A Variational Infinite Mixture for Probabilistic Inverse Dynamics Learning

Hany Abdulsamad, Peter Nickl, Pascal Klink, Jan Peters

Computer Science Department, Technische Universität Darmstadt

Abstract—Probabilistic regression techniques in control and robotics applications have to fulfill different criteria of data-driven adaptability, computational efficiency, scalability to high dimensions, and the capacity to deal with different modalities in the data. Classical regressors usually fulfill only a subset of these properties. In this work, we extend seminal work on Bayesian nonparametric mixtures and derive an efficient variational Bayes inference technique for infinite mixtures of probabilistic local polynomial models with well-calibrated certainty quantification. We highlight the model's power in combining data-driven complexity adaptation, fast prediction and the ability to deal with discontinuous functions and heteroscedastic noise. We benchmark this technique on a range of large real inverse dynamics datasets, showing that the infinite mixture formulation is competitive with classical Local Learning methods and regularizes model complexity by adapting the number of components based on data and without relying on heuristics. Moreover, to showcase the practicality of the approach, we use the learned models for online inverse dynamics control of a Barrett-WAM manipulator, significantly improving the trajectory tracking performance.

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

I. INTRODUCTION

Principled data-driven, adaptive and incremental learning is a desirable property in domains in which datasets are dynamic and accumulate slowly over time. For example, robots have to build models of their dynamics and the environment as they interact with the world. Moreover, these models have to be computationally efficient during both the learning and evaluation process. In the case of general-purpose robots, these models have also be able to 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 (ANN) [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, however the vanilla formulation of GPR [3] suffered from many drawbacks that have been successfully addressed by recent research, such as the smoothness assumption [6]–[8], scaling to large datasets [9]–[12] and difficulties modeling heteroscedastic noise [13]–[15]. Artificial Neural Networks, on the other hand, have proven themselves as very powerful easy-to-train universal approximators. They are still however susceptible to over-parameterization [16] and catastrophic forgetting [17]. Moreover, although major progress on the front of Bayesian Neural Networks (BNN) has been made [18]–[20], new evidence suggests that issues regarding the accuracy of uncertainty quantification still need to be tackled [21], [22]. Finally, Local Regression methods have had great success, particularly in the domain of robotics and control, precisely because of their flexibility, ability to model hard nonlinearities and heteroscedastic noise, and the possibility of incorporating new data online. Two categories of LR exist [23], lazy learners, that maintain all seen data points in memory [24], and memoryless learners that compress data by relying on basis functions in the input space and fitting and storing local parameterized models [25]–[27]. However, these methods are often difficult to tune as they possess many hyperparameters. A limited attempt to a Bayesian treatment is made in [28] to alleviate the need to tune the basis functions by constructing local nonparametric kernels and placing Gamma priors on the kernel width. However, this technique results in a
localized GP formulation that needs to retain the training data in memory, thus the computational issues of vanilla Gaussian Process Regression. A further Bayesian generalization of LR is suggested in [29], in which the authors couple the local models via the loss function, thus reinforcing global coordination, and treat the local models in a Gaussian Regression framework. All mentioned approaches however still rely on heuristics for pruning local models and fall short of formulating a full generative model in both the input and target space.

It is essential to distinguish between the mentioned LR methods and the so-called Mixture of Experts (MoE) techniques [30]–[32]. The distinction lies in the fact that MoEs rely on discriminative, explicitly input-dependent gating that divides the input space between competing experts. In contrast, LR takes a basis-function approach that resembles generative modeling of the input and allows cooperation between all experts during training, enabling each model to influence the global result.

