Journal of Machine Learning Research 27 (2026) 1-71 Submitted 4/25; Revised 1/26; Published 2/26

# LazyDINO: Fast, Scalable, and Efficiently Amortized Bayesian Inversion via Structure-Exploiting and Surrogate-Driven Measure Transport

**Lianghao Cao**[a*] LIANGHAO@CALTECH.EDU
**Joshua Chen**[b*] JOSHUA.CHEN@COLOSTATE.EDU
**Michael Brennan**[c] MCBRENN@MIT.EDU
**Thomas O'Leary-Roseberry**[d] OLEARY-ROSEBERRY.1@OSU.EDU
**Youssef Marzouk**[c] YMARZ@MIT.EDU
**Omar Ghattas**[e] OMAR@ODEN.UTEXAS.EDU
[a] *California Institute of Technology, Pasadena, CA 91125, USA.*
[b] *Colorado State University, Fort Collins, CO 80523, USA*
[c] *Massachusetts Institute of Technology, Cambridge, MA 02139, USA.*
[d] *The Ohio State University, Columbus, OH, 43210, USA.*
[e] *The University of Texas at Austin, Austin, TX 78712, USA.*

**Editor:** Stephan Mandt

## Abstract

We present LazyDINO, a transport map variational inference method for fast, scalable, and efficiently amortized solutions of high-dimensional nonlinear Bayesian inverse problems with expensive parameter-to-observable (PtO) maps. Our method consists of an offline phase, in which we construct a derivative-informed neural surrogate of the PtO map using joint samples of the PtO map and its Jacobian as training data. During the online phase, when given observational data, we rapidly approximate the posterior using surrogate-driven training of a lazy map, i.e., a structure-exploiting transport map with low-dimensional nonlinearity. Our surrogate construction is optimized for amortized Bayesian inversion using lazy map variational inference. We show that (i) the derivative-based reduced basis architecture minimizes an upper bound on the expected error in surrogate posterior approximation, and (ii) the derivative-informed surrogate training minimizes the expected error due to surrogate-driven variational inference. Our numerical results demonstrate that LazyDINO is highly efficient in cost amortization for Bayesian inversion. We observe a reduction of one to two orders of magnitude in offline cost for accurate online posterior approximation, compared to amortized simulation-based inference via conditional transport and to conventional surrogate-driven transport. In particular, LazyDINO consistently outperforms Laplace approximation using fewer than 1000 offline PtO map evaluations, while competing methods struggle and sometimes fail at 16,000 evaluations.

**Keywords:** Bayesian inverse problems, variational inference, measure transport, surrogate modeling, dimension reduction, operator learning

[*] Corresponding authors. These authors contributed equally as joint first authors to this work.

## 1. Introduction

We investigate the solution of nonlinear *Bayesian inverse problems* (BIPs), i.e., the inference of the uncertain parameters in computational models from noisy, indirect observations. Let $m \in \mathscr{M}$ denote the parameter and assume the observational data $\boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ is given by:

$$\boldsymbol{y} = \boldsymbol{\mathcal{G}}(m) + \boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}(0, \Gamma_n),$$

where $\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathbb{R}^{d_{\boldsymbol{y}}}$ is the parameter-to-observable (PtO) map and $\boldsymbol{n} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ is an unknown noise vector. Given a parameter prior distribution $\mu$, we seek to characterize the posterior distribution $\mu^{\boldsymbol{y}}$ defined via Bayes' rule

$$\underbrace{\mathrm{d}\mu^{\boldsymbol{y}}(m)}_{\text{Posterior}} \propto \underbrace{\exp\left(-\frac{1}{2}\left\|\Gamma_n^{-1/2}\left(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}\right)\right\|^2\right)}_{\text{Likelihood}} \underbrace{\mathrm{d}\mu(m)}_{\text{Prior}}.$$

We are particularly interested in continuum models of physical systems, e.g., parametric partial differential equations (PDEs), where the parameter $m \in \mathscr{M}$ can be arbitrarily high-dimensional, such as spatially or temporally varying functions, and the PtO map $\boldsymbol{\mathcal{G}}$ is defined through the solution of the governing equations (Stuart, 2010; Ghattas and Willcox, 2021). This type of BIP is challenging because likelihood evaluations are computationally intensive, and characterizing high-dimensional probability distributions is difficult due to the curse of dimensionality. These challenges severely limit our ability to obtain fast, accurate solutions to BIPs across a range of observational data, which is required, e.g., for real-time uncertainty quantification of predictive digital twins (Kapteyn et al., 2021) and for Bayesian optimal experimental design (Huan et al., 2024). These settings demand methods with *amortized computational cost*—that is, the expensive computation is performed offline, i.e., before acquiring the data, and posterior characterization ($\boldsymbol{y} \mapsto \mu^{\boldsymbol{y}}$) incurs a comparatively negligible cost once the data is available. In this work, we integrate recent advances in dimension reduction, neural operator learning, and measure transport to derive a fast, scalable, and efficiently amortized method for BIPs that is well-suited to modern computing frameworks.

### 1.1 Variational Inference Using Lazy Maps

We consider using transport map variational inference (TMVI, Marzouk et al. 2016) to approximate the posterior $\mu^{\boldsymbol{y}}$. This method seeks to construct a parameterized transport map $\mathcal{T}_{\boldsymbol{\theta}} : \mathscr{M} \to \mathscr{M}$ that maps a reference distribution, which we take to be the prior $\mu$, to the target Bayesian posterior $\mu^{\boldsymbol{y}}$. The map parameters can be found by minimizing the reverse KL divergence (rKL):

$$\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}(\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu \| \mu^{\boldsymbol{y}}), \tag{1}$$

where $(\cdot)_\sharp$ denotes the pushforward of a probability distribution. Once the transport map is constructed, it enables fast, on-demand approximate posterior sampling via map evaluations at reference samples. However, it can be difficult to represent expressive transport maps in high dimensions. For example, triangular maps (Baptista et al., 2024b) on $\mathbb{R}^n$ must describe $n$-variate functions and thus suffer from the curse of dimensionality. Kernel-based methods (Liu and Wang, 2016; Detommaso et al., 2018; Chen et al., 2019) require a large number of samples to be expressive in high dimensions. Flow-based methods (Rezende and Mohamed, 2015; Papamakarios et al., 2021) often increase expressiveness by adding layers, which is typically performed ad hoc and requires tuning.

Lazy maps (Brennan et al., 2020) are a class of transport maps that alleviate the curse of dimensionality by restricting the nonlinearity of the map to a relatively low-dimensional parameter latent space. Let $\mathcal{E}_r : \mathscr{M} \to \mathbb{R}^{d_r}$ be a linear encoder where $\mathbb{R}^{d_r}$ is the parameter latent space and $d_r \ll \dim(\mathscr{M})$. A lazy map has the following form:

$$\mathcal{T}_{\boldsymbol{\theta}} \coloneqq \mathcal{D}_r \circ \mathbf{T}_{\boldsymbol{\theta}} \circ \mathcal{E}_r + (\mathrm{Id}_{\mathscr{M}} - \mathcal{D}_r \circ \mathcal{E}_r), \tag{2}$$

where $\mathrm{Id}_{\mathscr{M}}$ denotes the identity map, $\mathcal{D}_r : \mathbb{R}^{d_r} \to \mathscr{M}$ is a linear decoder, and $\mathbf{T}_{\boldsymbol{\theta}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_r}$ is a latent space transport map parameterized by weights $\boldsymbol{\theta}$. The lazy map approximates the posterior in the latent space using TMVI, while the complementary space is filled by the prior. When the prior is Gaussian, the rKL minimization problem becomes

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{m\sim\mu}\left[\frac{1}{2}\left\|\Gamma_n^{-1/2}\left(\left(\boldsymbol{\mathcal{G}} \circ \mathcal{T}_{\boldsymbol{\theta}}\right)(m) - \boldsymbol{y}\right)\right\|^2 + \frac{1}{2}\left\|\mathbf{T}_{\boldsymbol{\theta}}(\mathcal{E}_r m)\right\|^2 - \log\left|\det \nabla \mathbf{T}_{\boldsymbol{\theta}}(\mathcal{E}_r m)\right|\right]. \tag{3}$$

A crucial step in constructing a lazy map is finding a parameter latent space that captures the discrepancy between the prior and posterior. Notably, the original lazy map uses the likelihood-informed subspace (Cui et al., 2014), which identifies the latent space based on likelihood sensitivity. By exploiting the structure of the BIPs, TMVI using lazy maps typically achieves high-quality posterior approximation more efficiently than TMVI without parameter reduction or with alternative reduction techniques.

Another fundamental challenge of TMVI, including when a lazy map is used, lies in the high cost of training the transport map, i.e., solving the stochastic and model-constrained rKL minimization problem in (3). Numerous evaluations of the PtO map $\boldsymbol{\mathcal{G}}$ and the actions of its Jacobian $D\boldsymbol{\mathcal{G}}$ are required within each optimization iteration. These evaluations involve repeated solutions of the governing equations of the computational models and their forward or adjoint sensitivities, which can be prohibitively expensive when these equations are, e.g., nonlinear PDEs. This cost barrier becomes further exacerbated when multiple posteriors need to be approximated for different instances of observational data.

### 1.2 Surrogate-Driven Lazy Map Variational Inference

Our work aims to remove the computational bottleneck of model solutions for lazy map variational inference. This is done by constructing a fast-to-evaluate ridge function (Pinkus, 2015) surrogate of the PtO map $\boldsymbol{\mathcal{G}}$ using a neural network latent representation $\mathbf{g}_{\boldsymbol{w}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$:

$$\boldsymbol{\mathcal{G}}(m) \approx \boldsymbol{V}\mathbf{g}_{\boldsymbol{w}}(\mathcal{E}_r m),$$

where $\boldsymbol{w}$ denotes the weights of the neural network and $\boldsymbol{V}$ is a (reduced) basis for the data space. This architecture is known as the reduced-basis or low-rank neural operators (Kovachki et al., 2023). Once the surrogate is constructed, we perform TMVI in the parameter latent space using the surrogate-driven rKL objective for a given observational data $\boldsymbol{y}$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\mathrm{Id}_{\mathbb{R}^{d_r}})}\left[\frac{1}{2}\left\|(\mathbf{g}_{\boldsymbol{w}} \circ \mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) - \boldsymbol{V}^*\boldsymbol{y}\right\|^2 + \frac{1}{2}\left\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right\|^2 - \log\left|\det \nabla \mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right|\right]. \tag{4}$$

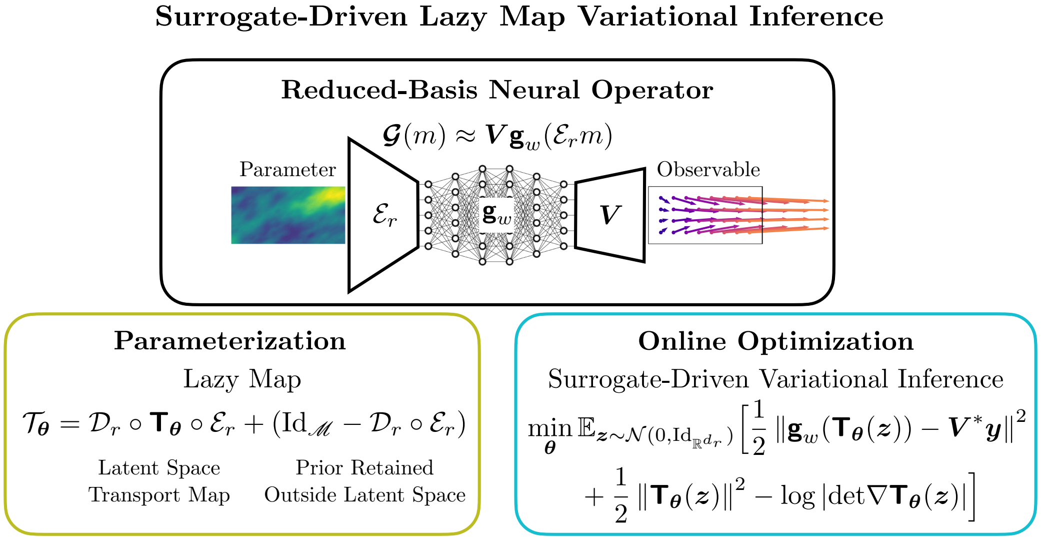
Surrogate-Driven Lazy Map Variational Inference
Reduced-Basis Neural Operator
$\mathcal{G}(m) \approx \boldsymbol{V}\mathbf{g}_w(\mathcal{E}_r m)$
Parameter
$\mathcal{E}_r$
$\mathbf{g}_w$
$\boldsymbol{V}$
Observable
Parameterization
Lazy Map
$\mathcal{T}_{\boldsymbol{\theta}} = \mathcal{D}_r \circ \mathbf{T}_{\boldsymbol{\theta}} \circ \mathcal{E}_r + (\mathrm{Id}_{\mathscr{M}} - \mathcal{D}_r \circ \mathcal{E}_r)$
Latent Space Transport Map
Prior Retained Outside Latent Space
Online Optimization
Surrogate-Driven Variational Inference
$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\mathrm{Id}_{\mathbb{R}^{d_r}})} \Big[\frac{1}{2}\left\|\mathbf{g}_w(\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})) - \boldsymbol{V}^*\boldsymbol{y}\right\|^2 + \frac{1}{2}\left\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right\|^2 - \log\left|\det\nabla\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right|\Big]$

Figure 1: Our amortized Bayesian Inversion method combines lazy map variational inference with a reduced-basis neural operator surrogate. The lazy map ensures scalability by restricting transport to a relatively low-dimensional latent space, while the neural network $\mathbf{g}_w$ replaces the high-fidelity but expensive PtO map $\mathcal{G}$ and rapidly solves the stochastic optimization for variational inference.

The forward and Jacobian evaluation costs of the neural network $\mathbf{g}_{\boldsymbol{w}}$ are significantly lower than those of the model and sensitivity solutions. This optimization problem is, therefore, far cheaper to solve compared to (3).

In terms of methodology, this work addresses two key questions regarding the surrogate-driven approach to lazy map variational inference in the context of amortized Bayesian inversion, where the surrogate is constructed without knowledge of the data $\boldsymbol{y}$.

(Q1) *How do we choose the shared parameter latent space of the surrogate and lazy map?*

We generalize the data-dependent, likelihood-informed parameter latent space of the lazy map to the setting of amortized Bayesian inversion. In particular, following Cui and Zahm (2021), samples of the PtO map Jacobian $D\mathcal{G}$ are used to identify a data-independent latent space that captures the expected prior-to-posterior update over a range of observational data $\boldsymbol{y}$. Notably, this latent space is also used in the derivative-informed projected neural network (`DIPNet`, O'Leary-Roseberry et al. 2022b), a reduced-basis neural operator chosen as our surrogate architecture.

(Q2) *How do we formulate surrogate training to improve the downstream lazy map variational inference?*

We train the surrogate using the derivative-informed learning method following O'Leary-Roseberry et al. (2024); Cao et al. (2025), where the surrogate emulates the latent representation of both the PtO map $\mathcal{G}$ and its Jacobian $D\mathcal{G}$:

$$\min_{\boldsymbol{w}} \mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{V}^*\mathcal{G}(m) - \mathbf{g}_{\boldsymbol{w}}(\mathcal{E}_r m)\right\|^2 + \left\|\boldsymbol{V}^* D\mathcal{G}(m)\mathcal{D}_r - \nabla\mathbf{g}_{\boldsymbol{w}}(\mathcal{E}_r m)\right\|_F^2\right].$$

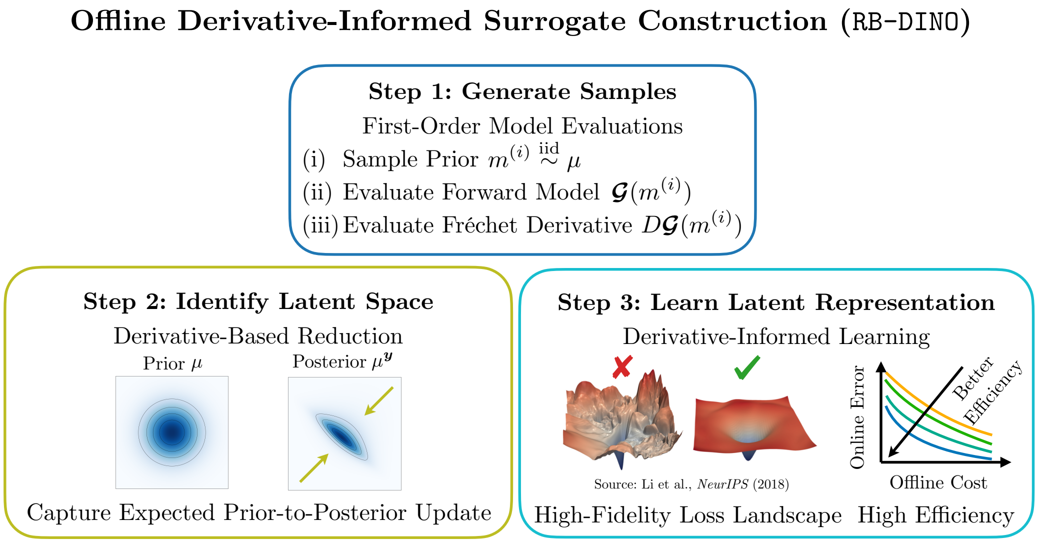
Offline Derivative-Informed Surrogate Construction (RB-DINO)
Step 1: Generate Samples
First-Order Model Evaluations
(i) Sample Prior $m^{(i)} \overset{\text{iid}}{\sim} \mu$
(ii) Evaluate Forward Model $\mathcal{G}(m^{(i)})$
(iii) Evaluate Fréchet Derivative $D\mathcal{G}(m^{(i)})$
Step 2: Identify Latent Space
Derivative-Based Reduction
Prior $\mu$
Posterior $\mu^{\boldsymbol{y}}$
Capture Expected Prior-to-Posterior Update
Step 3: Learn Latent Representation
Derivative-Informed Learning
✗
✓
Source: Li et al., *NeurIPS* (2018)
High-Fidelity Loss Landscape
Online Error
Offline Cost
Better Efficiency
High Efficiency

Figure 2: Our RB-DINO surrogate exploits first-order model evaluations. First, derivative-based reduction identifies a latent space that captures the expected prior-to-posterior update. Second, derivative-based learning trains the neural network latent representation to accurately approximate the variational inference loss landscape at low model evaluation costs. The loss landscape visualization is adapted from Li et al. (2018).

> Compared to conventional learning, derivative-informed learning additionally controls the errors in surrogate Jacobian predictions, which beninifts the downstream amortized Bayesian inversion. First, it enhances downstream surrogate-driven variational inference by minimizing distortion in the loss landscape due to surrogate approximation across a range of observations $\boldsymbol{y}$. Second, it improves surrogate training efficiency, leading to higher amortized Bayesian inversion efficiency, i.e., low online posterior approximation error at low offline model evaluation costs.

We refer to this surrogate construction, i.e., derivative-informed learning of `DIPNet`, as `RB-DINO` herein. We provide analysis that supports the use of `RB-DINO` for surrogate-driven lazy map variational inference in the context of amortized Bayesian inversion.

### 1.3 `LazyDINO` and Our Contributions

Combining lazy maps and `RB-DINO` surrogate construction yields a competitive method for amortized solutions to high-dimensional model-constrained BIPs, called `LazyDINO`. The method comprises an offline phase and an online phase:

*Offline Phase* Samples of the PtO map and its Jacobian are generated by solving the governing equations of the computational model and its forward or adjoint sensitivity. These samples are then used to construct a `RB-DINO` surrogate of the PtO map.

*Online Phase* When observational data are obtained, a latent space transport map is trained to rapidly approximate the posterior, where the training is driven by the surrogate instead of the PtO map. The transport map can then be used to produce approximate posterior samples. This process can be repeated for different observational data, effectively amortizing the construction cost of the `RB-DINO` surrogate.

Extensive numerical studies are conducted that compare `LazyDINO` against a range of baseline TMVI methods, including the Laplace approximation (LA), amortized simulation-based inference (A-SBI) via conditional transport, model-driven lazy maps, and lazy maps combined with conventional surrogate construction (`LazyNO`); see Section 1.4.4 and Table 2. These methods are tested on two infinite-dimensional PDE-constrained BIPs, each with multiple instances of observation data. Thorough evaluations of each method are reported, including posterior approximation tests using moment discrepancies, probability density-based metrics, and posterior visualizations, as well as timing comparisons. The implementations of `LazyDINO` and the numerical examples can be found via the link:

`https://github.com/dinoSciML/dinox`

We summarize the main contributions of `LazyDINO` below.

(C1) We show that the offline `RB-DINO` surrogate construction, i.e., derivative-informed learning of `DIPNet`, is optimized for amortized Bayesian inversion using lazy maps.

*Surrogate Architecture* Extending the analysis by Cui and Zahm (2021) and Cao et al. (2025), we derive an upper bound on the expected posterior approximation error when a ridge-function surrogate replaces the PtO map in the likelihood. We show that minimizing this bound yields `DIPNet` (O'Leary-Roseberry et al., 2022b), in which the parameter encoder is computed from samples of the PtO map Jacobian.

*Derivative-Informed Learning* We show that the expected gradient error (Theorem 5) and the expected optimality gap (Corollary 6) in surrogate-driven lazy map variational inference can be controlled by a weighted Sobolev norm of the surrogate approximation error, which is also the objective function of derivative-informed learning (O'Leary-Roseberry et al., 2024; Cao et al., 2025). In other words, derivative-informed learning minimizes the expected error in surrogate-driven lazy map variational inference.

(C2) We demonstrate that `LazyDINO` enables fast, scalable, and efficiently amortized solutions of high-dimensional Bayesian inverse problems.

*Scalability* The training costs of the surrogate and transport map in `LazyDINO` are independent of the parameter dimension as they share a single parameter latent space. To ensure scalability, we extend the lazy map formulation of Brennan et al. (2020) to infinite-dimensional parameter spaces and derive the corresponding latent space variational inference objective (Proposition 2).

*Fast Online Inference* Using a surrogate rKL objective, `LazyDINO` circumvents the bottleneck of model solutions and fully exploits GPU-based acceleration to rapidly approximate posteriors. Moreover, the optimize-then-sample approach of `LazyDINO` leads to faster online sampling than the typical inversion-to-sample approach of A-SBI.

*Superior Efficiency in Cost Amortization* `LazyDINO` achieves accurate online posterior approximation at low offline costs. Our numerical results show that `LazyDINO` reduces offline costs by one to two orders of magnitude to achieve the same online accuracy compared to `LazyNO` and A-SBI. Moreover, `LazyDINO` consistently outperforms LA at sparse offline model evaluations (<1,000). In contrast, `LazyNO` and A-SBI struggled to outperform LA and, in some cases, failed to do so with 16,000 offline model evaluations.

### 1.4 Related Works

In the following subsections, we discuss related work in dimension reduction, surrogate modeling, and variational inference for BIPs.

