf{X} \,|\, \mathbf{Z})\right] \\ &\quad + \mathbb{E}_{q(\mathbf{A}, \mathbf{c}, \mathbf{V})}\left[\log p(\mathbf{Y} \,|\, \mathbf{X}, \mathbf{Z})\right] + \mathrm{const} \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log r_{nk}. \end{aligned}$$

The expectations associated with the data likelihoods of $\mathbf{x}_n$ and $\mathbf{y}_n$ can be straightforwardly computed [71]. However, the expectations associated with infinite-dimensional categorical require more careful consideration [59]. We provide the necessary details in the appendix.

Variational Maximization Step. The M-step updates the remaining variational distributions given an approximation of the categorical posterior as follows

$$\begin{aligned} \log q(\mathbf{s}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{Z} \,|\, \boldsymbol{\pi}(\mathbf{s}))\right] + \log p(\mathbf{s}) + \mathrm{const}, \\ \log q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{X} \,|\, \mathbf{Z})\right] + \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) + \mathrm{const}, \\ \log q(\mathbf{A}, \mathbf{c}, \mathbf{V}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{Y} \,|\, \mathbf{X}, \mathbf{Z})\right] \\ &\quad + \log p(\mathbf{A}, \mathbf{c}, \mathbf{V}) + \mathrm{const}. \end{aligned}$$

Each of these updates reflects a conjugate computation of K log-posterior densities given a log-prior and a log-likelihood weighted by the responsibilities $r_{nk} = \mathbb{E}[z_{nk}]$. We provide more details and general computational recipes based on conjugacy rules in the appendix.

3.2 Posterior Predictive Distribution

For predicting the function value $\hat{\mathbf{y}}$ conditioned on a test query $\hat{\mathbf{x}}$, we marginalize the likelihood over the posterior parameters to get the joint posterior predictive density. To

make the marginalization tractable, we replace the true posterior with our approximate variational posterior $q(.\,|\,\mathcal{D})$ inferred under a training dataset $\mathcal{D}$. The conditional predictive for a single component $\mathbf{z} = k$ is a conditional multivariate Student's t-distribution of the form

$$p(\hat{\mathbf{y}}\,|\,\hat{\mathbf{x}}, \hat{\mathbf{z}} = k, \mathcal{D}) = \mathbb{E}_{q(\mathbf{A}, \mathbf{c}, \mathbf{V})}\left[p(\hat{\mathbf{y}}\,|\,\hat{\mathbf{x}}, \mathbf{A}_k, \mathbf{c}_k, \mathbf{V}_k)\right]$$
$$= \mathrm{T}\left(\mathbf{M}_k\hat{\mathbf{x}} + \boldsymbol{\theta}_k, a_k\boldsymbol{\Phi}_k, \eta_k + 1\right),$$

where we have defined

$$a_k = 1 - \mathbf{u}^\top\left(\mathbf{L}_k + \mathbf{u}\mathbf{u}^\top\right)^{-1}\mathbf{u},$$

with $\mathbf{u} = [\hat{\mathbf{x}},\,1]^\top$ and $\mathbf{L}_k = \mathrm{Block}\left(\mathbf{K}_k, \rho_k\right)$.

Additionally, the joint activation of a component k is a Student's t-distribution weighted by the expected categorical probability under the posterior stick-breaking process

$$p(\hat{\mathbf{x}}, \hat{\mathbf{z}} = k\,|\,\mathcal{D}) \propto \mathbb{E}_{q(\mathbf{s})}\left[p(\hat{\mathbf{z}} = k\,|\,\boldsymbol{\pi}(\mathbf{s}))\right]\mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})}\left[p(\hat{\mathbf{x}}\,|\,\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)\right]$$
$$= \frac{\gamma_k}{\gamma_k + \alpha_k}\prod_{l=1}^{k-1}\left(1 - \frac{\gamma_l}{\gamma_l + \alpha_l}\right)$$
$$\times \mathrm{T}\left(\boldsymbol{\mu}_k, \frac{\kappa_k}{1 + \kappa_k}\boldsymbol{\Psi}_k, \nu_k + 1\right).$$

These K-activation probabilities enable two prediction techniques. A *mode-prediction*, where the most likely active component is selected and used to perform prediction with the corresponding linear regression model, or a *mean-prediction*, that averages the predictions of all components weighted by their activation probabilities.

3.3 Computational Complexity

We calculate the training-time computational cost to be $\mathcal{O}(NK(d + m)^3)$, which can be straightforwardly reduced to $\mathcal{O}(LK(d + m)^3)$ by applying stochastic updates [70], where L is the batch size. This result shows linear scalability with the data, which is considerably more efficient than simple variants of GPR. The test-time complexity of a mean prediction is $\mathcal{O}(K(d^3 + dm))$, which combines the input membership query and the linear matrix transformation for every model k. This computation is, in contrast to GPR, independent of the training data size, hence the advantage of memoryless locally-parametric representations during real-time critical applications.

4 THE HIERARCHICAL INFINITE MIXTURE OF LOCAL LINEAR REGRESSION MODEL

The local regression model presented in Section 3 offers a very flexible and well-regularized alternative to previously developed approaches [30], [32], [42]. However, like similar local regression representations, it cannot account for shift-invariance across the input space. This drawback can cause it to generate duplicate regression components to account for similar local function trends across disconnected regions of the input space. This limitation results from the hierarchical design that directly couples activations and local function units via a one-to-one correspondence and enforces a uni-modal activation per regression component. This coupling can be observed in the definition of the likelihood in Section 3, where the activation and the local regression units share the same assignment variable.

It stands to reason that ILR does not offer enough flexibility and hinders parameter sharing between components. Therefore, we present a modified formulation of ILR that explicitly accounts for shift-invariance in the input space and provides the freedom to create regression units with multi-modal, theoretically infinitely-modal, activations, if needed. The resulting model, hierarchical infinite local regression (HILR), is an infinite mixture over infinite mixtures that shares similarities with existing representations developed for hierarchical clustering [72], [73], [74].

We start by describing the generative process of the hierarchical mixture over mixtures as depicted in Figure 3. HILR consists of two levels. At the upper level, a meta Dirichlet process generates the stick-breaking weights $\boldsymbol{\omega}$, the meta-activations $\{\boldsymbol{\tau}_m, \boldsymbol{\Lambda}_m\}_{m=1}^\infty$, and the shared slope and output precision matrices $\{\mathbf{A}_m, \mathbf{V}_m\}_{m=1}^\infty$

$$\boldsymbol{\omega}(\mathbf{t}) \sim \mathrm{GEM}(\beta),$$
$$\boldsymbol{\Lambda}_m \sim \mathrm{W}(\boldsymbol{\Psi}, \nu),\ \boldsymbol{\tau}_m \sim \mathrm{N}(\mathbf{m}, \lambda\boldsymbol{\Lambda}_m),$$
$$\mathbf{V}_m \sim \mathrm{W}(\boldsymbol{\Phi}, \eta),\ \mathbf{A}_m \sim \mathrm{MN}(\mathbf{M}, \mathbf{K}, \mathbf{V}_m),$$

where β is the concentration parameter of the upper-level DP and $\mathbf{t}$ are the generated stick lengths.