Following this introduction, it is our opinion that Local Regression with a proper fully Bayesian generative treatment has the potential to serve as an efficient general-purpose probabilistic function approximator with well-calibrated uncertainty estimates and drive many low-level applications in control and robotics that favor fast models and do not require deep representations. In the following, we set out to present a fully probabilistic graphical mixture model of Local Regression that maintains proper priors on all quantities. We start by presenting what we refer to as the Bayesian finite mixture that implicitly assumes a finite number of local regression models. This representation can then be extended to become an infinite mixture by relying on the paradigm of Bayesian Nonparametrics (BNP) [33] and ultimately results in a formulation related to all aforementioned local methods that alleviates the need for any heuristics. For learning these models, we derive a Variational-Bayes Expectation-Maximization (VBEM) scheme to efficiently infer the posterior parameters [34] and overcome the need for computationally heavy sampling methods. We benchmark this model on a range of toy tasks to highlight their strengths, as well as on real-world datasets for learning the inverse dynamics of robotic manipulators. Most importantly, we finally use the learned model to perform inverse dynamics control on a real Barrett-WAM manipulator, Figure 1.

The proposed approach is based on seminal work in the area of Bayesian Nonparametrics. We reference influential Markov-Chain-Monte-Carlo (MCMC) sampling techniques for Bayesian Nonparametric density estimation [35]–[38], which developed the first seeds of Bayesian Inference for Dirichlet Processes [39] under the Pólya-Urn sampling scheme [40]. Moreover, similar infinite mixture regression models have been proposed in [41]–[44], albeit while relying on expensive Gibbs sampling techniques and within a limited application scope. Furthermore, we build on fundamental work from Variational Inference (VI) [45], that presented the first VI algorithm for the Bayesian Finite Gaussian Mixture Model and on the contribution of [46], that developed the first VI technique for Infinite Gaussian Mixtures with stick-breaking priors, by relying on a truncated variational posterior representation.

II. PRELIMINARIES

In this section we introduce some concepts related to our approach such as Bayesian Linear Regression, Bayesian Mixture Models and Dirichlet Process.

**Bayesian Linear Regression.** We start by discussing the Bayesian treatment of a single component of a fully Bayesian Local Regression model, namely Bayesian Linear Regression [47]. The conditional data model takes a feature vector \( x \in \mathbb{R}^m \) as a random input variable and returns a response random variable \( y \in \mathbb{R}^d \) according to a linear mapping \( A : \mathbb{R}^m \rightarrow \mathbb{R}^d \) and additive zero-mean noise \( e \) with a precision matrix \( V \). \( y = Ax + e, e \sim N(0, V^{-1}) \). For a full Bayesian treatment we consider the parameters of this model to be random variables on which we place proper conjugate priors. In this case we place a Matrix-Normal-Wishart prior \( MN(A|M, V^{-1}, K^{-1}) W(V|P, \eta) \) on the matrix \( A \) and precision \( V \), where \( M \), the mean of \( A \), is a \( d \times m \) matrix and \( V \) and \( K \) are \( d \times d \) and \( m \times m \) that serve as row and column precisions of \( A \), respectively. The parameters of the Wishart distribution are the \( d \times d \) positive definite scale matrix \( P \) and the degrees of freedom \( \eta \). Given the conjugate nature of the chosen priors, the posterior of \( p(A, V|D) \) is

![Fig. 2: Two toy datasets learned with Infinite Local Regression (ILR). The bottom plots show the activation of the local regression models over the input space. Left, an example of how this technique deals with out-of-distribution uncertainty. Mean prediction (red) on training data of a Sinc function (grey dots) and the true mean function (black dashed). The shades of the blue area represent a predictive uncertainty of one and two standard deviations. In areas lacking training data the predictive uncertainty of ILR is large, the mean prediction falls back to the prior. Right, the ability of ILR to capture input-dependent noise is depicted. Mean prediction (red) on training data of a Sinc function (grey dots) and the true mean function (black dashed) corrupted by \( \pm 2\sigma(x) \) (dashed green). The blue lines represent the 2-\( \sigma \) confidence interval of the mean prediction. ILR is able to learn a strongly input-dependent noise function.](image-url)
Bayesian Finite Mixture Models. Gaussian mixtures are hierarchical latent variable models with universal approximation capabilities for arbitrary continuous densities. This insight is of central interest, when considered later in the context of density estimation for local linear models, that are themselves universal nonlinear function approximators [5]. A finite $K$-component Gaussian mixture of a random variable $x$ is a weighted linear combination of densities $p(x|\theta) = \sum_{k=1}^{K} p(z|\pi) p(x|\theta_k) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k^{-1})$, with $K$ unique mean vectors $\mu_k$ and precision $\Sigma_k$. The latent quantity $z \in \{1, \ldots, K\}$ is a discrete random variable, distributed according to a Categorical distribution $p(z_i) = \text{Cat}(\pi)$, subject to the mixing weights $\pi = \{\pi_1, \ldots, \pi_K\}$, satisfying $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