#### 1.4.1 The Laplace approximation

The Laplace approximation (LA) constructs a Gaussian approximation of the posterior, enabling efficient sampling and density evaluation. This makes the LA a sensible baseline for settings that require fast approximate posterior sampling. The LA requires maximum a posteriori (MAP) point estimation and covariance estimation using the Hessian of the negative log-posterior at the MAP point. The Hessian often has a low effective rank, allowing for efficient implementations in practice (Bui-Thanh et al., 2012; Isaac et al., 2015; Petra et al., 2014). Details on LA are included in Appendix F.

#### 1.4.2 Dimension reduction for Bayesian inverse problems

A common likelihood-independent dimension-reduction technique is the Karhunen–Loève expansion, which represents the parameter as a sum of a small number of prior-covariance eigenfunctions; see, e.g., Marzouk and Najm (2009). Likelihood-based dimension reduction techniques use the gradient of the log-likelihood to identify a low-dimensional parameter subspace that captures the update from prior to posterior; the posterior is then replaced with the conditional prior in the complementary subspace. These methods generally provide more targeted dimension reduction and greater efficiency (Cui et al., 2014; Chen and Ghattas, 2019; Zahm et al., 2020, 2022; Bigoni et al., 2022) than methods based exclusively on the prior. They are naturally linked to the parameter sensitivity of the likelihood function and, thus, the Fisher information, averaged over the prior or posterior. Zahm et al. (2022) show that forward Kullback–Leibler divergence (fKL) from the approximate posterior to the true posterior has a derivative-based upper bound, and that the subspace minimizing this bound can be found through the solution of an eigenproblem. Cui and Zahm (2021) establish a similar result in expectation over the data, where the operator whose leading eigendirections yield the optimal subspace is computed by averaging the Fisher information over the prior. These derivative-based dimension reductions are shown to be effective in many Bayesian inverse problems; see, e.g., Flath et al. (2011); Bui-Thanh and Ghattas (2012b,a, 2013); Isaac et al. (2015); Chen et al. (2017); Chen and Ghattas (2019).

#### 1.4.3 Surrogate models for Bayesian inverse problems

Substantial work has been done on using surrogate models to accelerate solutions of BIPs, such as polynomial approximation (Marzouk and Najm, 2009; Marzouk et al., 2007; Marzouk and Xiu, 2009; Farcas et al., 2020) and model-order reduction (Galbally et al., 2010; Lieberman et al., 2010; Cui et al., 2015). Surrogate models are often used within multifidelity posterior sampling algorithms (Peherstorfer et al., 2018; Lykkegaard et al., 2023; Cao et al., 2023).

This work focuses on neural network surrogates. Since our algorithmic framework can be applied to infinite-dimensional BIPs, we note the connection to neural operators (Kovachki et al., 2023) that map between function spaces, with architectures and training that are agnostic to the discretization of these spaces. Notable architectures include reduced-basis neural networks using linear (Hesthaven and Ubbiali, 2018; Bhattacharya et al., 2021; O'Leary-Roseberry et al., 2022b,a) or nonlinear (Lu et al., 2021) dimension reduction, and neural network integral kernels such as the Fourier neural operator and its variants (Li et al., 2021; Cao et al., 2024; Lanthaler et al., 2024). Notable training formulations include physics-informed learning (Li et al., 2024; Wang et al., 2021), with loss functions based on the residual of the implicit equation (e.g., PDE residuals).

Our surrogate architecture is based on the derivative-based dimension reduction, i.e., `DIPNet` of O'Leary-Roseberry et al. (2022b). Additionally, we use the derivative-informed learning method for surrogate training, i.e., `DINO` of O'Leary-Roseberry et al. (2022b). This method has been successfully applied to surrogate-driven solutions of PDE-constrained optimization under uncertainties (Luo et al., 2023), Bayesian optimal experimental design (Go and Chen, 2024, 2025), and BIPs (Cao et al., 2025). This method has also been used by Qiu et al. (2024) to train the deep operator network architecture by Lu et al. (2021).

### 1.4.4 Transport map parameterizations and amortized inference

The `LazyDINO` algorithm performs TMVI (El Moselhy and Marzouk, 2012; Rezende and Mohamed, 2015) to solve a Bayesian inference problem in a latent space. It assumes no particular map parameterization; rather, it wraps around any transport map class. We briefly review popular transport map parameterizations. Normalizing flows (Rezende and Mohamed, 2015; Tabak and Turner, 2013; Papamakarios et al., 2021; Kobyzev et al., 2020) form a broad class of methods that construct transport maps through compositions of neural networks with specific parameterizations. Autoregressive flows (Dinh et al., 2016; Kingma et al., 2016; Papamakarios et al., 2017; Huang et al., 2018; De Cao et al., 2020; Jaini et al., 2019), a popular subclass, compose autoregressive or triangular maps to allow efficient computation of Jacobian determinants (Daniels and Velikova, 2010; Wehenkel and Louppe, 2019). Several works seek an approximation to the Knothe–Rosenblatt (KR) rearrangement (Knothe, 1957; Rosenblatt, 1952). This diffeomorphic triangular map exists between any two distributions that are absolutely continuous with respect to a common measure, using orthonormal basis expansions such as sparse polynomials (El Moselhy and Marzouk, 2012; Marzouk et al., 2016; Spantini et al., 2018; Baptista et al., 2024b; Zech and Marzouk, 2022b; Westermann and Zech, 2025) or neural networks (Zech and Marzouk, 2022a). One distinguishing feature of `LazyDINO` is its use of PtO map Jacobian evaluations during training. Other recent works also incorporate derivative information, such as by adding a Fisher divergence term to the training objective (Zeghal et al., 2022; Brehmer et al., 2020), thus exploiting the differentiability of the log-likelihood. Finally, we note the recent popularity of inference methods that amortize transport-based posterior approximations (Durkan et al., 2018; Papamakarios et al., 2019; Greenberg et al., 2019; Papamakarios and Murray, 2018; Baptista et al., 2024a). Amortized simulation-based Inference (A-SBI, Ganguly et al. 2023) approaches parameterize transport maps to treat the conditioning variable (i.e., the observational data) as a functional input and generate samples from the corresponding posterior. Many transport map parametrizations have been studied for A-SBI; conditional normalizing flows, in particular, have achieved broad adoption; see, e.g., Abdelhamed et al. (2019); Algren et al. (2023); Schopmans and Friederich (2024); Li et al. (2025); Yang et al. (2024). In our numerical examples, we compare `LazyDINO` with a conditional normalizing flow implementation of A-SBI.

### 1.5 Notation

*Space, Vector, and Norms* We use bold symbols to denote finite-dimensional vectors, e.g., $\boldsymbol{x} \in \mathbb{R}^{d_{\boldsymbol{x}}}$, where $d_{\boldsymbol{x}}$ denotes the dimension. We denote the 2-norm on finite-dimensional vector spaces as $\|\cdot\|$. We denote the Frobenius matrix norm as $\|\cdot\|_F$. We use math script to denote separable Hilbert spaces that have high or infinite dimensions, e.g., $\mathscr{X}$, with inner product $\langle\cdot,\cdot\rangle_{\mathscr{X}}$, norm $\|\cdot\|_{\mathscr{X}}$, and element $x \in \mathscr{X}$.

*Linear Operators on Hilbert Spaces* We denote the Banach space of bounded linear operators from $\mathscr{X}_1$ to $\mathscr{X}_2$ as $B(\mathscr{X}_1, \mathscr{X}_2)$. We denote its subset of Hilbert–Schmidt (HS) operators as $\mathrm{HS}(\mathscr{X}_1, \mathscr{X}_2)$. We define $B(\mathscr{X}) := B(\mathscr{X}, \mathscr{X})$ and similar for HS operators. We denote the set of positive, self-adjoint, and trace class operators on $\mathscr{X}$ as $B_1^+(\mathscr{X})$. When $\mathscr{X}$ is a finite-dimensional vector space, $B_1^+(\mathscr{X})$ consists of symmetric positive definite matrices.

*Weighted Inner Products and Norms* We denote the inner-product and norm weighted by a positive and self-adjoint operator $\mathcal{A} : \mathscr{X} \to \mathscr{X}$ as $\langle x_1, x_2\rangle_{\mathcal{A}} := \langle \mathcal{A}^{1/2}x_1, \mathcal{A}^{1/2}x_2\rangle_{\mathscr{X}}$ and $\|x\|_{\mathcal{A}} := \sqrt{\langle \mathcal{A}^{1/2}x, \mathcal{A}^{1/2}x\rangle_{\mathscr{X}}}$ and $\mathcal{A}^{1/2}$ denotes the self-adjoint square root of $\mathcal{A}$. We note that the operator square root is not required in numerical implementation.

*Probability Measures on Hilbert Spaces* The set of probability distributions defined using Borel $\sigma$-algebra on $\mathscr{X}$ is denoted by $\mathscr{P}(\mathscr{X})$. The density between two probability distributions (i.e., Radon–Nikodym derivative) $\mu_1, \mu_2 \in \mathscr{P}(\mathscr{X})$ evaluated at $x \in \mathscr{X}$ is denoted as $(\mathrm{d}\mu_1/\mathrm{d}\mu_2)(x)$. For probability distributions on finite-dimensional vector spaces, we do not distinguish between a distribution $\mu$ and its density function $\pi(\boldsymbol{x}) = (\mathrm{d}\mu/\mathrm{d}\mu_L)(\boldsymbol{x})$, where $\mu_L$ is the Lebesgue measure. We use $\boldsymbol{x} \sim \pi$ and $\boldsymbol{x} \overset{\text{iid}}{\sim} \pi$ to denote a $\pi$-distributed random variable and independent and identically distributed samples from $\pi$, respectively.

*Diffeomorphism and Pushforward of Measures* We denote the diffeomorphism group of $\mathscr{X}$ as $\mathrm{Diff}^1(\mathscr{X}) := \big\{\mathcal{T} : \mathscr{X} \to \mathscr{X} \mid \mathcal{T} \text{ is an automorphism, and } \mathcal{T}, \mathcal{T}^{-1} \in C^1(\mathscr{X})\big\}$. For $\mathcal{T} \in \mathrm{Diff}^1(\mathscr{X})$, we denote by $\mathcal{T}_\sharp\mu$ and $\mathcal{T}^\sharp\mu$ the pushforward and pullback of probability distributions in the sense that $\mathcal{T}_\sharp\mu = \mu \circ \mathcal{T}^{-1}$ and $\mathcal{T}^\sharp\mu = \mu \circ \mathcal{T}$.

### 1.6 Outline of the Paper

The remainder of this work proceeds as follows: Section 2 introduces the lazy map variational inference method for solving infinite-dimensional Bayesian inversion. Section 3 introduces the optimized surrogate construction for amortized Bayesian inverse for lazy map variational inference. Section 4 describes the `LazyDINO` algorithm in detail, including documentation of all offline and online procedures and its role in enabling amortized inference. Section 5 defines the setup for numerical experiments, including the two infinite-dimensional PDE-constrained BIPs and all metrics utilized to measure the posterior approximation errors. Section 6 presents the numerical results and discusses the performance of `LazyDINO` relative to competing methods. Finally, we give concluding remarks in Section 7. We have additional results and discussions in the various appendices.

## 2. Solving Bayesian Inverse Problems Using Lazy Maps

In this section, we provide a precise formulation of lazy map variational inference (LMVI) for solving Bayesian inverse problems (BIPs), building on the conceptual outline in Section 1.1. We note that the following description of the lazy map and its optimization largely follows that of Brennan et al. (2020) and extends it to Hilbert spaces.

This section is organized as follows. Section 2.1 defines the BIPs considered in this work. Sections 2.2 to 2.4 describe posterior approximation via ridge functions and the resulting inference problem in a parameter latent space. Sections 2.5 and 2.6 introduce LMVI that approximates the target posterior in the latent space.

### 2.1 Bayesian Inverse Problems

We denote the parameter of interest $m \in \mathscr{M}$, where $\mathscr{M}$ is a separable Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathscr{M}}$. Let $\boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ denote observational data, and $\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathbb{R}^{d_{\boldsymbol{y}}}$ denote the parameter-to-observable (PtO) map. We begin with the following standard assumptions.

**Assumption 1 (Gaussian Prior)** *We consider a Gaussian prior distribution $\mu = \mathcal{N}(0, \mathcal{C})$ with a covariance operator $\mathcal{C} \in B_1^+(\mathscr{M})$.*

**Assumption 2 (Additive Gaussian Noise)** *We assume the observed data has the following distribution:*

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mathcal{G}}(m), \Gamma_n), \tag{5}$$

*where $\Gamma_n \in B_1^+(\mathbb{R}^{d_{\boldsymbol{y}}})$.*

We consider the inverse problem of recovering the parameter $m$ from the observational data vector $\boldsymbol{y}$. We are interested in characterizing the posterior distribution satisfying Bayes' rule, which we denote by $\mu^{\boldsymbol{y}} \in \mathscr{P}(\mathscr{M})$:

$$\frac{\mathrm{d}\mu^{\boldsymbol{y}}}{\mathrm{d}\mu}(m) = \frac{1}{Z^{\boldsymbol{y}}} \exp(-\Phi^{\boldsymbol{y}}(m)), \qquad \Phi^{\boldsymbol{y}}(m) := \frac{1}{2} \|\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}\|^2_{\Gamma_n^{-1}}. \tag{6}$$

Here, we define the *potential* $\Phi^{\boldsymbol{y}} : \mathscr{M} \to \mathbb{R}$, i.e. the negative log-likelihood, and the normalization constant $Z^{\boldsymbol{y}} := \mathbb{E}_{m \sim \mu}\left[\exp(-\Phi^{\boldsymbol{y}}(m))\right]$.

**Remark 1** *We address two concerns regarding the BIP setting. Firstly, to infer parameters with non-Gaussian priors, one can perform inference in transformed coordinates distributed according to a Gaussian prior and subsequently sample from the posterior via the inverse transform; see, e.g., Soize and Ghanem (2004); Sraj et al. (2016). Secondly, even though we only consider the BIP arising from a single observation, this work straightforwardly extends to a collection of observations, e.g., $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mathcal{G}}(m), \Gamma_n)$, in which case the potential in* (6) *becomes the sum of the negative log-likelihood of each observation. This extension is non-trivial for many A-SBI methods.*

We define the following Cameron–Martin spaces, i.e., separable Hilbert spaces with prior and noise precision–weighted inner products,

$$H_{\mathcal{C}} := \{m \in \mathscr{M} : \|m\|_{\mathcal{C}^{-1}} < \infty\}, \quad H_{\Gamma_n} := \left\{\boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}} : \|\boldsymbol{y}\|_{\Gamma_n^{-1}} < \infty\right\},$$

where $H_{\mathcal{C}}$ and $H_{\Gamma_n}$ are equipped with inner products $\langle\cdot,\cdot\rangle_{\mathcal{C}^{-1}}$ and $\langle\cdot,\cdot\rangle_{\Gamma_n^{-1}}$, respectively. Note that $H_{\mathcal{C}}$ is continuously embedded in $\mathscr{M}$ and is isomorphic to $\mathscr{M}$ with respect to the identity map only when $\mathscr{M}$ is finite-dimensional.

In this work, we assume the PtO map is $H_\mu^1$-differentiable in the following sense.

**Assumption 3 ($H_\mu^1$-Differentiable PtO Map)** *We assume the PtO map to live in the Sobolev space with Gaussian measure* $H_\mu^1(\mathscr{M}; H_{\Gamma_n}) := \{\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathbb{R}^{d_{\boldsymbol{y}}} : \|\boldsymbol{\mathcal{G}}\|_{H_\mu^1(\mathscr{M};H_{\Gamma_n})} < \infty\}$ *where*

$$\|\boldsymbol{\mathcal{G}}\|^2_{H_\mu^1(\mathscr{M},H_{\Gamma_n})} := \mathbb{E}_{m\sim\mu}\left[\|\boldsymbol{\mathcal{G}}(m)\|^2_{\Gamma_n^{-1}} + \|D_H\boldsymbol{\mathcal{G}}(m)\|^2_{\mathrm{HS}(H_{\mathcal{C}},H_{\Gamma_n})}\right], \tag{7}$$

*and* $D_H\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathrm{HS}(H_{\mathcal{C}}, H_{\Gamma_n})$ *is the stochastic derivative or the Malliavin derivative of* $\boldsymbol{\mathcal{G}}$ *that satisfies*

$$\lim_{t\to 0}\left\|t^{-1}\left(\boldsymbol{\mathcal{G}}(m+t\delta m) - \boldsymbol{\mathcal{G}}(m)\right) - D_H\boldsymbol{\mathcal{G}}(m)\delta m\right\|_{\Gamma_n^{-1}} = 0 \quad \forall \delta m \in H_{\mathcal{C}} \quad a.s. \tag{8}$$

If the Fréchet derivative $D\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathrm{HS}(\mathscr{M}, \mathbb{R}^{d_{\boldsymbol{y}}})$ exists, we have the following equality a.s.,

$$D_H\boldsymbol{\mathcal{G}}(m) = D\boldsymbol{\mathcal{G}}(m)|_{H_{\mathcal{C}}}, \quad D_H\boldsymbol{\mathcal{G}}(m)^* = \mathcal{C}D\boldsymbol{\mathcal{G}}(m)^*\Gamma_n^{-1}, \tag{9}$$

where $|_{H_{\mathcal{C}}}$ denotes the restriction of the function domain from $\mathscr{M}$ to $H_{\mathcal{C}}$. We do not distinguish between $D\boldsymbol{\mathcal{G}}(m)$ and $D_H\boldsymbol{\mathcal{G}}(m)$ when it is clear that the domain is $H_{\mathcal{C}}$.

### 2.2 Subspace Decomposition of Bayesian Inverse Problems

Let $\mathcal{P} \in B(\mathscr{M})$ be a rank-$d_r$ linear projection, which induces a unique decomposition of $\mathscr{M}$ into its image (i.e., the range space) and kernel (i.e., the null space):

$$\mathscr{M} = \mathrm{Im}(\mathcal{P}) \oplus \mathrm{Ker}(\mathcal{P}), \quad m = \underbrace{\mathcal{P}m}_{m_r} + \underbrace{(\mathrm{Id}_{\mathscr{M}} - \mathcal{P})m}_{m_\perp} \quad \forall m \in \mathscr{M},$$

where $\oplus$ denotes the direct sum as defined above. We denote the prior and posterior marginals in $\mathrm{Im}(\mathcal{P})$ by the pushforward $\mu_r := \mathcal{P}_\sharp\mu$ and $\mu_r^{\boldsymbol{y}} := \mathcal{P}_\sharp\mu^{\boldsymbol{y}}$, respectively. A probability distribution on $\mathscr{M}$ can be decomposed into its marginal probability in $\mathrm{Im}(\mathcal{P})$ and its conditional probability in $\mathrm{Ker}(\mathcal{P})$ in the following sense. For any measurable subset $\mathscr{A} \subseteq \mathscr{M}$ and its decomposition $\mathscr{A} = \mathscr{A}_r \oplus \mathscr{A}_\perp$, where $\mathscr{A}_r \subseteq \mathrm{Im}(\mathcal{P})$ and $\mathscr{A}_\perp \subseteq \mathrm{Ker}(\mathcal{P})$, the prior and posterior probability concentrations on $\mathscr{A}$ are given by

$$\mu(\mathscr{A}) = \int_{\mathscr{A}_r}\mu_{\perp|r}(\mathscr{A}_\perp|m_r)\mathrm{d}\mu_r(m_r), \quad \mu^{\boldsymbol{y}}(\mathscr{A}) = \int_{\mathscr{A}_r}\mu^{\boldsymbol{y}}_{\perp|r}(\mathscr{A}_\perp|m_r)\mathrm{d}\mu_r^{\boldsymbol{y}}(m_r), \tag{10}$$

where $\mu_{\perp|r}(\cdot|m_r), \mu^{\boldsymbol{y}}_{\perp|r}(\cdot|m_r) \in \mathscr{P}(\mathrm{Ker}(\mathcal{P}))$ are the prior and posterior conditionals in $\mathrm{Ker}(\mathcal{P})$. In particular, the prior marginal $\mu_r$, hereafter referred to as the *subspace prior*, and conditional $\mu_{\perp|r}(\cdot|m_r)$ has closed forms given by:

$$\mu_r = \mathcal{N}\left(0, \mathcal{P}\mathcal{C}\mathcal{P}^*\right), \quad \mu_{\perp|r}(\cdot|m_r) = \mathcal{N}\left(\mathcal{C}\mathcal{P}^*(\mathcal{P}\mathcal{C}\mathcal{P}^*)^{-1}m_r - m_r, \mathcal{C}\mathcal{P}^* - \mathcal{P}\mathcal{C}\right), \tag{11}$$

where $\mathcal{P}^*$ is the Hermitian adjoint of $\mathcal{P}$. These forms can be simplified for specific choices of $\mathcal{P}$, which is discussed in Section 2.4.

### 2.3 Posterior Approximation Using Ridge Functions

We proceed under the assumption that the projection $\mathcal{P}$ has been chosen such that the data $\boldsymbol{y}$ are uninformative of the parameter in $\mathrm{Ker}(\mathcal{P})$, i.e., the difference between the prior and the posterior is small in $\mathrm{Ker}(\mathcal{P})$. The process for choosing $\mathcal{P}$ will be delineated in Section 3. Under this assumption, we consider a ridge function approximation of the PtO map:

$$\widetilde{\boldsymbol{\mathcal{G}}} : \mathrm{Im}(\mathcal{P}) \to \mathbb{R}^{d_{\boldsymbol{y}}}, \quad \widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P} \approx \boldsymbol{\mathcal{G}}. \tag{12}$$

An example of this ridge function is the conditional expectation of the PtO map, where the projected parameter is lifted into the full space $\mathscr{M}$ by filling $\mathrm{Ker}(\mathcal{P})$ with the prior conditional:

$$\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}(\mathcal{P}m) \coloneqq \mathbb{E}_{m_\perp \sim \mu_{\perp|r}(\cdot|m_r)}\left[\boldsymbol{\mathcal{G}}(\mathcal{P}m + m_\perp)\right]. \tag{13}$$

For a given projection, this ridge function is optimal with respect to the Bochner norm on $L^2_\mu(\mathscr{M}; H_{\Gamma_n})$ (Zahm et al., 2020),

$$\mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}\left(\mathcal{P}m\right)\right\|^2_{\Gamma_n^{-1}}\right] = \inf_{\widetilde{\boldsymbol{\mathcal{G}}}:\mathrm{Im}(\mathcal{P})\to\mathbb{R}^{d_{\boldsymbol{y}}}} \mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right].$$

We refer to $\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{P}$ as the *optimal ridge function.*

A ridge function approximation of the PtO map induces an approximate posterior $\widetilde{\mu}^{\boldsymbol{y}} \in \mathscr{P}(\mathscr{M})$ given by

$$\widetilde{\Phi}^{\boldsymbol{y}}(\mathcal{P}m) \coloneqq \frac{1}{2}\left\|\widetilde{\boldsymbol{\mathcal{G}}}(\mathcal{P}m) - \boldsymbol{y}\right\|^2_{\Gamma_n^{-1}}, \quad \frac{\mathrm{d}\widetilde{\mu}^{\boldsymbol{y}}}{\mathrm{d}\mu}(m) = \frac{1}{\widetilde{Z}^{\boldsymbol{y}}}\exp\left(-\widetilde{\Phi}^{\boldsymbol{y}}(\mathcal{P}m)\right), \tag{14}$$

where $\widetilde{\Phi}^{\boldsymbol{y}} \circ \mathcal{P} \approx \Phi^{\boldsymbol{y}}$. Since the ridge function does not act in $\mathrm{Ker}(\mathcal{P})$, the approximate posterior conditional $\widetilde{\mu}^{\boldsymbol{y}}_{\perp|r}$ is proportional to the prior conditional $\mu_{\perp|r}$, and the following holds by Bayes' rule

$$\frac{\mathrm{d}\widetilde{\mu}^{\boldsymbol{y}}_r}{\mathrm{d}\mu_r}(m_r) = \frac{1}{\widetilde{Z}^{\boldsymbol{y}}_r}\exp(-\widetilde{\Phi}^{\boldsymbol{y}}(m_r)), \quad \widetilde{\mu}^{\boldsymbol{y}}_{\perp|r} = \frac{\widetilde{Z}^{\boldsymbol{y}}_r}{\widetilde{Z}^{\boldsymbol{y}}}\mu_{\perp|r}, \tag{15}$$

where the *subspace posterior*, $\widetilde{\mu}^{\boldsymbol{y}}_r \in \mathscr{P}(\mathrm{Im}(\mathcal{P}))$, is a marginal of $\widetilde{\mu}^{\boldsymbol{y}}$.