HILR's ability to account for shift-invariance is a result of allowing multi-modal activations. That way, every regression unit can be associated with multiple local regions in the input space. This structure is realized by endowing every upper-level component m with its own local DP. These lower-level DPs, with a concentration parameter α, generate the weights $\boldsymbol{\pi}_m$, the activation centers $\{\boldsymbol{\mu}_{mk}\}_{k=1}^\infty$ and the shift coefficients $\{\mathbf{c}_{mk}\}_{k=1}^\infty$

$$\boldsymbol{\pi}_m(\mathbf{s}_m) \sim \mathrm{GEM}(\alpha),$$
$$\boldsymbol{\mu}_{mk} \sim \mathrm{N}(\boldsymbol{\tau}_m, \kappa\boldsymbol{\Lambda}_m),\ \mathbf{c}_{mk} \sim \mathrm{N}(\boldsymbol{\theta}, \rho\mathbf{V}_m).$$

We assume here that the lower-level centers $\boldsymbol{\mu}_{mk}$ and coefficients $\mathbf{c}_{mk}$ are tied via their respective upper-level precision matrices $\boldsymbol{\Lambda}_m$ and $\mathbf{V}_m$.

Finally, given all necessary parameters, the upper- and lower-level labels $\mathbf{h}_n, \mathbf{z}_n$ and the data pairs $\mathbf{x}_n, \mathbf{y}_n$ are sampled according to the likelihoods

$$\mathbf{h}_n \sim \mathrm{Cat}(\boldsymbol{\omega}(\mathbf{t})),\ \mathbf{z}_n \sim \mathrm{Cat}(\boldsymbol{\pi}(\mathbf{s}), \mathbf{h}_n),$$
$$\mathbf{x}_n \sim \mathrm{N}(\boldsymbol{\mu}_{h_n, z_n}, \boldsymbol{\Lambda}_{h_n}),\ \mathbf{y}_n \sim \mathrm{N}(\mathbf{A}_{h_n}\mathbf{x}_n + \mathbf{c}_{h_n, z_n}, \mathbf{V}_{h_n}).$$

Notice how the upper-level generative process resembles the structure of ILR. However, in this model, the meta-activations $(\boldsymbol{\tau}_m, \boldsymbol{\Lambda}_m)$ induce another hierarchy over the lower-level multi-modal activation centers $\boldsymbol{\mu}_{mk}$. Moreover, the upper-level DP only accounts for the shared slope of the local regression units $\mathbf{A}_m$, whereas the shift coefficients $\mathbf{c}_{mk}$ are induced at the lower level, as they are influenced by the individual activation centers $\boldsymbol{\mu}_{mk}$.

Similar to the procedure in Section 3, we aim to infer a posterior over the parameters of this model given a set of inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and outputs $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$. To that end, we derive a structured variational Bayes expectation-maximization algorithm for this model. We start the derivation by defining the complete data likelihood, the infinite prior, and the mean-field posterior factorization.
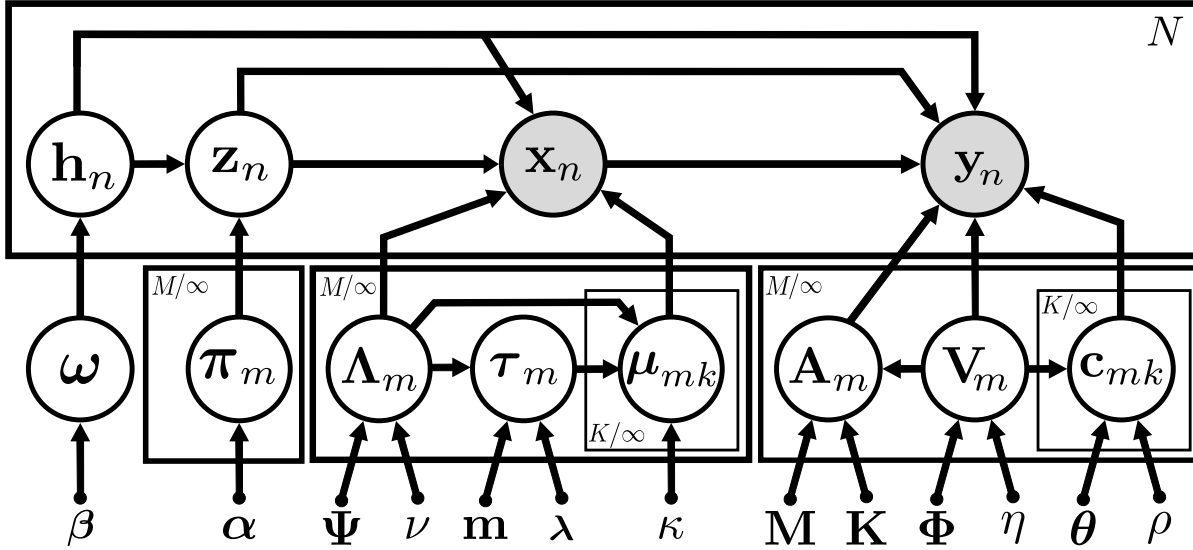
Fig. 3: A unified plate notation for hierarchical infinite tied mixtures of Bayesian local regression models. This model outlines a two-level architecture that allows sharing of parameters between single components in order to compress the representation. It can be interpreted as a local regression model with multi-modal activations. Each unit of the upper level is itself a mixture of local regression models that share the same slope $\mathbf{A}_m$ and output precision $\mathbf{V}_m$. Each of these m different slopes can be activated at k unique lower-level input regions centered around $\boldsymbol{\mu}_{mk}$ and tied via a shared input precision $\boldsymbol{\Lambda}_m$. Both the upper- and lower-level mixtures are governed by independent Dirichlet process priors.

Complete Data Likelihood. The likelihood model is a two-level precision-tied joint density over the observations $(\mathbf{X}, \mathbf{Y})$ and the one-hot upper- and lower-labels $(\mathbf{H}, \mathbf{Z})$

$$p(.) = p(\mathbf{H})\, p(\mathbf{Z} \,|\, \mathbf{H})\, p(\mathbf{X} \,|\, \mathbf{H}, \mathbf{Z})\, p(\mathbf{Y} \,|\, \mathbf{H}, \mathbf{Z}, \mathbf{X})$$
$$= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{h}_n \,|\, \boldsymbol{\omega}(\mathbf{t}))$$
$$\times \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{z}_n \,|\, \boldsymbol{\pi}(\mathbf{s}), \mathbf{h}_n)$$
$$\times \prod_{n=1}^{N} \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \mathrm{N}(\mathbf{x}_n \,|\, \boldsymbol{\mu}_{mk}, \boldsymbol{\Lambda}_m)^{z_{nk} \times h_{nm}}$$
$$\times \prod_{n=1}^{N} \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \mathrm{N}(\mathbf{y}_n \,|\, \mathbf{A}_m \mathbf{x}_n + \mathbf{c}_{mk}, \mathbf{V}_m)^{z_{nk} \times h_{nm}},$$

where the conditioning of the lower-level labels $\mathbf{z}_n$ on the upper-level labels $\mathbf{h}_n$ manifests as a one-hot selector of the corresponding lower-level categoricals

$$\mathrm{Cat}(\mathbf{z}_n \,|\, \boldsymbol{\pi}(\mathbf{s}), \mathbf{h}_n) = \prod_{m=1}^{\infty} \mathrm{Cat}(\mathbf{z}_n \,|\, \boldsymbol{\pi}_m(\mathbf{s}_m))^{h_{nm}}$$
$$= \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \pi_{mk}^{z_{nk} \times h_{nm}}.$$