The Bayesian extension [45] of this model introduces a conjugate Normal-Wishart prior $H(\lambda)$ on the means and precision matrices $\mu_k, \Sigma_k$, where $\lambda$ is the hyperparameter vector. Furthermore, a conjugate Dirichlet prior, with an $\alpha$ concentration parameter, is placed on the mixing weights $\pi \sim \text{Dir}(\lambda K)$. This Bayesian perspective has proven very effective in regularizing the shortcomings of GMMs, by allowing superfluous components to fall back to their priors, instead of severely over-fitting to small clusters, thus consequently inducing a sparsity effect over $K$ [49].

Dirichlet Process and Stick-Breaking. A Dirichlet process is a distribution over probability measures $G$. We write $G \sim \text{DP}(\alpha, H)$, where $\alpha$ is the concentration parameter and $H$ is the base measure [48], [50]. Intuitively, a Dirichlet process is a distribution over distributions, which means that each draw $G$ is itself a distribution. The draws from a DP are discrete distributions. The base distribution $H$ is the mean of the DP and the concentration parameter $\alpha$ can be interpreted as an inverse variance. The larger $\alpha$, the smaller the variance and the DP will concentrate more of its mass around the mean $H$.

We will rely on the stick-breaking construction [51] of a DP as an algorithmic realization. Stick-breaking delivers an infinite sequence of mixture weights $\pi_k$ of an infinite mixture model from the probabilistic process $\pi_k = v_k \prod_{l=1}^{k-1} (1 - v_l)$, where $v_k \sim \text{Beta}(1, \alpha)$. This process is sometimes denoted as $\pi \sim \text{GEM}(\alpha)$ [48]. The stick-breaking procedure describes how the random variables $v_k$, representing stick lengths, are drawn from a Beta distribution and combined to obtain the mixture weights $\pi_k$. If the concentration parameter $\alpha$ increases, the magnitude of the mixing weights $\pi_k$ decreases on average and the number of probable active components increases. This representation of DPs can be used to replace the priors placed on the Finite Gaussian Mixture Model [46]. In such a setting the sampled measures $G \sim \text{DP}(\alpha, H(\lambda))$ are the parameters $\theta_k \sim H(\lambda)$ of an infinite number of clusters, associated with an infinite number of weights $\pi_k$ generated by the stick-breaking. The clustering effect comes into play due to the discrete nature of the DP. Eventually, the same parameters will be sampled over and over forcing the associated data points to cluster.

III. VARIATIONAL BAYES FOR INFINITE MIXTURE OF LOCAL REGRESSORS

Using the previously presented concepts of Bayesian Linear Regression, Bayesian Linear Models and Dirichlet Process, we now construct a fully Bayesian Infinite Mixture of Local Regressors (ILR). Our approach to solving the regression task is mainly a Bayesian joint density estimation task. Our aim is to find the posterior over all regression parameters $p(\theta|D)$ and use the predictive marginal at prediction time. We use the generative model as depicted in Figure 3

$$\pi \sim \text{GEM}(\alpha), \quad z_i \sim \text{Cat}(\pi), \quad \theta_k \sim H(\lambda),$$

$$x_i \sim N(\mu_{z_i}, \Sigma_{z_i}^{-1}), \quad y_i \sim N(A_{z_i} x_i, V_{z_i}^{-1}),$$

where $\theta_k = \{\mu_k, \Sigma_k, A_k, V_k\}$. Notice here that the defined densities over the input space $X$ naturally play the role of the basis functions or so-called receptive fields as in the Receptive Field Weighted Regression [26] and Locally Weighted Projected Regression [27] algorithms.

