Inter-Domain Gaussian Processes with Arbitrary Mean Functions in Factor Graphs

Alex Ledbetter — *TU Eindhoven, Netherlands*
Hoang Minh Huu Nguyen — *TU Eindhoven, Netherlands*
Lucas C. van Laake — *TU Eindhoven, Netherlands*
Irene Kuling — *TU Eindhoven, Netherlands*
Yoeri van de Burgt — *TU Eindhoven, Netherlands*
Thijs van de Laar — *TU Eindhoven, Netherlands*

Abstract

The modeling flexibility and expressivity of inter-domain Gaussian processes, enabled by applying arbitrary linear operators to a latent Gaussian process, is significant and desirable in robotics and probabilistic numerics communities. However, inter-domain Gaussian processes have not yet been available in a probabilistic message-passing formulation. In this paper, we formalize how to enable inter-domain observations for Gaussian processes in a variational message-passing framework. We develop a decoupled inter-domain variational sparse Gaussian process (dID-VSGP) model for univariate latent Gaussian processes with arbitrary mean functions, and derive the mean-field variational message-passing update rules that allow inference in this model. We validate our derivations in a shape exploration and modeling task, and by solution to a linear stochastic partial differential equation representative of physics-informed exploration. We confirm that our message-passing implementation maintains the same scaling complexity as the VSGP analytical solution. Our results further unify the analytical and message-passing approaches to variational inference and enable inter-domain observations in factor-graph Gaussian processes.

Proceedings of the 2nd International Conference on Probabilistic Numerics (ProbNum) 2026, Lappeenranta, Finland. PMLR Volume TBA.

1 Introduction

Gaussian processes (GPs) provide a flexible, non-parametric framework for probabilistic modeling (Rasmussen and Williams, 2006). They are particularly well suited for object contour estimation in robotics, where objects can be represented as implicit surface potentials that encode shape, position, and orientation. By specifying an appropriate mean function, a GP can capture the far-field behavior of such implicit surface representations (Dragiev et al., 2011).

Incorporating derivative information further enhances the implicit surface representation. By conditioning on gradient observations, one obtains an implicit surface field that describes, at any point in space, the direction toward the nearest point on the object's surface (Dragiev et al., 2011). This directional information can be exploited for tasks such as robotic grasp planning and control (Toussaint, 2009; Dragiev et al., 2013; Solak et al., 2002).

More generally, Gaussian processes can be extended to accommodate arbitrary linear operators and inter-domain observations (Lázaro-Gredilla and Figueiras-Vidal, 2009; Matsumoto and Sullivan, 2024), enabling the incorporation of (coupled) differential equations and facilitating physics-informed modeling approaches (Särkkä, 2011). The term *inter-domain* was coined to recognize that, in general, the transformed function can be defined over the original function domain and additional transformed domains.

A key limitation of standard GPs is their computational complexity. The kernel matrix scales with the number of observations, making matrix inversion prohibitively expensive for large datasets. This challenge particularly limits real-time estimation of time-series models with streaming data. To address this issue, a range of sparse GP methods have been proposed (Snelson and Ghahramani, 2005; Quiñonero-Candela and Rasmussen, 2005), including the variational sparse Gaussian process (VSGP) model upon which our work is based (Titsias, 2009; Nguyen et al., 2025), which enhances tractability for large datasets by approximating the kernel matrix using a reduced set of inducing points.

Several software libraries support GP modeling, including scikit-learn (Pedregosa et al., 2011), GPflow (Matthews et al., 2017), and GPyTorch (Gardner et al., 2018). While some of these frameworks provide support for derivative observations, real-time robotic applications demand highly efficient, analytic inference schemes. Such efficiency can be achieved through message passing on a factor graph. A factor node in the graph may represent a Gaussian process submodel with arbitrary mean function under possibly linear transformations. Inference then amounts to message passing on the graph, where pre-computed messages can be re-used across models. This message-passing approach to inference can be implemented within a reactive probabilistic programming framework (Bagaev and De Vries, 2023).

Our contributions in this paper are two-fold:

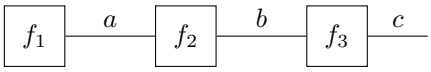

Figure 1: An example Forney-style factor graph.

1. We derive variational message updates for a novel decoupled inter-domain variational sparse Gaussian process (dID-VSGP) factor node for transformed univariate latent Gaussian processes with arbitrary mean functions.
2. We demonstrate the generality of our dID-VSGP node for free-form probabilistic modeling by applying the derived updates in two distinct contexts; namely, object contour estimation including gradient observations, and estimation of physics-informed models involving differential equations.

Section 2 reviews variational message passing and Gaussian processes. Section 3 presents the novel GP node and theoretical message update results. Section 4 presents the simulation results. Section 5 discusses strengths and limitations of the proposed approach and opportunities for future developments. Section 6 concludes the presented work.

2 Background

We briefly review Forney-style factor-graphs (FFGs) and variational message passing in Section 2.1, and Gaussian processes in Section 2.2, that together provide the necessary material to understand the innovations in this paper.

2.1 Variational Message Passing over Factor-Graphs

A Forney-style factor graph is a graphical representation of a factorized function,

$$\mathrm{f}(\boldsymbol{z}) = \prod_a \mathrm{f}_a(\boldsymbol{z}_a)\,, \tag{1}$$

where a variable z_i is represented by an edge i, and a factor f_a by a node a (Forney, 2001; Loeliger, 2004). An edge i connects to a node a if the variable z_i is an argument of the node function f_a. See Fig. 1 for an illustration of the FFG of an example function $\mathrm{f}(a,b,c)$ that can be factored as $\mathrm{f}(a,b,c) = \mathrm{f}_1(a)\mathrm{f}_2(a,b)\mathrm{f}_3(b,c)$. As a notational convention we will represent sequences, vectors and matrices by a bold script. Furthermore, we will use f to denote a factor function, and f to denote a variable over functions (as modeled by a Gaussian process).

In a probabilistic model, the factors (graph nodes) represent conditional and prior probability distributions, so that the model becomes a joint probability distribution over the modeled variables (graph edges). As an example, consider the graph in Fig. 2, which represents the model,

$$p(y,x,f) = p(y|f)p(f|x)p(x)\,. \tag{2}$$

We use a small black square to indicate an observation $y = \hat{y}$. Given the observation, inference can then be performed by message passing on the FFG,

$$\begin{aligned} p(x|y=\hat{y}) &\propto \int p(y=\hat{y}|f)p(f|x)p(x)\,\mathrm{d}f \qquad (3)\\ &= \underbrace{p(x)}_{\overrightarrow{\nu}(x)} \underbrace{\int \overbrace{p(y=\hat{y}|f)}^{\overleftarrow{\nu}(f)} p(f|x)\,\mathrm{d}f}_{\overleftarrow{\nu}(x)}\,, \end{aligned}$$

where a message $\nu(z_i)$ is a function of the edge variable z_i that summarizes — by marginalization — all factors on one side of edge i. For example, the message $\overleftarrow{\nu}(x)$ summarizes the factors $p_{y|f}$ and $p_{f|x}$ on the right-hand side of edge x by marginalizing out f.

Although an FFG is principally undirected, we impose an edge direction to anchor the message direction (forward or backward). Message computations can be nested, and a message-passing schedule specifies the order of consecutive message computations (message updates). Message updates can be pre-derived for specific factor functions, and stored in a lookup table, such that they can be reused across models for increased efficiency.

When exact message passing becomes intractable due to, for example, inclusion of non-conjugate factors, such as the one we develop in this paper, one can resort to approximations such as *variational* message-passing. In this paper we will derive variational message updates for the dID-VSGP node.

Variational message updates minimize a variational free energy (VFE) objective,

$$\mathcal{F}[q] = \mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{q(\boldsymbol{z})}{\mathrm{f}(\boldsymbol{z})}\right]\,, \tag{4}$$

under additional constraints (e.g. factorization) on the variational distribution q, (Senöz et al., 2021). Under a mean-field factorization of the variational distribution,

$$q(\boldsymbol{z}) \triangleq \prod_i q(z_i)\,, \tag{5}$$

the update for a variational message from a node a to an edge i becomes,

$$\log \nu(z_i) = \mathbb{E}_{q(\boldsymbol{z}_{a\setminus i})}[\log \mathrm{f}_a(\boldsymbol{z})]\,, \tag{6}$$

with $q(\boldsymbol{z}_{a\setminus i})$ the local variational factorization of the variables on the edges connected to node a, excluding the term $q(z_i)$ Dauwels (2007).

The product of colliding messages on an edge computes an update to the variational, marginal distribution, e.g.

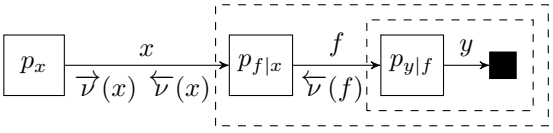

Figure 2: An example Forney-style factor graph with indicated messages. The small black box indicates an observed value. The dashed lines indicate boxes that are summarized by messages.

for variable z_i,

$$q(z_i) \propto \overrightarrow{\nu}(z_i)\overleftarrow{\nu}(z_i)\,. \tag{7}$$

The updates per Eqs. (6) and (7) are computed for all variables z_i and iterated until convergence. The computation of Eq. (6) can often be derived in analytical form, making inference fast and scalable.

2.2 Gaussian Processes

A univariate Gaussian process (GP) can be interpreted as a distribution over scalar functions,

$$f \sim GP(m(\boldsymbol{x}), k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}'))\,, \tag{8}$$

with a mean function m and kernel function $k_{\boldsymbol{\theta}}$ that is parameterized by a hyper-parameter $\boldsymbol{\theta}$. More precisely, a GP is a set of random variables, any finite subset of which is normally distributed (Rasmussen and Williams, 2006). By univariate, we mean that the output of the function is scalar-valued, while the input $\boldsymbol{x}$ can be multivariate.