Infinite Conjugate Prior. We assume a factorized two-level precision-tied conjugate infinite mixture prior

$$p(\mathbf{t}, \mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V}) = p(\mathbf{t})\, p(\mathbf{s})$$
$$\times p(\boldsymbol{\mu} \,|\, \boldsymbol{\tau}, \boldsymbol{\Lambda})\, p(\boldsymbol{\tau} \,|\, \boldsymbol{\Lambda})\, p(\boldsymbol{\Lambda})$$
$$\times p(\mathbf{A} \,|\, \mathbf{V})\, p(\mathbf{c}|\mathbf{V})\, p(\mathbf{V}).$$

The meta-activation prior is a normal-Wishart distribution over the meta-centers $\boldsymbol{\tau}_m$ and precision matrices $\boldsymbol{\Lambda}_m$

$$p(\boldsymbol{\tau} \,|\, \boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) = \prod_{m=1}^{\infty} \mathrm{N}(\boldsymbol{\tau}_m \,|\, \mathbf{m}_0, \lambda_0 \boldsymbol{\Lambda}_m)\, \mathrm{W}(\boldsymbol{\Lambda}_m \,|\, \boldsymbol{\Psi}_0, \nu_0),$$

while the activation centers $\boldsymbol{\mu}_{mk}$ are sampled from a conditional normal distribution

$$p(\boldsymbol{\mu} \,|\, \boldsymbol{\tau}, \boldsymbol{\Lambda}) = \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \mathrm{N}(\boldsymbol{\mu}_{mk} \,|\, \boldsymbol{\tau}_m, \kappa_0 \boldsymbol{\Lambda}_m).$$

The mappings $\mathbf{A}_m$ and precision matrices $\mathbf{V}_m$ are sampled form a matrix-normal-Wishart

$$p(\mathbf{A} \,|\, \mathbf{V})p(\mathbf{V}) = \prod_{m=1}^{\infty} \mathrm{MN}(\mathbf{A}_m \,|\, \mathbf{M}_0, \mathbf{K}_0, \mathbf{V}_m)\, \mathrm{W}(\mathbf{V}_m \,|\, \boldsymbol{\Phi}_0, \eta_0).$$

while the biases $\mathbf{c}_{mk}$ are drawn from a K-tied conditional normal distribution

$$p(\mathbf{c} \,|\, \mathbf{V}) = \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \mathrm{MN}(\mathbf{c}_{mk} \,|\, \boldsymbol{\theta}_0, \rho_0 \mathbf{V}_m).$$

Finally, the stick-breaking priors $p(\mathbf{t}, \mathbf{s})$ follow the definitions from Section 3

$$p(\mathbf{t}) = \prod_{m=1}^{\infty} \mathrm{Beta}(t_m \,|\, 1, \beta_0),$$
$$p(\mathbf{s}) = \prod_{m=1}^{\infty} \prod_{k=1}^{\infty} \mathrm{Beta}(s_{mk} \,|\, 1, \alpha_0).$$

Truncated Mean-Field Posterior. We assume a structured decomposition of the posterior that leads to conjugate computation while maintaining the dependencies between the discrete labels, the input activations, and the regression parameters, respectively. Moreover, we apply the truncation

scheme from [59] to establish a tractable posterior approximation while maintaining an infinite prior

$$p(.\,|\,\mathcal{D}) \approx q(\mathbf{H})\,q(\mathbf{Z}\,|\,\mathbf{H})\,q(\mathbf{t})\,q(\mathbf{s})\,q(\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{\Lambda})\,q(\mathbf{A},\mathbf{c},\mathbf{V})$$

$$\begin{aligned}
&= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{h}_n\,|\,\mathbf{g}_n) \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{z}_n\,|\,\mathbf{r}_n,\mathbf{h}_n)\\
&\times \prod_{m=1}^{M-1} \mathrm{Beta}(t_m\,|\,\delta_m,\beta_m)\\
&\times \prod_{m=1}^{M-1}\prod_{k=1}^{K-1} \mathrm{Beta}(s_{mk}\,|\,\gamma_{mk},\alpha_{mk})\\
&\times \prod_{m=1}^{M} \mathrm{N}(\boldsymbol{\tau}_m\,|\,\mathbf{m}_m,\lambda_m\boldsymbol{\Lambda}_m)\,\mathrm{W}(\boldsymbol{\Lambda}_m\,|\,\boldsymbol{\Psi}_m,\nu_m)\\
&\times \prod_{m=1}^{M} \mathrm{MN}(\mathbf{A}_m\,|\,\mathbf{M}_m,\mathbf{K}_m,\mathbf{V}_m)\,\mathrm{W}(\mathbf{V}_m\,|\,\boldsymbol{\Phi}_m,\eta_m)\\
&\times \prod_{m=1}^{M}\prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_{mk}\,|\,\boldsymbol{\tau}_m,\kappa_{mk}\boldsymbol{\Lambda}_m)\,\mathrm{N}(\mathbf{c}_{mk}\,|\,\boldsymbol{\theta}_{mk},\rho_{mk}\mathbf{V}_k),
\end{aligned}$$

where $\mathbf{g}_n$ and $\mathbf{r}_n$ are the upper- and lower-level posterior responsibilities, respectively.

Variational Expectation Step. The E-step computes the joint posterior categorical over joint labels $\mathbf{H}$ and $\mathbf{Z}$

$$\begin{aligned}
\log q(\mathbf{Z}\,|\,\mathbf{H}) &= \mathbb{E}_{q(\mathbf{s})}\left[\log p(\mathbf{Z}\,|\,\mathbf{H})\right]\\
&\quad + \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log p(\mathbf{X}\,|\,\mathbf{H},\mathbf{Z})\right]\\
&\quad + \mathbb{E}_{q(\mathbf{A},\mathbf{c},\mathbf{V})}\left[\log p(\mathbf{Y}\,|\,\mathbf{H},\mathbf{Z},\mathbf{X})\right] + \mathrm{const}\\
&= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K} h_{nm}z_{nmk}\log r_{nmk},
\end{aligned}$$

$$\begin{aligned}
\log q(\mathbf{H}) &= \mathbb{E}_{q(\mathbf{t})}\left[\log p(\mathbf{H})\right] + \log q(\mathbf{Z}\,|\,\mathbf{H}) + \mathrm{const}\\
&= \sum_{n=1}^{N}\sum_{m=1}^{M} h_{nm}\log g_{nm},
\end{aligned}$$

where these expectations can computed in a similar fashion to Section 3. More details can be found in the appendix.