The quality of the ridge function $\widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P}$ can be understood through statistical distances between the posterior $\mu$ and the approximate posterior $\widetilde{\mu}^{\boldsymbol{y}}$, which are covered in Section 3.1.

### 2.4 Latent Bayesian Inverse Problems Induced by Ridge Functions

Let $\Psi_r = \{\psi_j \in \mathscr{M}\}_{j=1}^{d_r}$ denote a basis for $\mathrm{Im}(\mathcal{P})$, i.e., $\mathrm{span}(\Psi_r) = \mathrm{Im}(\mathcal{P})$. We refer to $\Psi_r$ as a *reduced basis* on $\mathscr{M}$. The reduced basis defines an encoder $\mathcal{E}_r$ and decoder $\mathcal{D}_r$ pair:

$$\begin{cases} \mathcal{E}_r : \mathscr{M} \ni \sum_{j=1}^{d_r} \boldsymbol{x}_j\psi_j + m_\perp \mapsto \boldsymbol{x} \in \mathbb{R}^{d_r}, \\ \mathcal{D}_r : \mathbb{R}^{d_r} \ni \boldsymbol{x} \mapsto \sum_{j=1}^{d_r} \boldsymbol{x}_j\psi_j \in \mathrm{Im}(\mathcal{P}), \end{cases} \qquad \begin{cases} \mathcal{P} = \mathcal{D}_r \circ \mathcal{E}_r, \\ \mathrm{Id}_{d_r} = \mathcal{E}_r \circ \mathcal{D}_r, \end{cases} \tag{16}$$

where $\mathrm{Id}_{d_r}$ is the identity matrix in $\mathbb{R}^{d_r}$. We refer to $\mathbb{R}^{d_r} = \mathcal{E}_r(\mathscr{M})$ as the *latent parameter vector space.* The *latent prior* $\pi \in \mathscr{P}(\mathbb{R}^{d_r})$ and *latent posterior* $\widetilde{\pi}^{\boldsymbol{y}} \in \mathscr{P}(\mathbb{R}^{d_r})$ are defined as the pushforward of the prior marginal $\mu_r$ and the subspace posterior $\widetilde{\mu}_r^{\boldsymbol{y}}$, respectively, by the encoder $\mathcal{E}_r$, and they satisfy Bayes' rule of the latent parameters:

$$\begin{cases} \pi := \mathcal{E}_{r\sharp}\mu_r = \mathcal{N}(0, \mathcal{E}_r\mathcal{C}\mathcal{E}_r^*) \\ \widetilde{\pi}^{\boldsymbol{y}} := \mathcal{E}_{r\sharp}\widetilde{\mu}_r^{\boldsymbol{y}} \end{cases}, \quad \widetilde{\pi}^{\boldsymbol{y}}(\boldsymbol{x}) = \frac{1}{\widetilde{Z}_r^{\boldsymbol{y}}} \exp\left(-\widetilde{\Phi}^{\boldsymbol{y}}(\mathcal{D}_r\boldsymbol{x})\right)\pi(\boldsymbol{x}). \tag{17}$$

Given a latent posterior sample $\boldsymbol{x} \sim \widetilde{\pi}^{\boldsymbol{y}}$ and a prior conditional samples $m_\perp \sim \mu_{\perp|r}(\cdot|\mathcal{D}_r\boldsymbol{x})$, then $m = \mathcal{D}_r\boldsymbol{x} + m_\perp$ is distributed according to the approximate posterior $\widetilde{\mu}^{\boldsymbol{y}}$.

While sampling the latent posterior $\widetilde{\pi}^{\boldsymbol{y}}$ is not straightforward for nonlinear BIPs, we consider a class of $H_{\mathcal{C}}$–orthonormal reduced bases that simplifies the sampling of the prior conditional $\mu_{\perp|r}$. These reduced bases satisfy

$$\langle\psi_j, \psi_k\rangle_{\mathcal{C}^{-1}} = \delta_{jk}, \quad j,k = 1,\ldots,d_r, \quad \mathcal{E}_r m = \sum_{j=1}^{d_r} \langle m, \psi_j\rangle_{\mathcal{C}^{-1}}\,\boldsymbol{e}_j, \tag{18}$$

where $\delta_{jk}$ is the Kronecker delta, and $\boldsymbol{e}_j$ is the unit vector in the latent space $\mathbb{R}^{d_r}$. Through the definition of the Hermitian adjoint on $\mathscr{M}$, we have $\mathcal{E}_r^* = \mathcal{C}^{-1}\mathcal{D}_r$ and $\mathcal{D}_r^* = \mathcal{E}_r\mathcal{C}$, which implies $\mathcal{E}_r\mathcal{C}\mathcal{E}_r^* = \mathrm{Id}_{\mathbb{R}^{d_r}}$ and $\mathcal{P}^* = \mathcal{C}^{-1}\mathcal{P}\mathcal{C}$ due to (16). Consequently, (11) and (17) yield

$$\pi = \mathcal{N}(\boldsymbol{0}, \mathrm{Id}_{\mathbb{R}^{d_r}}), \qquad \text{(Whitened Latent Prior)} \tag{19}$$

$$\mu_{\perp|r}(\cdot|m_r) \equiv \mu_\perp, \qquad \text{(Independence of Marginals)} \tag{20}$$

where $\mu_\perp = (\mathrm{Id}_{\mathscr{M}} - \mathcal{P})_\sharp\mu$ is the prior marginal in $\mathrm{Ker}(\mathcal{P})$. As a result of (20), sampling of the conditional $m_\perp \sim \mu_{\perp|r}(\cdot|m_r)$ can be accomplished via sampling the full prior:

$$m_\perp = m_{\mathrm{pr}} - \mathcal{P}m_{\mathrm{pr}}, \quad m_{\mathrm{pr}} \sim \mu. \tag{21}$$

### 2.5 Lazy Map Variational Inference

We consider TMVI that seeks a diffeomorphic deterministic coupling between a target and a reference distribution. For our BIP in (6), we aim to find a transport map $\mathcal{T} \in \mathrm{Diff}^1(\mathscr{M})$ that couples our reference, the prior $\mu \in \mathscr{P}(\mathscr{M})$, to the target, the posterior $\mu^{\boldsymbol{y}} \in \mathscr{P}(\mathscr{M})$:

$$\mathcal{T}_\sharp\mu = \mu^{\boldsymbol{y}}, \quad \mathcal{T}^\sharp\mu^{\boldsymbol{y}} = \mu.$$

Given $\mathcal{T}$, sampling from the posterior $m_{\mathrm{post}} \sim \mu^{\boldsymbol{y}}$ is accomplished by evaluating $m_{\mathrm{post}} = \mathcal{T}(m_{\mathrm{pr}})$, where $m_{\mathrm{pr}} \sim \mu$. Since $\mathcal{T}$ is typically unavailable in closed form for nonlinear BIPs, we consider classes of transport maps parametrized by weights $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ such that $\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu$ and $\mu$ are mutually absolutely continuous. These weights are found via the solution of a stochastic optimization problem, whose goal is to find $\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu \approx \mu^{\boldsymbol{y}}$. The reverse Kullback-Leibler (rKL) divergence of the approximating distribution from the target posterior $\mu^{\boldsymbol{y}}$

$$\mathcal{D}_{\mathrm{KL}}(\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu||\mu^{\boldsymbol{y}}) = \mathcal{D}_{\mathrm{KL}}(\mu||\mathcal{T}_{\boldsymbol{\theta}}{}^\sharp\mu^{\boldsymbol{y}}) = \mathbb{E}_{m\sim\mu}\left[\log\left(\frac{\mathrm{d}\mu}{\mathrm{d}(\mu^{\boldsymbol{y}} \circ \mathcal{T}_{\boldsymbol{\theta}})}(m)\right)\right]$$

is often used to measure the error of transport map posterior approximation and thereby employed as the objective function for optimization (Marzouk et al., 2016; Blei et al., 2017). This objective is equivalent to the *evidence lower bound* objective function.

To make TMVI tractable when $\mathscr{M}$ has high or infinite dimensions, we use *lazy maps* of Brennan et al. (2020), which leverages the subspace decomposition as in Section 2.4:

$$\mathbf{T}_{\boldsymbol{\theta}} \coloneqq \mathbf{T}(\cdot,\boldsymbol{\theta}) \in \mathscr{T} \subset \mathrm{Diff}^1(\mathbb{R}^{d_r}), \quad \mathcal{T}_{\boldsymbol{\theta}} \coloneqq \overbrace{(\mathrm{Id}_{\mathscr{M}} - \mathcal{P})}^{\text{Identity in } \mathrm{Ker}(\mathcal{P})} + \underbrace{\mathcal{D}_r \circ \mathbf{T}_{\boldsymbol{\theta}} \circ \mathcal{E}_r}_{\text{Nonlinear Transport in } \mathrm{Im}(\mathcal{P})}, \quad \text{(Lazy Map)} \tag{22}$$

where a latent space nonlinear transport is used to represent the coupling of the prior and the posterior in $\mathrm{Im}(\mathcal{P})$ while the prior is preserved in $\mathrm{Ker}(\mathcal{P})$. Here we consider a parametrized class $\mathscr{T} \subset \mathrm{Diff}^1(\mathbb{R}^{d_r})$ with weights $\boldsymbol{\theta} \subseteq \mathbb{R}^{d_{\boldsymbol{\theta}}}$ that allows $\mathbf{T}_{\boldsymbol{\theta}\sharp}\pi$ and $\pi$ to be mutually absolutely continuous and assume the diffeomorphic property is achieved by constraining the map to satisfy $\det \nabla_{\boldsymbol{z}} \mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z}) > 0$ a.e.; see Marzouk et al. (2016).

The following proposition shows an equivalence in rKL between a lazy map defined on the parameter space $\mathscr{M}$, and a transport map defined on the latent space $\mathbb{R}^{d_r}$ through the optimal ridge function as in (17).

**Proposition 2** *Given a linear projection $\mathcal{P}$ defined using a $H_{\mathcal{C}}$-orthonormal reduced basis and a latent space transport $\mathbf{T}_{\boldsymbol{\theta}} \in \mathscr{T}$, we have*

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}(\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu||\mu^{\boldsymbol{y}}) &= \mathcal{D}_{\mathrm{KL}}(\mathbf{T}_{\boldsymbol{\theta}\sharp}\pi||\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}) + C_1 \\ &= \mathbb{E}_{\boldsymbol{z}\sim\pi}\Big[\Big(\widetilde{\Phi}^{\boldsymbol{y}}_{\mathrm{opt}} \circ \mathcal{D}_r \circ \mathbf{T}_{\boldsymbol{\theta}}\Big)(\boldsymbol{z}) + \frac{1}{2}\left\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right\|^2 - \log\det\nabla_{\boldsymbol{z}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\Big] + C_2\end{aligned} \tag{23}$$

*where $\mathcal{T}_{\boldsymbol{\theta}}$ is the lazy map in* (22), $\pi = \mathcal{N}(\mathbf{0}, \mathrm{Id}_{\mathbb{R}^{d_r}})$ *is the whitened latent prior in* (19), $\widetilde{\Phi}^{\boldsymbol{y}}_{\mathrm{opt}}$ *and $\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}$ are the approximate potential and latent posterior induced by $\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{P}$ in* (13), *and $C_1$ and $C_2$ are constants that do not depend on $\boldsymbol{\theta}$.*

The proof of Proposition 2 is provided in Appendix C. Due to this result, we may formulate LMVI as the following optimization problem with an equivalent latent representation

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{\boldsymbol{\theta}}} \mathcal{D}_{\mathrm{KL}}(\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu||\mu^{\boldsymbol{y}}), \qquad \text{(Lazy Map Variational Inference)} \tag{24a}$$

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{\boldsymbol{\theta}}} \mathcal{D}_{\mathrm{KL}}(\mathbf{T}_{\boldsymbol{\theta}\sharp}\pi||\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}). \qquad \text{(Equivalent Latent Representation)} \tag{24b}$$

The latent representation can be treated as a TMVI problem that seeks $\mathbf{T} \in \mathrm{Diff}^1(\mathbb{R}^{d_r})$ defined as follows.

$$\mathbf{T}_\sharp\pi = \widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}, \quad \mathbf{T}_\sharp\pi(\boldsymbol{x}) = (\pi \circ \mathbf{T}^{-1})(\boldsymbol{x})|\det\nabla\mathbf{T}^{-1}(\boldsymbol{x})|, \qquad \text{(Latent Space Pushforward)}$$

$$\mathbf{T}^\sharp\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}} = \pi, \quad \mathbf{T}^\sharp\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}(\boldsymbol{z}) = (\widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}} \circ \mathbf{T})(\boldsymbol{z})|\det\nabla\mathbf{T}(\boldsymbol{z})|. \qquad \text{(Latent Space Pullback)}$$

Given such a transport map, sampling $\boldsymbol{x} \sim \widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}$ is accomplished by evaluating $\boldsymbol{x} = \mathbf{T}(\boldsymbol{z})$, where $\boldsymbol{z} \sim \pi$. In turn, approximate posterior, or *pushforward*, samples $m \sim \widetilde{\mu}^{\boldsymbol{y}}_{\mathrm{opt}}$ can be drawn simply by lifting $\boldsymbol{x}$ into $\mathscr{M}$ following (21).

| BIP Name | Approximation | | Parameter space | Prior and Posterior | |
|---|---|---|---|---|---|
| Original | None | | $\mathscr{M}$ | $\mu, \mu^{\boldsymbol{y}}$ | (6) |
| Subspace | $\boldsymbol{\mathcal{G}} \approx \widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P}$ | (12) | $\mathrm{Im}(\mathcal{P})$ | $\mu_r, \widetilde{\mu}_r^{\boldsymbol{y}}$ | (15) |
| Latent | $\boldsymbol{\mathcal{G}} \approx \widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P}$<br>$\mathcal{P} = \mathcal{D}_r \circ \mathcal{E}_r$ | (12)<br>(16) | $\mathbb{R}^{d_r}$ | $\pi, \widetilde{\pi}^{\boldsymbol{y}}$ | (17) |
| Lazy map<br>Latent Representation | $\boldsymbol{\mathcal{G}} \approx \widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{P}$<br>$\mathcal{P} = \mathcal{D}_r \circ \mathcal{E}_r$ | (13)<br>(16) | $\mathbb{R}^{d_r}$ | $\pi, \widetilde{\pi}^{\boldsymbol{y}}_{\mathrm{opt}}$ | (17) |

Table 1: A summary of BIPs discussed in Section 2.

### 2.6 Stochastic Optimization of Lazy Map and Challenges

For a given transport map parametrization $\mathbf{T}_{\boldsymbol{\theta}} \in \mathscr{T}$, the map parameter vector $\boldsymbol{\theta}$ is typically found via gradient-based stochastic optimization, which in turn requires evaluating Monte Carlo (MC) estimates of the gradient of the objective with respect to $\boldsymbol{\theta}$. The shifted rKL objective, denoted as $\mathcal{L}^{\boldsymbol{y}} : \mathbb{R}^{d_{\boldsymbol{\theta}}} \to \mathbb{R}$, can be expressed as an expectation of a single-sample MC estimator, $\mathcal{L}_1^{\boldsymbol{y}} : \mathscr{M} \times \mathbb{R}^{d_{\boldsymbol{\theta}}} \to \mathbb{R}$, defined as follows:

$$\mathcal{L}^{\boldsymbol{y}}(\theta) := \mathbb{E}_{m\sim\mu}\left[\mathcal{L}_1^{\boldsymbol{y}}(m, \boldsymbol{\theta})\right] := \mathcal{D}_{\mathrm{KL}}(\mathcal{T}_{\boldsymbol{\theta}\sharp}\mu || \mu^{\boldsymbol{y}}) - C_2, \tag{25}$$

$$\mathcal{L}_1^{\boldsymbol{y}}(\mathcal{D}_r\boldsymbol{z} + m_\perp, \theta) = \Phi^{\boldsymbol{y}}\left(\left(\mathcal{D}_r \circ \mathbf{T}_\theta\right)(\boldsymbol{z}) + m_\perp\right) + \frac{1}{2}\left\|\mathbf{T}_\theta(\boldsymbol{z})\right\|^2 - \log\det\nabla_{\boldsymbol{z}}\mathbf{T}_\theta(\boldsymbol{z}), \tag{26}$$

where $\boldsymbol{z} \sim \pi$ and $m_\perp \sim \mu_\perp$. The single-sample MC gradient estimator with respect to the map parameters $\boldsymbol{\theta}$ takes the following form

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}_1^{\boldsymbol{y}}(\mathcal{D}_r\boldsymbol{z} + m_\perp, \theta) = \nabla_{\boldsymbol{\theta}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top(\mathcal{E}_r \circ D_H\Phi^{\boldsymbol{y}})\left((\mathcal{D}_r \circ \mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) + m_\perp\right) \\ + \nabla_{\boldsymbol{\theta}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \nabla_{\boldsymbol{\theta}}(\log\det\nabla_{\boldsymbol{z}}\mathbf{T}_\theta)(\boldsymbol{z}),\end{aligned} \tag{27}$$

where $D_H\Phi^{\boldsymbol{y}}(m) := D_H\boldsymbol{\mathcal{G}}(m)^*(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y})$ is the prior-preconditioned gradient of the potential; see (9) for its full form.

The MC gradient estimator of the rKL objective is then

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}}\widehat{\mathcal{L}}^{\boldsymbol{y}}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\nabla_{\boldsymbol{\theta}}\mathcal{L}_1^{\boldsymbol{y}}(m^{(j)}, \boldsymbol{\theta}), \quad m^{(j)} \overset{\mathrm{iid}}{\sim} \mu.$$

**Remark 3** *In the numerical results, starting in Section 6, the lazy map optimization is implemented with an alternative form of the rKL objective where samples from $\mu_\perp$ are estimated as $\mathbb{E}[\mu_\perp] = 0$:*

$$\mathcal{L}^{\boldsymbol{y}}(\theta) := \mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\mathcal{L}_1^{\boldsymbol{y}}(\mathcal{D}_r\boldsymbol{z}, \boldsymbol{\theta})\right].$$

*This leads to a different TMVI problem induced by the ridge function $\boldsymbol{\mathcal{G}} \circ \mathcal{P}$ instead of $\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{P}$ in Proposition 2. We empirically found that it performs better under a limited computational budget, likely due to a superior bias–variance trade-off in TMVI.*

The computation cost of evaluating the second and third terms in (27) only depends on the parametrization of the transport map. Notably, triangular transport maps (Marzouk et al., 2016) and parameterization built as compositions of triangular maps (e.g., inverse autoregressive flows by Kingma et al. 2017), are structured such that these terms are efficiently computable. The first term requires evaluating the PtO map and its prior-preconditioned gradient. Optimizing for an accurate lazy map is prohibitively expensive when the PtO map is expensive to evaluate.

## 3. Optimized Surrogate Construction for Lazy Map Variational Inference

Now we detail the ideas sketched out in Section 1.2: the construction of a fast-to-evaluate neural network ridge function surrogate $\widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P} \approx \boldsymbol{\mathcal{G}}$ that leads to a small and controlled expected error in surrogate-driven LMVI. Under the setting of amortized Bayesian inversion, this expectation is taken over the marginal distribution of data, denoted as $\gamma \in \mathscr{P}(\mathbb{R}^{d_{\boldsymbol{y}}})$ with $\gamma(\boldsymbol{y}) \propto Z^{\boldsymbol{y}}$ as in (5). Our strategy for constructing this surrogate is given as follows.

1. Minimizing an upper bound on $\mathbb{E}_{\boldsymbol{y}\sim\gamma}[\mathcal{D}_{\mathrm{KL}}(\mu^{\boldsymbol{y}}||\widetilde{\mu}^{\boldsymbol{y}})]$, the expected forward KL divergence (fKL) from the posterior $\mu^{\boldsymbol{y}}$ defined by the PtO map to the approximate posterior $\widetilde{\mu}^{\boldsymbol{y}}$ defined by the surrogate.
2. Minimizing an upper bound on the expected optimality gap $\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\sqrt{\mathcal{L}^{\boldsymbol{y}}(\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger}) - \mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta}^{\boldsymbol{y},\dagger})}\right]$ for surrogate-driven LMVI, where $\boldsymbol{\theta}^{\boldsymbol{y},\dagger}$ is the true minimizer of the rKL objective and $\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger}$ is the minimizer found via the ridge function surrogate.

In this section, we show that the resulting ridge function surrogate is `DIPNet` (O'Leary-Roseberry et al., 2022b), trained using the derivative-informed learning method (O'Leary-Roseberry et al., 2024), and that it leads to a latent-space surrogate rKL objective.

### 3.1 Error Analysis for Surrogate-Driven Lazy Map Variational Inference

For any ridge function $\widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P}$ and a pair of encoder $\mathcal{E}_r$ and decoder $\mathcal{D}_r$ with $\mathcal{P} = \mathcal{D}_r \circ \mathcal{E}_r$, we define a finite-dimensional latent representation $\mathbf{g} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$ given by

$$\mathbf{g} \coloneqq \boldsymbol{V}^* \circ \widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{D}_r, \quad \widetilde{\boldsymbol{\mathcal{G}}} = \boldsymbol{V} \circ \mathbf{g} \circ \mathcal{E}_r. \tag{28}$$

Here $\boldsymbol{V} \in \mathrm{HS}(\mathbb{R}^{d_{\boldsymbol{y}}}, H_{\Gamma_n})$ is a full-rank matrix with columns consists of $H_{\Gamma_n}$-orthonormal basis and $\boldsymbol{V}^* = \boldsymbol{V}^\top \Gamma_n^{-1}$ is its Hermitian adjoint that satisfies $\boldsymbol{V}^*\boldsymbol{V} = \mathrm{Id}_{d_{\boldsymbol{y}}}$. Note that $\boldsymbol{V}^*$ is a whitening transformation on the data space. In this subsection, we take $\widetilde{\boldsymbol{\mathcal{G}}}$ and $\mathbf{g}$ as the surrogate PtO map and its latent representation.