The set of training variables are values of the latent function, $\boldsymbol{f} = \{f_i\}_{i=1}^N$, evaluated at a set of training inputs, $X = \{\boldsymbol{x}_i\}_{i=1}^N$. Correlation between function values of nearby inputs is modeled by a kernel function. This kernel function ensures that the correlation patterns expressed by nearby values exhibit the linear, smooth, periodic, or other properties of the chosen kernel.

For GP regression, we do not typically observe a latent function value directly, but a noisy proxy instead. For a univariate GP,

$$y = f(\boldsymbol{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)\,. \tag{9}$$

This relationship is then captured by the observation model $p(y \mid f, \boldsymbol{\theta})$.

From the training variables, inducing inputs and observations $\hat{y}$, one can maximize the log-marginal likelihood $\log p(y = \hat{y} \mid \boldsymbol{\theta})$ to fit the kernel hyper-parameters $\boldsymbol{\theta}$ (Rasmussen and Williams, 2006). Predictions can then be made for test variables, f_*, at arbitrary test inputs, $\boldsymbol{x}_*$.

In Section 3, we will extend this Gaussian process formulation to allow for modeling of inter-domain functions, $\tilde{\boldsymbol{f}} = \mathcal{L}f$, where $\mathcal{L}$ is a linear operator, and for sparse approximations to reduce computational complexity.

3 Methods

In this section, we address the problems defined in Section 1 by deriving the dID-VSGP prior in Section 3.1, the associated node function in Section 3.2, and the corresponding variational message-passing update rules in Section 3.3. Finally, we derive the posterior predictive distribution for the node in Section 3.4.

3.1 Sparse Inter-Domain GP Prior

A GP can be generalized to allow for application of deterministic linear operators, $\mathcal{L}$, to transform the function f (Särkkä, 2011). For example, for a function $f(x, y, z)$, $\mathcal{L}$ could be a gradient operator, $\mathcal{L}f = \nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right]^{\mathrm{T}}$, producing a vector field, or $\mathcal{L}$ can house an inter-domain stack of operators like $\mathcal{L} = \left[\mathcal{I}, \nabla\right]^{\mathrm{T}}$, where $\mathcal{I}$ is the identity operator, to model the joint distribution over the latent function and function gradient (Lázaro-Gredilla and Figueiras-Vidal, 2009). Note that while the operator, $\mathcal{L}$, may produce tensor-valued outputs in general, these are represented in vectorized form without loss of information.

We indicate transformed functions by a tilde,

$$\tilde{\boldsymbol{f}}(\boldsymbol{x}) \triangleq \mathcal{L}f(\boldsymbol{x}) \in \mathbb{R}^P\,. \tag{10a}$$

The GP for the transformed function then becomes (Matsumoto and Sullivan, 2024),

$$\tilde{\boldsymbol{f}} \sim GP(\mathcal{L}m(\boldsymbol{x}), \mathcal{L}_1\mathcal{L}_2 k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}'))\,, \tag{11a}$$

where the subscript under $\mathcal{L}$ indicates the argument position that the operator applies to.

Eq. (11) represents an exact, inter-domain GP formulation. An exact GP formulation has efficiency limitations, and so we utilize a sparse GP approximation to mitigate these. In sparse GP approximations, including the VSGP formulation, a limited, pseudo-observation inducing-dataset of size $M \ll N$, with inducing inputs $X_{\boldsymbol{u}} = \{\boldsymbol{x}_{\boldsymbol{u}}^{(i)}\}_{i=1}^M$ and inducing variables $\boldsymbol{u} = \{u_i\}_{i=1}^M$, are utilized to reduce computational complexity (Snelson and Ghahramani, 2005). Most sparse Gaussian-process variants approximate the joint distribution over training and test variables by assuming that the train and test functions $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{f}}_*$ are conditionally independent given the inducing variables $\boldsymbol{u}$ (Quiñonero-Candela and Rasmussen, 2005). This means that in practice, we will learn the inducing variables $\boldsymbol{u}$ that best explain the observed transformed function values $\tilde{\boldsymbol{f}}$, and then use these inducing variables to make predictions for the test function $\tilde{\boldsymbol{f}}_*$.

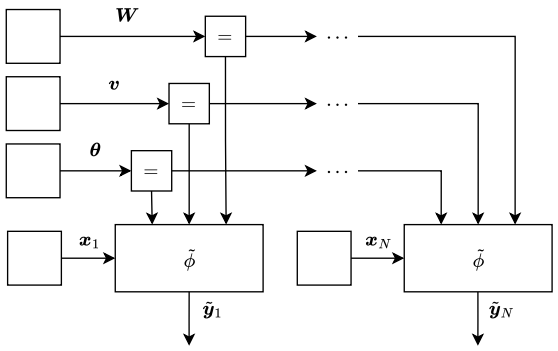

Figure 3: An FFG utilizing the developed dID-VSGP nodes to perform regression to a dataset consisting of input, $X = \{x_i\}_{i=1}^M$, and transformed output, $\tilde{Y} = \{\tilde{y}_i\}_{i=1}^M$, pairs. Note that there is no obligation that the $\tilde{\phi}(\cdot)$ node functions are transformed by the same linear operator, only that the observations, $\tilde{y}_i$, associated with each node correspond to the same transformation.

For notational convenience we introduce the following shorthands,

$$\boldsymbol{m}_{\boldsymbol{u}} \triangleq \left[\boldsymbol{m}\left(\boldsymbol{x}_{\boldsymbol{u}}^{(1)}\right), \ldots, \boldsymbol{m}\left(\boldsymbol{x}_{\boldsymbol{u}}^{(M)}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{M} \tag{12a}$$

$$\tilde{\boldsymbol{m}}(\boldsymbol{x}) \triangleq \mathcal{L}m(\boldsymbol{x}) \in \mathbb{R}^{P} \tag{12b}$$

$$\tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta}) \triangleq \mathcal{L}_1 k_{\boldsymbol{\theta}}(\boldsymbol{x}, X_{\boldsymbol{u}}) \in \mathbb{R}^{P \times M} \tag{12c}$$

$$\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}, \boldsymbol{\theta}) \triangleq \mathcal{L}_2 k_{\boldsymbol{\theta}}(X_{\boldsymbol{u}}, \boldsymbol{x}) \in \mathbb{R}^{M \times P} \tag{12d}$$

$$\tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{\theta}) \triangleq \mathcal{L}_1 \mathcal{L}_2 k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}^{P \times P} \tag{12e}$$

$$\boldsymbol{K}_{\boldsymbol{uu}}(\boldsymbol{\theta}) \triangleq k_{\boldsymbol{\theta}}(X_{\boldsymbol{u}}, X_{\boldsymbol{u}}) \in \mathbb{R}^{M \times M}\,. \tag{12f}$$

We then construct the joint distribution over $\tilde{\boldsymbol{f}}$, $\tilde{\boldsymbol{f}}_*$, and $\boldsymbol{u}$ as

$$p(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{f}}_*, \boldsymbol{u} \mid \boldsymbol{x}, \boldsymbol{x}_*, \boldsymbol{\theta}) = \mathcal{N}\left(\begin{bmatrix}\tilde{\boldsymbol{f}} \\ \tilde{\boldsymbol{f}}_* \\ \boldsymbol{u}\end{bmatrix} \middle| \begin{bmatrix}\tilde{\boldsymbol{m}}(\boldsymbol{x}) \\ \tilde{\boldsymbol{m}}(\boldsymbol{x}_*) \\ \boldsymbol{m}_{\boldsymbol{u}}\end{bmatrix}, \begin{bmatrix}\tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{\theta}) & \tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}, \boldsymbol{x}_*, \boldsymbol{\theta}) & \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta}) \\ \tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}_*, \boldsymbol{x}, \boldsymbol{\theta}) & \tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}_*, \boldsymbol{x}_*, \boldsymbol{\theta}) & \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}_*, \boldsymbol{\theta}) \\ \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}, \boldsymbol{\theta}) & \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}_*, \boldsymbol{\theta}) & \boldsymbol{K}_{\boldsymbol{uu}}(\boldsymbol{\theta})\end{bmatrix}\right). \tag{13}$$

From here we can utilize standard Gaussian conditioning and Schur complement formulas (Petersen and Pedersen, 2012) to express the conditional distribution,

$$\begin{aligned} &p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \\ &\quad = \mathcal{N}(\tilde{\boldsymbol{f}} \mid \tilde{\boldsymbol{m}}(\boldsymbol{x}) + \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta}) \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{\theta})(\boldsymbol{u} - \boldsymbol{m}_{\boldsymbol{u}}), \\ &\qquad \tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{\theta}) - \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta}) \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{\theta}) \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}, \boldsymbol{\theta}))\,. \end{aligned} \tag{14}$$

We can substitute $\boldsymbol{x}_*$ and $\tilde{\boldsymbol{f}}_*$ for $\boldsymbol{x}$ and $\tilde{\boldsymbol{f}}$ to obtain the conditional distribution for the test function. Then, the distribution in Eq. (14) represents the transformed sparse Gaussian process prior for single input/output pairs, $\boldsymbol{x}/\tilde{\boldsymbol{f}}$ and $\boldsymbol{x}_*/\tilde{\boldsymbol{f}}_*$, respectively. Because, in general, applied linear operators can span both latent and transformed spaces, we refer to this prior as a decoupled *inter-domain* variational sparse Gaussian process (dID-VSGP) prior. In this paper, the latent Gaussian process is univariate with arbitrary mean function.