Variational Maximization Step. The M-step updates the variational gating, activations, and regression parameters

$$\log q(\mathbf{t}) = \mathbb{E}_{q(\mathbf{H})}\left[\log p(\mathbf{H})\right] + \log p(\mathbf{t}) + \mathrm{const},$$

$$\log q(\mathbf{s}) = \mathbb{E}_{q(\mathbf{H},\mathbf{Z})}\left[\log p(\mathbf{Z}\,|\,\mathbf{H})\right] + \log p(\mathbf{s}) + \mathrm{const},$$

$$\begin{aligned}
\log q(\boldsymbol{\mu}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log p(\mathbf{X}\,|\,\mathbf{H},\mathbf{Z})\right]\\
&\quad + \mathbb{E}_{q(\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log p(\boldsymbol{\mu}\,|\,\boldsymbol{\tau},\boldsymbol{\Lambda})\right] + \mathrm{const},
\end{aligned}$$

$$\begin{aligned}
\log q(\boldsymbol{\tau},\boldsymbol{\Lambda}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\mu})}\left[\log p(\mathbf{X}\,|\,\mathbf{H},\mathbf{Z})\right] + \log p(\boldsymbol{\tau},\boldsymbol{\Lambda})\\
&\quad + \mathbb{E}_{q(\boldsymbol{\mu})}\left[\log p(\boldsymbol{\mu}\,|\,\boldsymbol{\tau},\boldsymbol{\Lambda})\right] + \mathrm{const},
\end{aligned}$$

$$\begin{aligned}
\log q(\mathbf{c}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{A},\mathbf{V})}\left[\log p(\mathbf{Y}\,|\,\mathbf{H},\mathbf{Z},\mathbf{X})\right]\\
&\quad + \mathbb{E}_{q(\mathbf{V})}\left[\log p(\mathbf{c}\,|\,\mathbf{V})\right] + \mathrm{const},
\end{aligned}$$

$$\begin{aligned}
\log q(\mathbf{A},\mathbf{V}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{c})}\left[\log p(\mathbf{Y}\,|\,\mathbf{H},\mathbf{Z},\mathbf{X})\right] + \log p(\mathbf{A},\mathbf{V})\\
&\quad + \mathbb{E}_{q(\mathbf{c})}\left[\log p(\mathbf{c}\,|\,\mathbf{V})\right] + \mathrm{const}.
\end{aligned}$$

As previously stated, these updates resemble posterior computations weighted by the responsibilities $g_{nm} = \mathbb{E}[h_{nm}]$ and $r_{nmk} = \mathbb{E}[z_{nmk}]$. The appendix provides further details.

4.1 Posterior Predictive Distribution

Prediction with HILR is akin to that with ILR, as described in Section 3.2. We briefly state the conditional predictive for a component $\mathbf{h} = m$ and an activation $\mathbf{z} = k$ as

$$\begin{aligned}
p(\hat{\mathbf{y}}\,|\,\hat{\mathbf{x}},\hat{\mathbf{h}}{=}m,\hat{\mathbf{z}}{=}k,\mathcal{D}) &= \mathbb{E}_{q(\mathbf{A},\mathbf{c},\mathbf{V})}\left[p(\hat{\mathbf{y}}\,|\,\hat{\mathbf{x}},\mathbf{A}_m,\mathbf{c}_{mk},\mathbf{V}_m)\right]\\
&= \mathrm{T}\left(\mathbf{M}_m\hat{\mathbf{x}} + \boldsymbol{\theta}_{mk}, a_{mk}\boldsymbol{\Phi}_m, \eta_m + 1\right),
\end{aligned}$$

where the computation of a_{mk} is similar to that in Section 3.2. Further, the weight of the $k-$th activation of the $m-$th component is computed as follows

$$\begin{aligned}
p(\hat{\mathbf{x}},\hat{\mathbf{h}}{=}m,\hat{\mathbf{z}}{=}k\,|\,\mathcal{D}) &\propto \mathbb{E}_{q(\mathbf{t})}\left[p(\hat{\mathbf{h}}{=}m\,|\,\boldsymbol{\omega}(\mathbf{t}))\right]\\
&\quad \times \mathbb{E}_{q(\mathbf{s}_m)}\left[p(\hat{\mathbf{z}}{=}k\,|\,\boldsymbol{\pi}_m(\mathbf{s}_m))\right]\\
&\quad \times \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[p(\hat{\mathbf{x}}\,|\,\boldsymbol{\mu}_{mk},\boldsymbol{\Lambda}_m)\right]\\
&= \frac{\delta_m}{\delta_m + \beta_m}\prod_{l=1}^{m-1}\left(1 - \frac{\delta_l}{\delta_l + \beta_l}\right)\\
&\quad \times \frac{\gamma_{mk}}{\gamma_{mk} + \alpha_{mk}}\prod_{l=1}^{k-1}\left(1 - \frac{\gamma_{ml}}{\gamma_{ml} + \alpha_{ml}}\right)\\
&\quad \times \mathrm{T}\left(\boldsymbol{\mu}_{mk}, \frac{\kappa_{mk}}{1 + \kappa_{mk}}\boldsymbol{\Psi}_m, \nu_m + 1\right).
\end{aligned}$$

5 EMPIRICAL EVALUATION

We evaluate the presented models on a range of tasks. Our goals are (1) to highlight the advantages of ILR and HILR, such as dealing with out-of-distribution predictions, recovering an input-dependent noise function, hierarchical gating, sharing parameters, and the ability to perform Bayesian sequential updates, (2) to benchmark the models on high dimensional datasets from real robots, and (3) to deploy the models in a real-world scenario to further empirically demonstrate its validity. A reference implementation can be found under https://github.com/hanyas/mimo.

Out-of-distribution Uncertainty. In Figure 1 (left), we apply ILR on a synthetic dataset with two large gaps. We observe how the predictive uncertainty strongly reflects the lack of training data in these regions and how the mean prediction falls back to the prior values. This example highlights the model's reasonable uncertainty quantification. The out-of-distribution behavior of ILR is strongly influenced by the discrete gating and a query's activation probability that jointly define the overall membership weights.

Heteroscedastic Noise. We test on two different problems with input-dependent noise, the cosmic microwave background (CMB) [75], Figure 1 (right), and a synthetic dataset from a stochastic sinc function $y(x) = \mathrm{sinc}(x) + \epsilon$, where the noise ϵ is distributed according to zero-mean normal with a standard deviation $\sigma_\epsilon(x) = 0.05 + 0.2(1 + \sin(2x))/(1 + e^{-0.2x})$. In Figure 4 (left), we see that ILR can approximate the nonlinear functions well. In particular, the heteroscedastic noise function is recovered in great detail.