For the optimal ridge function $\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}$ in (13) associated with a particular PtO map $\boldsymbol{\mathcal{G}}$ , we define the following optimal latent representations $\mathbf{g}_{\mathrm{opt}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$ according to (28):

$$\mathbf{g}_{\mathrm{opt}} \coloneqq \boldsymbol{V}^* \circ \widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{D}_r, \quad \widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} = \boldsymbol{V} \circ \mathbf{g}_{\mathrm{opt}} \circ \mathcal{E}_r. \tag{29}$$

The following theorem provides an upper bound on the expected fKL between the posterior and the approximated posterior, averaged over the data distribution.

**Theorem 4 (Posterior Approximation)** *Given $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M}; H_{\Gamma_n})$ and a projector $\mathcal{P} \in B(\mathscr{M})$ defined via an $H_{\mathcal{C}}$-orthonormal reduced basis as in* (18)*, we have the following inequality for the approximate posterior $\widetilde{\mu}^{\boldsymbol{y}}$ in* (15) *defined via any ridge function $\widetilde{\boldsymbol{\mathcal{G}}} \circ \mathcal{P} \in L^2_\mu(\mathscr{M}; H_{\Gamma_n})$:*

$$\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\mathcal{D}_{\mathrm{KL}}(\mu^{\boldsymbol{y}} \| \widetilde{\mu}^{\boldsymbol{y}})\right] \leq \underbrace{\mathrm{Tr}_{H_{\mathcal{C}}}\left((\mathrm{Id}_{H_{\mathcal{C}}} - \mathcal{P})\,\mathcal{H}_A\,(\mathrm{Id}_{H_{\mathcal{C}}} - \mathcal{P})\right)}_{\text{Parameter reduction error}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\left\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \mathbf{g}(\boldsymbol{z})\right\|^2\right]}_{\text{Latent representation error}},$$

*where $\mathrm{Tr}_{H_{\mathcal{C}}} : B_1^+(H_{\mathcal{C}}) \to \mathbb{R}$ returns the trace of operators on $H_{\mathcal{C}}$, and $\mathcal{H}_A \in B_1^+(H_{\mathcal{C}})$ is the expected prior–preconditioned Gauss–Newton Hessian of the potential:*

$$\mathcal{H}_A := \mathbb{E}_{m\sim\mu}\left[D_H\boldsymbol{\mathcal{G}}(m)^* D_H\boldsymbol{\mathcal{G}}(m)\right]. \tag{30}$$

*Here $D_H\boldsymbol{\mathcal{G}}(m)^* \in \mathrm{HS}(H_{\Gamma_n}, H_{\mathcal{C}})$ denotes the Hermitian adjoint of the stochastic derivative $D_H\boldsymbol{\mathcal{G}}(m)$ in* (8)*.*

The bound decomposes the expected error into terms involving the parameter reduction error that depends on the choice of $\mathcal{P}$ and the discrepancy between the surrogate latent representation $\mathbf{g}$ and the optimal latent representation $\mathbf{g}_{\mathrm{opt}}$ of $\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}$. The proof of Theorem 4 can be found in Appendix D.1.

For surrogate-driven LMVI, understanding the expected discrepancy between the posteriors and their transport targets (i.e., the surrogate approximated posteriors $\widetilde{\mu}^{\boldsymbol{y}}$) is insufficient, as the transport map is constructed through the process of gradient-based stochastic optimization. The accuracy of the rKL gradient approximation is also important. The following theorem establishes upper bounds on the error of the surrogate objective gradient.

**Theorem 5 (Surrogate rKL Objective Gradient)** *Given $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M}; H_{\Gamma_n})$, a linear projector $\mathcal{P} \in B(\mathscr{M})$ defined using an $H_{\mathcal{C}}$-orthonormal reduced basis as in* (18)*. Assume we have a latent space transport $\mathbf{T}_{\boldsymbol{\theta}} \in \mathscr{T}$ with an essentially bounded density between $\mathbf{T}_{\boldsymbol{\theta}\sharp}\pi$ and $\pi$ and an essentially-bounded Jacobian with respect to $\boldsymbol{\theta}$. We have the following error upper bound for the approximate gradient of rKL objective $\widetilde{\boldsymbol{\mathcal{L}}}^{\boldsymbol{y}}(\boldsymbol{\theta})$ given by a ridge function,*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\left\|\nabla_{\boldsymbol{\theta}}\boldsymbol{\mathcal{L}}^{\boldsymbol{y}}(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{\mathcal{L}}}^{\boldsymbol{y}}(\boldsymbol{\theta})\right\|\right] \lesssim \Big(\mathbb{E}_{\boldsymbol{z}\sim\pi}\Big[&\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \mathbf{g}(\boldsymbol{z})\|^2 \\ &+ \|\nabla\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \nabla\mathbf{g}(\boldsymbol{z})\|_F^2\Big]\Big)^{1/2},\end{aligned} \tag{31}$$

*where $\lesssim$ denotes bounded up to a multiplicative constant.*

The proof of Theorem 5 is presented in Appendix D.2. Our result states that the expected gradient error is controlled by the latent representation error measured in a $\pi$-weighted Sobolev norm on $H^1_\pi(\mathbb{R}^{d_r}; \mathbb{R}^{d_y})$, which additionally contain the expected error in the Jacobian compared to the error measure using $\pi$-weighted Bochner norm on $L^2_\pi(\mathbb{R}^{d_r}; \mathbb{R}^{d_y})$ in Theorem 4. Notably, the two error measures are generally not equivalent, and the Sobolev norm is stronger than the Bochner norm. This result reflects the fact that the gradient of the rKL objective involves the Jacobian of the PtO map, and the surrogate Jacobian accuracy affects the optimization of lazy maps. To further explore the consequences of surrogate Jacobian misfit, we consider the following corollary on the expected optimality gap for surrogate-driven LMVI under stronger assumptions.

**Corollary 6 (Optimality Gap)** *Suppose the assumptions in Theorem 5 hold. Let* $\boldsymbol{\theta}^{\boldsymbol{y},\dagger}$ *and* $\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger}$ *denote* $\gamma$*-measurable functions that return the second order stationary points of* $\mathcal{L}^{\boldsymbol{y}}$ *and* $\widetilde{\mathcal{L}}^{\boldsymbol{y}}$ $\gamma$*-a.e., respectively. Let* $B_r(\boldsymbol{x})$ *denote a ball of radius* $r$ *centered at* $\boldsymbol{x}$*. We assume that* $R^{\boldsymbol{y}}$ *and* $\lambda^{\boldsymbol{y}}$ *are* $\gamma$*-essentially bounded from below by some positive constants such that (i)* $\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger} \in B_{R^{\boldsymbol{y}}}(\boldsymbol{\theta}^{\boldsymbol{y},\dagger})$ $\gamma$*-a.e., and (ii)* $\nabla^2_{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta}) \succeq \lambda^{\boldsymbol{y}}\mathrm{Id}_{\mathbb{R}^{d_r}}$ *for all* $\boldsymbol{\theta} \in B_{R^{\boldsymbol{y}}}(\boldsymbol{\theta}^{\boldsymbol{y},\dagger})$ $\gamma$*-a.e.*

*We have the following upper bound on the optimality gap:*

$$\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\sqrt{\mathcal{L}^{\boldsymbol{y}}(\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger}) - \mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta}^{\boldsymbol{y},\dagger})}\right] \lesssim \Big(\mathbb{E}_{\boldsymbol{z}\sim\pi}\Big[\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \mathbf{g}(\boldsymbol{z})\|^2 + \|\nabla\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \nabla\mathbf{g}(\boldsymbol{z})\|_F^2\Big]\Big)^{1/2}. \tag{32}$$

This result states that if the rKL objective is locally strongly convex near the true rKL minimizers and the minimizers found by the surrogate lands within those locally convex regions, we can bound the expected optimality gap by the latent representation error measured by the $\pi$-weighted Sobolev norm. The assumptions in Corollary 6 are commonly employed in the optimization and machine learning literature, either directly via local strong convexity or through the Polyak–Łojasiewicz inequality; see, for example, Bottou et al. 2018. These two results together demonstrate the need to control the surrogate Jacobian error in the context of LMVI.

Motivated by these results, we delineate the procedure for constructing a surrogate model for LMVI in the following subsections.

### 3.2 Minimize the Parameter Reduction Error: Derivative-Informed Subspace

We seek an $H_{\mathcal{C}}$-orthonormal reduced basis $\{\psi_j\}_{j=1}^{d_r}$ such that $\mathrm{span}(\{\psi_j\}_{j=1}^{d_r}) = \mathrm{Im}(\mathcal{P})$ and the parameter reduction error term in Theorem 4 is minimized. This can be accomplished by finding the parameter subspace that the PtO map is most sensitive to in expectation. This subspace is often referred to as the derivative-informed subspace or active subspace (Zahm et al., 2020), and it can be computed from the dominant $d_r$ eigenfunctions arising from the following eigenvalue problem in $H_{\mathcal{C}}$,

$$\mathcal{H}_A\psi_j = \lambda_j\psi_j, \quad \langle\psi_j,\psi_k\rangle_{\mathcal{C}^{-1}} = \delta_{jk}, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq 0, \tag{33}$$

where $\mathcal{H}_A$ is the prior-preconditioned Gauss–Newton Hessian in (30). Under a stronger Fréchet differentiability assumption, the eigenvalue problem in $H_{\mathcal{C}}$ is equivalent to a more common form of a generalized eigenvalue problem in $\mathscr{M}$ due to (9):

$$\mathbb{E}_{m\sim\mu}\left[D\boldsymbol{\mathcal{G}}(m)^*\Gamma_n^{-1}D\boldsymbol{\mathcal{G}}(m)\right]\psi_j = \lambda_j\mathcal{C}^{-1}\psi_j, \quad \langle\psi_k,\psi_j\rangle_{\mathcal{C}^{-1}} = \delta_{jk}, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq 0. \tag{34}$$

The minimum value of the parameter reduction error is

$$\min_{\substack{\mathcal{P}\in\{\text{rank}-d_r \text{ linear} \\ \text{projection on } \mathscr{M}\}}} \mathrm{Tr}_{H_{\mathcal{C}}}\left((\mathrm{Id}_{H_{\mathcal{C}}} - \mathcal{P})\,\mathcal{H}_A\,(\mathrm{Id}_{H_{\mathcal{C}}} - \mathcal{P})\right) = \sum_{j>d_r}\lambda_j. \tag{35}$$

This derivative-based reduced basis leads to an expected parameter reduction error proportional to the eigenvalue tail sum in (33) corresponding to the discarded eigenfunctions. Existing bounds for the truncated Karhunen–Loéve expansion of the parameter are strictly higher than (35); see Zahm et al. (2020); Cao et al. (2025).

### 3.3 Minimize the Latent Representation Error: Conventional Learning

We first consider a neural operator ridge function using a neural network latent representation $\mathbf{g}_{\mathrm{NN}} : \mathbb{R}^{d_r} \times \mathbb{R}^{d_{\boldsymbol{w}}} \to \mathbb{R}^{d_{\boldsymbol{y}}}$:

$$\mathbf{g}_{\boldsymbol{w}}(\boldsymbol{x}) \coloneqq \mathbf{g}_{\mathrm{NN}}(\boldsymbol{x}, \boldsymbol{w}), \qquad \text{(Neural Latent Representation)} \qquad (36a)$$

$$\widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}}(\mathcal{P}m) \coloneqq \boldsymbol{V}\mathbf{g}_{\mathrm{NN}}(\mathcal{E}_r m, \boldsymbol{w}), \qquad \text{(Neural Ridge Function)} \qquad (36b)$$

where $\boldsymbol{w} \in \mathbb{R}^{d_{\boldsymbol{w}}}$ consists of trainable weights. Neural ridge function architecture using the derivative-informed subspace (34) is known as `DIPNet` (O'Leary-Roseberry et al., 2022b).

Motivated by Theorem 4, it would be sensible to find the neural network weights by minimizing the latent representation error, which also minimizes the upper bound on the expected surrogate posterior approximation error:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} \mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\left\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\right\|^2\right]. \qquad (37)$$

However, estimating this objective function requires a nested MC method due to the Ker($\mathcal{P}$) marginalization in $\mathbf{g}_{\mathrm{opt}}$; see definitions in (12) and (28). Specifically, $\boldsymbol{z}^{(j)} \sim \pi$, $1 \leq j \leq N_{\mathrm{out}}$, are used to estimate the objective, and $m_{\perp}^{(j,k)} \sim \mu_{\perp}$, $1 \leq k \leq N_{\mathrm{in}}$, are used to estimate the output of the optimal latent representation at each $\boldsymbol{z}^{(j)}$. However, the inner MC is unnecessary when $\mathcal{P}$ is chosen as in Section 3.2, since the PtO map is insensitive to changes in Ker($\mathcal{P}$); see, e.g., Zahm et al. 2022, Corollary 7.5. Therefore, we consider the conventional supervised learning method with error measure using the norm on the Bochner space $L^2_\mu(\mathscr{M}, H_{\Gamma_n})$, i.e., a mean squared error objective:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} \mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right], \qquad \text{(Conventional } L^2_\mu \text{ Learning)} \quad (38)$$

$$\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} \mathbb{E}_{(\boldsymbol{z},m_\perp)\sim\pi\otimes\mu_\perp}\left[\left\|\underbrace{\boldsymbol{V}^*\boldsymbol{\mathcal{G}}(\mathcal{D}_r\boldsymbol{z} + m_\perp)}_{\approx\,\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z})} - \mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\right\|^2\right]. \qquad \text{(Equiv. Latent Represent.)} \quad (39)$$

The equivalent latent representation of the operator learning objective reveals that this objective can be derived from the neural latent representation error (37) using a single sample ($m_\perp \sim \mu_\perp$) estimate of the marginalization in $\mathbf{g}_{\mathrm{opt}}$. For notational convenience, we will use $\boldsymbol{g}^{(j)}$ to denote the iid *whitened PtO samples* used to estimate the conventional $L^2_\mu$ operator learning objective:

$$\boldsymbol{g}^{(j)} \coloneqq \boldsymbol{V}^*\boldsymbol{\mathcal{G}}(m^{(j)}) \in \mathbb{R}^{d_{\boldsymbol{y}}}, \quad m^{(j)} \overset{\mathrm{iid}}{\sim} \mu. \qquad \text{(Whitened PtO Sample)} \qquad (40)$$

We refer to `DIPNet` surrogates trained using the conventional $L^2_\mu$ operator learning method in (38) as `RB-NO` (reduced basis neural operator) in contrast to surrogates construction introduced in the following subsection.

### 3.4 Minimizing the Expected Optimality Gap: Derivative-Informed Learning

While the conventional $L^2_\mu$ learning problem presented in Section 3.3 is suitable for constructing a neural operator ridge function, Theorem 5 shows that controlling latent representation error in $H^1_\pi$ controls both the expected gradient error, as well as the expected optimality gap between the exact and surrogate variational inference objective functions. To this end, we consider minimizing the latent representation error measured by the $\pi$-weighted Sobolev norm on $H^1_\pi(\mathbb{R}^{d_r};\mathbb{R}^{d_{\boldsymbol{y}}})$:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} \mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\left\|\mathbf{g}_{\text{opt}}(\boldsymbol{z}) - \mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\right\|^2 + \left\|\nabla\mathbf{g}_{\text{opt}}(\boldsymbol{z}) - \nabla_{\boldsymbol{z}}\mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\right\|_F^2\right]. \tag{41}$$

However, as discussed in the previous subsection, estimating the objective function above also requires a nested MC method. To circumvent this issue, we adopt an derivative-informed $H^1_\mu$ operator learning objective following O'Leary-Roseberry et al. (2024); Cao et al. (2025):

$$\begin{aligned}\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} &\mathbb{E}_{m\sim\mu}\Big[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}} \\ &+ \left\|D\boldsymbol{\mathcal{G}}(m) - D(\widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}}\circ\mathcal{P})(m)\right\|^2_{\text{HS}(H_{\mathcal{C}},H_{\Gamma_n})}\Big]\end{aligned} \qquad \text{(Der.-Informed } H^1_\mu \text{ Learning)} \tag{42}$$

$$\begin{aligned}\min_{\boldsymbol{w}\in\mathbb{R}^{d_{\boldsymbol{w}}}} &\mathbb{E}_{(\boldsymbol{z},m_\perp)\sim\pi\otimes\mu_\perp}\Big[\left\|\boldsymbol{V}^*\boldsymbol{\mathcal{G}}(\mathcal{D}_r\boldsymbol{z}+m_\perp) - \mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\right\|^2 \\ &+ \|\underbrace{\boldsymbol{V}^*\circ D\boldsymbol{\mathcal{G}}(\mathcal{D}_r\boldsymbol{z}+m_\perp)\circ\mathcal{D}_r}_{\approx\nabla\mathbf{g}_{\text{opt}}(\boldsymbol{z})} - \nabla_{\boldsymbol{z}}\mathbf{g}_{\boldsymbol{w}}(\boldsymbol{z})\|^2_F\Big]\end{aligned} \qquad \text{(Equiv. Latent Represent.)} \tag{43}$$

The equivalent latent representation reveals that the derivative-informed learning objective can be derived from (41) using a single-sample ($m_\perp\sim\mu_\perp$) estimate for the marginalization in both $\mathbf{g}_{\text{opt}}$ and $\nabla_{\boldsymbol{z}}\mathbf{g}_{\text{opt}}$; see Appendix B for a discussion of this marginalization. We refer to `DIPNet` surrogates trained using the derivative-informed $H^1_\mu$ operator learning method as `RB-DINO` (reduced basis derivative-informed neural operator).

We emphasize that one only needs samples of the latent representation of the derivative for $H^1_\mu$ operator learning compared to $L^2_\mu$ operator learning,

$$\boldsymbol{J}_r^{(j)} := \boldsymbol{V}^*\circ D\boldsymbol{\mathcal{G}}(m^{(j)})\circ\mathcal{D}_r \in \mathbb{R}^{d_{\boldsymbol{y}}\times d_r} \quad m^{(j)} \overset{\text{iid}}{\sim} \mu. \qquad \text{(Whitened Latent Jacobian)} \tag{44}$$

For notational convenience, we use $\boldsymbol{J}_r^{(j)}$ to denote the iid samples of the *whitened latent Jacobian sample* of the PtO map.

### 3.5 Surrogate-Driven Lazy Map Variational Inference in the Latent Space

We use a trained ridge function surrogate $\widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}}\circ\mathcal{P}$ to replace the PtO map $\boldsymbol{\mathcal{G}}$ and its Jacobian $D\boldsymbol{\mathcal{G}}$ evaluations during stochastic optimization of the latent space transport $\mathbf{T}_{\boldsymbol{\theta}}$. Specifically, a single-sample estimate of the surrogate rKL $\widetilde{\mathcal{L}}_1^{\boldsymbol{y}} : \mathscr{M}\times\mathbb{R}^{d_{\boldsymbol{\theta}}}\to\mathbb{R}$, replacing (26), and its gradient, replacing  (27), can be equivalently represented in the latent space as $\widetilde{\mathcal{L}}_{1,r}^{\boldsymbol{y}} : \mathbb{R}^{d_r}\times\mathbb{R}^{d_{\boldsymbol{\theta}}}\to\mathbb{R}$:

$$\widetilde{\mathcal{L}}_1^{\boldsymbol{y}}(m,\boldsymbol{\theta};\boldsymbol{w}) \equiv \widetilde{\mathcal{L}}_{1,r}^{\boldsymbol{y}}(\mathcal{E}_r m,\boldsymbol{\theta};\boldsymbol{w}) \tag{45a}$$

$$\widetilde{\mathcal{L}}_{1,r}^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta};\boldsymbol{w}) = \frac{1}{2}\left\|(\mathbf{g}_{\boldsymbol{w}}\circ\mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) - \boldsymbol{V}^*\boldsymbol{y}\right\|^2 + \frac{1}{2}\left\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right\|^2 - \log\det\nabla_{\boldsymbol{z}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{45b}$$

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}_{1,r}^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta};\boldsymbol{w}) = \nabla_{\boldsymbol{\theta}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top\Big(&\nabla_{\boldsymbol{z}}\mathbf{g}_{\boldsymbol{w}}\circ\mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z})^\top((\mathbf{g}_{\boldsymbol{w}}\circ\mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) - \boldsymbol{V}^*\boldsymbol{y})\Big) \\ &+ \nabla_{\boldsymbol{\theta}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z}) - \nabla_{\boldsymbol{\theta}}(\log\det\nabla_{\boldsymbol{z}}\mathbf{T}_\theta)(\boldsymbol{z})\end{aligned} \tag{45c}$$

Consequently, the surrogate-driven lazy maps variational inference proceeds entirely in the parameter latent space. After the latent space transport map $\mathbf{T}_{\boldsymbol{\theta}}$ is optimized, we use the map to produce latent space posterior samples, i.e., $\boldsymbol{x} = \mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})$ with $z \sim \pi$, and they can be lifted to the full parameter space via sampling the prior as in (21).

**Remark 7 (Model Misspecification)** *In this work, we develop an efficient solver for BIPs with a given prior and PtO map. While model misspecification is not considered, we address it in this remark. Theoretical analysis indicates that our prior-based surrogate is robust: the surrogate posterior error is bounded by the $L^2_\mu$ surrogate error, provided the PtO map satisfies regularity conditions such as Lipschitz continuity—See Marzouk and Xiu (2009); Cao et al. (2023). Consequently, minimizing the surrogate error over the prior effectively minimizes the posterior error, even for highly unlikely observations far in the tails of the data distribution. However, the tightness of this error bound depends on the expected misfit between the observational data and model predictions. In cases of severe misspecification, a large expected misfit degrades the bound. This could be addressed via active learning, specifically by localizing samples to high-likelihood regions, provided the costs of the required online model evaluations can be managed efficiently.*

## 4. `LazyDINO` for Amortized Bayesian Inversion

In this section, we present a high-level overview of the steps involved in `LazyDINO` for amortized Bayesian inversion, using schematics and brief descriptions. We refer the reader to Appendix E for a more detailed exposition of these steps.

### 4.1 Offline Phase: `RB-DINO` Surrogate Construction

In Figure 3, we provide a schematic for the offline surrogate construction. In this phase, one first defines the prior $\mu$ and PtO map $\boldsymbol{\mathcal{G}}$ that determines the class of BIPs to be solved by `LazyDINO`. Subsequently, encoders and decoders for the parameter are constructed as delineated in Section 3.2. The training samples are then generated and reduced to their latent representations as in (40) and (44). In particular, the full PtO map Jacobian samples are never formed. Instead, they are compressed matrix-free using the parameter decoder $\mathcal{D}_r$. Generating training samples is often computationally costly, as PtO map evaluations require model solutions, and PtO map Jacobian evaluations require computing forward or adjoint model sensitivities. Once the training samples are collected, a given neural latent representation $\mathbf{g}_{\boldsymbol{w}}$ is trained using the derivative-informed learning method in the latent space (41). We refer to O'Leary-Roseberry et al. (2024); Cao et al. (2025) for more implementation details and theory on `RB-DINO` surrogate construction.