Note that we make a design choice to keep the inducing variables in the latent function space (i.e. in the space of f instead of $\tilde{\boldsymbol{f}}$), while enabling inter-domain observations via the transformed kernels in Eqs. (12c) to (12e), hence the label "decoupled" (Wu et al., 2021). This choice has some benefits and limitations which we address in Section 5.

3.2 Node Function Derivation

We consider a generative process whereby noisy "inter-domain" observations, (Lázaro-Gredilla and Figueiras-Vidal, 2009), of a linearly-transformed univariate Gaussian process are obtained as

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\epsilon}\,, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{W}^{-1}\right)\,. \tag{15}$$

We define the following shorthand notation for the reparameterization to transformed inducing points, and prior mean and covariance as

$$\boldsymbol{v} \triangleq \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{\theta})\boldsymbol{u} \in \mathbb{R}^{M} \tag{16a}$$

$$\tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}) \triangleq \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta})(\boldsymbol{v} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{\theta})\boldsymbol{m}_{\boldsymbol{u}}) \tag{16b}$$

$$\begin{aligned} \tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\theta}) \triangleq\ & \tilde{\boldsymbol{K}}_{\boldsymbol{xx'}}(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{\theta}) \\ & - \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}, \boldsymbol{\theta}) \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{\theta}) \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}, \boldsymbol{\theta})\,. \end{aligned} \tag{16c}$$

With reference to the Gaussian process prior developed in Eq. (14), augmented by the shorthand notation in Eq. (16), we can define a generative model associated with the generative process in Eq. (15) as

$$p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}) = \mathcal{N}(\tilde{\boldsymbol{f}} \mid \tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}), \tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\theta})) \tag{17a}$$

$$p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W}) = \mathcal{N}(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W}^{-1}) \tag{17b}$$

$$p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{f}}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{W}, \boldsymbol{\theta}) = p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W})\, p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})\,. \tag{17c}$$

This factorized generative model contains two terms, namely, $p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W})$ and $p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})$. Following the design choices made by Nguyen et al. (2025), we now seek to define a single composite node-function, with tractable update rules, associated with this generative model. Similar to Nguyen et al. (2025), we take inspiration from Titsias (2009) and establish the variational approximation to the true posterior distribution to be

$$\begin{aligned} &q(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{f}}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{W}, \boldsymbol{\theta}) \\ &\qquad = q(\tilde{\boldsymbol{y}}) p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}) q(\boldsymbol{x}) q(\boldsymbol{v}) q(\boldsymbol{W}) q(\boldsymbol{\theta})\,, \end{aligned} \tag{18}$$

which results in a convenient cancellation in Eq. (19).

We can then evaluate the variational free energy associated with this approximation. Defining $\boldsymbol{s}' = \{\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{f}}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{W}, \boldsymbol{\theta}\}$ and $\boldsymbol{s} = \{\tilde{\boldsymbol{y}}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{W}, \boldsymbol{\theta}\}$, we have

$$\begin{aligned} \mathcal{F}[q] &= \mathbb{E}_{q(\boldsymbol{s}')}\left[\log \frac{q(\boldsymbol{s}')}{p(\boldsymbol{s}')}\right] \\ &= \mathbb{E}_{q(\boldsymbol{s})}[\log q(\boldsymbol{s})] \\ &\quad - \mathbb{E}_{q(\boldsymbol{s})}\left[\mathbb{E}_{p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})}[\log p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W})]\right], \end{aligned} \tag{19}$$

where the term $p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})$ cancels out from the numerator and denominator in $q(\boldsymbol{s}')/p(\boldsymbol{s}')$.

If we define

$$\log \tilde{\boldsymbol{\phi}}(\boldsymbol{s}) \triangleq \mathbb{E}_{p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})}[\log p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W})]\,, \tag{20}$$

then we can express Eq. (19) as

$$\mathcal{F}[q] = \mathbb{E}_{q(\boldsymbol{s})}\left[\log \frac{q(\boldsymbol{s})}{\tilde{\boldsymbol{\phi}}(\boldsymbol{s})}\right], \tag{21}$$

where we can recognize a variational free energy functional associated with a mean-field variational approximation for a generative model of a single factor, $\tilde{\boldsymbol{\phi}}(\boldsymbol{s})$.

This collapsed-prior observation model, $\tilde{\boldsymbol{\phi}}(\boldsymbol{s})$, is exactly the composite node-function we were seeking to define. By Eq. (20) and by Jensen's inequality, we can interpret the logarithm of this composite factor to represent a lower-bound approximation to a true $\tilde{\boldsymbol{f}}$-marginalized composite log factor, $\log \mathbb{E}_{p(\tilde{\boldsymbol{f}} \mid \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})}[p(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{f}}, \boldsymbol{W})]$.

Finally, evaluating the exponential of the expectation in Eq. (20), we can establish the node function for the dID-VSGP node to be

$$\tilde{\phi}(\boldsymbol{s}) = \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{W}\tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{\theta})\right)\right) \times \mathcal{N}\left(\tilde{\boldsymbol{y}} \mid \tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta}), \boldsymbol{W}^{-1}\right) . \tag{22}$$

See Fig. 4 for a diagram of the derived composite node construction.

3.3 Variational Message-Passing Update Rules

Now that we have derived a single probabilistic factor node, we can derive the variational message-passing update rules that allow use of thise node in a factor graph.

The variational message-passing update rules for the dID-VSGP node, as depicted in Fig. 5, are summarized in Table 1. Below we define the shorthands that are used in the table.

We use a wide tilde for expressions that additionally involve an expectation,

$$\widetilde{\boldsymbol{m}} = \mathbb{E}_{q(\boldsymbol{x})}[\tilde{\boldsymbol{m}}(\boldsymbol{x})] . \tag{23}$$

Shorthands for the other variables follow the same tilde conventions. When a particular kernel parameter $\hat{\boldsymbol{\theta}}$ is substituted, we omit this term from the expression,

$$\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}) = \mathcal{L}_2 k_{\hat{\boldsymbol{\theta}}}(X_{\boldsymbol{u}}, \boldsymbol{x}) \tag{24a}$$
$$\widetilde{\boldsymbol{K}}_{\boldsymbol{ux}} = \mathbb{E}_{q(\boldsymbol{x})}\left[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\right] \tag{24b}$$
$$\boldsymbol{K}_{\boldsymbol{uu}} = k_{\hat{\boldsymbol{\theta}}}(X_{\boldsymbol{u}}, X_{\boldsymbol{u}}) . \tag{24c}$$

We additionally omit the subscript $\boldsymbol{u}$ from the Gaussian process prior covariance,

$$\tilde{\boldsymbol{A}}(\boldsymbol{x}) = \tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \hat{\boldsymbol{\theta}}) \tag{25a}$$
$$\widetilde{\boldsymbol{A}}(\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{x})}\left[\tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\theta})\right] \tag{25b}$$
$$\widetilde{\boldsymbol{A}} = \mathbb{E}_{\boldsymbol{x}}\left[\tilde{\boldsymbol{A}}(\boldsymbol{x}, \hat{\boldsymbol{\theta}})\right] , \tag{25c}$$

and mean,

$$\tilde{\boldsymbol{b}}(\boldsymbol{x}) = \tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x}, \overline{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}) \tag{26a}$$
$$\widetilde{\boldsymbol{b}}(\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{x})}\left[\tilde{\boldsymbol{b}}(\boldsymbol{x}, \overline{\boldsymbol{v}}, \boldsymbol{\theta})\right] . \tag{26b}$$

Finally we summarize some unwieldy expressions in dedicated terms,

$$\boldsymbol{\eta} = \mathbb{E}_{\boldsymbol{x}}\left[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\overline{\boldsymbol{W}}\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{m}}(\boldsymbol{x}) + \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\overline{\boldsymbol{m}}_{\boldsymbol{u}}\right)\right] \tag{27a}$$

$$\boldsymbol{U} = \mathbb{E}_{\boldsymbol{x}}\left[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\overline{\boldsymbol{W}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\right] , \tag{27b}$$

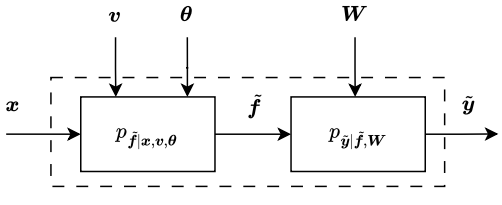

Figure 4: Composite dID-VSGP node

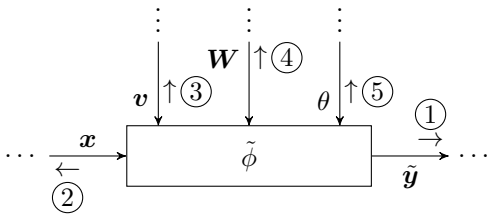

Figure 5: Factor graph representation of the dID-VSGP node with indicated messages.

and as a function of state and parameters,

$$\boldsymbol{S}(\boldsymbol{x}) = \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{v}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}) \tag{28a}$$
$$\boldsymbol{S}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\left[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x},\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{v}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x},\boldsymbol{\theta})\right] \tag{28b}$$
$$\boldsymbol{B}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\left[\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}},\boldsymbol{\theta})\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}},\boldsymbol{\theta})^{\mathrm{T}}\right] \tag{28c}$$
$$\boldsymbol{D} = \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}} + \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{S}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}}\left[(\tilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x}))(\tilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x}))^{\mathrm{T}}\right] . \tag{28d}$$

For some kernel families these terms can be expressed analytically. In other cases we resort to quadrature methods to evaluate expectations.