Hierarchical Parameter Sharing. In Figure 4 (right), we test HILR's ability to share slope parameters via multimodal activations. We consider a dataset stemming from a periodic triangle signal overlayed with additive noise. HILR decides to activate two upper- and two lower-level regions to match the data structure, despite having more degrees of freedom at each level.
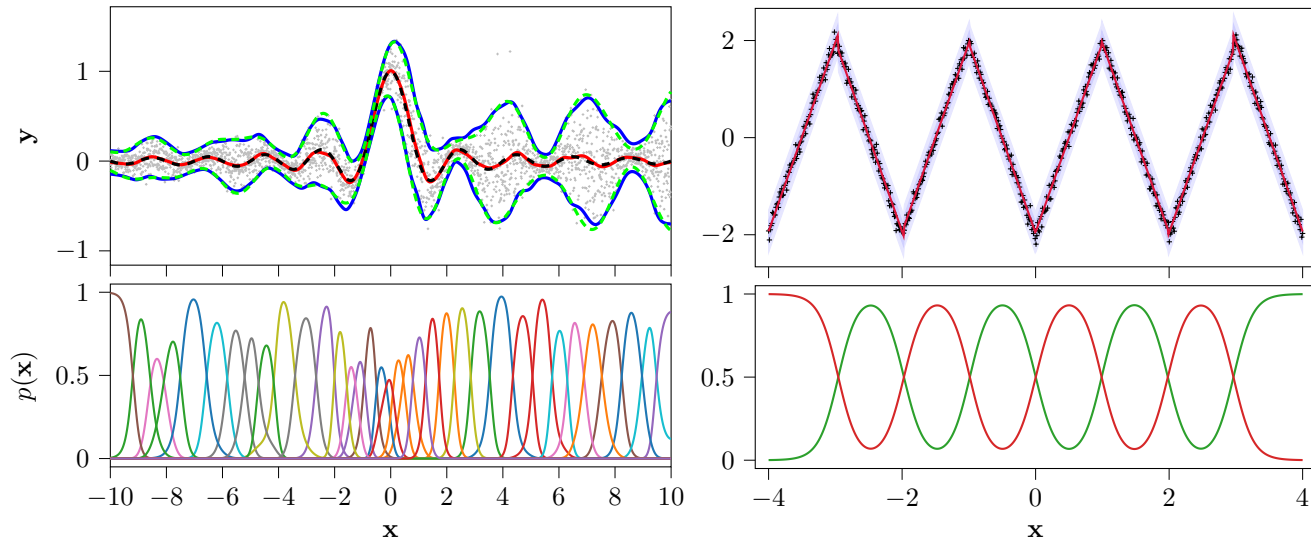
Fig. 4: Left, a challenging heteroscedastic example of a sinc function heavily overlayed with input-dependent noise. The figure shows the mean prediction (red) on the training data (dots) and the true mean function (dashed black) corrupted by noise (dashed green). The blue dashed lines represent the recovered noise process. Right, hierarchical local regression with HILR using parameter sharing in shift-invariant functions. The top figure shows the *mode-prediction* (red) along with two standard deviations of predictive uncertainty (shaded blue). The bottom plot highlights the multi-modal activation, leading to shared slope information over non-adjacent regions.
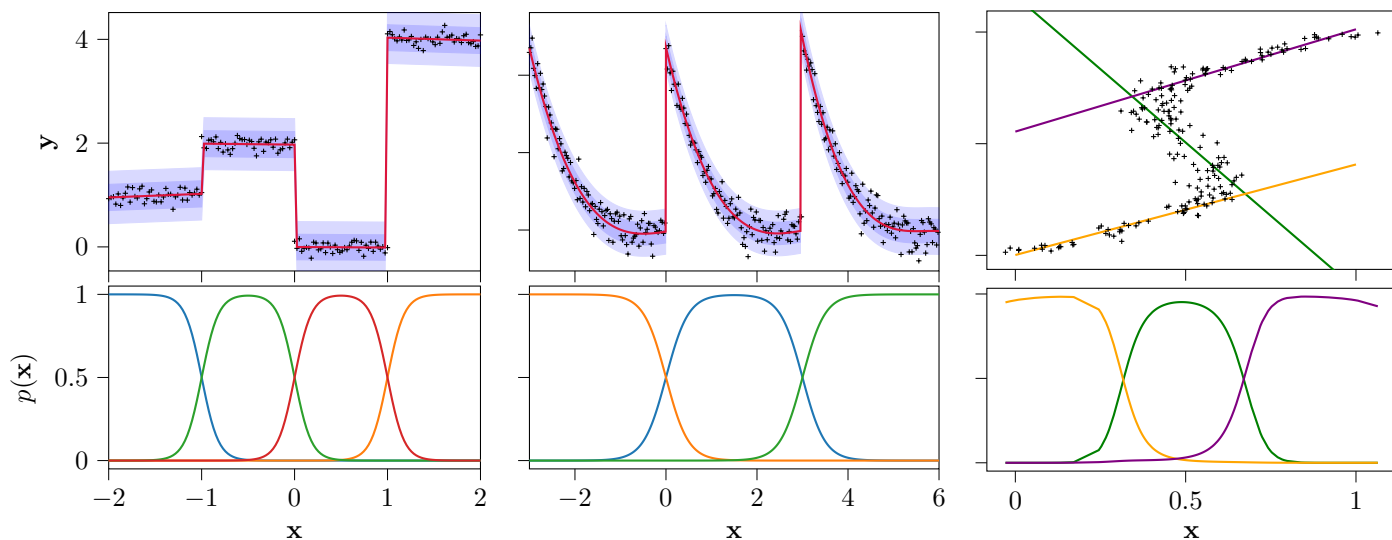
Fig. 5: Left and middle, learning discontinuous functions with ILR. The top figures show the *mode-prediction* (red) and two standard deviations confidence (shaded blue). The left example is a simple step function that can be captured with linear features, while in the middle, we use a polynomial transformation of the input for more flexibility. Right, we tackle inverse mapping problems with ILR. This example includes scattered data that maps the input **x** to multiple output values **y**. A discriminative modeling approach fails in these scenarios, as it tries to capture the ambiguous mean of the function $f : \mathbf{x} \to \mathbf{y}$. By approximating the joint density over both input and output and using *mode-prediction*, ILR can reconstruct these non-unique relations via local linear approximations. The bottom plots show the activation over the input space.

Discontinuous (polynomial) Functions. In Figure 5 (left), a combination of step functions is fitted using mode-prediction, as described in Section 3.2. By using polynomial input features, more expressive local models can be realized. Figure 5 (middle) depicts an example of cubic regressors, which are still linear in the parameters, fitted to data sampled from noisy cubic polynomials.

Inverse Mapping. One crucial advantage of generative over discriminative modeling is the ability to deal with non-unique inverse mapping problems. Such scenarios arise when the same input can be mapped to multiple output values. Joint modeling of the input-output data allows for flexible conditioning and alleviates the directional graph constraints. In Figure 5 (right), we show a simple example of how ILR can learn these mappings, adapted from [64].