### 4.2 Online Phase: Rapid `LazyDINO` Transport Map Construction

Once a `RB-DINO` surrogate PtO map is constructed, it is used in place of the PtO map in the lazy map training, which removes the computational bottleneck of model solutions and makes rapid online inference possible. The process for lazy map construction for a single instance of observational data $\boldsymbol{y}$, is shown in Figure 4. This process involves defining the transport map architecture $\mathbf{T}_{\boldsymbol{\theta}}$, and the associated latent-space rKL objective (45a). The transport map is then optimized with respect to its map parameters $\boldsymbol{\theta}$. The trained latent

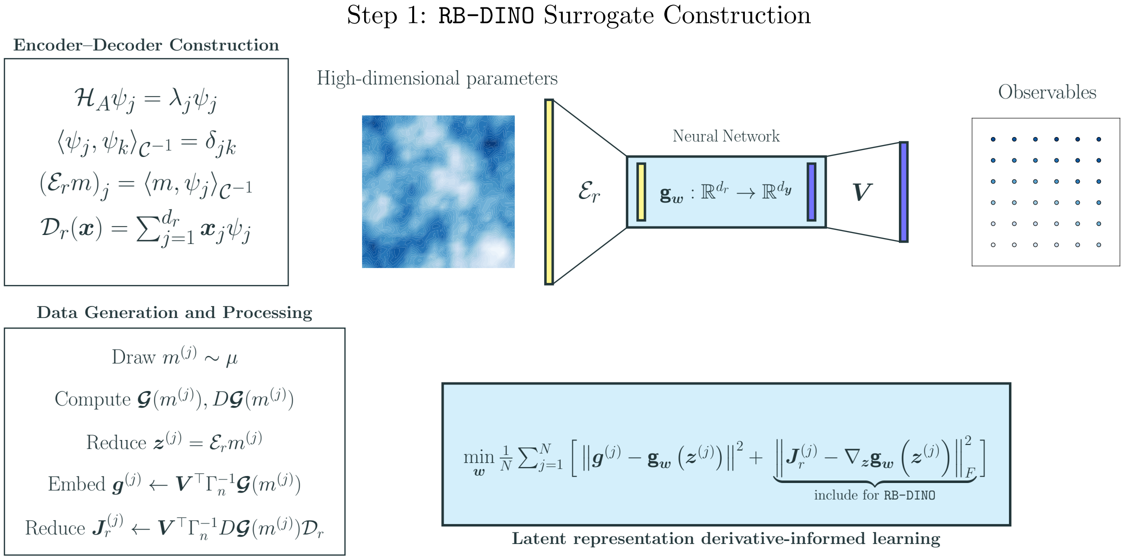
Step 1: RB-DINO Surrogate Construction
Encoder–Decoder Construction
$\mathcal{H}_A\psi_j = \lambda_j\psi_j$
$\langle\psi_j, \psi_k\rangle_{\mathcal{C}^{-1}} = \delta_{jk}$
$(\mathcal{E}_r m)_j = \langle m, \psi_j\rangle_{\mathcal{C}^{-1}}$
$\mathcal{D}_r(\boldsymbol{x}) = \sum_{j=1}^{d_r} \boldsymbol{x}_j\psi_j$
High-dimensional parameters
$\mathcal{E}_r$
Neural Network
$\mathbf{g}_{\boldsymbol{w}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$
$\boldsymbol{V}$
Observables
Data Generation and Processing
Draw $m^{(j)} \sim \mu$
Compute $\boldsymbol{\mathcal{G}}(m^{(j)}), D\boldsymbol{\mathcal{G}}(m^{(j)})$
Reduce $\boldsymbol{z}^{(j)} = \mathcal{E}_r m^{(j)}$
Embed $\boldsymbol{g}^{(j)} \leftarrow \boldsymbol{V}^\top\Gamma_n^{-1}\boldsymbol{\mathcal{G}}(m^{(j)})$
Reduce $\boldsymbol{J}_r^{(j)} \leftarrow \boldsymbol{V}^\top\Gamma_n^{-1}D\boldsymbol{\mathcal{G}}(m^{(j)})\mathcal{D}_r$
$\min_{\boldsymbol{w}} \frac{1}{N}\sum_{j=1}^{N}\left[\left\|\boldsymbol{g}^{(j)} - \mathbf{g}_{\boldsymbol{w}}\left(\boldsymbol{z}^{(j)}\right)\right\|^2 + \underbrace{\left\|\boldsymbol{J}_r^{(j)} - \nabla_{\boldsymbol{z}}\mathbf{g}_{\boldsymbol{w}}\left(\boldsymbol{z}^{(j)}\right)\right\|_F^2}_{\text{include for RB-DINO}}\right]$
Latent representation derivative-informed learning

Figure 3: Overview of the offline RB-DINO construction.

space transport map pushes the whitened latent prior $\pi$ in (19) to an approximation of the latent posterior induced by the optimal ridge function $\mathsf{T}_{\boldsymbol{\theta}\sharp}\pi \approx \widetilde{\pi}^{\boldsymbol{y}}_{\text{opt}}$ as in Proposition 2. These samples can then be decoded to generate samples in the full space using the prior $\mu$.

A major point of emphasis for LazyDINO is that the computationally expensive aspects of the method are limited to the offline phase. The RB-DINO surrogate replaces the expensive-to-evaluate and often implicitly defined PtO map with a fast-to-evaluate explicit function. In practice, this leads to potentially enormous speedups across all computations associated with the likelihood and its gradient.

Likewise, the transport map approximation depicted in Figure 4 occurs in a relatively low-dimensional parameter latent space. It exploits modern, optimized computing kernels, such as batch computations, fast white-noise sampling, automatic differentiation, and compile-time-optimized explicit calculations, with all precompiled before training.

### 4.3 LazyDINO as an Amortized Inference Method

The latent space transport maps can be rapidly constructed during the online phase, making LazyDINO a compelling method for real-time inference and a competitive alternative to A-SBI methods (Ganguly et al., 2023) for solving BIPs with the same PtO map but different instances of observational data.

Since we create a surrogate for the PtO map, and *not* the likelihood, the cost of RB-DINO surrogate construction can be amortized by instancing a new likelihood for any new observational data. In this amortization process shown in Figure 5, the construction of the RB-DINO surrogate is amortized over many different BIPs defined by the same PtO map and prior. With this in mind, LazyDINO is an ideal method for settings where many BIPs are solved for the same setting. This class of problems appears in predictive digital twins, state estimation, and optimal experimental design.

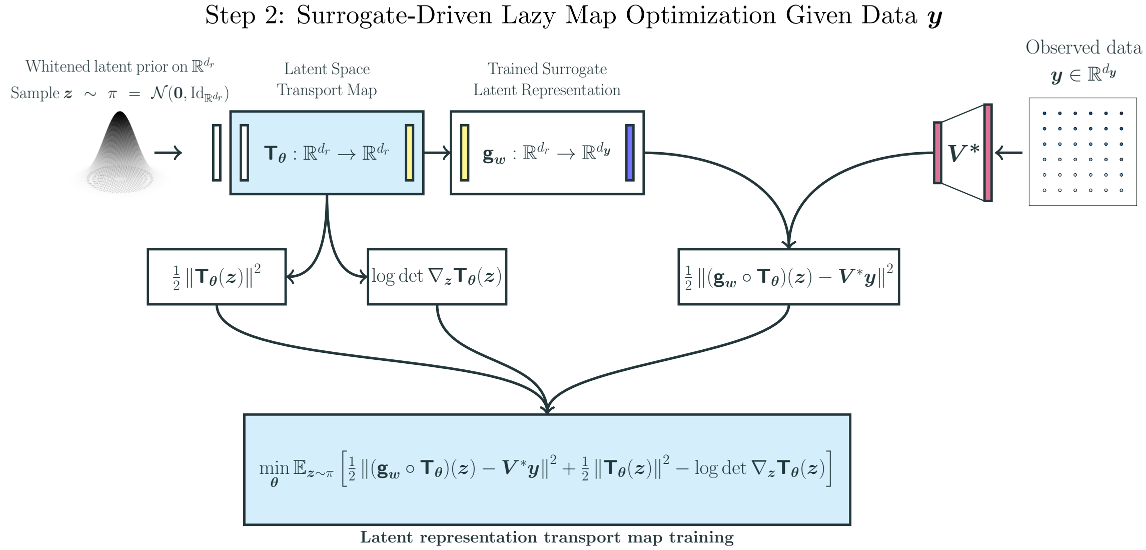
Step 2: Surrogate-Driven Lazy Map Optimization Given Data $\boldsymbol{y}$
Whitened latent prior on $\mathbb{R}^{d_r}$
Sample $\boldsymbol{z} \sim \pi = \mathcal{N}(\mathbf{0}, \mathrm{Id}_{\mathbb{R}^{d_r}})$
Latent Space
Transport Map
$\mathbf{T}_{\boldsymbol{\theta}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_r}$
Trained Surrogate
Latent Representation
$\mathbf{g}_{\boldsymbol{w}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$
Observed data
$\boldsymbol{y} \in \mathbb{R}^{d_{\boldsymbol{y}}}$
$\boldsymbol{V}^*$
$\frac{1}{2}\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2$
$\log\det\nabla_{\boldsymbol{z}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})$
$\frac{1}{2}\|(\mathbf{g}_{\boldsymbol{w}} \circ \mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) - \boldsymbol{V}^*\boldsymbol{y}\|^2$
$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\frac{1}{2}\|(\mathbf{g}_{\boldsymbol{w}} \circ \mathbf{T}_{\boldsymbol{\theta}})(\boldsymbol{z}) - \boldsymbol{V}^*\boldsymbol{y}\|^2 + \frac{1}{2}\|\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2 - \log\det\nabla_{\boldsymbol{z}}\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$
Latent representation transport map training

Figure 4: Overview of the online latent representation lazy map training.

**LazyDINO vs. A-SBI** Given joint samples of the latent prior and simulated observational data, the A-SBI methods optimize for a conditional transport map that matches the pullback distributions, $\boldsymbol{y}^{(j)} \mapsto \mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{y}^{(j)}, \cdot)^\sharp\pi$, to posteriors at the simulated data samples using an fKL objective. Sample generation and the construction of the transport map are both performed offline. When the observational data $\boldsymbol{y}^\dagger$ is available, the approximate posterior sampling is performed using inversion at latent prior samples $\mathbf{T}_{\boldsymbol{\theta}}(\boldsymbol{y}^\dagger, \cdot)^{-1}(\boldsymbol{z}^{(j)})$, $\boldsymbol{z}^{(j)} \overset{\text{iid}}{\sim} \pi$, without the computational bottleneck of model simulations, making it an amortized inference method. Transport map construction in A-SBI incurs a similar cost compared to RB-NO surrogate construction, requiring PtO map samples with noise perturbation, i.e., $\boldsymbol{y}^{(j)} := \boldsymbol{\mathcal{G}}(m^{(j)}) + \boldsymbol{n}^{(j)}$ with $m^{(j)} \overset{\text{iid}}{\sim} \mu$ and $\boldsymbol{n} \overset{\text{iid}}{\sim} \mathcal{N}(0, \Gamma_n)$. The training also only requires quantities related to the latent prior that are easy to evaluate.

Despite ostensibly relaxed training requirements, A-SBI has a much lower sample efficiency than LazyDINO. Given limited PtO map samples, A-SBI attempts to directly approximate *all posteriors* (i.e., posteriors for all instances of observational data). In contrast, LazyDINO invests these samples in surrogate construction to gain almost unlimited access to all *surrogate-approximated posteriors*. As a result, LazyDINO leads to a much smaller transport map approximation error than A-SBI. Furthermore, due to the high efficiency of RB-DINO, the transport map approximation error of A-SBI is much higher than the surrogate posterior approximation error in LazyDINO, leading to more than two orders of magnitude lower sample efficiency of A-SBI observed in our numerical results in Section 6.

All A-SBI optimization problems are solved offline, which is often regarded as an advantage. In contrast, LazyDINO requires solving an optimization problem for online posterior sampling at each instance of observational data. However, for popular transport map parametrizations such as conditional normalizing flows, sampling requires solving a root-finding problem in $\mathbb{R}^{d_r}$ for map inversion, which incurs a non-negligible cost in practice. The inversion-to-sample approach of A-SBI can be more costly than the optimize-to-sample ap-

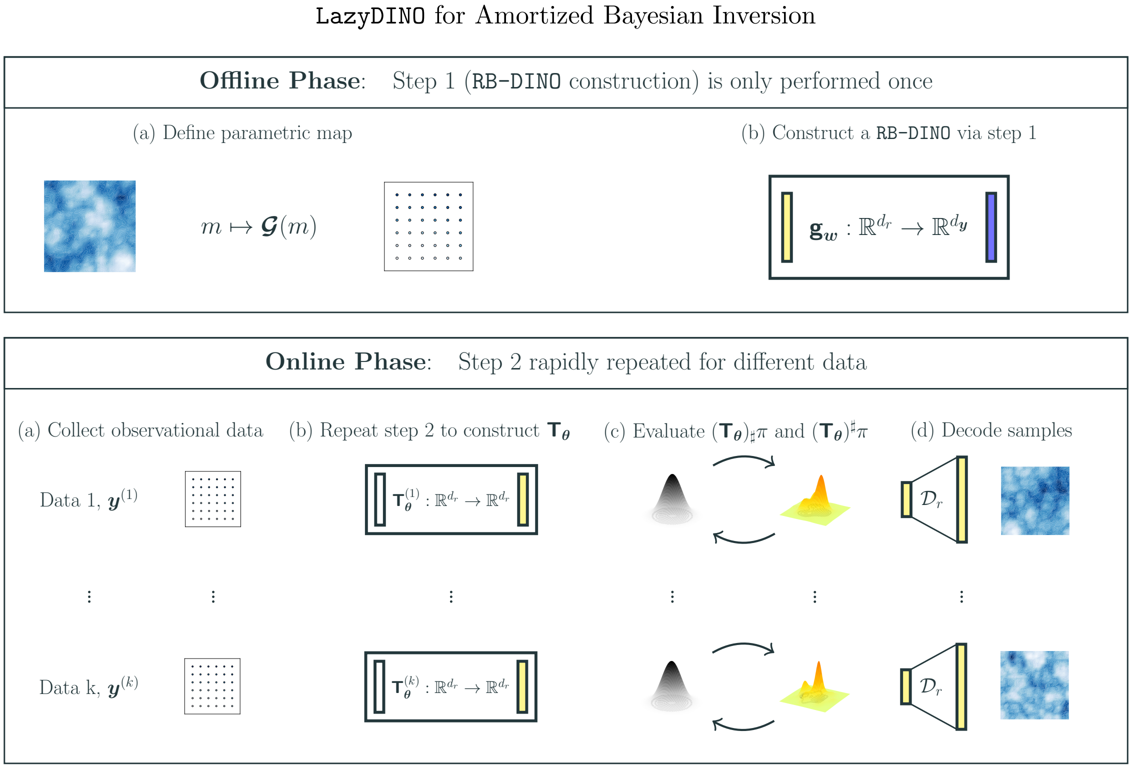
LazyDINO for Amortized Bayesian Inversion
Offline Phase: Step 1 (RB-DINO construction) is only performed once
(a) Define parametric map
$m \mapsto \boldsymbol{\mathcal{G}}(m)$
(b) Construct a RB-DINO via step 1
$\mathbf{g}_{\boldsymbol{w}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_{\boldsymbol{y}}}$
Online Phase: Step 2 rapidly repeated for different data
(a) Collect observational data
(b) Repeat step 2 to construct $\mathbf{T}_{\boldsymbol{\theta}}$
(c) Evaluate $(\mathbf{T}_{\boldsymbol{\theta}})_\sharp \pi$ and $(\mathbf{T}_{\boldsymbol{\theta}})^\sharp \pi$
(d) Decode samples
Data 1, $\boldsymbol{y}^{(1)}$
$\mathbf{T}_{\boldsymbol{\theta}}^{(1)} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_r}$
$\mathcal{D}_r$
Data k, $\boldsymbol{y}^{(k)}$
$\mathbf{T}_{\boldsymbol{\theta}}^{(k)} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_r}$
$\mathcal{D}_r$

Figure 5: Overview of LazyDINO amortization procedure.

proach of LazyDINO in some situations, e.g., when a large number of approximate posterior samples is needed for each instance of observational data. We provide concrete numerical evidence to support these claims in Section 6.3.

## 5. Setup of the Numerical Studies

In this section, we describe the setup of the numerical examples, including the two BIP examples, the neural network architectures, the training procedures, and the posterior error measures. We first define the two PDE problems and their associated inverse problems in Sections 5.1 and 5.2. Then, we define the architecture and training of the surrogate and transport map in Sections 5.3 and 5.4, respectively. Then, we introduce the error measures for the surrogate and posterior approximations in Section 5.5.

For the BIP examples, we consider 2D nonlinear elliptic PDEs with $\Omega \subset \mathbb{R}^2$. The Gaussian priors are defined using Matérn covariance operators given by:

$$\mathcal{C} = (-\gamma \nabla \cdot (\boldsymbol{A} \nabla) + \delta \mathrm{Id}_{\mathscr{M}})^{-2}. \tag{46}$$

Here $\gamma, \delta > 0$ are scalar parameters that control the marginal variance and correlation lengths of the random field samples, and $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite matrix that induces anisotropy in the random field samples. In both cases, we employ Robin boundary conditions to control boundary artifacts in the samples; see (Villa et al., 2021,

| | Method<br>Feature | MCMC (Truth) | LA-Baseline | LazyMap | A-SBI | LazyNO | LazyDINO |
|---|---|---|---|---|---|---|---|
| Offline | Parallel training sample generation | – | – | – | $\boldsymbol{z}^{(j)}$,<br>$\boldsymbol{y}^{(j)}$ | $\boldsymbol{z}^{(j)}$,<br>$\boldsymbol{g}^{(j)}$ | $\boldsymbol{z}^{(j)}$,<br>$\boldsymbol{g}^{(j)}$,<br>$\boldsymbol{J}_r^{(j)}$ |
| | Construct surrogate using training samples | – | – | – | ✗ | $L^2_\mu$ training | $H^1_\mu$ training |
| | Posterior approximation using training samples | – | – | – | ✓ | ✗ | ✗ |
| Online Cost | Full amortization: No training, no model solution, and no surrogate evaluation | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| | Bottleneck amortization: No model solution | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| | Parallel efficiency: Vectorized explicit surrogate evaluations | – | – | – | – | ✓ | ✓ |
| | Parallel efficiency: Embarrassingly parallel model solutions | ✗ | ✗ | ✓ | – | – | – |
| Online Sampling | Direct sampling using trained transport map with no map inversion required | – | – | ✓ | ✗ | ✓ | ✓ |
| | Fully parallelizable sampling | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 2: **Feature Comparison for Posterior Approximation Methods**. We emphasize that both LazyNO and LazyDINO train transport maps using the surrogate rKL objective online. In contrast, A-SBI trains transport maps using the fKL objective offline.

Equation 37) and (Villa and O'Leary-Roseberry, 2024). In both cases, the parameter space $\mathscr{M}$ is approximated using linear triangular finite elements, while the state space $\mathscr{U} \subset H^1(\Omega;\mathbb{R})$ or $H^1(\Omega;\mathbb{R}^2)$ is approximated using quadratic triangular finite elements. The Newton-Raphson method is used for nonlinear solves, and a sparse direct solver is used for the linear solve in each Newton iteration. The PtO map $\boldsymbol{\mathcal{G}}$ is defined by composing the PDE solution operator $\mathcal{F} : \mathscr{M} \to \mathscr{U}$ and an observation operator $\boldsymbol{\mathcal{O}} : \mathscr{U} \to \mathbb{R}^{d_{\boldsymbol{y}}}$. The inverse problems arise from generating four synthetic observations: each is obtained by sampling the prior, evaluating the PtO map, and adding additive white noise.

### 5.1 Example I: Inference of Diffusivity in a Nonlinear Reaction–Diffusion PDE

For our first example, we consider the following nonlinear reaction–diffusion PDE for $u : \Omega = [0,1]^2 \to \mathbb{R}$:

$$\begin{aligned} -\nabla_{\boldsymbol{s}} \cdot \exp(m(\boldsymbol{s}))\nabla u(\boldsymbol{s}) + u(\boldsymbol{s})^3 &= 0, \quad \boldsymbol{s} \in (0,1)^2, && \text{(47a)}\\ \exp(m(\boldsymbol{s}))\nabla u(\boldsymbol{s}) \cdot \boldsymbol{n} &= 0, \quad \boldsymbol{s} \in \partial\Omega_{\text{left}} \cup \partial\Omega_{\text{right}}, && \text{(47b)}\\ u(\boldsymbol{s}) &= 1, \quad \boldsymbol{s} \in \partial\Omega_{\text{top}}, && \text{(47c)}\\ u(\boldsymbol{s}) &= 0, \quad \boldsymbol{s} \in \partial\Omega_{\text{bottom}}, && \text{(47d)} \end{aligned}$$

where $\partial\Omega_{\text{left}}$, $\partial\Omega_{\text{right}}$, $\partial\Omega_{\text{top}}$, and $\partial\Omega_{\text{bottom}}$ denote the left, right, top and bottom boundaries of the unit square, and $\boldsymbol{n}$ is the outward unit normal vector. The inverse problem is to find the log-diffusivity field $m : (0,1)^2 \to \mathbb{R}$ that best matches noisy observations of $u$ at a set of spatial positions.

We use a regular grid with $40 \times 40$ cells, which yields 1,681 and 3,362 degrees of freedom (DoFs) for discretized parameters and states, respectively. We choose $\boldsymbol{A} = \mathrm{Id}_{\mathbb{R}^2}$, $\gamma = 0.03$, and $\delta = 3.33$ for the prior, which leads to a pointwise marginal variance around 9 and a spatial correlation length of around 0.1. The observation operator uses $d_{\boldsymbol{y}} = 25$ randomly sampled interior points and produces the locally averaged state values around them. The noise distribution has covariance $\Gamma_n = 1.94 \times 10^{-3}\mathrm{Id}_{\mathbb{R}^{d_{\boldsymbol{y}}}}$, which corresponds to a signal-to-noise ratio of around 500. The synthetic data set is visualized in Figure 6.