3.4 Posterior Predictive Distribution

The (approximate) posterior predictive distribution is obtained by marginalizing the inducing variables under the variational posteriors $q(\boldsymbol{v})$ and $q(\boldsymbol{\theta})$ from the transformed prior for the test variables given in Eq. (14). After making the shorthand substitutions found in Eq. (16) this becomes

$$q(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{x}_*) = \int p(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{x}_*, \boldsymbol{v}, \boldsymbol{\theta})\, q(\boldsymbol{v})q(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v}\mathrm{d}\boldsymbol{\theta} . \tag{29}$$

Provided we have inferred a posterior distribution for $q(\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v} \mid \boldsymbol{\mu}_{\boldsymbol{v}}, \boldsymbol{\Sigma}_{\boldsymbol{v}})$, and have an optimized belief for $q(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$, we evaluate this integral to be

$$q(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{x}_*) = \mathcal{N}(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{\mu}_{\tilde{\boldsymbol{f}}_*}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{f}}_*}) \tag{30}$$
$$\boldsymbol{\mu}_{\tilde{\boldsymbol{f}}_*} = \tilde{\boldsymbol{m}}(\boldsymbol{x}_*) + \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}_*)(\boldsymbol{\mu}_{\boldsymbol{v}} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{x}_*)\boldsymbol{m}_{\boldsymbol{u}}) \in \mathbb{R}^P$$
$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{f}}_*} = \tilde{\boldsymbol{K}}_{\boldsymbol{xx}'}(\boldsymbol{x}_*, \boldsymbol{x}_*) + \tilde{\boldsymbol{K}}_{\boldsymbol{xu}}(\boldsymbol{x}_*)\left(\boldsymbol{\Sigma}_{\boldsymbol{v}} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\right)\tilde{\boldsymbol{K}}_{\boldsymbol{u}*}(\boldsymbol{x}_*) \in \mathbb{R}^{P\times P} .$$

This distribution is evaluated at test locations in the simulations in Section 4.

4 Results

In all of the following experiments, the VSGP nodes have been implemented in RxInfer v4.7.1, (Bagaev et al., 2023), an efficient reactive factor-graph message-passing probabilistic programming-language (FGMP-PPL) package written in Julia. Upon inclusion of observed data, variational message passing proceeds until convergence of the VFE.

Table 1: Update rules for the dID-VSGP node under a naive (mean-field) variational approximation and point-mass constraint on $\boldsymbol{\theta}$. Shorthand terms are specified below. The overbar notation indicates an expectation $\overline{x} = \mathbb{E}_{q(x)}[x]$, and P indicates the dimensionality of $\tilde{\boldsymbol{y}}$.

①	$\overrightarrow{\nu}(\tilde{\boldsymbol{y}}) \propto \mathcal{N}\Big(\tilde{\boldsymbol{y}} \vert \widetilde{\boldsymbol{m}} + \widetilde{\boldsymbol{K}}_{\boldsymbol{ux}}^{\mathrm{T}}(\overline{\boldsymbol{v}} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}), \overline{\boldsymbol{W}}^{-1}\Big)$
②	$\log \overleftarrow{\nu}(\boldsymbol{x}) = -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\big(\tilde{\boldsymbol{A}}(\boldsymbol{x}) + \boldsymbol{S}(\boldsymbol{x})\big)\Big) - \frac{1}{2}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x})) + C$
③	$\overleftarrow{\nu}(\boldsymbol{v}) \propto \mathcal{N}_{\xi}(\boldsymbol{v} \vert \boldsymbol{\eta}, \boldsymbol{U})$
④	$\overleftarrow{\nu}(\boldsymbol{W}) \propto \mathcal{W}\Big(\boldsymbol{W} \vert (\tilde{\boldsymbol{A}} + \boldsymbol{D})^{-1}, P + 2\Big)$
⑤	$\log \overleftarrow{\nu}(\boldsymbol{\theta}) = -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\big(\tilde{\boldsymbol{A}}(\boldsymbol{\theta}) + \boldsymbol{B}(\boldsymbol{\theta}) + \boldsymbol{S}(\boldsymbol{\theta})\big)\Big) + \tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\,\tilde{\boldsymbol{b}}(\boldsymbol{\theta}) + C$
	$U = \frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\big(\tilde{\boldsymbol{A}} + \boldsymbol{D}\big)\Big) + \frac{P}{2}\log(2\pi) - \frac{1}{2}\overline{\log \boldsymbol{W}}$

4.1 ISP/ISF Shape Modelling With and Without Gradient Observations

In this section, we showcase usage of the developed dID-VSGP node for the task of simultaneously modeling an implicit surface potential (ISP) and an implicit surface field (ISF) as described in the introduction. Briefly, the ISP defines, at all points in the input space, a scalar potential whose level sets describe the shape, while the ISF corresponds to the gradient of this potential over the input space, indicating the surface normals at points in the zero-level set and pointing away from the shape at points beyond the surface.

To begin, we define a pair of linear operators that allow simultaneous modeling of a 2D ISP and ISF as

$$\mathcal{L}_p = \mathcal{I}\,, \quad \mathcal{L}_f = \nabla_{\boldsymbol{x}}\,, \tag{31}$$

where $\mathcal{I}$ is the identity operator, and $\nabla_{\boldsymbol{x}}$ is the gradient operator with respect to the input space. With these applied operators, our noisy observations for the generative process defined in Eq. (15) become

$$\tilde{\boldsymbol{y}}_{p,i} \in \mathbb{R}, \quad \tilde{\boldsymbol{y}}_{p,i} \sim \mathcal{N}(0, 1\mathrm{e}{-3}) \tag{32}$$

$$\tilde{\boldsymbol{y}}_{f,i} \in \mathbb{R}^2, \quad \tilde{\boldsymbol{y}}_{f,i} \sim \mathcal{N}(\boldsymbol{n}(\boldsymbol{x}_i), 1\mathrm{e}{-3}\boldsymbol{I})\,, \tag{33}$$

where a 0-value ISP observation implies surface contact, and $\boldsymbol{n}$-value ISF observation provides the surface-normal direction at the corresponding cartesian location, $\boldsymbol{x}_i$. We define a set of inducing inputs, X_u, as an equally-spaced grid of 225 points spread across the x-y plane, distributed along each axis from -4 to 4, with $\boldsymbol{v} \in \mathbb{R}^{225}$, and we utilize a mean function, $\tilde{\boldsymbol{m}}(\boldsymbol{x}) = 20.0$, for both nodes. For further details on the shape and data-generation process, see Section B.1. For simplicity, we utilize fixed inputs, and process noises. The numerical priors for $\boldsymbol{v}$, $\boldsymbol{W}_p$, and $\boldsymbol{W}_f$ are somewhat arbitrary, but affect how strongly the observations influence the latent ISP and ISF. We can then define the factors that form a joint factorized generative model that we perform inference over as

$$p(y_{p,i} \mid \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_p, \boldsymbol{\theta}) = \tilde{\boldsymbol{\phi}}_p(y_{p,i}, \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_p, \boldsymbol{\theta}) \tag{34a}$$

$$p(\tilde{\boldsymbol{y}}_{f,i} \mid \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_f, \boldsymbol{\theta}) = \tilde{\boldsymbol{\phi}}_f(\tilde{\boldsymbol{y}}_{f,i}, \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_f, \boldsymbol{\theta}) \tag{34b}$$

$$p(\boldsymbol{x}_i) = \delta(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \tag{34c}$$

$$p(\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v} \mid 200 \cdot \boldsymbol{1}, 200\boldsymbol{I}) \tag{34d}$$

$$p(\boldsymbol{W}_p) = \delta(\boldsymbol{W}_p - 1\mathrm{e}3) \tag{34e}$$

$$p(\boldsymbol{W}_f) = \delta(\boldsymbol{W}_f - 1\mathrm{e}3\boldsymbol{I}) \tag{34f}$$

$$p(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - [1.0, 1.6]^T)\,. \tag{34g}$$

Critically, in the above setup, we have listed quantities with subscript p and f, pertaining to ISP and ISF, respectively. We utilize only the items with subscript p for the results presented in the left-hand plot of Fig. 6, and use both in the right-hand plot, to demonstrate the benefit obtained from including surface-normal observations, enabled by this newly developed transformed VSGP node. After convergence of the variational message-passing scheme, the resulting posteriors are used in Eq. (30) to predict the ISP / ISF values over a dense grid over inputs on x-y axes from -4 to 4. Finally, in the left-hand plot of Fig. 6, we can see that the surface contact measurements are satisfied, but without surface-normal information, or more contact measurements, the shape does not reliably conform to the true object shape, while in the right-hand plot, we can see that the predicted zero-level set of the ISP is much more aligned with the true object shape. Converged optimal kernel hyperparameters are $\boldsymbol{\theta} = [-0.173, 1.568]$ and $\boldsymbol{\theta} = [0.989, 1.498]$ for the left and right plots, respectively.