Bayesian Sequential Updates. In Figure 6, we demonstrate a sequential learning problem. Data from a chirp signal arrives in three batches so that the posterior of previous batches serves as the prior for the next learning phase. ILR successfully captures the data trend, and no significant
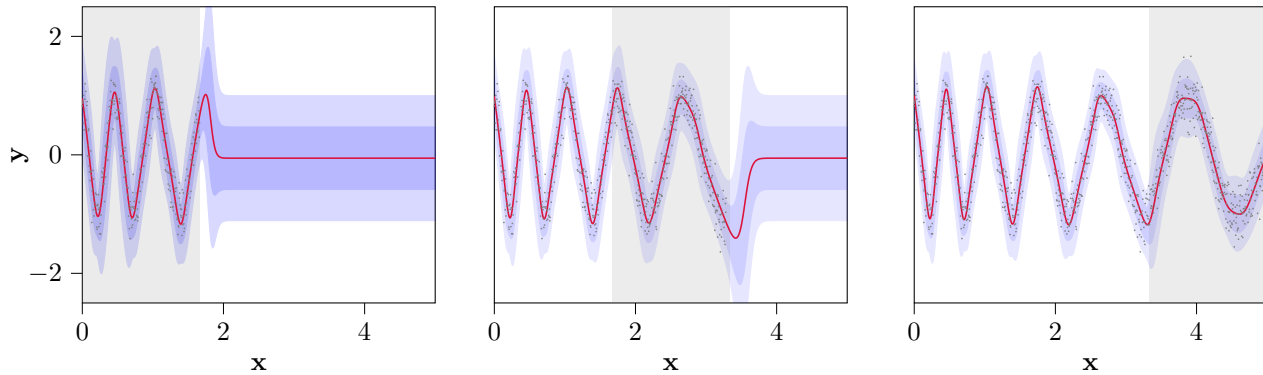
Fig. 6: Bayesian sequential updates. Mean (red) and a two standard deviations interval (shaded blue) of the predictive distribution fitted to sequentially arriving data (three batches) from the chirp dataset (gray dots). For the second and third plots, the posterior fitted to the previous batches is used as a prior to perform a Bayesian sequential update. There is no catastrophic forgetting of previously learned knowledge, and in regions with no data, the prediction falls back to the prior.

TABLE 1: Accuracy on the SARCOS dataset

	MSE	NMSE	Experts
ILR	$(4.80 \pm 0.30) \times 10^{-1}$	$(3.40 \pm 0.20) \times 10^{-3}$	1700
HILR	$(5.30 \pm 0.40) \times 10^{-1}$	$(3.90 \pm 0.30) \times 10^{-3}$	1450
LGR*	86.00×10^{-1}	50.00×10^{-3}	7000
LWPR	$(26.00 \pm 0.30) \times 10^{-1}$	$(18.00 \pm 0.20) \times 10^{-3}$	32 000
GPR	N/A	N/A	-
SGPR	$(8.50 \pm 0.03) \times 10^{-1}$	$(6.000 \pm 0.008) \times 10^{-3}$	-
rBCM	$(4.52 \pm 0.05) \times 10^{-1}$	$(2.600 \pm 0.030) \times 10^{-3}$	315
gPoE	$(4.60 \pm 0.06) \times 10^{-1}$	$(3.000 \pm 0.075) \times 10^{-3}$	315

TABLE 2: Accuracy on the Barrett-WAM dataset

	MSE	NMSE	Experts
ILR	$(2.90 \pm 0.50) \times 10^{-1}$	$(7.0 \pm 0.5) \times 10^{-3}$	1350
HILR	$(3.10 \pm 0.65) \times 10^{-1}$	$(8.0 \pm 0.6) \times 10^{-3}$	1110
LGR*	7.70×10^{-1}	17.0×10^{-3}	3270
LWPR	$(10.00 \pm 1.50) \times 10^{-1}$	$(37.0 \pm 10.0) \times 10^{-3}$	2900
GPR	$(1.00 \pm 0.01) \times 10^{-1}$	$(2.30 \pm 0.01) \times 10^{-3}$	-
SGPR	$(1.80 \pm 0.05) \times 10^{-1}$	$(6.30 \pm 0.02) \times 10^{-3}$	-
rBCM	$(3.80 \pm 0.35) \times 10^{-1}$	$(19.00 \pm 1.80) \times 10^{-3}$	100
gPoE	$(3.40 \pm 0.13) \times 10^{-1}$	$(16.00 \pm 0.60) \times 10^{-3}$	100

catastrophic forgetting is observed. The approximation errors due to the mean-field posterior factorization assumption have little influence because the posterior updates are localized in the input domain.

Robot Inverse Dynamics. Next, we use ILR and HILR to learn the inverse dynamics of anthropomorphic manipulators. The dynamics of these manipulators are governed by the following equation

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}),$$

where $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ are joint angles, velocities and accelerations, and $\mathbf{u}$ are torques. $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ are the Coriolis and centripetal forces, and $\mathbf{G}(\mathbf{q})$ is the gravity force. $\epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ are general unmodelled nonlinearities such as sticktion/friction and hydraulic and tendon/cable dynamics, that motivate a data-driven approach to learn the mapping $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \rightarrow \mathbf{u}$. Later, we use the learned ILR model for online inverse dynamics control.

We use the mean squared error (MSE), normalized mean squared error (NMSE), and the number of local experts as evaluation criteria. These measures cover the prediction accuracy as well as the complexity of the learned model. We compare to popular (probabilistic) methods such as local Gaussian regression (LGR) [42], locally weighted projection regression (LWPR) [32], Gaussian process regression (GPR) [3] and sparse Gaussian process regression (SGPR) [10], and two scalable Gaussian process product of experts: the robust Bayesian committee machine (rBCM) [12] and the generalized product of experts (gPoE) [11].

We benchmark the prediction accuracy of all regression techniques on a high-dimensional dataset collected from a 7-DoF (degrees of freedom) anthropomorphic SARCOS arm

[32]. The dataset consists of 44484 training points and 4449 test cases. Overall there are 21 input variables $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ mapping to 7 motor torques $\mathbf{u}$. We also benchmark on an inverse dynamics dataset from a 4-DoF Barrett-WAM manipulator, mapping from a 12-D to 4-D space. This dataset contains 25000 training and 5000 test pairs.

Table 1 and Table 2 list the results for both datasets. We report the MSE, NMSE, and the number of active models over all joints, averaged over five seeds. We compute the mean and standard deviation for every cell in the table, except for LGR*, because of the unreasonable training times achieved while using the authors' code. When evaluating GPR on the SARCOS dataset, we faced GPU-memory constraints (32 GB), and we have discarded this evaluation. For rBCM and gPoE, we assigned an expert to every 1000 data points, repeated for every output dimension.

The results show that ILR and HILR outperform the related local regression methods LWPR and LGR, both in terms of prediction accuracy and number of active experts. Nonetheless, GPR is still the gold standard when the kernel size is within memory limits. Interestingly, the figures also indicate that ILR and HILR are competitive with sparse Gaussian process regression (SGPR) and the two product of experts rBCM and gPoE. Finally, the evidence reveals that HILR tends to activate roughly $10-15\%$ fewer components than ILR. This observation hints that HILR may be taking advantage of shift-invariance patterns in the data and avoiding duplicate regression units. However, this hypothesis is hard to validate due to the data's high dimensionality.

Real Inverse Dynamics Control. Finally, we demonstrate the validity of the learned dynamics in a real-robot control scenario. In this evaluation, we tackle two exper-

TABLE 3: Tracking error and torque contributed by the PD-controller during the Barrett-WAM real robot task.

		PD	Analytic+PD	ILR+PD
T1	MSE	2.33×10^{-2}	2.16×10^{-2}	1.03×10^{-3}
	PD-Torque	8.25×10^{0}	7.12×10^{0}	1.40×10^{0}
T2	MSE	2.60×10^{-2}	2.55×10^{-2}	9.17×10^{-4}
	PD-Torque	8.71×10^{0}	7.41×10^{0}	1.33×10^{0}
T3	MSE	2.94×10^{-2}	3.08×10^{-2}	8.96×10^{-4}
	PD-Torque	9.38×10^{0}	8.06×10^{0}	1.38×10^{0}

iments. In the first, we focus on the Bayesian sequential learning aspect, whereas the second highlights our model's ability to deal with massive amounts of data. Both experiments deploy an ILR model on the Barrett-WAM to track an 8-shaped trajectory in the xy-plane of the end-effector.