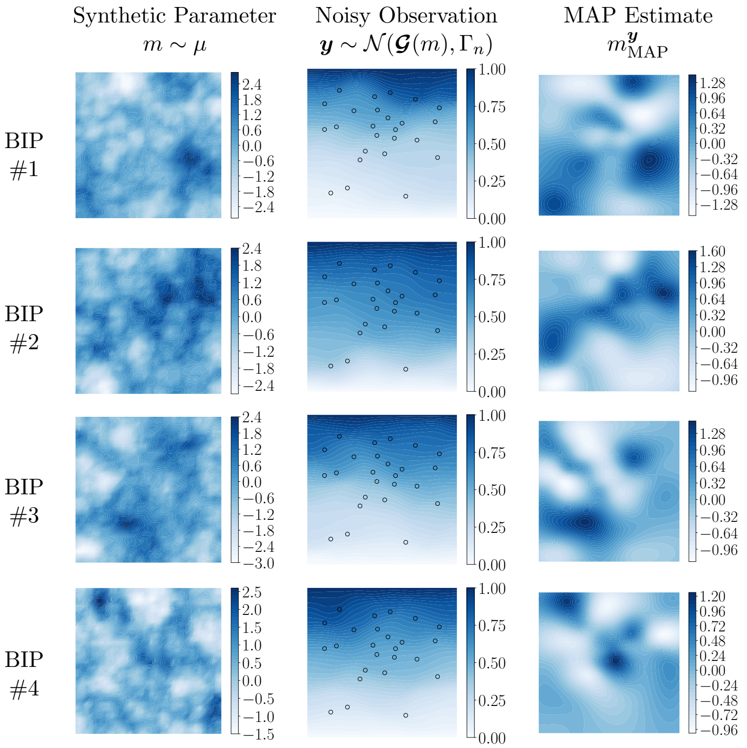

Figure 6: **Example I.** Setup for inferring the diffusivity field in a nonlinear reaction–diffusion PDE detailed in Section 5.1. For each BIP (#1–4), we show the data-generating synthetic parameter $m$ drawn from the prior, the synthetic data $\boldsymbol{y}$ placed on top of the PDE solution at $m$, and the MAP estimate $m^{\boldsymbol{y}}_{\text{MAP}}$.

### 5.2 Example II: Inference of a Heterogeneous Hyperelastic Material Property

For our second example, we consider the uniaxial tensile test of a hyperelastic thin film. The inverse problem aims to recover the Young's modulus field, which characterizes spatially varying material strength, from measurements of the material deformation.

Let $\Omega = (0,2) \times (0,1)$ be a normalized material domain. The material coordinates $\boldsymbol{s} \in \Omega$ of the reference configuration are mapped to the spatial coordinates $\boldsymbol{s} + \boldsymbol{u}(\boldsymbol{s})$ of the deformed configuration, where $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ is the material displacement. The strain energy of the hyperelastic material $\mathcal{W}_e$ depends on the deformation gradient $\boldsymbol{F} = \mathrm{Id}_{\mathbb{R}^{2\times 2}} + \nabla \boldsymbol{u}$. We consider the neo-Hookean model for the strain energy density:

$$\mathcal{W}_e(\boldsymbol{F}) = \frac{\mu_e}{2}(\mathrm{tr}(\boldsymbol{F}^\top \boldsymbol{F}) - 3) + \frac{\lambda_e}{2}(\ln\det(\boldsymbol{F}))^2 - \mu_e \ln\det(\boldsymbol{F}). \tag{48}$$

Here, $\lambda_e$ and $\mu_e$ are the Lamé parameters, and they are related to Young's modulus $E_Y$ and Poisson ratio $\nu_P$ under the plane strain assumption:

$$\lambda_e = \frac{E_Y \nu_P}{(1+\nu_P)(1-2\nu_P)}\,, \qquad \mu_e = \frac{E_Y}{2(1+\nu_P)}. \tag{49}$$

We assume $\nu_P = 0.4$, and a spatially-varying normalized Young's modulus, $E_Y : \Omega \to (E_{Y\min}, E_{Y\max})$. We represent $E_Y$ through a parameter field $m : \Omega \to \mathbb{R}$ as follows

$$E_Y(m(\boldsymbol{s})) = \frac{1}{2}\left(E_{Y\max} - E_{Y\min}\right)\left(\mathrm{erf}(m(\boldsymbol{s})) + 1\right) + E_{Y\min},$$

where $\mathrm{erf} : \mathbb{R} \to (-1,1)$ is the error function and take $E_{Y\min} = 1$ and $E_{Y\max} = 7$. The first Piola–Kirchhoff stress tensor is given by $\boldsymbol{P}_e(m, \boldsymbol{F}) = 2\partial\mathcal{W}_e(m,\boldsymbol{F})/\partial\boldsymbol{F}$. Assuming a quasi-static model with negligible body forces, the balance of linear momentum leads to the following nonlinear PDE:

$$\nabla_{\boldsymbol{s}} \cdot \boldsymbol{P}_e(m, \boldsymbol{F})(\boldsymbol{s}) = \boldsymbol{0}, \qquad \boldsymbol{s} \in \Omega; \tag{50a}$$

$$\boldsymbol{u}(\boldsymbol{s}) = \boldsymbol{0}\,, \qquad \boldsymbol{s} \in \partial\Omega_{\text{left}}; \tag{50b}$$

$$\boldsymbol{u}(\boldsymbol{s}) = 3/2\,, \qquad \boldsymbol{s} \in \partial\Omega_{\text{right}}; \tag{50c}$$

$$\boldsymbol{P}_e(m, \boldsymbol{F})(\boldsymbol{s}) \cdot \boldsymbol{n} = \boldsymbol{0}, \qquad \boldsymbol{s} \in \partial\Omega_{\text{top}} \cup \partial\Omega_{\text{bottom}}; \tag{50d}$$

where $\partial\Omega_{\text{top}}$, $\partial\Omega_{\text{right}}$, $\partial\Omega_{\text{bottom}}$, and $\partial\Omega_{\text{left}}$ denote the material domain's top, right, bottom, and left boundary.

We use a regular grid with 64 × 32 cells, which leads to 2,145 and 16,770 DoFs for the discretized parameter and state, respectively. For the prior covariance in (46), we induce spatial anisotropy via the following matrix

$$\boldsymbol{A} = \begin{bmatrix} \theta_1 \sin(\alpha)^2 + \theta_2 \cos(\alpha)^2 & (\theta_1 - \theta_2)\sin(\alpha)\cos(\alpha) \\ (\theta_1 - \theta_2)\sin(\alpha)\cos(\alpha) & \theta_1\cos(\alpha)^2 + \theta_2 \sin(\alpha)^2 \end{bmatrix},$$

where $\theta_1 = 2$ and $\theta_2 = 1/2$, $\alpha = \arctan(2)$. We choose $\gamma = 0.3$ and $\delta = 3.3$, which lead to a pointwise marginal variance of around 1 and spatial correlations of around 2 and 0.5 perpendicular to and along the left-to-right, bottom-to-top diagonal of the material domain, respectively. We define the observation operator using 32 equally spaced interior points and take the locally averaged state values around those points as observations. This leads to $d_{\boldsymbol{y}} = 64$. The noise distribution has covariance $\Gamma_n = 2.86 \times 10^{-3}\mathrm{Id}_{d_{\boldsymbol{y}}}$, which corresponds to a signal-to-noise ratio of around 500. We visualize the synthetic data set in Figure 7.

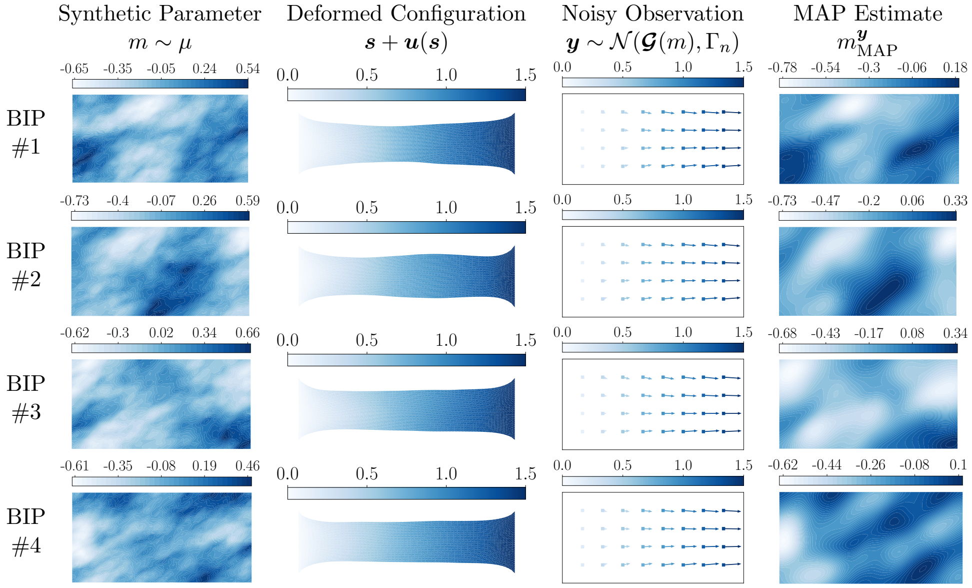

Figure 7: **Example II.** Setup for inferring a heterogeneous hyperelastic material property detailed in Section 5.2. For each BIP (#1–4), we visualize the synthetic parameter $m$ drawn from the prior, the corresponding deformed configuration, the synthetic displacement data $\boldsymbol{y}$, and the MAP estimate $m^{\boldsymbol{y}}_{\mathrm{MAP}}$.

### 5.3 Surrogate Architecture and Training

**Reduced Basis** For both problems, we chose the dimension of the parameter latent space to be $d_r = 200$ and used 1000 MC samples to compute the reduced basis. We fixed the reduced-basis dimension for all the studied variational inference methods. Our numerical examples are focused on comparing our proposed `LazyDINO` with other variational inference methods. Therefore, the effects of varying $d_r$ and MC sample sizes for reduced basis construction are not studied in this work. The eigenvalue decay and basis functions for both examples are visualized in Appendix I.

**Architecture and Training** For both examples, we choose a multi-layer perceptron (MLP) as $\mathbf{g}_w$ with 7 hidden layers, each with a width of 400 and a Gaussian error linear unit (GELU) activation function. The loss is minimized using Adam. We found that many other training tricks, such as batch normalization or learning rate decay scheduling, were not necessary to produce good generalization for `RB-DINO`. For conventional $L^2_\mu$ training, we find that learning rate decay scheduling and adaptive total epoch sizes are needed to maintain training stability and prevent overfitting; see Appendix H for details.

### 5.4 Transport Map Architecture and Training

**Architecture** We chose inverse autoregressive flow (Kingma et al., 2017) as our latent space transport map, with 30 transport map layers and random input permutation. Each transport map layer is an autoregressive MLP, with inputs masked to ensure triangular dependence. Each MLP has 4 hidden layers, each with a width of 400 and GELU activation.

For A-SBI, the conditional normalizing flow is constructed as a masked autoregressive flow with the same architecture, but with a larger input dimension to account for conditioning on observations. The tanh activation is used as it empirically leads to stable training.

**Training** The training procedures for all methods are kept as similar as possible. For `LazyDINO` and `LazyNO`, the transport map is trained using Adamax (Kingma and Ba, 2017) with a scheduled batch size and learning rate and with sample replacement. In particular, the batch size increases and the learning rate decreases with the number of iterations. For `LazyMap`, the batch size is fixed with sample replacement. Note that each optimization iteration involves evaluating the PtO map and its Jacobian, which we refer to as training samples in our results. For our numerical examples, we use the checkpoints at different iterations to examine `LazyMap` at different training sample sizes. For A-SBI, the batch size is fixed without sample replacement, and the learning rate is fixed. Optimization terminates when the validation error stagnates. See details of the training procedures in Appendix H.

### 5.5 Error Measures

**Surrogate Approximation Error** The approximation errors of the neural ridge function surrogate $\widetilde{\boldsymbol{\mathcal{G}}}_{\boldsymbol{w}} \circ \mathcal{P}$ for the PtO map approximation, $\boldsymbol{E}_{\mathbf{g}}$, and the PtO map latent Jacobian approximation, $\boldsymbol{E}_{\nabla\mathbf{g}}$, are defined as follows.

$$\boldsymbol{E}_{\mathbf{g}} = \sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\left[\frac{\|\mathbf{g}_w(\boldsymbol{z}^{(j)}) - \boldsymbol{g}^{(j)}\|^2}{\|\boldsymbol{g}^{(j)}\|^2}\right]} \qquad \text{(Relative PtO Map Error)}$$

$$\boldsymbol{E}_{\nabla\mathbf{g}} = \sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{j=1}^{N_{\mathrm{MC}}}\left[\frac{\|\boldsymbol{J}_r^{(j)} - \nabla\mathbf{g}_w(\boldsymbol{z}^{(j)})\|_F^2}{\|\boldsymbol{J}_r^{(j)}\|_F^2}\right]} \qquad \text{(Relative Latent Jacobian Error)}$$

where the whitened PtO samples $\boldsymbol{g}^{(j)}$ and the whitened latent Jacobian $\boldsymbol{J}_r^{(j)}$ are defined in (40) and (44), respectively. We compute the errors using $N_{\mathrm{MC}}$ =5k samples of the prior, whitened PtO evaluations, and its latent Jacobian evaluations.

**Posterior Approximation Error** Posterior approximation accuracy can be assessed in many ways. It is important to assess posterior accuracy in high-probability regions using moment and mode estimates. Additionally, probability divergences measure the overall deviation from the posterior. For our numerical examples, we assess the quality of the posterior approximation under a nonlinear transport map $\mathcal{T}$ using moment discrepancies and density-based diagnostics, as described below.

*Moment Discrepancies* Let $\overline{m}^{\boldsymbol{y}} \in \mathscr{M}$, $\mathcal{C}^{\boldsymbol{y}} \in \mathrm{HS}(\mathscr{M})$ denote the mean and covariance of $\mu^{\boldsymbol{y}}$ and $\mathcal{S}_{25}^{\boldsymbol{y}} \in \mathbb{R}^{25\times25\times25}$ denote the skewness of $\mu^{\boldsymbol{y}}$ in the leading 25 latent space coordinates. Let $\overline{m}^{\mathcal{T}}$, $\mathcal{C}^{\mathcal{T}}$, and $\mathcal{S}_{25}^{\mathcal{T}}$ denote the same quantities for $\mathcal{T}_\sharp\mu$. We consider the following relative error in the moments

$$\begin{aligned}
\boldsymbol{E}_{\mathrm{mean}} &= \left\|\overline{m}^{\boldsymbol{y}} - \overline{m}^{\mathcal{T}}\right\|_{\mathscr{M}} / \left\|\overline{m}^{\boldsymbol{y}}\right\|_{\mathscr{M}} && \text{(Relative Mean Error)}\\
\boldsymbol{E}_{\mathrm{cov}} &= \left\|\mathcal{C}^{\boldsymbol{y}} - \mathcal{C}^{\mathcal{T}}\right\|_{\mathrm{HS}(\mathscr{M})} / \left\|\mathcal{C}^{\boldsymbol{y}}\right\|_{\mathrm{HS}(\mathscr{M})} && \text{(Relative Covariance Error)}\\
\boldsymbol{E}_{\mathrm{skew}} &= \left\|\mathcal{S}_{25}^{\boldsymbol{y}} - \mathcal{S}_{25}^{\mathcal{T}}\right\|_F / \left\|\mathcal{S}_{25}^{\boldsymbol{y}}\right\|_F && \text{(Relative Skewness Error)}
\end{aligned}$$

The central moments of both $\mu^{\boldsymbol{y}}$ and $\mathcal{T}_\sharp\mu$ are estimated using samples, where samples from $\mu^{\boldsymbol{y}}$ are obtained using up to $5 \times 10^6$ MCMC samples using a simplified manifold MCMC method (Beskos et al., 2017). The discrepancies are reported as percentages.

Since all central moments must converge as a posterior estimator converges to the posterior, analyzing moment discrepancies of varying orders together is more helpful than analyzing them independently. Estimating higher-order statistics becomes progressively more challenging, so we consider only the first three moments and compute the skewness in a low-dimensional parameter subspace.

*Density-Based Diagnostics* Let $\Phi_{\mathcal{T}}(m) \coloneqq \log\left(\frac{\mathrm{d}\mu}{\mathrm{d}(\mathcal{T}_\sharp\mu)}(m)\right)$ denote the log density of the prior with respect to the pushforward distribution; see Appendix G for explicit forms. Let $\widetilde{w}(m) = \exp(-2\Phi^{\boldsymbol{y}}(m) + 2\Phi_{\mathcal{T}}(m))$ denote the unnormalized density of the posterior with respect to the pushforward density and $w(m)$ denote its normalization by $\mathbb{E}_{m\sim\mathcal{T}_\sharp\mu}[\widetilde{w}(m)]$. We consider the following quantities related to the quality of each posterior approximation:

$$\begin{aligned}
\boldsymbol{E}_{\text{fKL}} &= \mathbb{E}_{m\sim\mathcal{T}_\sharp\mu}\left[\Phi^{\boldsymbol{y}}(m) - \Phi_{\mathcal{T}}(m)\right] + C_1 && \text{(Shifted rKL Divergence)}\\
\boldsymbol{E}_{\text{rKL}} &= \mathbb{E}_{m\sim\mathcal{T}_\sharp\mu}\left[w(m)(-\Phi^{\boldsymbol{y}}(m) + \Phi_{\mathcal{T}}(m))\right] + C_2 && \text{(Shifted ANIS fKL Diverg.)}\\
\text{ESS}_N\% &= \frac{\left(\frac{1}{N}\sum_{j=1}^N \widetilde{w}(m^{(j)})\right)^2}{\frac{1}{N}\sum_{j=1}^N \widetilde{w}(m^{(j)})^2} \times \frac{100\%}{N},\ m^{(j)} \overset{\text{iid}}{\sim} \mathcal{T}_\sharp\mu && \text{(ANIS Effective Sample Per.)}\\
\boldsymbol{E}_{\text{MAP}} &= \left\|m^{\boldsymbol{y}}_{\text{MAP}} - m^{\mathcal{T}_{\text{MAP}}}\right\|_{\mathscr{M}} / \left\|m^{\boldsymbol{y}}_{\text{MAP}}\right\|_{\mathscr{M}} && \text{(Relative MAP Point Error)}
\end{aligned}$$

We use both rKL and fKL to measure the posterior approximation error. The former measures the optimality gap due to surrogate approximation error in `LazyNO` and `LazyDINO`. On the other hand, rKL can be small when the pushforward is overly concentrated; thus, we also report the fKL. The auto-normalized importance sampling weights (Agapiou et al., 2017) are used to compute a biased but consistent estimator for the fKL. We note that rKL is shifted by the normalization constant, and both rKL and fKL are additionally shifted by a constant for visualizations in the logarithmic scale. We use $10^5$ iid samples from the pushforward to estimate the expectations in the rKL and fKL.

We also consider the effective sample size (ESS) percentage estimated using $N$ iid samples from the pushforward distribution, denoted $\text{ESS}_N\%$. We take $N = 10^5$ in our numerical studies. This is a commonly used diagnostic to assess the quality of approximate posterior sampling, and many numerical methods struggle to achieve large effective sample sizes (Polo and Vicente, 2020). This diagnostics is related to the forward $\chi^2$ divergence by $\chi^2(\mu^{\boldsymbol{y}}||\mathcal{T}_\sharp\mu) \approx 1 - \text{ESS}_N\%/100$; see Sanz-Alonso and Wang (2020).

## 6. Numerical Results

In this section, we present numerical results for the two BIP problems in Sections 5.1 and 5.2. We compare the posterior errors for `LazyDINO`, `LazyNO`, `LazyMap`, A-SBI, and the Laplace approximation (LA) as a baseline. First, we compare the cost-efficiency `RB-DINO` and `RB-NO` surrogate construction in Section 6.1. Then, we provide comparisons of the posterior approximation quality in Section 6.2. A timing comparison for transport map training and sampling is provided in Section 6.3. Lastly, we compare the posterior marginals obtained by LazyDINO and other methods using corner plots. In Appendix I, we include additional visualizations of the posterior mean, MAP, posterior pointwise variance, and expanded corner plots to support the results in this section.

**Remark 8** *When reporting the surrogate training results, we compare computational costs measured as the number of nonlinear PDE solves. The additional costs associated with Jacobian data generation are measured relative to the nonlinear PDE solves. Since these additional costs are negligible due to the use of sparse direct solvers (Cao et al., 2025, Section 4.3), this point of comparison is not considered for the posterior approximation error.*

## 6.1 Neural Ridge Function Generalization

We begin by assessing the results of RB-DINO and RB-NO surrogate construction. The results for Example I can be found in Figure 8, while the results for Example II can be found in Figure 9. Overall, the trend demonstrates that the derivative-informed learning method leads to a significant cost reduction in learning the latent representation of the PtO map and its Jacobian compared to conventional supervised learning. These results are consistent with those of Cao et al. (2025). We expect that the improvement of the PtO map approximation leads to better fidelity in the surrogate approximated posterior through Theorem 4, and the combination of improved PtO map approximation and Jacobian approximation leads to more accurate transport map training through Corollary 6.

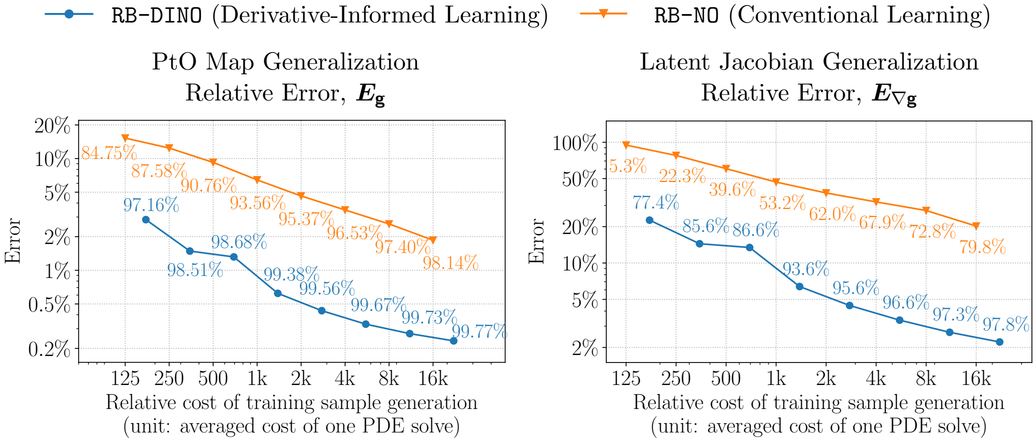

Figure 8: **Example I: Neural Ridge Function Generalization.** The surrogate error versus training sample generation costs is visualized. The percentage accuracy, 100% × (1 − error), is overlaid. The results demonstrate that RB-DINO is 64× more cost-efficient than RB-NO. We note a statistical anomaly of RB-DINO at 500 samples.

## 6.2 Posterior Approximation Error

We investigate the posterior error metrics described in Section 5.5. For all plots in this section, the markers for BIP #1–4 are labeled as ⋎ , ⋏ , ≺ , and ≻. The average error is plotted in a darker color, and a line is drawn between each average to visualize the trend.

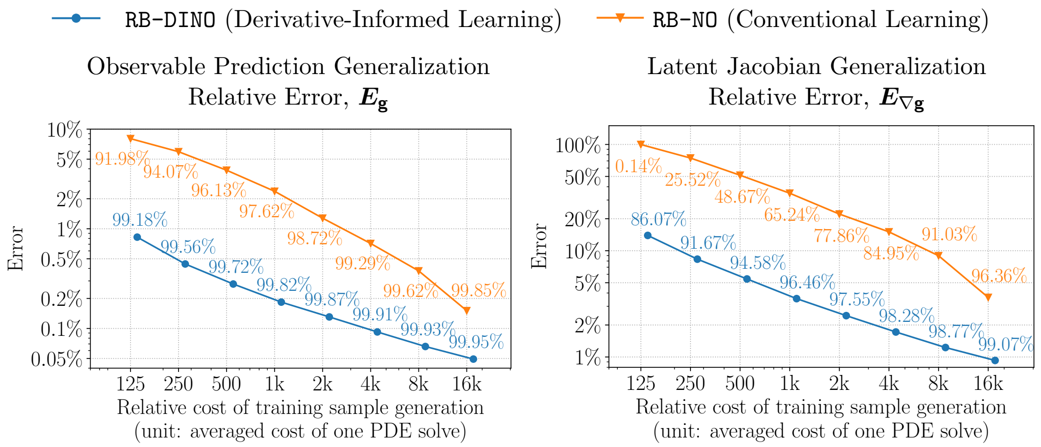

Figure 9: **Example II: Neural Ridge Function Generalization.** In a similar pattern as Figure 8, RB-DINO enjoys 8–32× higher cost-efficiency over RB-NO.