4.2 Physics-Informed Factors as Partial Differential Equations

In this section, we showcase solving a linear stochastic partial differential equation (SPDE) to demonstrate the ability enabled by the newly developed node to include uncertain physics-informed factors in a FGMP-PPL. We define the SPDE as

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x})\,, \tag{35}$$

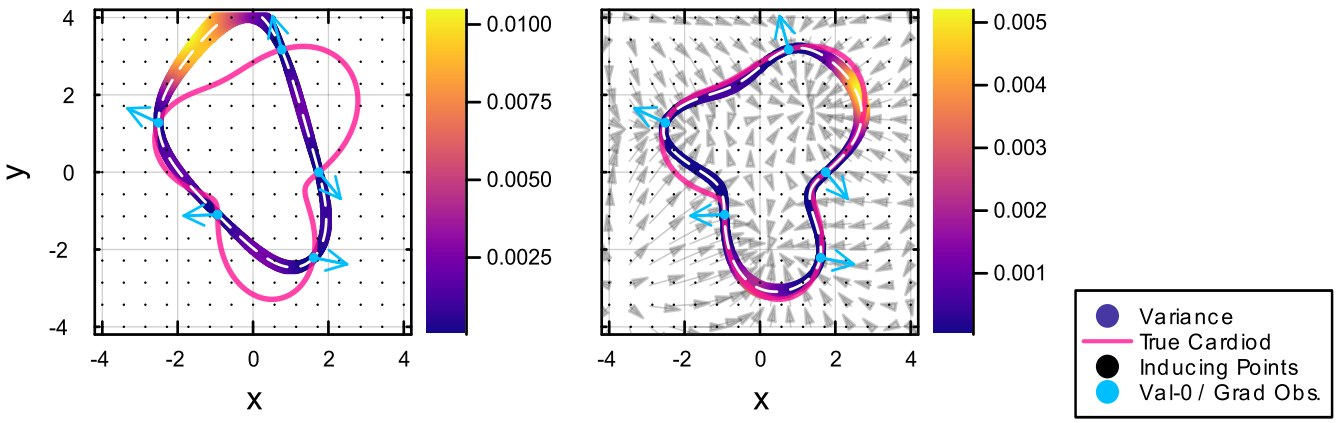

Figure 6: 2D Implicit Surface Regression. The simulation associated with the right plot utilizes inter-domain gradient observations, while the left plot does not. The colored band labeled "Variance" indicates where the zero-level set is plausible. Specifically, the width of the band is associated with the uncertainty about the zero level set; the band is plotted where $|\mu| <= 3\sigma$, where μ and σ^2 are the ISP predicted mean and variance per Eq. (30). The color of the band quantifies the predicted variance, σ^2, directly and represents uncertainty associated with the prediction itself. The white dashed line over this band denotes the zero level set, $\{x : f(x) = 0\}$. The blue markers labeled "Val-0 / Grad Obs" indicate the surface-contact observations, while the blue arrows stemming from these markers indicate the surface-normal observations, but are only used in the right-hand simulation. Finally, the gray arrows in the right-hand plot indicate the negative posterior predictive ISF per Eq. (30), utilizing $\tilde{\phi}_f$ per Eq. (34b).

with

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \cos(\boldsymbol{x}_1)\cos(\boldsymbol{x}_2) \\ -\sin(\boldsymbol{x}_1)\sin(\boldsymbol{x}_2) \end{bmatrix}. \tag{36}$$

We can then define a pair of linear operators, $\mathcal{L}_e$ and $\mathcal{L}_c$ that together encode onto the underlying GP function a partial differential equation and a boundary constraint, respectively, as

$$\mathcal{L}_e = \nabla_{\boldsymbol{x}}, \quad \mathcal{L}_c = \begin{bmatrix} \mathcal{I} \\ \nabla_{\boldsymbol{x}} \end{bmatrix}. \tag{37}$$

Similar to Section 4.1, under these linear operators, our noisy observations per Eq. (15) become

$$\tilde{\boldsymbol{y}}_{e,i} \in \mathbb{R}^2, \quad \tilde{\boldsymbol{y}}_{e,i} \sim \mathcal{N}(\boldsymbol{g}(\boldsymbol{x}_i), 1\text{e}{-4}\boldsymbol{I}) \tag{38}$$

$$\tilde{\boldsymbol{y}}_c \in \mathbb{R}^3, \quad \tilde{\boldsymbol{y}}_c \sim \mathcal{N}([0, \boldsymbol{g}(\boldsymbol{0})^{\mathrm{T}}]^{\mathrm{T}}, 1\text{e}{-6}\boldsymbol{I}), \tag{39}$$

where a PDE constraint is applied at each collocation point, $\boldsymbol{x}_i$, in Eq. (38), and a boundary constraint $f(\boldsymbol{x}_c = 0) = 0$, along with the PDE constraint, is applied in Eq. (39). We define a set of inducing inputs, X_u, as an equally-spaced grid of 400 points spread across the x-y plane, distributed along each axis from -5 to 5, with $\boldsymbol{v} \in \mathbb{R}^{400}$, and we set our input PDE collocation points, $\boldsymbol{x}_i$, to be the same as the inducing inputs. We utilize a mean function, $\tilde{\boldsymbol{m}}(\boldsymbol{x}) = 0.0$ in both nodes, and for simplicity, we utilize fixed inputs, process noise, and kernel hyperparameters. The numerical priors for $\boldsymbol{v}$, $\boldsymbol{W}_p$, and $\boldsymbol{W}_f$ are again somewhat arbitrary, with $p(\boldsymbol{v})$ being uninformative and the process-noises being low. We can then define the factors that form a joint factorized generative model that we perform inference over as

$$p(\tilde{\boldsymbol{y}}_{e,i} \mid \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_e, \boldsymbol{\theta}) = \tilde{\phi}_e(\tilde{\boldsymbol{y}}_{e,i}, \boldsymbol{x}_i, \boldsymbol{v}, \boldsymbol{W}_e, \boldsymbol{\theta}) \tag{40a}$$

$$p(\tilde{\boldsymbol{y}}_c \mid \boldsymbol{x}_c, \boldsymbol{v}, \boldsymbol{W}_c, \boldsymbol{\theta}) = \tilde{\phi}_c(\tilde{\boldsymbol{y}}_c, \boldsymbol{x}_c, \boldsymbol{v}, \boldsymbol{W}_c, \boldsymbol{\theta}) \tag{40b}$$

$$p(\boldsymbol{x}_i) = \delta(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \tag{40c}$$

$$p(\boldsymbol{x}_c) = \delta(\boldsymbol{x}_c - \boldsymbol{0}) \tag{40d}$$

$$p(\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v} \mid \boldsymbol{0}, 1\text{e}{-12}\boldsymbol{I}) \tag{40e}$$

$$p(\boldsymbol{W}_e) = \delta(\boldsymbol{W}_e - 1\text{e}4\boldsymbol{I}) \tag{40f}$$

$$p(\boldsymbol{W}_c) = \delta(\boldsymbol{W}_c - 1\text{e}6\boldsymbol{I}) \tag{40g}$$

$$p(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - [0.5, 0.5]^T). \tag{40h}$$

As we can see in Fig. 7, the predicted scalar field is a nearly-perfect match. We find the final MSE evaluated over a dense 80 by 80 grid of inputs to be 6e−7.

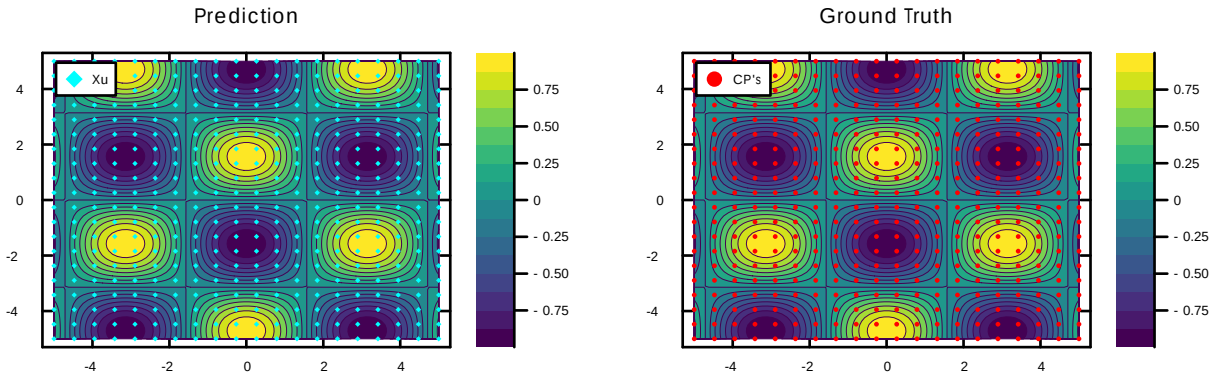

Figure 7: The figure on the left shows the predicted scalar field, $f_\star(x)$, inferred from the PDE: $\nabla_x f(x) = g(x)$ with $g(x) = [\cos(x_1)\cos(x_2), -\sin(x_1)\sin(x_2)]^{\mathrm{T}}$. The inducing points, X_u, are also plotted. In the right figure is plotted the ground truth scalar field, $f(x) = \sin(x_1)\cos(x_2)$. The PDE collocation points (CP's) are also plotted, and were chosen to match the inducing points for simplicity.

5 Discussion

5.1 Scaling of the dID-VSGP Node

The baseline univariate variational sparse Gaussian process (VSGP) model has computational complexity $\mathcal{O}(NM^2)$ (Titsias, 2009), where N is the number of observations and M the number of inducing points. Extensions to multi-output settings that model D outputs as independent Gaussian processes (e.g., via an identity operator of size D) incur a cost of $\mathcal{O}(DNM^2)$ (Nguyen et al., 2025), as each output introduces a separate inducing-point projection despite sharing kernel hyperparameters. In contrast, the proposed dID-VSGP node applies arbitrary linear operators to a shared latent GP, with inducing variables defined solely in the latent space. Although the transformed observations are P-dimensional, they remain jointly governed by a single GP with a shared inducing-variable covariance factorization, and no additional inducing variables are introduced. As a result, the overall computational complexity remains $\mathcal{O}(NM^2)$, rather than $\mathcal{O}(PNM^2)$.

5.2 Inter-Domain Inducing Variables

As mentioned in Section 3.1, the derivations utilize inducing variables defined in the latent function space.