In the following, we will tackle the problem of inferring the posterior $p(\theta, z, v, \Sigma, A, V)$, where $v$ are the lengths generated during the stick-breaking process GEM($\alpha$). We opt here for a Variational Bayes approach because VI approaches tend to be more computationally efficient in comparison to MCMC methods, especially considering the high dimensional posterior space in terms of the number of components and number of parameters per component.

Joint Likelihood Function. For the general case of multivariate regression with fully populated covariance matrices we use the following likelihood function

$$p(Y, X | z, \mu, \Sigma, A, V) = \prod_{n=1}^{N} \prod_{k=1}^{K} N(y_n|A_k x_n, V_k^{-1})^{z_{nk}} N(x_n|\mu_k, \Sigma_k^{-1})^{z_{nk}},$$

where the dimensions of all quantities follow the notation of Bayesian Linear Regression.
**Infinite Conjugate Prior.** We assume the factorized conjugate infinite mixture prior \( p(v, \mu, \Sigma, \Lambda, V) = \) 
\[
\prod_{k=1}^{\infty} \text{Beta}(v_k|\gamma_{0,1}, \gamma_{0,2}) \prod_{k=1}^{\infty} \text{MN}(A_k|M_0, V_k^{-1}, K_0^{-1}) 
\]
\[
W(V_k|P_0, \eta_0) \prod_{k=1}^{\infty} N\left(\mu_k|m_0, (\lambda_k \Sigma_k)^{-1}\right) W(\Sigma_k|L_0, \nu_0).
\]
This prior introduces a Normal-Wishart distribution on the cluster means \( \mu_k \) and precision matrices \( \Sigma_k \), while a Matrix-Normal-Wishart prior is placed on the regression coefficients \( A_k \) and the precision matrices \( V_k \). The parameter \( \pi_k \) in the Categorical distribution is substituted by the expression of the stick-breaking \( \pi_k(v) = v_k \prod_{j=1}^{k-1} (1 - v_j) \). The parameters \( v = \{v_1, \ldots, v_k\} \) are independently Beta distributed. By definition, the first hyperparameter of the Beta distribution \( \gamma_{0,1} \) is set to 1. The second hyperparameter \( \gamma_{0,2} \) is called concentration parameter and has a crucial impact on the number of components that are learned from the data.

**Mean-Field Factorization.** For tractable inference we assume a mean field factorization of the posterior \( p(z, v, \mu, \Sigma, \Lambda, V|x) = \prod_{k=1}^{N} q(z_n) q(v, \mu, \Sigma, \Lambda, V) \), which makes the assumptions that latent variables \( z \) are conditionally independent from the other parameters.

**Truncated Variational Posterior.** The posterior is a result of the free form optimization of the variational distributions using the mean-field equation [52]. We follow [46] by allowing a truncated form of the posterior, while maintaining an infinite prior. This truncation is considered in contrast to approaches that truncate the Dirichlet process [37]. During evaluation we choose a very high truncation threshold that is seldom reached. The resulting approximate variational posterior has the following conjugate form \( q^*(z, v, \mu, \Sigma, \Lambda, V) = \) 
\[
= \prod_{n=1}^{N} \text{Cat}(z_n|r_n) \prod_{k=1}^{K} \text{Beta}(v_k|\gamma_{k,1}, \gamma_{k,2}) \prod_{k=1}^{K} \text{MN}(A_k|M_k, V_k^{-1}, K_k^{-1}) 
\]
\[
N\left(\mu_k|m_k, (\lambda_k \Sigma_k)^{-1}\right) W(\Sigma_k|L_k, \nu_k) \prod_{k=1}^{K} \text{MN}(A_k|M_k, V_k^{-1}, K_k^{-1}) W(V_k|P_k, \eta_k) \).
\]