We cut off vertical-axis errors at 200–300%, depending on the plot, for readability since the scale of the errors across methods varies widely. The LA is computed once per BIP, and the horizontal lines labeled as LA are included only to facilitate visual comparison. A conservative estimate of the cost of LA is similar to that for 100 training samples.

We begin by comparing moment discrepancies for Example I and II in Figure 10 and Figure 11, respectively. In general, the LA provided a reasonable baseline point of comparison. In Example I, it consistently outperformed all other methods in terms of covariance error. In all other cases, however, LazyDINO provides higher-quality moment predictions than LA, particularly with plentiful training samples. Compared to the remaining methods, LazyDINO leads to the best moment matching across the board. LazyNO typically is the second best, though in some cases A-SBI or LazyMap performed comparably or slightly better. A-SBI and LazyMap are more than an order of magnitude worse than LazyDINO in all metrics. The poor performance of LazyMap can be explained by the sample intensity required for reliable transport map training. In particular, LazyMap here has a limited budget of up to 128k high-fidelity samples, whereas LazyDINO and LazyNO required 16 million total surrogate-generated samples across all iterations. This point of comparison demonstrates the essential benefit of the LazyDINO approach—by first building a reliable surrogate PtO map over the prior using a fixed number of samples, we can later enable an optimization algorithm requiring orders of magnitude more samples.

In Figures 12 and 13, we compare the performance of the different methods through the density-based diagnostics defined in Section 5.5. As with the moment discrepancy results, we observe consistent superior performance of LazyDINO compared to the other methods. Notably, LazyDINO yields an effective sample size that is orders of magnitude higher than those of the other methods in the largest sample case for both examples.

### 6.3 Timing Comparisons

So far, we compared various methods based on sample complexity. In this subsection, we consider time-complexity comparisons. We begin by comparing LazyDINO with LazyMap,

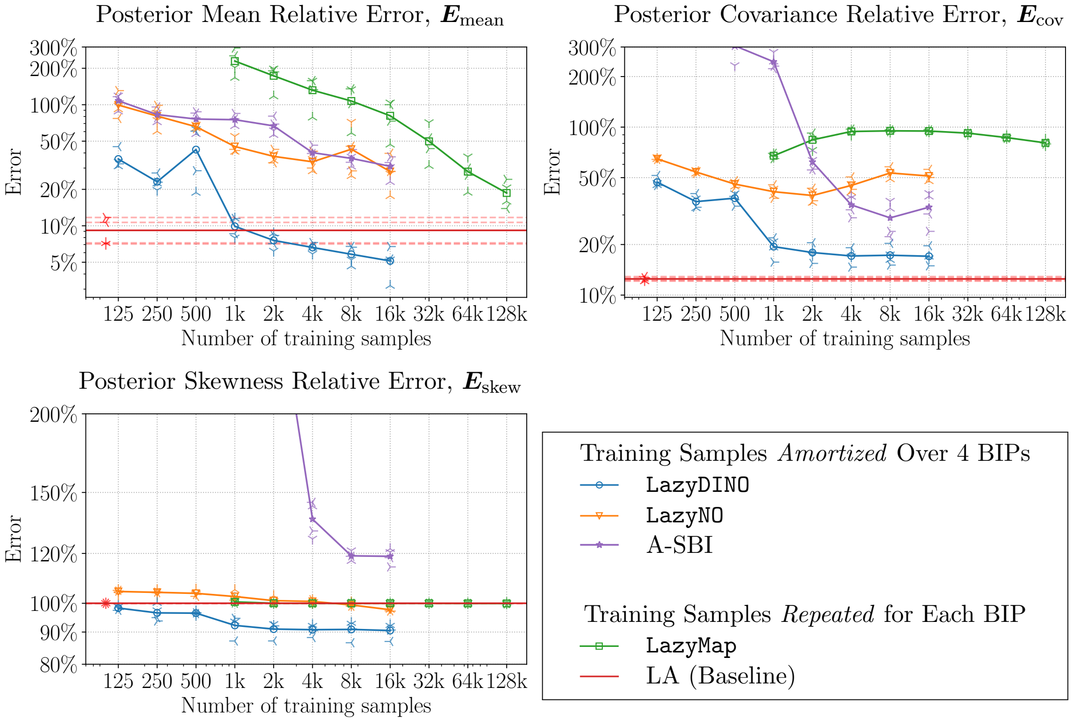

Figure 10: **Example I: Moment Discrepancies.** (`LazyDINO` vs. `LazyNO`): Apart from the statistical anomaly at 500 samples, `LazyDINO` is 64× more sample-efficient measured in mean error, 4× in covariance error, and 64× in skewness error. The discrepancy is even more pronounced beyond 500 samples. (`LazyDINO` vs. A-SBI): `LazyDINO` is 64× more sample-efficient measured in mean error, and A-SBI is uncompetitive in covariance and skewness error. (`LazyDINO` vs. `LazyMap`): Even when discounting the repeated model solution costs of `LazyMap` over different BIPs, `LazyDINO` still achieves orders of magnitude higher sample-efficiency. (`LazyDINO` vs. LA): `LazyDINO` achieves lower error in mean and skewness, particularly in the large-sample regime.

recognizing that the latter can be made more efficient with parallelism. Then, we compare the total online cost of A-SBI and `LazyDINO`, demonstrating that, in addition to being much more accurate than A-SBI, `LazyDINO` has a lower overall cost for amortized Bayesian inversion. All GPU computations were performed on Nvidia A100 GPUs, and all CPU computations were performed on Intel Xeon Gold 6248R 3.00GHz CPUs.

**`LazyMap` vs. `LazyDINO`** The efficiency of `LazyDINO` is affected by its offline phase. In contrast, `LazyMap` does not have an offline phase and thus might be competitive in posterior approximation if it exploits parallelism within each transport map training iteration. Here, we compare the performance of `LazyMap` and `LazyDINO` under a similar total computational budget, while allowing parallel evaluation of the PtO map and its Jacobian.

In Tables 3 and 4, we provide a comparison of solution time and posterior mean accuracy for the two methods, given access to 20 concurrent CPU evaluations of the PtO map and its Jacobian action. We provide the times for training sample generation for `LazyDINO` and

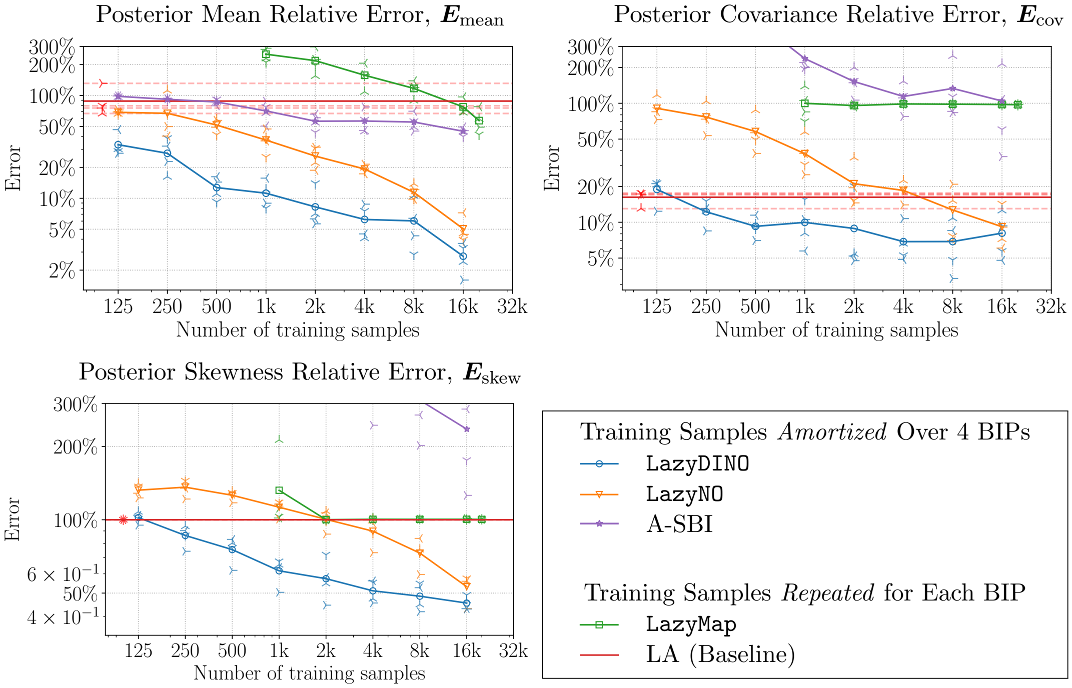

Figure 11: **Example II: Moment Discrepancies.** Similar trends as Example I in Figure 10 are observed. (`LazyDINO` vs. `LazyNO`): `LazyDINO` consistently outperforms by 2–64× in sample-efficiency. (`LazyDINO` vs. A-SBI): The best-performing A-SBI at 16k samples is still less accurate compared to `LazyDINO` at 125 samples. (`LazyDINO` vs. `LazyMap`): `LazyMap` exhausts the training budget before performing comparably to `LazyDINO`. (`LazyDINO` vs. LA): At 250 samples, `LazyDINO` has lower errors than LA.

the total transport map training times for `LazyDINO` and `LazyMap`. The reported times of `LazyDINO` offline phase are amortized across four instances of observational data. All computations repeated for each instance are reported as averages, rounded to the nearest 10 seconds. Each iteration of `LazyMap` transport map training is computed with a 200-sample MC gradient estimator. The entry labeled `LazyMap` (16k) refers to training with a total of 80 iterations, and `LazyMap` (128k) refers to 640 iterations. In contrast, the surrogate and its Jacobian actions are evaluated 16 million times during `LazyDINO` transport map training.

Overall, these results demonstrate that `LazyDINO` still performs substantially better than `LazyMap` with 20-way parallelism for similar end-to-end computational costs. Moreover, while we only considered mean error as a performance metric in the reported results, `LazyDINO`'s efficiency gains in other metrics are even higher, as shown in Section 6.2. A key takeaway from these results: for extremely model-query-intensive algorithms such as transport map training, surrogates are necessary, and since training with high-fidelity models is so expensive, one can invest significantly in offline computations and still achieve an order-of-magnitude reduction in total computational costs.

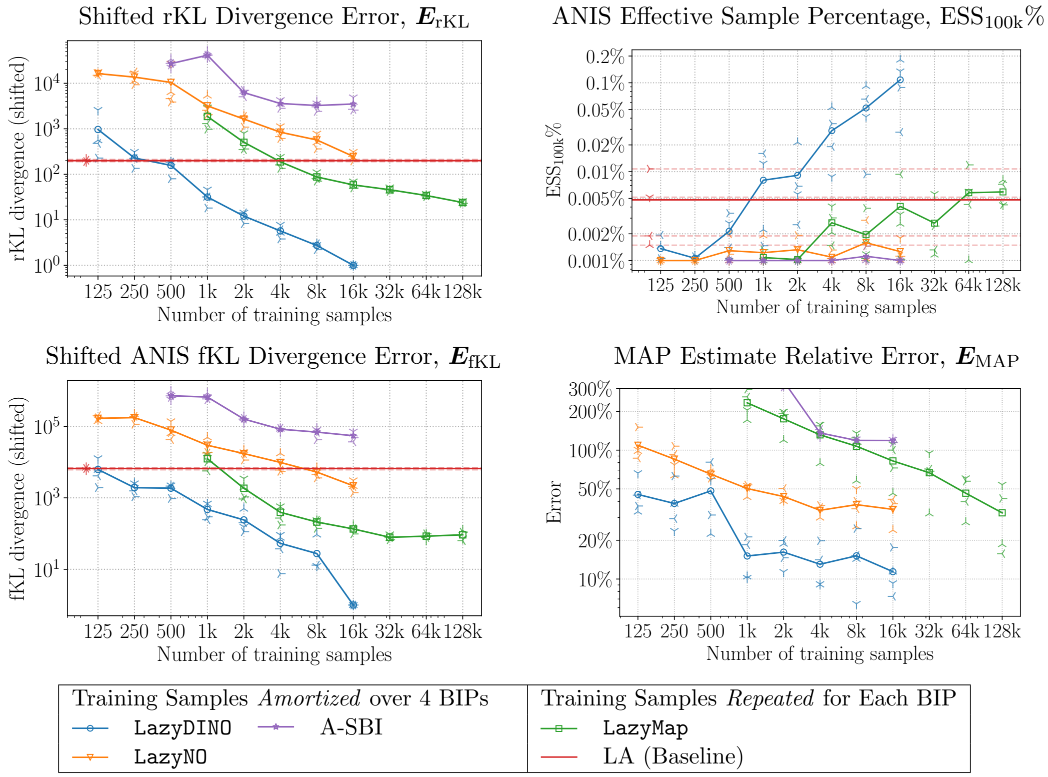

Figure 12: **Example I: Density-Based Diagnostics.** Higher values of $\mathrm{ESS}_{100\mathrm{k}}\%$ and lower values of all other diagnostics imply better posterior approximation. Similar trends to the moment discrepancy comparisons in Figure 10 are observed. `LazyDINO` enjoys over 8–128× higher sample efficiency. While $\mathrm{ESS}_{100\mathrm{k}}\%$ is low across the board, `LazyDINO` produces around 100 effective samples while other methods only achieve around 1–6.

**A-SBI vs.** `LazyDINO` In typical formulations of A-SBI, sampling requires inverting the transport map. While this need not be the case, the alternative requires inverting the transport map during training, which is often considered too expensive. Since our aim is fast online inference, we compare against the typical A-SBI formulation, which is fast to train. For transport map parameterizations such as IAFs, the scalability of this inversion can be preserved. In the case of IAFs, the cost is dominated by $d_r$-dimensional root-finding problems, which can be solved relatively quickly, e.g., via the bisection method. In contrast, sampling with `LazyDINO` involves only explicit evaluations of the transport map and, thus, produces samples in significantly less time.

In Table 5, we report the average times for the online phase to generate 1 million iid approximate posterior samples for four instances of the observational data studied in Examples I and II. The total online costs of `LazyDINO`, which include first training a transport map for each instance of observational data and then sampling using map evaluations, are lower than those of A-SBI, which only involves sampling by inverting the transport map already trained offline. We note that these results depend on the transport map architecture

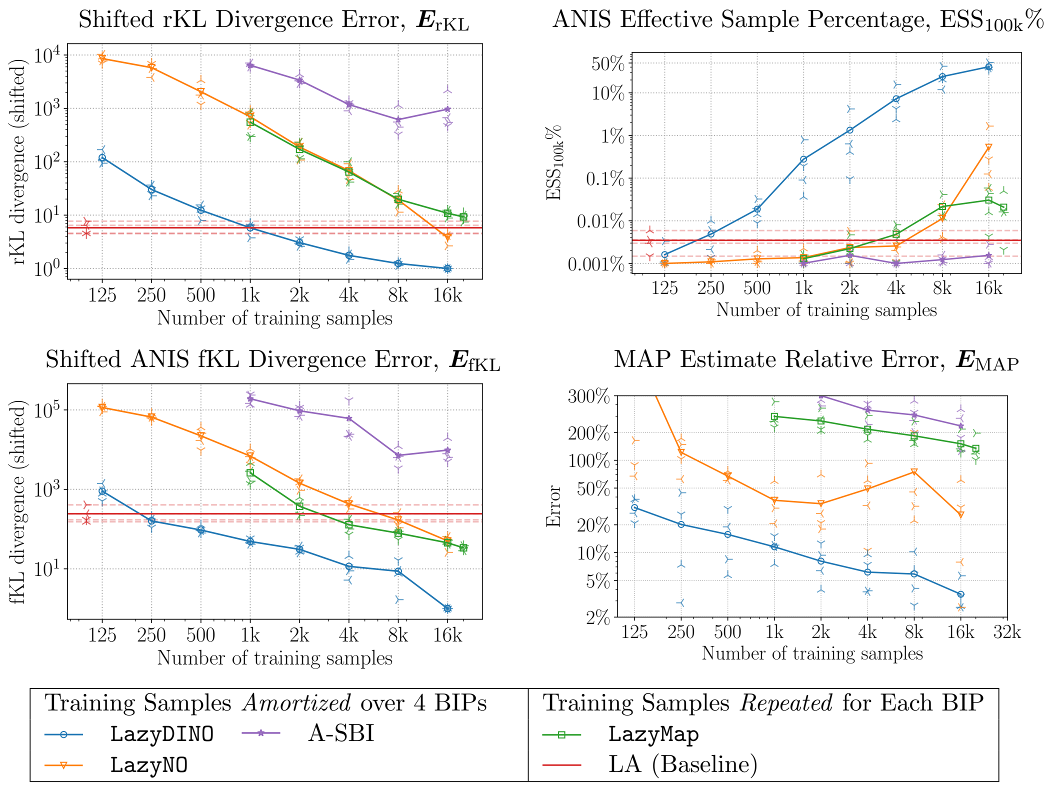

Figure 13: **Example II: Density-Based Diagnostics.** We observe similar trends as in Example I. Notably, `LazyDINO` achieves nearly 50% ANIS effective sample percentage as a 100k-sample independent sampler with 16k training samples, 50 to 50k times the percentage achieved for competing methods.

and the quality of implementation; however, the overall point about the additional expense of inverting maps in A-SBI is broadly applicable.

**Remark 9** *We emphasize that the reported times for transport map training (460s and 750s for Examples I and II, respectively) reflect intensive optimization runs involving 16 million evaluations of the variational inference objective and its gradient to ensure a highly accurate posterior approximation benchmark. However, since marginal gains in accuracy diminish with optimization iterations, the optimization can be stopped much earlier to achieve faster online Bayesian inversion while retaining high posterior accuracy.*

### 6.4 Visualization of Discrepancy in Marginals

In this section, we examine the relative performance of different methods using their 2D marginal kernel density estimates and 1D marginal histograms. We plot the progression of marginals as the training sample size increases. Figures 14 to 17 shows the marginal plots for Example I. They are consistent with the previous results: `LazyDINO` produces better approximations than the other TMVI methods and overtakes the LA with a relatively small number of training samples.

| | | **LMVI Method** | | | |
|---|---|---|---|---|---|
| | **Category (unit)** | `LazyMap` (16k) | `LazyMap` (128k) | `LazyDINO` (1k) | `LazyDINO` (16k) |
| Algorithm Steps | Amortized PtO evaluations (sec) | – | – | 130/4 | 1,950/4 |
| | Parallel amortized PtO evaluations (sec) | – | – | 6.5 /4 | 97.5/4 |
| | Amortized Jacobian (sec) | – | – | 50/4 | 750/4 |
| | Parallel amortized Jacobian (sec) | – | – | 2.5/4 | 37.5/4 |
| | Amortized DINO training (sec) | – | – | 80/4 | 1,220 /4 |
| | TMVI training (sec) | 2,710 | 21,560 | 460 | 460 |
| | Parallel TMVI training (sec) | 135.5 | 1078 | – | – |
| Total | Time per BIP (sec) | 2,710 | 21,560 | 525 | 1,440 |
| | Parallel time per BIP (sec) | 135.5 | 1078 | 482.45 | 798.75 |
| | Relative mean error achieved (%) | 80 | 20 | 10 | 5 |

Table 3: **Example I: `LazyDINO` vs. `LazyMap` Timing Comparison.** The total sequential execution times are similar for `LazyMap` (16k) and `LazyDINO` (16k), and the total 20-way parallel execution times are similar for `LazyMap` (128k) and `LazyDINO` (16k). In both cases, the relative mean error achieved is much lower for `LazyDINO`. Moreover, `LazyDINO` (1k) achieves a smaller relative mean error than `LazyMAP` (128k) in less time. The TMVI training time for `LazyDINO` is the online transport map training time per observational data.

We also investigate posterior marginals for Example II in Figures 18 to 21. These results show that the target marginals exhibit greater non-Gaussianity than Example I, and `LazyDINO` consistently outperforms all other methods. Notably, due to the non-Gaussianity, the LA does not produce accurate marginal approximations, while `LazyDINO` matches the true marginals well with a small number of samples.

This concludes our numerical study. In almost every point of comparison, `LazyDINO` yielded the most accurate estimation of the posterior distribution as evidenced by moment discrepancies, density-based diagnostics, and posterior marginals. Notably, `LazyDINO` gives faithful posterior estimates for orders of magnitude fewer samples than competing methods. While the LA performed well on some metrics, given limited samples (e.g., MAP and covariance estimation), it assumes posterior Gaussianity, which is undesirable for highly nonlinear BIPs. In Appendix I, we also provide additional visualizations of the posterior mean, MAP, posterior pointwise variance, and expanded posterior marginal plots that reinforce the superiority of `LazyDINO` observed from the numerical results in this section.

| | **Category (unit)** | **Method** | | | |
|---|---|---|---|---|---|
| | | `LazyMap` (1k) | `LazyMap` (16k) | `LazyDINO` (1k) | `LazyDINO` (16k) |
| Algorithm Steps | Amortized PtO evaluations (sec) | – | – | 2,150/4 | 34,200/4 |
| | Parallel amortized PtO evaluations (sec) | – | – | 107.5/4 | 1,710/4 |
| | Amortized Jacobian (sec) | – | – | 220/4 | 3,440/4 |
| | Parallel amortized Jacobian (sec) | – | – | 11/4 | 172/4 |
| | Amortized DINO training (sec) | – | – | 120/4 | 1,840/4 |
| | TMVI training (sec) | 2,390 | 38,500 | 750 | 750 |
| | Parallel TMVI training (sec) | 119.5 | 1,925 | – | – |
| Total | Time per BIP (sec) | 2,390 | 38,500 | 1,372.5 | 10,620 |
| | Parallel time per BIP (sec) | 119.5 | 1,925 | 809.625 | 1,680.5 |
| Relative mean error achieved (%) | | 230 | 90 | 12 | 3.5 |

Table 4: **Example II: `LazyDINO` vs. `LazyMap` Timing Comparison.** This timing comparison is similar to Table 3. For both sample sizes, `LazyDINO` achieves a much lower relative mean error. Moreover, `LazyDINO` (1k) achieves a much smaller error than `LazyDINO` (16k) in less time.

| **Example I** | **Method** | |
|---|---|---|
| **Time (sec)** | A-SBI (16k) | `LazyDINO` (16k) |
| 1 Million Samples | 1130 | 60 |
| Online Training | — | 460 |
| Offline Training | 930 / 4 | 1220/4 |
| Total per BIP | 1362.5 | 825 |

| **Example II** | **Method** | |
|---|---|---|
| **Time (sec)** | A-SBI (16k) | `LazyDINO` (16k) |
| 1 Million Samples | 1150 | 60 |
| Online Training | — | 750 |
| Offline Training | 960 / 4 | 1840/4 |
| Total per BIP | 1390 | 1270 |

Table 5: **Online Cost of A-SBI vs. `LazyDINO`.** The inversion-to-sample approach of A-SBI incurs more online costs optimize-then-sample approach of `LazyDINO` for generating 1 million samples. The bottleneck in sampling with A-SBI is due to transport map inversion. For fair comparison, we employ similar IAF architectures for both methods. Times are averaged across the four observation instances and rounded to 10 seconds.