For the first experiment, we collect 30000 training samples (roughly 1 minute) from multiple trajectories with different velocity profiles and Bayesian sequential learning over 15 batches for multiple seeds. Figure 7 depicts the progression of the learning process. We then select the best model and perform model-based control to track held-out test trajectories with unseen velocity profiles. ILR predicts the feed-forward torques while supported by a low-gain PD-controller for safety considerations. We compare this controller to one with access to the analytical dynamics and the same low-gain PD-controller and to a *model-free* PD-controller. Figure 8 depicts the tracking performance of the different controllers on two test trajectories.

In the second experiment, we learn a model covering a larger region of the state-action space. We generate a larger Barrett-WAM dataset consisting of 150000 training examples (roughly 5 minutes). We test the model learned by ILR on held-out test trajectories and compare it to the same controller constellation as in the first experiment. As benchmarking criteria, we evaluate the MSE with respect to the desired trajectory and the mean torque contribution of the PD-controller to the overall control signal. The rationale is that a good inverse dynamics model will consistently produce a low MSE without relying on the PD-controller's assistance. Table 3 shows the benchmarks on three test trajectories. The results indicate that ILR significantly improves tracking performance with minimal contribution from the PD-controller. Moreover, we are able to consistently achieve a prediction frequency of $2000 \, \text{Hz}$ during both tasks, although the Barrett-WAM robot requires only $500 \, \text{Hz}$.

6 CONCLUSION

In this paper, we presented two probabilistic hierarchical local regression models, ILR and HILR, and derived an efficient variational inference technique for data-driven learning. These representations are based on infinite mixtures from Bayesian nonparametrics. We situate our contributions as the next iteration in a large family of local linear regression techniques such as RFWR, LWPR, and LGR. We have shown that placing Dirichlet process priors on Bayesian mixtures of local regression units can regularize model complexity with minor loss in performance and without relying on heuristics. Moreover, we have highlighted the advantage
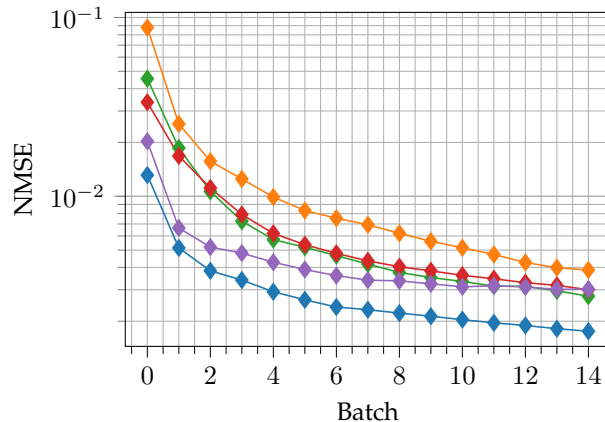

Fig. 7: 8-Shaped trajectory learning. Bayesian sequential updates on a dataset collected from a Barrett-WAM. We plot the NMSE for five different seeds on accumulated data over the number of batches. The NMSE consistently improves with new data, and no catastrophic forgetting is observed.
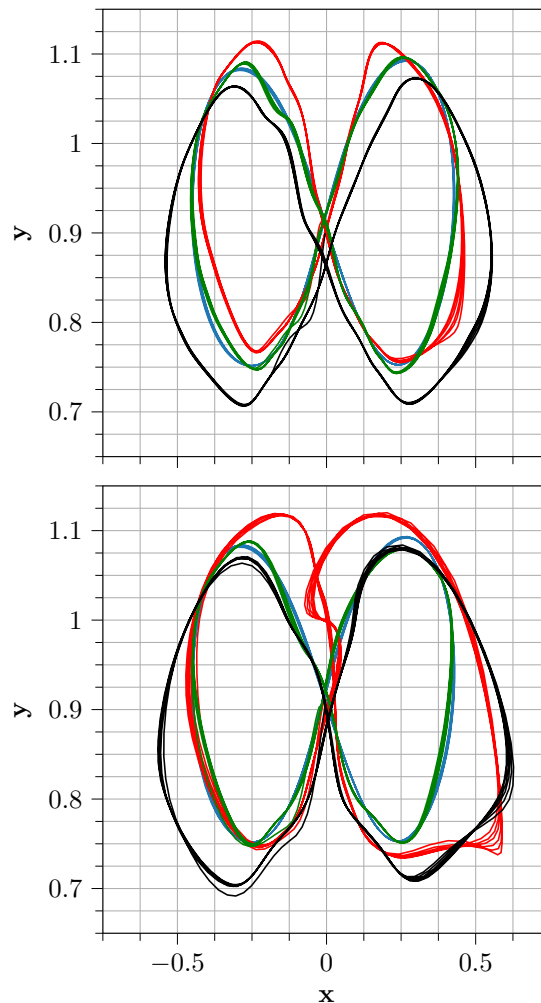

Fig. 8: 8-shaped trajectory tracking on the Barrett-WAM. We compare three controllers on two test trajectories (blue), a low-gain PD (black), a low-gain PD + feed-forward torques from an analytical model (red), and a low-gain PD + feed-forward torques from ILR (green). The results indicate that ILR delivers the best tracking performance.

of the two-level architecture of HILR in compressing the learned models and reducing the number of active experts. Empirical evaluation indicates that the models outperform LWPR and LGR, and are competitive with sparse GPR and products of experts. Finally, we have empirically confirmed the practicality of this approach for online inverse dynamics control on a Barrett-WAM robot.

Nonetheless, these presented concepts still suffer from multiple drawbacks. The mean-field assumption is a source of significant errors in posterior inference. Collapsed formulations of Dirichlet process priors promise better approximations [76]. In addition, Bayesian mixture models are influenced by a large number of hyperparameters, which cannot be directly optimized via empirical Bayes [77], leading to lower predictive performance when compared to optimized GPR. Nonetheless, the ELBO offers an objective for hyperparameter optimization. Naive gradient-based techniques have proven too brittle due to their reliance on Euclidean distance metrics. A natural-gradient approach appears to be a suitable alternative in the future.

Further development of hierarchical local regression may focus on treating ILR and HILR as layers in a multi-layered representation. This extension would allow the models to benefit from intermediate nonlinear projections into high dimensional spaces that have proven powerful in deep neural networks. Another practical consideration is to incorporate physical inductive biases such as inverse dynamics [78] to facilitate learning meaningful quantities.

ACKNOWLEDGMENTS

REFERENCES

[1] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," *Neural Computation*, 2005.

[2] D. Büchler, R. Calandra, B. Schölkopf, and J. Peters, "Control of musculoskeletal systems using learned dynamics models," *IEEE Robotics and Automation Letters*, 2018.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, 2006.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 2016.

[5] L. Wasserman, *All of Nonparametric Statistics*, 2006.

[6] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold Gaussian processes for regression," in *International Joint Conference on Neural Networks*, 2016.

[7] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *International Conference on Artificial Intelligence and Statistics*, 2016.