**Variational Expectation Step.** In the E-step the responsibilities are computed, while other variational parameters are fixed. The responsibilities are variational parameters of the categorical \( q^*(z) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_n^* k \) and are calculated as
\[
r_{nk} \propto V_k^k \Sigma_k^k \exp \left[ -\frac{m}{2 \nu_k} + E_q[\log v_k] \right]
\]
\[
- \frac{\nu_k}{2} (x_n - m_k)^T L_k (x_n - m_k) - \frac{1}{2} \text{Tr}(K_k^{-1} X_n X_n^T)
\]
\[
- \frac{\eta_k}{2} (y_n - M_k X_n)^T P_k (y_n - M_k X_n)
\]
\[
+ \sum_{j=1}^{d} E_q[\log(1 - v_j)]
\],

with the help of the following expressions
\[
E_q[\log v_k] = \psi(\gamma_{k,1}) - \psi(\gamma_{k,1} + \gamma_{k,2}),
\]
\[
E_q[\log(1 - v_k)] = \psi(\gamma_{k,2}) - \psi(\gamma_{k,1} + \gamma_{k,2}),
\]
\[
\log \hat{\Sigma}_k = \sum_{j=1}^{m} \psi \left( \frac{\nu_k + 1 - j}{2} \right) + \log |\Sigma_k|,
\]
\[
\log \hat{V}_k = \sum_{j=1}^{d} \psi \left( \eta_k + 1 - j \right) + \log |V_k|,
\]

where \( \psi \) is the Digamma function.

**Variational Maximization Step.** The M-step updates the remaining variational parameters as follows
\[
\gamma_{k,1} = \sum_{n=1}^{N} r_{nk} + \gamma_{0,1}, \quad \gamma_{k,2} = \sum_{n=1}^{N} \sum_{j=1}^{K} r_{nj} + \gamma_{0,2},
\]
\[
\lambda_k = \lambda_0 + N_k, \quad m_k = \frac{1}{\lambda_k} (N_k \bar{x}_k + \lambda_0 m_0),
\]
\[
v_k = v_0 + N_k, \quad \eta_k = \eta_0 + N_k,
\]
\[
L_k^{-1} = L_0^{-1} + N_k \bar{X}_k + \frac{\lambda_0 N_k}{\lambda_k} (m_0 - \bar{x}_k)(m_0 - \bar{x}_k)^T,
\]
\[
K_k = \sum_{n=1}^{N} r_{nk} X_n X_n^T + K_0,
\]
\[
M_k = \left[ \sum_{n=1}^{N} r_{nk} y_n X_n^T + M_0 K_0 \right] K_k^{-1},
\]
\[
P_k^{-1} = P_0^{-1} + M_0 K_0 M_0^T + \sum_{n=1}^{N} r_{nk} y_n y_n^T - M_k K_k M_k^T,
\]

where the data statistics are defined as
\[
N_k = \sum_{n=1}^{N} r_{nk}, \quad \bar{x}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n,
\]
\[
S_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^T.
\]

**Posterior Predictive Distribution.** For predicting the function value \( \hat{y} \) conditioned on a test query \( \hat{x} \) we marginalize over the posterior parameters \( \theta \) to get the joint posterior predictive density. To make the marginalization tractable, we replace the true posterior by our approximate variational posterior inferred under a training dataset \( D \) [52].