The key advantage of this design choice is that it enables multiple dID-VSGP nodes, constructed with different linear operators, to share a common set of inducing variables within a single graph. If, instead, the linear operator were also applied to the inducing variables in Eq. (13), each inter-domain VSGP node would require its own operator-specific inducing space. This would prevent the nodes from jointly updating a shared latent representation of the underlying process.

The main limitation from this design choice, however, is that it limits expressivity of the inducing variables to capture low-frequency (integral) or higher-order derivative structure (Alvarez et al., 2012; Wilk et al., 2020). Some extensions, most notably the convolutional GP from Wilk et al. (2017), also depend critically on defining a specific domain for the inducing variables.

Therefore, while our results show adequate expressivity for first-order gradient information, some problems will benefit from the enhanced expressivity offered by inter-domain inducing variables. As such, future work could investigate how to incorporate inter-domain inducing variables into the message-passing VSGP architecture.

5.3 Arbitrary Linear Operators

Finally, we specify the caveat that while application of arbitrary deterministic linear operators to a Gaussian process is allowed in general, their implementation must be supported by the chosen kernel function. For example, the Matérn kernel is only finitely differentiable (e.g., a Matérn kernel with smoothness parameter ν is $\lfloor \nu \rfloor$-times mean-square differentiable), and therefore differential operators of order exceeding this smoothness are not well-defined. In contrast, the squared exponential kernel is infinitely differentiable, and thus supports differential operators of arbitrary order.

5.4 Arbitrary Mean-Function

A key advantage of specifying a mean function on the latent Gaussian process is that it is closed under linear operators: if $f \sim GP(m, k)$, then $\mathcal{L}f \sim GP(\mathcal{L}m, \mathcal{L}_1\mathcal{L}_2 k)$, so transformed quantities automatically inherit the mean $\mathcal{L}m$. This is particularly convenient in operator-based models, where a single latent mean induces consistent means for all derived processes (e.g., gradients or other operators), avoiding the need to construct them manually. While this does not enforce hard constraints, it provides a simple and coherent way to align prior structure across the inter-domain processes.

We note that while exploration of learnable, parameterized mean functions for Gaussian processes is described by Fortuin et al. (2020), our derived VSGP nodes do not currently expose to inference the parameters of the latent mean function, which prevents this learning. While this appears feasible, it is left as future work.

6 Conclusion

This paper has described extensions to prior work modeling a univariate variational sparse Gaussian process (VSGP) as a node in a Forney-style factor-graph (Nguyen et al., 2025). Specifically, we extend this work to (a) enable inclusion of an arbitrary mean function in the GP prior and (b) to enable application of arbitrary deterministic linear inter-domain (ID) operators to the underlying function modeled by the GP. These extensions enable a vastly larger set of possible observation types and result in a new dID-VSGP node, where "dID" indicates decoupled inter-domain latent-space inducing variables. We derive the mean-field variational message-passing update rules for the dID-VSGP node and then showcase its use in two characteristic problems, namely implicit-surface shape modeling, and physics-informed free-form probabilistic modeling which allows approximate solutions to stochastic partial differential equations to be included and inferred per noisy observations in a factor graph. The choice to utilize decoupled inducing points impose a trade-off whereby some modeling expressivity is exchanged for inference flexibility. We further discussed that the chosen linear operators should be supported by the chosen kernel, and the convenience obtained by inclusion of the mean-function within the GP prior specification for complex operators.

In conclusion, inter-domain observations, variational message passing, and a sparse Gaussian process come together to form an expressive and scalable function-modeling tool, and this paper describes how to implement it as a node in Forney-style factor graphs.

Acknowledgements

This work was funded by the Eindhoven Artificial Intelligence Systems Institute (EAISI), under their Exploratory Multidisciplinary AI Research Program (EMDAIR), as part of the Embodied AI for Continuous Human-Like Learning Project (EMBODeAI). The authors express their sincere appreciation to the EAISI institute, BIASlab, the Reshape Lab, and the Neuromorphic Engineering Lab for their valuable assistance.

We also thank the two anonymous reviewers for their insightful comments and suggestions, which have improved the quality of this paper.

References

M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for Vector-Valued Functions: a Review, Apr. 2012. URL http://arxiv.org/abs/1106.6251. arXiv:1106.6251 [stat].

D. Bagaev and B. De Vries. Reactive Message Passing for Scalable Bayesian Inference. *Scientific Programming*, 2023:1–26, May 2023. ISSN 1875-919X, 1058-9244. doi: 10.1155/2023/6601690. URL https://www.hindawi.com/journals/sp/2023/6601690/.

D. Bagaev, A. Podusenko, and B. De Vries. RxInfer: A Julia package for reactive real-time Bayesian inference. *JOSS*, 8(84):5161, 2023. ISSN 2475-9066. doi: 10.21105/joss.05161. URL https://joss.theoj.org/papers/10.21105/joss.05161.

J. Dauwels. On Variational Message Passing on Factor Graphs. In *IEEE International Symposium on Information Theory*, pages 2546–2550, Nice, France, June 2007. doi: 10.1109/ISIT.2007.4557602. URL http://ieeexplore.ieee.org/abstract/document/4557602.

S. Dragiev, M. Toussaint, and M. Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *2011 IEEE International Conference on Robotics and Automation*, pages 2845–2850, Shanghai, China, May 2011. IEEE. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980395. URL http://ieeexplore.ieee.org/document/5980395/.

S. Dragiev, M. Toussaint, and M. Gienger. Uncertainty aware grasping and tactile exploration. In *2013 IEEE International Conference on Robotics and Automation*, pages 113–119, Karlsruhe, Germany, May 2013. IEEE. ISBN 978-1-4673-5643-5 978-1-4673-5641-1. doi: 10.1109/ICRA.2013.6630564. URL http://ieeexplore.ieee.org/document/6630564/.

G. Forney. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, 47(2):520–548, Feb. 2001. ISSN 0018-9448. doi: 10.1109/18.910573. URL https://ieeexplore.ieee.org/abstract/document/910573.

V. Fortuin, H. Strathmann, and G. Rätsch. Meta-Learning Mean Functions for Gaussian Processes, Feb. 2020. URL http://arxiv.org/abs/1901.08098. arXiv:1901.08098 [stat].

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration, Sept. 2018. URL https://arxiv.org/abs/1809.11165v6.

H.-A. Loeliger. An introduction to factor graphs. *Signal Processing Magazine, IEEE*, 21(1):28–41, Jan. 2004. doi: 10.1109/MSP.2004.1267047. URL https://ieeexplore.ieee.org/document/1267047.

M. Lázaro-Gredilla and A. Figueiras-Vidal. Inter-domain Gaussian Processes for Sparse Inference using Inducing Features. pages 1087–1095, Dec. 2009.

T. Matsumoto and T. J. Sullivan. Images of Gaussian and other stochastic processes under closed, densely-defined, unbounded linear operators. *Anal. Appl.*, 22(03):619–633, Apr. 2024. ISSN 0219-5305, 1793-6861. doi: 10.1142/S0219530524400025. URL http://arxiv.org/abs/2305.03594. arXiv:2305.03594 [math].

A. G. d. G. Matthews, M. v. d. Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian Process Library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017. ISSN 1533-7928. URL http://jmlr.org/papers/v18/16-537.html.

H. M. H. Nguyen, I. Senoz, and B. De Vries. A Factor Graph Approach to Variational Sparse Gaussian Processes. *IEEE Open Journal of Signal Processing*, 2025.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

K. B. Petersen and M. S. Pedersen. The matrix cookbook. Technical report, 2012. URL http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf.

J. Quiñonero-Candela and C. E. Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1194909.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL http://mitpress.mit.edu/books/chapters/026218253Xchap2.pdf.

I. Senöz, T. van de Laar, D. Bagaev, and B. de Vries. Variational Message Passing and Local Constraint Manipulation in Factor Graphs. *Entropy*, 23(7):

807, July 2021. ISSN 1099-4300. doi: 10.3390/e23070807. URL https://www.mdpi.com/1099-4300/23/7/807.

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. URL https://papers.nips.cc/paper_files/paper/2005/hash/4491777b1aa8b5b32c2e8666dbe1a495-Abstract.html.

E. Solak, R. Murray-smith, W. Leithead, D. Leith, and C. Rasmussen. Derivative Observations in Gaussian Process Models of Dynamic Systems. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/hash/5b8e4fd39d9786228649a8a8bec4e008-Abstract.html.

S. Särkkä. Linear Operators and Stochastic Partial Differential Equations in Gaussian Process Regression. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792, pages 151–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-21737-1 978-3-642-21738-8. doi: 10.1007/978-3-642-21738-8_20. URL http://link.springer.com/10.1007/978-3-642-21738-8_20. Series Title: Lecture Notes in Computer Science.

M. K. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

M. Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1049–1056, Montreal Quebec Canada, June 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553508. URL https://dl.acm.org/doi/10.1145/1553374.1553508.

M. v. d. Wilk, C. E. Rasmussen, and J. Hensman. Convolutional Gaussian Processes, Sept. 2017. URL http://arxiv.org/abs/1709.01894. arXiv:1709.01894 [stat].

M. v. d. Wilk, V. Dutordoir, S. T. John, A. Artemev, V. Adam, and J. Hensman. A Framework for Interdomain and Multioutput Gaussian Processes, Mar. 2020. URL http://arxiv.org/abs/2003.01115. arXiv:2003.01115 [stat].

L. Wu, A. Miller, L. Anderson, G. Pleiss, D. Blei, and J. Cunningham. Hierarchical Inducing Point Gaussian Process for Inter-domain Observations, June 2021. URL http://arxiv.org/abs/2103.00393. arXiv:2103.00393 [cs].