[8] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," in *Advances in Neural Information Processing Systems*, 2017.

[9] R. Herbrich, N. D. Lawrence, and M. Seeger, "Fast sparse Gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems*, 2003.

[10] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *International Conference on Artificial Intelligence and Statistics*, 2009.

[11] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of Gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.

[12] M. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *International Conference on Machine Learning*, 2015.

[13] M. Bauer, M. van der Wilk, and C. E. Rasmussen, "Understanding probabilistic sparse Gaussian process approximations," in *Advances in Neural Information Processing Systems*, 2016.

[14] A. G. d. G. Matthews, "Scalable Gaussian process inference using variational methods," Ph.D. dissertation, University of Cambridge, 2017.

[15] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic Gaussian process regression," in *International Conference on Machine Learning*, 2005.

[16] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *International Conference on Machine Learning*, 2007.

[17] H. Liu, Y.-S. Ong, and J. Cai, "Large-scale heteroscedastic regression via Gaussian process," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[18] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.

[19] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, 1989.

[20] R. Neal, "Bayesian learning for neural networks," Ph.D. dissertation, University of Toronto, 1994.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, 2015.

[22] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017.

[23] M. E. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, "Fast and scalable Bayesian deep learning by weight-perturbation in Adam," *International Conference on Machine Learning*, 2018.

[24] S. Sun, G. Zhang, J. Shi, and R. Grosse, "Functional variational Bayesian neural networks," *International Conference on Learning Representations*, 2019.

[25] J. Watson, J. A. Lin, P. Klink, J. Pajarinen, and J. Peters, "Latent derivative Bayesian last layer networks," in *International Conference on Artificial Intelligence and Statistics*, 2021.

[26] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux-effortless Bayesian deep learning," *Advances in Neural Information Processing Systems*, 2021.

[27] F. Wenzel, K. Roth, B. Veeling, J. Swiatkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the Bayes posterior in deep neural networks really?" *International Conference on Machine Learning*, 2020.

[28] A. Y. Foong, D. R. Burt, Y. Li, and R. E. Turner, "On the expressiveness of approximate inference in Bayesian neural networks," *Advances in Neural Information Processing Systems*, 2020.

[29] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, 1997.

[30] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, 1998.

[31] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, 2002.

[32] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, 2005.

[33] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, 1989.

[34] L. Xu, M. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," *Advances in Neural Information Processing Systems*, 1994.

[35] O. Nelles and R. Isermann, "Basis function networks for interpolation of local linear models," *Conference on Decision and Control*, 1996.

[36] P. Moerland, "Classification using localized mixtures of experts," 1999.

[37] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, 1991.

[38] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, 1994.

[39] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of Gaussian process experts," in *Advances in Neural Information Processing Systems*, 2002.

[40] J. A. Ting, F. Meier, S. Vijayakumar, and S. Schaal, "Locally weighted regression for control," in *Encyclopedia of Machine Learning and Data Mining*, 2010.

[41] J. A. Ting, M. Kalakrishnan, S. Vijayakumar, and S. Schaal, "Bayesian kernel shaping for learning control," in *Advances in Neural Information Processing Systems*, 2009.

[42] F. Meier, P. Hennig, and S. Schaal, "Incremental local Gaussian regression," in *Advances in Neural Information Processing Systems*, 2014.

[43] H. Abdulsamad, P. Nickl, P. Klink, and J. Peters, "A variational infinite mixture for probabilistic inverse dynamics learning," *IEEE International Conference on Robotics and Automation*, 2021.

[44] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, *Bayesian Nonparametrics*, 2010.

[45] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University College London, 2003.

[46] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts," in *Advances in Neural Information Processing Systems*, 2006.

[47] C. Yuan and C. Neubauer, "Variational mixture of Gaussian process experts," in *Advances in Neural Information Processing Systems*, 2009.

[48] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, 1995.

[49] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, 2000.

[50] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association*, 2001.

[51] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems*, 1999.

[52] D. Blackwell, J. B. MacQueen *et al.*, "Ferguson distributions via pólya urn schemes," *The Annals of Statistics*, 1973.

[53] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, 1973.

[54] P. Mueller, A. Erkanli, and M. West, "Bayesian curve fitting using multivariate normal mixtures," *Biometrika*, 1996.

[55] B. Shahbaba and R. M. Neal, "Nonlinear models using Dirichlet process mixtures," *The Journal of Machine Learning Research*, 2009.

[56] L. Hannah, D. M. Blei, and W. B. Powell, "Dirichlet process mixtures of generalized linear models," *The Journal of Machine Learning Research*, 2011.

[57] C. Gadd, S. Wade, and A. Boukouvalas, "Enriched mixtures of generalised Gaussian process experts," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[58] H. Attias, "A variational Bayesian framework for graphical models," in *Advances in Neural Information Processing Systems*, 2000.

[59] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, 2006.

[60] E. Angelino, M. J. Johnson, and R. Adams, "Patterns of scalable Bayesian inference," *Foundations and Trends in Machine Learning*, 2016.

[61] T. Minka, "Bayesian linear regression," 2000, accessed: 2020-07-28. [Online]. Available: https://tminka.github.io/papers/linear.html

[62] M. J. Beal and Z. Ghahramani, "Variational Bayesian learning of directed graphical models with hidden variables," *Bayesian Analysis*, 2006.

[63] J. Rousseau and K. Mengersen, "Asymptotic behaviour of the posterior distribution in overfitted mixture models," *Journal of the Royal Statistical Society*, 2011.

[64] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.

[65] Y. W. Teh, "Dirichlet process," in *Encyclopedia of Machine Learning and Data Mining*, 2010.

[66] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, 1994.

[67] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov chain Monte Carlo*, 2011.

[68] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, 2017.

[69] M. Opper and D. Saad, *Advanced Mean Field Methods: Theory and Practice*, 2001.

[70] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, 2013.

[71] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006.

[72] H. Z. Yerebakan, B. Rajwa, and M. Dundar, "The infinite mixture of infinite Gaussian mixtures," in *Advances in Neural Information Processing Systems*, 2014.

[73] T. V. Nguyen, D. Phung, X. Nguyen, S. Venkatesh, and H. Bui, "Bayesian nonparametric multilevel clustering with group-level contexts," in *International Conference on Machine Learning*, 2014.

[74] V. Huynh, D. Q. Phung, S. Venkatesh, X. Nguyen, M. D. Hoffman, and H. H. Bui, "Scalable nonparametric bayesian multilevel clustering." in *Conference on Uncertainty in Artificial Intelligence*, 2016.

[75] C. L. Bennett, M. Halpern, G. Hinshaw, N. Jarosik, A. Kogut, M. Limon, S. S. Meyer, L. Page, D. N. Spergel, G. S. Tucker *et al.*, "First year Wilkinson microwave anisotropy probe (WMAP) observations: Preliminary maps and basic results," *The Astrophysical Journal Supplement Series*, 2003.

[76] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational Dirichlet process mixture models." in *International Joint Conferences on Artificial Intelligence*, 2007.

[77] J. S. Maritz and T. Lwin, "Empirical Bayes methods," *Monographs on Statistics and Applied Probability*, 1989.

[78] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *IEEE international conference on robotics and automation*, 2010.

[79] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, 2008.