**Computational Complexity.** Given the previous equations, we can deduce that the training-time computational cost is \( O(NK(d + m)^3) \) and can be straightforwardly reduced to \( O(MK(d + m)^3) \) by applying stochastic updates [53], where \( M \) is the batch size. This result shows linear scalability with the data, which is considerably more efficient than simple variants of GPR. The testing-time complexity of a mean prediction is \( O(K(d^2 + dm)) \) which combines the input membership query and the linear matrix transformation for every model \( k \). This computation is, in contrast to GPs, independent of the training data size, hence the advantage of memoryless representations during real-time critical applications.
IV. Empirical Evaluation

We evaluate different aspects of the proposed model on a range of benchmark tasks. Our goals are 1) to highlight some of the advantages of ILR, such as dealing with out-of-distribution predictions, recovering an input-dependent noise function, and the ability to perform Bayesian sequential updates, 2) to benchmark the model on real robotic high dimensional datasets, and 3) to deploy the model in a real scenario to further empirically demonstrate its validity. The source code can be found under https://github.com/hanyas/mimo.

Out-of-distribution Uncertainty. In Figure 2 (left), we apply ILR to a synthetic Sine dataset with two large gaps. We observe that the predictive uncertainty strongly reflects the lack of training data in these regions. Furthermore, the mean prediction falls back to the prior values quickly. This example highlights the reasonable quantification of uncertainty by the model. Uncertainty is low, where the mean prediction is accurate and very high in regions where the prior dominates.

The out-of-distribution behavior of ILR is strongly influenced by the input gating and a query’s membership probability.

Input-dependent Noise Function. Following [15], we generate data from \( y(x) = \text{sinc}(x) + \epsilon, \ x \in [-10, 10] \), where the noise term \( \epsilon \) is distributed according to a normal distribution \( N(0, \sigma_x^2(x)) \) with input-dependent standard deviation \( \sigma_x(x) = 0.05 + 0.2(1 + \sin(2\pi x)) / (1 + e^{-0.2x}) \). Figure 2 (right) shows that ILR can approximate the nonlinear and highly heteroscedastic function well. In particular, the input-dependent noise function is recovered in great detail.

Discontinuous Data and Local Polynomials. In Figure 4 (left), a combination of step functions is fitted using the mode of the predictive distribution. By applying a polynomial feature transformation to the input space, more expressive local regressors can be realized. Figure 4 (right) depicts an example of cubic regressors, which are still linear in the parameters, fitted to data sampled from noisy cubic polynomials.

Bayesian Sequential Updates. In Figure 5 we demonstrate a sequential learning problem. Data from the Chirp dataset arrives in three batches, and the posterior of the later batches becomes the prior for the current learning batch. ILR successfully captures data trend and no significant catastrophic forgetting of previously learned knowledge can be observed. The approximation errors resulting from the mean-field assumptions have little influence because the posterior updates are localized in the input domain.

### Table I: Accuracy on the SARCOS inverse dynamics dataset.

| Model | MSE | NMSE | Models |
|-------|-----|-----|--------|
| ILR   | \( (4.80 \pm 0.30) \times 10^{-1} \) | \( (3.40 \pm 0.20) \times 10^{-3} \) | 1700 |
| LGPR* | \( (86.00 \pm 0.00) \times 10^{-1} \) | \( (50.00 \pm 0.00) \times 10^{-3} \) | 7000 |
| LWPR  | \( (26.00 \pm 0.30) \times 10^{-1} \) | \( (18.00 \pm 0.20) \times 10^{-3} \) | 32000 |
| GPR†  | \( (6.10 \pm 0.20) \times 10^{-1} \) | \( (4.10 \pm 0.07) \times 10^{-3} \) | - |
| SGPR  | \( (8.50 \pm 0.03) \times 10^{-1} \) | \( (6.00 \pm 0.008) \times 10^{-3} \) | - |