A Message Update Derivations

We consider the update rules for a univariate Gaussian Process (GP) that is transformed by a linear operator, $\mathcal{L}$. A GP $f \sim GP(m, k)$ with mean function $m(\boldsymbol{x})$ and kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ is then transformed by this operator to $\tilde{\boldsymbol{f}} \sim GP(\mathcal{L}m, \mathcal{L}_1\mathcal{L}_2 k)$, where $\mathcal{L}_i$ applies the operator to the i'th function argument Matsumoto and Sullivan (2024).

We consider the observation model,

$$p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{f}}) = \mathcal{N}\left(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{f}}, \boldsymbol{W}^{-1}\right) . \tag{41}$$

As derived in the main text, the process model for $\tilde{\boldsymbol{f}}$ conditioned on the inducing points $\boldsymbol{u}$ has a covariance,

$$\tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\theta}) = \overbrace{\mathcal{L}_1\mathcal{L}_2 k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x})}^{\tilde{\boldsymbol{K}}_{\boldsymbol{x}\boldsymbol{x}'}(\boldsymbol{x},\boldsymbol{x},\boldsymbol{\theta})} - \underbrace{\mathcal{L}_1 k_{\boldsymbol{\theta}}(\boldsymbol{x}, X_{\boldsymbol{u}})}_{\tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{\theta})^{\mathrm{T}}} \underbrace{k_{\boldsymbol{\theta}}(X_{\boldsymbol{u}}, X_{\boldsymbol{u}})^{-1}}_{\boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}(\boldsymbol{\theta})^{-1}} \underbrace{\mathcal{L}_2 k_{\boldsymbol{\theta}}(X_{\boldsymbol{u}}, \boldsymbol{x})}_{\tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{\theta})} , \tag{42}$$

where we introduced shorthand notations for the transformed kernels, and expressed their explicit dependence on the input $\boldsymbol{x}$ and the kernel hyperparameters $\boldsymbol{\theta}$. For the process mean, we define shorthands for the transformed inducing points $\boldsymbol{v} \triangleq \boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}(\boldsymbol{\theta})^{-1}\boldsymbol{u}$, and the vector constructed by applying the mean function to the inducing inputs $\boldsymbol{m}_{\boldsymbol{u}} \triangleq m(X_{\boldsymbol{u}})$. The main text then derives the process mean,

$$\tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}) \triangleq \overbrace{\mathcal{L}m(\boldsymbol{x})}^{\tilde{\boldsymbol{m}}(\boldsymbol{x})} + \tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{v} - \boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}^{-1}(\boldsymbol{\theta})\boldsymbol{m}_{\boldsymbol{u}}) . \tag{43}$$

The process model is then expressed as,

$$p(\tilde{\boldsymbol{f}}|\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}) = \mathcal{N}\left(\tilde{\boldsymbol{f}}|\tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}), \tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\theta})\right) . \tag{44}$$

For notational convenience we will drop the $\boldsymbol{u}$ subscripts from the $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{A}}$ terms. As detailed in the main text, the dID-VSGP node function is then derived as,

$$\begin{aligned}\log \tilde{\phi}(\tilde{\boldsymbol{y}}, \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{W}, \boldsymbol{\theta}) &= \mathbb{E}_{p(\tilde{\boldsymbol{f}}|\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta})}\left[\log p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{f}})\right] \\ &= -\frac{1}{2}\operatorname{tr}\left(\boldsymbol{W}\tilde{\boldsymbol{A}}(\boldsymbol{x}, \boldsymbol{\theta})\right) - \frac{P}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{W}| - \\ &\quad \frac{1}{2}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta}))^{\mathrm{T}}\boldsymbol{W}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{\theta})) ,\end{aligned} \tag{45}$$

with P the dimensionality of $\tilde{\boldsymbol{y}}$.

The variational message updates below are derived under a mean-field factorization of the variational distribution, $q(\cdot) \triangleq q(\tilde{\boldsymbol{y}})q(\boldsymbol{x})q(\boldsymbol{v})q(\boldsymbol{W})q(\boldsymbol{\theta})$, and a point-mass constraint on the kernel hyperparameter belief, $q(\boldsymbol{\theta}) \triangleq \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$. For notational convenience, we write expectations by an overbar, $\overline{x} \triangleq \mathbb{E}_{q(\boldsymbol{x})}[\boldsymbol{x}]$, and omit the q's from the expectation, $\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x}} \triangleq \mathbb{E}_{q(\tilde{\boldsymbol{y}})q(\boldsymbol{x})}$.

A.1 Forward Message for $\tilde{y}$

Throughout the derivations we absorb all terms that are independent of the message argument in a constant

term C. The forward variational message for the output becomes,

$$\begin{aligned}\log \overrightarrow{\nu}(\tilde{\boldsymbol{y}}) &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{v},\boldsymbol{W},\boldsymbol{\theta}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= -\frac{1}{2}\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))\Big] + C \\ &= -\frac{1}{2}\Big(\tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\Big] - \\ &\quad \mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\Big]^{\mathrm{T}}\overline{\boldsymbol{W}}\tilde{\boldsymbol{y}}\Big) + C\,,\end{aligned} \tag{46}$$

where we overloaded notation by omitting the substituted kernel $\hat{\boldsymbol{\theta}}$ from the $\tilde{\boldsymbol{b}}$ (and later the $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{A}}$) terms. We then encounter an expectation,

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\Big] = \underbrace{\overline{\tilde{\boldsymbol{m}}(\boldsymbol{x})}}_{\widetilde{\boldsymbol{m}}} + \underbrace{\overline{\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})}^{\mathrm{T}}}_{\widetilde{\boldsymbol{K}}_{\boldsymbol{ux}}}(\overline{\boldsymbol{v}} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m}_{\boldsymbol{u}})\,. \tag{47}$$

Substituting, the message then becomes,

$$\overrightarrow{\nu}(\tilde{\boldsymbol{y}}) \propto \mathcal{N}\Big(\tilde{\boldsymbol{y}}|\widetilde{\boldsymbol{m}} + \widetilde{\boldsymbol{K}}_{\boldsymbol{ux}}^{\mathrm{T}}(\overline{\boldsymbol{v}} - \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m}_{\boldsymbol{u}}), \overline{\boldsymbol{W}}^{-1}\Big)\,, \tag{48}$$

where we remind ourselves of the notation,

$$\widetilde{\boldsymbol{m}} = \mathbb{E}_{q(\boldsymbol{x})}[\mathcal{L}m(\boldsymbol{x})] \tag{49a}$$
$$\widetilde{\boldsymbol{K}}_{\boldsymbol{ux}} = \mathbb{E}_{q(\boldsymbol{x})}\big[\mathcal{L}_2 k_{\hat{\boldsymbol{\theta}}}(X_{\boldsymbol{u}},\boldsymbol{x})\big] \tag{49b}$$
$$\boldsymbol{K}_{\boldsymbol{uu}} = k_{\hat{\boldsymbol{\theta}}}(X_{\boldsymbol{u}}, X_{\boldsymbol{u}}) \tag{49c}$$
$$\boldsymbol{m}_{\boldsymbol{u}} = m(X_{\boldsymbol{u}})\,. \tag{49d}$$

A.2 Backward Message for x

The backward message for the input becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{x}) &= \mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{v},\boldsymbol{W},\boldsymbol{\theta}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\tilde{\boldsymbol{A}}(\boldsymbol{x},\hat{\boldsymbol{\theta}})\Big) - \\ &\quad \frac{1}{2}\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{v}}\Big[(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))\Big] + C\,.\end{aligned} \tag{50}$$

We rewrite the expectation of the second term (again omitting the substituted kernel $\hat{\boldsymbol{\theta}}$),

$$\begin{aligned}\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{v}}[\cdot] = -\tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\overbrace{\mathbb{E}_{\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\Big]}^{\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}})} - \mathbb{E}_{\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\Big]^{\mathrm{T}}\overline{\boldsymbol{W}}\tilde{\boldsymbol{y}} + \\ \operatorname{tr}\Big(\overline{\boldsymbol{W}}\mathbb{E}_{\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v})^{\mathrm{T}}\Big]\Big) + C\,.\end{aligned} \tag{51}$$

Using the covariance relation $\overline{\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}} = \overline{\boldsymbol{v}}\,\overline{\boldsymbol{v}}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{v}}$, the expectation within the trace becomes,

$$\mathbb{E}_{\boldsymbol{v}}\big[(\cdot)(\cdot)^{\mathrm{T}}\big] = \underbrace{\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}})}_{\tilde{\boldsymbol{b}}(\boldsymbol{x})}\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}})^{\mathrm{T}} + \underbrace{\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{v}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})}_{\boldsymbol{S}(\boldsymbol{x})}\,. \tag{52}$$

Substituting back, the log-message then becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{x}) = -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\Big(\tilde{\boldsymbol{A}}(\boldsymbol{x}) + \boldsymbol{S}(\boldsymbol{x})\Big)\Big) - \\ \frac{1}{2}(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x})) + C\,,\end{aligned} \tag{53}$$

with the shorthand notation,

$$\tilde{\boldsymbol{A}}(\boldsymbol{x}) = \tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x},\hat{\boldsymbol{\theta}}) \tag{54a}$$
$$\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x}) = \mathcal{L}_2 k_{\hat{\boldsymbol{\theta}}}(X_{\boldsymbol{u}},\boldsymbol{x}) \tag{54b}$$
$$\widetilde{\boldsymbol{b}}(\boldsymbol{x}) = \tilde{\boldsymbol{b}}_{\boldsymbol{u}}(\boldsymbol{x},\overline{\boldsymbol{v}},\hat{\boldsymbol{\theta}})\,. \tag{54c}$$

A.3 Backward Message for v

The backward message towards the inducing points becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{v}) &= \mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{W},\boldsymbol{\theta}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= -\frac{1}{2}\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x}}\Big[(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))\Big] + C\,.\end{aligned} \tag{55}$$

Substituting $\tilde{\boldsymbol{b}}$ and isolating the terms that depend on $\boldsymbol{v}$,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{v}) = -\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\overbrace{\mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\overline{\boldsymbol{W}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\Big]}^{\boldsymbol{U}}\boldsymbol{v} + \\ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{\eta} + C\,,\end{aligned} \tag{56}$$

with,

$$\boldsymbol{\eta} = \mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\overline{\boldsymbol{W}}\Big(\widetilde{\boldsymbol{y}} - \tilde{\boldsymbol{m}}(\boldsymbol{x}) + \tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\overline{\boldsymbol{m}}_{\boldsymbol{u}}\Big)\Big]\,. \tag{57}$$

This leads to a canonically parameterized Gaussian message,

$$\overleftarrow{\nu}(\boldsymbol{v}) \propto \mathcal{N}_{\xi}(\boldsymbol{v}|\boldsymbol{\eta},\boldsymbol{U})\,, \tag{58}$$

with weighted mean $\boldsymbol{\eta}$ and precision $\boldsymbol{U}$.