### Table II: Accuracy on the Barrett inverse dynamics dataset.

| Model | MSE | NMSE | Models |
|-------|-----|-----|--------|
| ILR   | \( (2.90 \pm 0.50) \times 10^{-1} \) | \( (7.0 \pm 0.5) \times 10^{-3} \) | 1350 |
| LGPR* | \( (7.70 \pm 0.00) \times 10^{-1} \) | \( (17.0 \pm 0.0) \times 10^{-3} \) | 3270 |
| LWPR  | \( (10.00 \pm 1.50) \times 10^{-1} \) | \( (37.0 \pm 10) \times 10^{-3} \) | 29000 |
| GPR†  | \( (1.000 \pm 0.003) \times 10^{-1} \) | \( (2.30 \pm 0.01) \times 10^{-3} \) | - |
| SGPR  | \( (1.80 \pm 0.005) \times 10^{-1} \) | \( (6.30 \pm 0.02) \times 10^{-3} \) | - |

Next we learn the inverse dynamics of real anthropomorphic manipulators governed by \( M(q) \ddot{q} + C(q, \dot{q}) + G(q) + \epsilon(q, \dot{q}, \ddot{q}) = u \), where \( q, \dot{q}, \ddot{q} \) are joint angles, velocities and accelerations and \( u \) are the torques. \( M(q) \) is the inertia matrix, \( C(q, \dot{q}) \) are the Coriolis and centrifugal forces and \( G(q) \) is the gravity force. \( \epsilon(q, \dot{q}, \ddot{q}) \) are general unmodelled nonlinearities such as sticktion/friction and hydraulic and tendon/cable dynamics, thus we take a data-driven approach to learn the mapping \( q, \dot{q}, \ddot{q} \rightarrow u \). The learned model is then used for model-based control. We use the Mean Squared Error (MSE), Normalized Mean Squared Error (NMSE) and the number of active models 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 LGPR [29], LWPR [27], GPR [3] and SGPR [10].

SARCOS and Barrett-WAM Datasets. We benchmark the prediction accuracy of ILR on a relatively high-dimensional dataset collected from a 7-DoF anthropomorphic SARCOS arm [27]. The dataset consists of 44484 training points and 4449 test cases. Overall there are 21 input variables, \( q, \dot{q}, \ddot{q} \) mapping to the torques of the 7 motors \( u \). We also benchmark on a real inverse dynamics dataset of a 4-DoF Barrett-WAM manipulator, Figure 1, a mapping from a 12-D to 4-D space. The dataset contains 25000 training pairs and 5000 test cases. Table I and Table II list the results for both datasets. We report the average MSE, NMSE, and number of active models over all joints. The results are obtained by running five seeds and computing the means and standard deviations 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, and had to limit the dataset to 35000 instead of 44484. The results of this evaluation show that ILR clearly outperforms LWPR and LGR on both datasets, in terms of prediction accuracy and number of used models. On the SARCOS dataset, ILR even outperforms GPR. We attribute this result however, to the fact that GPR had to work with a reduced training set\(^*\), and acknowledge that GPR is still the gold standard on medium-sized datasets. Finally, the results indicate that ILR is, in general, very competitive with SGPR.

**Real Inverse Dynamics Control.** We demonstrate the validity of the learned dynamics captured by ILR by using the learned model in an online desired-trajectory tracking scenario on a Barrett-WAM, while applying a low-gain PD-controller and to the tracking precision of the "model-free" PD-controller. Figure 7 shows a qualitative comparison of the different controllers on two test trajectories. For the second task, we construct a similar scenario, albeit to learn a model that covers a larger region of the state-action space and computing quantitative precision benchmarks. We generate a larger real Barrett dataset consisting of 150-thousand training examples (roughly 5 minutes). The movements are distinct sinusoidal joint-space trajectories with slow and fast velocity profiles. We repeat the process of the previous task and run ILR on held-out test trajectories with the same low-gain PD-controller and compare with the analytical model and "model-free" PD. As benchmarking criteria, we evaluate the joint-angle MSE w.r.t. desired trajectory and the mean torque contributed by the low-gain PD-controller to the overall control signal. The rationale behind is as follows; A good inverse dynamics model will consistently produce a low MSE while not relying on the PD-controller’s assistance in the background. Table III shows the benchmark quantities for 3 test trajectories. The results indicate that ILR outperforms both control strategies and achieves good tracking with little contribution from the PD-controller. During both tasks, we were able to consistently achieve a prediction frequency of 2000 Hz, while the Barrett-WAM requires "only" 500 Hz. Supplementary material contains a video demonstration of executed trajectories.

**V. CONCLUSION**

We presented and successfully applied a computationally efficient Variational Bayes technique for learning of inverse dynamics, which is based on the principles of infinite mixtures and Bayesian Nonparametrics. We situated this approach as the next iteration in a large family of local regression techniques such as RFWR, LWPR, and LGR. We showed that placing Dirichlet Process priors on Bayesian mixtures of local regression models can regularize model complexity with little loss in performance and without relying on heuristics. Empirical evaluations showed that the model offers well-calibrated uncertainty quantification, outperforms LWPR and LGR, and is competitive with Sparse GPR. Moreover, we have shown the practicality of using this approach for online inverse dynamics control on a Barrett-WAM. Further research will consider higher model compression by constructing hierarchical models that enable parameter sharing between local regressors.
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 dynamic models,” IEEE RA-L, 2018.

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

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

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

[6] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, “Manifold Gaussian processes for regression,” in IJCNN. IEEE, 2016.

[7] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, “Deep kernel learning,” in AISTATS, 2016.

[8] H. Salimbeni and M. Deisenroth, “Doubly stochastic variational inference for deep Gaussian processes,” in NIPS, 2017.

[9] R. Herbrich, N. D. Lawrence, and M. Seeger, “Fast sparse Gaussian process methods: The informative vector machine,” in NIPS, 2003.

[10] M. Titsias, “Variational learning of inducing variables in sparse gaussian processes;” in AISATS, 2009.

[11] F. Meier, P. Hennig, and S. Schaal, “Incremental local Gaussian regression,” in ICML, 2007.

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

[13] Q. V. Le, A. J. Smola, and S. Canu, “Heteroscedastic Gaussian process regression,” in ICML, 2005.

[14] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, “Most likely heteroscedastic Gaussian process regression,” in ICML, 2007.

[15] H. Liu, Y.-S. Ong, and J. Cai, “Large-scale heteroscedastic regression via Gaussian process,” arXiv preprint:1811.01179, 2020.

[16] J. Frankle and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” in ICLR, 2019.

[17] M. Mccloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in Psychology of Learning and Motivation, Elsevier, 1989.

[18] R. M. Neal, Bayesian Learning for Neural Networks. Springer, 2012.

[19] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in NIPS, 2017.

[20] 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,” ICML, 2018.

[21] 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?” ICML, 2020.

[22] A. Y. Foong, D. R. Burt, Y. Li, and R. E. Turner, “On the expressiveness of approximate inference in bayesian neural networks,” arXiv, 2019.

[23] J. A. Ting, F. Meier, S. Vijayakumar, and S. Schaal, “Locally weighted regression for control,” in Encyclopedia of Machine Learning and Data Mining. Springer, 2010.

[24] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally weighted learning for control,” Artificial Intelligence Review, 1997.

[25] O. Nelles and R. Isermann, “Basis function networks for interpolation of local linear models,” Conference on Decision and Control, 1996.

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

[27] S. Vijayakumar, A. D’Souza, and S. Schaal, “Incremental online learning in high dimensions,” Neural Computation, 2005.

[28] J. A. Ting, M. Kalakrishnan, S. Vijayakumar, and S. Schaal, “Bayesian kernel shaping for learning control,” in NIPS, 2009.

[29] F. Meier, P. Hennig, and S. Schaal, “Incremental local Gaussian regression,” in NIPS, 2014.

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

[31] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the EM algorithm,” Neural Computation, 1994.

[32] C. E. Rasmussen and Z. Ghahramani, “Infinite mixtures of Gaussian process experts,” in NIPS, 2001.

[33] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, Bayesian Nonparametrics. Cambridge University Press, 2010.