A.4 Backward Message for W

The backward message towards the precision becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{W}) &= \mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= -\frac{1}{2}\operatorname{tr}\Big(\boldsymbol{W}\overbrace{\mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{A}}(\boldsymbol{x},\hat{\boldsymbol{\theta}})\Big]}^{\tilde{\boldsymbol{A}}}\Big) + \frac{1}{2}\log|\boldsymbol{W}| \\ &\quad - \frac{1}{2}\operatorname{tr}\Big(\boldsymbol{W}\underbrace{\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v}}\Big[(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\hat{\boldsymbol{\theta}}))^{\mathrm{T}}\Big]}_{\boldsymbol{D}}\Big) \\ &\quad + C\,.\end{aligned} \tag{59}$$

Expanding the expected quadratic form and using the definition of covariance,

$$\begin{aligned}\boldsymbol{D} = \boldsymbol{\Sigma}_{\tilde{\boldsymbol{y}}} + \mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{v}}\tilde{\boldsymbol{K}}_{\boldsymbol{ux}}(\boldsymbol{x})\Big] + \\ \mathbb{E}_{\boldsymbol{x}}\Big[(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x}))(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{b}}(\boldsymbol{x}))^{\mathrm{T}}\Big]\,.\end{aligned} \tag{60}$$

The message then becomes,

$$\overleftarrow{\nu}(\boldsymbol{W}) \propto \mathcal{W}\Big(\boldsymbol{W}|(\widetilde{\boldsymbol{A}} + \boldsymbol{D})^{-1}, P + 2\Big)\,. \tag{61}$$

A.5 Backward Message for $\boldsymbol{\theta}$

The backward message towards the kernel hyperparameter becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{\theta}) &= \mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v},\boldsymbol{W}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\overbrace{\mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{A}}(\boldsymbol{x},\boldsymbol{\theta})\Big]}^{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})}\Big) - \\ &\quad \frac{1}{2}\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v}}\Big[(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta}))^{\mathrm{T}}\overline{\boldsymbol{W}}(\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta}))\Big] + C\,.\end{aligned} \tag{62}$$

We rewrite the expectation of the second term as a function of $\boldsymbol{\theta}$,

$$\begin{aligned}\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v}}[\cdot] &= -\tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\overbrace{\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta})\Big]}^{\tilde{\boldsymbol{b}}(\boldsymbol{\theta})} - \\ &\quad \mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta})\Big]^{\mathrm{T}}\overline{\boldsymbol{W}}\tilde{\boldsymbol{y}} + \\ &\quad \operatorname{tr}\Big(\overline{\boldsymbol{W}}\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta})\tilde{\boldsymbol{b}}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{\theta})^{\mathrm{T}}\Big]\Big) + C\,.\end{aligned} \tag{63}$$

The expectation within the trace becomes,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x},\boldsymbol{v}}\big[(\cdot)(\cdot)^{\mathrm{T}}\big] &= \overbrace{\mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}},\boldsymbol{\theta})\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}},\boldsymbol{\theta})^{\mathrm{T}}\Big]}^{\boldsymbol{B}(\boldsymbol{\theta})} + \\ &\quad \underbrace{\mathbb{E}_{\boldsymbol{x}}\Big[\tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{v}}\tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{\theta})\Big]}_{\boldsymbol{S}(\boldsymbol{\theta})}\,.\end{aligned} \tag{64}$$

The message then becomes,

$$\begin{aligned}\log \overleftarrow{\nu}(\boldsymbol{\theta}) &= -\frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\Big(\widetilde{\boldsymbol{A}}(\boldsymbol{\theta})+\boldsymbol{B}(\boldsymbol{\theta})+\boldsymbol{S}(\boldsymbol{\theta})\Big)\Big) + \\ &\quad \tilde{\boldsymbol{y}}^{\mathrm{T}}\overline{\boldsymbol{W}}\,\widetilde{\boldsymbol{b}}(\boldsymbol{\theta}) + C\,,\end{aligned} \tag{65}$$

with the shorthands,

$$\widetilde{\boldsymbol{A}}(\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{x})}\Big[\tilde{\boldsymbol{A}}_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{\theta})\Big] \tag{66a}$$

$$\widetilde{\boldsymbol{b}}(\boldsymbol{\theta}) = \mathbb{E}_{q(\boldsymbol{x})}\Big[\tilde{\boldsymbol{b}}(\boldsymbol{x},\overline{\boldsymbol{v}},\boldsymbol{\theta})\Big] \tag{66b}$$

$$\tilde{\boldsymbol{K}}_{\boldsymbol{u}\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{\theta}) = \mathcal{L}_2 k_{\boldsymbol{\theta}}(X_{\boldsymbol{u}},\boldsymbol{x})\,. \tag{66c}$$

A.6 Average Energy

Using the results of the backward message for the precision, the average energy follows as,

$$\begin{aligned}U &= -\mathbb{E}_{\tilde{\boldsymbol{y}},\boldsymbol{x},\boldsymbol{v},\boldsymbol{W},\boldsymbol{\theta}}\Big[\log \tilde{\phi}(\cdot)\Big] \\ &= \frac{1}{2}\operatorname{tr}\Big(\overline{\boldsymbol{W}}\Big(\widetilde{\boldsymbol{A}}+\boldsymbol{D}\Big)\Big) + \frac{P}{2}\log(2\pi) - \frac{1}{2}\overline{\log \boldsymbol{W}}\,.\end{aligned} \tag{67}$$

B Simulation Setup Details

B.1 ISP / ISF Simulation Data Generation

We generate synthetic observations from a parametric closed curve in polar coordinates. Let $\theta \in [0, 2\pi)$ denote the latent angular parameter. The radial profile is defined as a harmonic deformation,

$$\begin{aligned}r(\theta) &= R - a_2\cos(2\theta) \\ &\quad + a_1\sin(\theta+\phi_1) + a_3\sin(3\theta+\phi_3),\end{aligned} \tag{68}$$

where $(R, a_2, a_1, \phi_1, a_3, \phi_3)$ are fixed shape parameters. In our experiment we use the parameterization

$$\begin{aligned}&(R, a_2, a_1, \phi_1, a_3, \phi_3) \\ &\qquad = (2.5,\ 0.5,\ 0.45,\ 0.5,\ 0.73,\ -0.7)\,.\end{aligned} \tag{69}$$

The cartesian coordinates for the surface of the radial profile, as a function of angular position are

$$\mathbf{x}(\theta) = \begin{bmatrix}x(\theta)\\ y(\theta)\end{bmatrix} = r(\theta)\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix}. \tag{70}$$

The surface normals for the radial profile are obtained from the tangent vector,

$$\frac{d\mathbf{x}}{d\theta} = \begin{bmatrix}\dot{x}(\theta)\\ \dot{y}(\theta)\end{bmatrix}, \tag{71}$$

with

$$\dot{x}(\theta) = \dot{r}(\theta)\cos\theta - r(\theta)\sin\theta, \tag{72}$$

$$\dot{y}(\theta) = \dot{r}(\theta)\sin\theta + r(\theta)\cos\theta, \tag{73}$$

and

$$\dot{r}(\theta) = 2a_2\sin(2\theta) + a_1\cos(\theta+\phi_1) + 3a_3\cos(3\theta+\phi_3). \tag{74}$$

The unit surface normals along the radial profile are then given by,

$$\mathbf{n}(\theta) = \frac{1}{\left\|\frac{d\mathbf{x}}{d\theta}\right\|}\begin{bmatrix}\dot{y}(\theta)\\ -\dot{x}(\theta)\end{bmatrix}. \tag{75}$$

We create a synthetic dataset consisting of five noise-free surface-contact and surface-normal observations by defining a set of equally spaced angles $\{\theta_i\}_{i=1}^{5}$ over the interval $[0,\ 1.7\pi]$, i.e.

$$\theta_i = \frac{i-1}{4}(1.7\pi), \quad i = 1,\ldots,5, \tag{76a}$$

$$\mathbf{x}_i = \mathbf{x}(\theta_i), \tag{76b}$$

$$\mathbf{y}_i = \begin{bmatrix}0\\ \mathbf{n}(\theta_i)\end{bmatrix} \in \mathbb{R}^3, \tag{76c}$$

where the scalar ISP observation is fixed at zero to indicate surface contact, and the remaining components correspond to the associated unit surface normal.

We define a set of inducing points as an equally-spaced grid of 15 points spread across the x-y plane, distributed along each axis from -4 to 4.