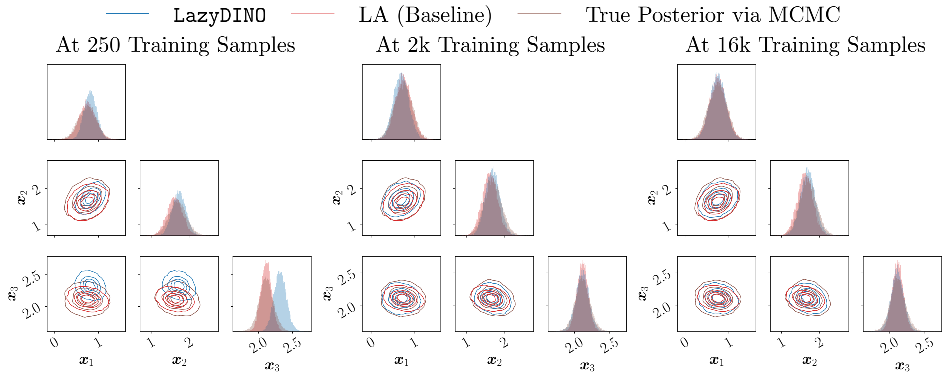

Figure 14: **Example I: LazyDINO vs. LA Posterior Marginals.** At 250 samples, we observe clear discrepancies in the marginals for both methods, with LA being more accurate. At 2k samples, LazyDINO is more accurate, especially for the low probability contours.

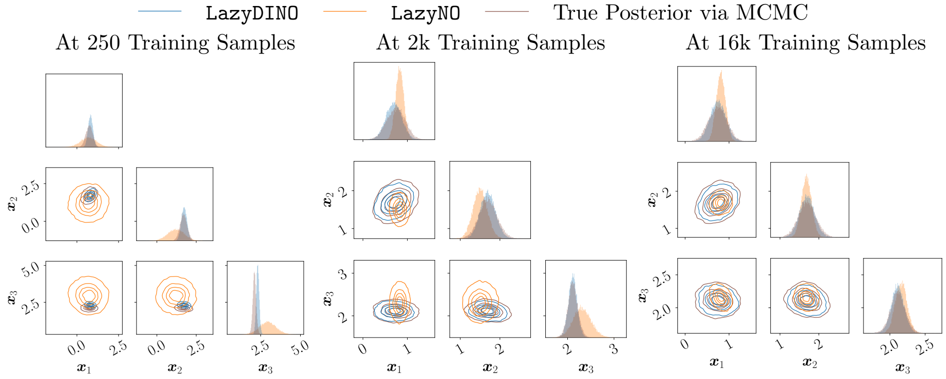

Figure 15: **Example I: LazyDINO vs. LazyNO Posterior Marginals.** LazyNO is less accurate than LazyDINO. LazyNO's posterior marginals are under-concentrated at the small-sample regime, and over-concentrated at the high-sample regime.

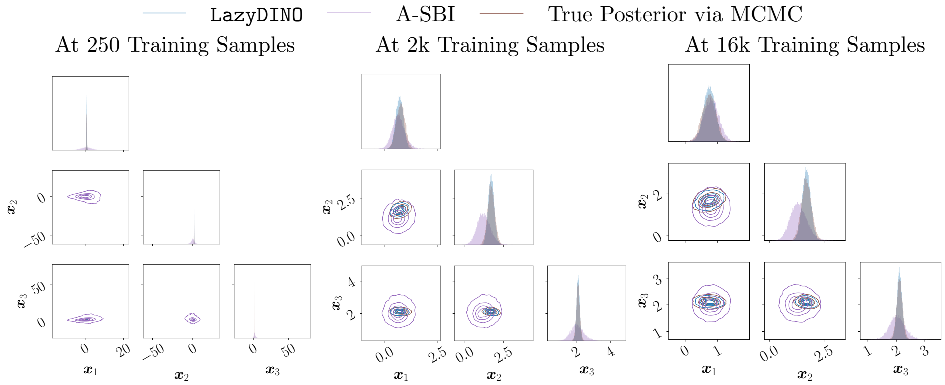

Figure 16: **Example I: LazyDINO vs. A-SBI Posterior Marginals.** A-SBI underestimates the posterior uncertainty and yields a low-quality approximation with 16k samples.

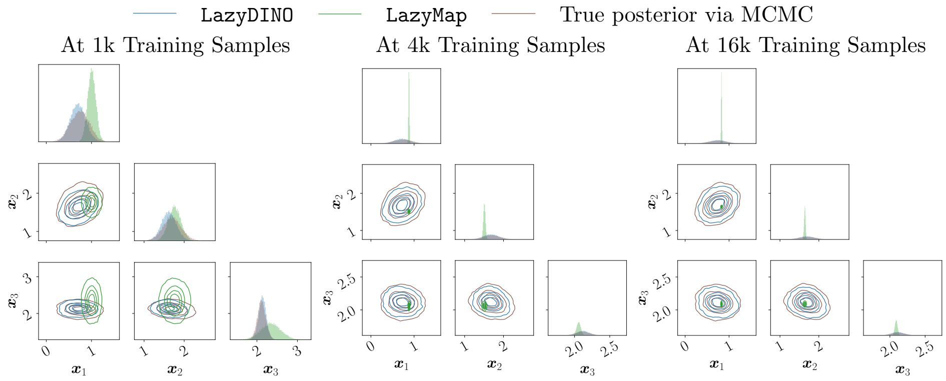

Figure 17: **Example I: `LazyDINO` vs. `LazyMap` Posterior Marginals.** `LazyMap` produces a low-quality approximation for 1k samples and drastically overestimates the posterior uncertainty with more samples.

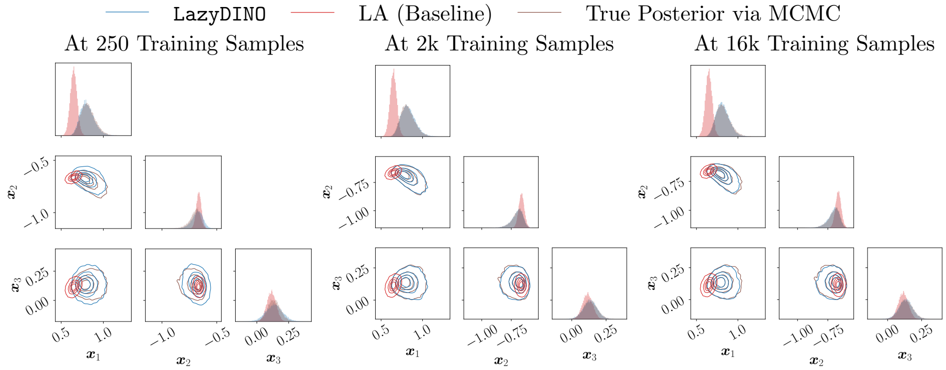

Figure 18: **Example II: `LazyDINO` vs. LA Posterior Marginals.** Due to the posterior's non-Gaussianity, LA struggles to match the marginals. In contrast, `LazyDINO` is accurate at 2k samples. In general, the marginal of the LA differs from the MAP of the marginal, which may explain the LA's inconsistency with the true posterior marginal.

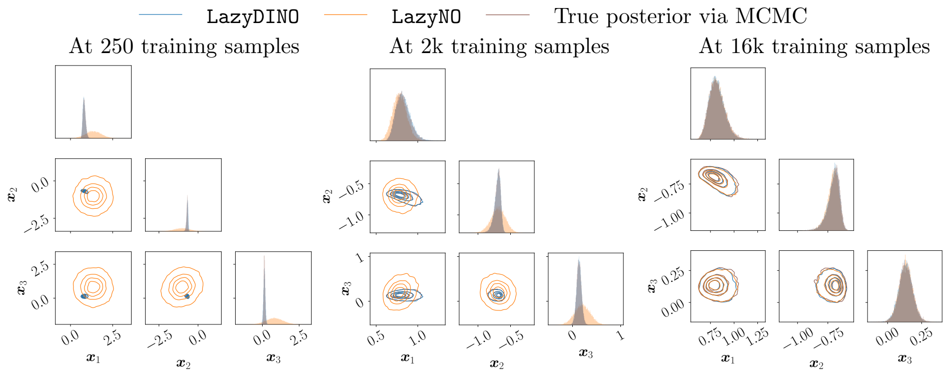

Figure 19: **Example II: `LazyDINO` vs. `LazyNO` Posterior Marginals.** With 250 training samples, `LazyNO` drastically underestimate the posterior uncertainty. With 16k samples, both methods appear to be accurate. `LazyNO` requires orders of magnitude more samples to have comparable performance to `LazyDINO`.

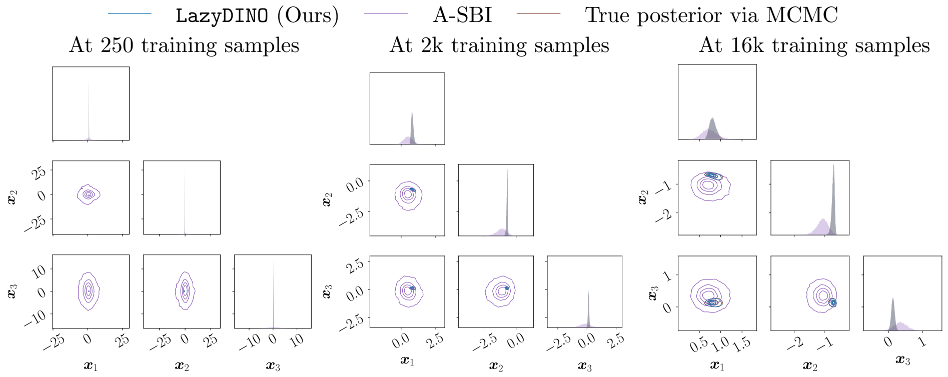

Figure 20: **Example II: `LazyDINO` vs. A-SBI Posterior Marginals.** A-SBI produces samples that are highly under-concentrated. A-SBI is simultaneously off in capturing the peak locations of the marginals while overestimating the parametric uncertainty.

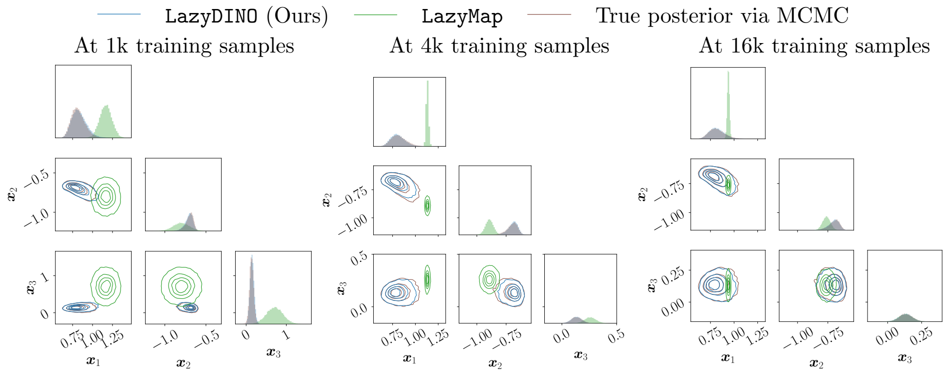

Figure 21: **Example II: `LazyDINO` vs. `LazyMap` Marginals.** `LazyMap` fails to capture the high-probability region at 1k and 4k samples and over-concentrate at 16k samples.

## 7. Conclusion

We present LazyDINO, a fast, scalable, and efficiently amortized method for high-dimensional Bayesian inversion with expensive PtO maps. The method comprises an offline and an online phase. During the offline phase, we generate joint samples of the PtO map and its Jacobian to construct a RB-DINO surrogate via derivative-based dimension reduction and derivative-informed learning methods. During the online phase, when observational data are given, we rapidly approximate the posterior via surrogate-driven optimization of lazy maps, i.e., structure-exploiting transport maps with relatively low-dimensional nonlinearity. The trained lazy map is used for approximate posterior sampling and density evaluations.

We provide theoretical results demonstrating that the RB-DINO surrogate construction is ideally suited for amortized Bayesian inversion via lazy map variational inference. In Theorem 4, we show that the conventional supervised learning of the DIPNet surrogate architecture minimizes an upper bound on the expected posterior approximation error when the ridge function surrogate replaces the PtO map. This architecture constrains the surrogate approximation to the parameter subspace that captures the prior-to-posterior update. In Theorem 5 and Corollary 6, we show that the derivative-informed learning of the surrogate minimizes the expected gradient error and optimality gap due to surrogate-driven transport map optimization. Surrogate Jacobian accuracy thus controls the quality of the trained lazy map and the posterior approximation.

The LazyDINO method has several desirable traits.

1. *Scalability.* The surrogate and transport map training in LazyDINO are independent of the parameter dimension, as their latent representations reside in the same relatively low-dimensional derivative-informed subspace (Figures 3 and 4).

2. *Fast Online Inference.* Using a cheap-to-evaluate surrogate rKL objective for transport map optimization, LazyDINO fully exploits GPU-based accelerations to rapidly approximate posteriors (Tables 3 and 4). While our method requires solving an optimization problem to sample, we demonstrate that it leads to faster online posterior sampling than the typical inversion-to-sample approach of A-SBI in the large-sample regime (Table 5).

3. *High Posterior Accuracy at Low Offline Costs.* First, the RB-DINO surrogate and the lazy map are co-designed to efficiently exploit the structure of the BIP. Second, the derivative-informed learning method is highly cost-efficient and outperforms conventional supervised learning by one to two orders of magnitude (Figures 8 and 9). Consequently, LazyDINO requires a much smaller cost in offline computation to achieve high accuracy in online posterior approximation across multiple instances of the observational data.

Our numerical experiments consider two challenging infinite-dimensional PDE-constrained BIPs: (I) inferring the diffusivity field in a nonlinear reaction-diffusion equation, and (II) inferring the Young's modulus field of a hyperelastic material under deformation. In both cases, we observed one to two orders of magnitude in offline cost reduction to achieve similar accuracy in online posterior approximation compared to alternative amortized inference methods such as LazyNO and A-SBI via conditional transport. Moreover, LazyDINO consistently outperforms Laplace approximation even in a small-sample offline training regime

(250–1k), except for covariance approximation for Example I. In contrast, `LazyNO` and `A-SBI` struggle to outperform Laplace approximation and, in some cases, failed at 16k offline training samples.

The efficiency gains achieved via `LazyDINO` motivate further study. For instance, `LazyDINO` does not assume that the observations $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{d_{\boldsymbol{y}}})$ are iid. On the contrary, in the examples studied, the data arise from spatially distributed observations of a PDE solution; their distributions thus vary in space and may involve correlated observational errors. Yet our framework can easily be extended to the setting of multiple independent realizations of non-iid observations, $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(k)})$, where each $\boldsymbol{y}^{(j)} = (\boldsymbol{y}_1^{(j)}, \ldots, \boldsymbol{y}_{d_{\boldsymbol{y}}}^{(j)})$. This is a natural benefit of building surrogates of the PtO map: the same surrogate can be used to approximate each term in a (block) product likelihood, unlike, e.g., the A-SBI methods considered in this work. Given its significantly reduced online costs, `LazyDINO` can be used for real-time uncertainty quantification. In particular, the `RB-DINO` surrogate construction can be applied to rapidly forward-propagate the `LazyDINO` posterior samples and obtain posterior predictive samples for quantities of interest relevant to downstream statistical decision-making.

## Acknowledgment

This work was partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under awards DE-SC0021239 (JC, TOR, and OG), DE-SC0023171 (JC, TOR, and OG), and DE-SC0023187 (MB, YM); the National Science Foundation under awards OAC-2313033, DMS-2324643, and RISE-2425922 (JC, TOR, and OG); the Air Force Office of Scientific Research under MURI grants FA9550-21-1-0084 and FA9550-24-1-0327 (JC, TOR, and OG); a Department of Defense Vannevar Bush Faculty Fellowship held by Andrew M. Stuart (LC); and the SciAI Center, funded by the Office of Naval Research, under Grant Number N00014-23-1-2729 (LC).

## References

Abdelrahman Abdelhamed, Marcus A Brubaker, and Michael S Brown. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3165–3173, 2019.

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. doi: 10.1214/17-STS611.

Malte Algren, Tobias Golling, Manuel Guth, Chris Pollard, and John Andrew Raine. Flow away your differences: Conditional normalizing flows as an improvement to reweighting. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2304.14963.

Ricardo Baptista, Lianghao Cao, Joshua Chen, Omar Ghattas, Fengyi Li, Youssef M. Marzouk, and J. Tinsley Oden. Bayesian model calibration for block copolymer self-assembly: Likelihood-free inference and expected information gain computation via measure transport. *Journal of Computational Physics*, 503:112844, 2024a. doi: 10.1016/j.jcp.2024.112844.

Ricardo Baptista, Youssef Marzouk, and Olivier Zahm. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, 24 (6):2063–2108, 2024b. doi: 10.1007/s10208-023-09630-x.

Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E. Farrell, and Andrew M. Stuart. Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017. doi: 10.1016/j.jcp.2016.12.041.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural network for parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021. doi: 10.5802/smai-jcm.74.

Daniele Bigoni, Youssef Marzouk, Clémentine Prieur, and Olivier Zahm. Nonlinear dimension reduction for surrogate modeling using gradient information. *Information and Inference: A Journal of the IMA*, 11(4):1597–1639, 2022. doi: 10.1093/imaiai/iaac006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

Vladimir I. Bogachev. *Gaussian Measures*, volume 62 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island, 1998. ISBN 978-0-8218-9264-3.

Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization*, 28(4):3312–3343, 2018. doi: 10.1137/17M1154679.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173.

Johann Brehmer, Gilles Louppe, Juan Pavez, and Kyle Cranmer. Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy of Sciences*, 117(10):5242–5249, 2020. doi: 10.1073/pnas.1915980117.

Michael Brennan, Daniele Bigoni, Olivier Zahm, Alessio Spantini, and Youssef Marzouk. Greedy inference with structure-exploiting lazy maps. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8330–8342. Curran Associates, Inc., 2020.

Tan Bui-Thanh and Omar Ghattas. Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves. *Inverse Problems*, 28(5):055002, 2012a. doi: 10.1088/0266-5611/28/5/055002.

Tan Bui-Thanh and Omar Ghattas. Analysis of the Hessian for inverse scattering problems. Part I: Inverse shape scattering of acoustic waves. *Inverse Problems*, 28(5):055001, 2012b. doi: 10.1088/0266-5611/28/5/055001.

Tan Bui-Thanh and Omar Ghattas. Analysis of the Hessian for inverse scattering problems. Part III: Inverse medium scattering of electromagnetic waves. *Inverse Problems and Imaging*, 7(4):1139–1155, 2013. doi: 10.3934/ipi.2013.7.1139.

Tan Bui-Thanh, Carsten Burstedde, Omar Ghattas, James Martin, Georg Stadler, and Lucas C. Wilcox. Extreme-scale UQ for Bayesian inverse problems governed by PDEs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, Washington, DC, USA, 2012. IEEE Computer Society Press. ISBN 9781467308045.

Lianghao Cao, Thomas O'Leary-Roseberry, Prashant K Jha, J Tinsley Oden, and Omar Ghattas. Residual-based error correction for neural operator accelerated infinite-dimensional Bayesian inverse problems. *Journal of Computational Physics*, 486:112104, 2023. doi: 10.1016/j.jcp.2023.112104.

Lianghao Cao, Thomas O'Leary-Roseberry, and Omar Ghattas. Derivative-informed neural operator acceleration of geometric MCMC for infinite-dimensional Bayesian inverse problems. *Journal of Machine Learning Research*, 26(78):1–68, 2025. URL `http://jmlr.org/papers/v26/24-0745.html`.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024. doi: 10.1038/s42256-024-00844-4.

Peng Chen and Omar Ghattas. Hessian-based sampling for high-dimensional model reduction. *International Journal for Uncertainty Quantification*, 9(2), 2019. doi: 10.1615/Int.J.UncertaintyQuantification.2019028753.

Peng Chen, Umberto Villa, and Omar Ghattas. Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 327:147–172, 2017. doi: 10.1016/j.cma.2017.08.016.

Peng Chen, Keyi Wu, Joshua Chen, Tom O'Leary-Roseberry, and Omar Ghattas. Projected Stein variational newton: A fast and scalable Bayesian inference method in high dimensions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

P.C. Chowdhury. The truncated lanczos algorithm for partial solution of the symmetric eigenproblem. *Computers & Structures*, 6(6):439–446, 1976. doi: 10.1016/0045-7949(76)90037-7.

T Cui, J Martin, Y M Marzouk, A Solonen, and A Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, October 2014. doi: 10.1088/0266-5611/30/11/114015.

Tiangang Cui and Olivier Zahm. Data-free likelihood-informed dimension reduction of Bayesian inverse problems. *Inverse Problems*, 37(4):045009, 2021. doi: 10.1088/1361-6420/abeafb.

Tiangang Cui, Youssef M. Marzouk, and Karen E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015. doi: 10.1002/nme.4748.

Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6):906–917, 2010. doi: 10.1109/TNN.2010.2044803.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1263–1273. PMLR, 22–25 Jul 2020.

Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. Inexact newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982. doi: 10.1137/0719025.

Gianluca Detommaso, Tiangang Cui, Youssef Marzouk, Alessio Spantini, and Robert Scheichl. A Stein variational Newton method. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv preprint*, 2016. doi: 10.48550/arXiv.1605.08803.

Conor Durkan, George Papamakarios, and Iain Murray. Sequential neural methods for likelihood-free inference. *arXiv preprint*, 2018. doi: 10.48550/arXiv.1811.08723.

Stanley C. Eisenstat and Homer F. Walker. Choosing the forcing terms in an inexact newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996. doi: 10.1137/0917003. URL `https://doi.org/10.1137/0917003`.

Tarek A. El Moselhy and Youssef M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, October 2012. doi: 10.1016/j.jcp.2012.07.022.

I-G Farcas, Jonas Latz, Elisabeth Ullmann, Tobias Neckel, and H-J Bungartz. Multilevel adaptive sparse Leja approximations for Bayesian inverse problems. *SIAM Journal on Scientific Computing*, 42(1):A424–A451, 2020. doi: 10.1137/19M126029.

Pearl H. Flath, Lucas C. Wilcox, Volkan Akçelik, Judy Hill, Bart van Bloemen Waanders, and Omar Ghattas. Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations. *SIAM Journal on Scientific Computing*, 33(1):407–432, 2011. doi: 10.1137/090780717.

D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81:1581–1608, 3 2010. doi: 10.1002/nme.2746.

Ankush Ganguly, Sanjana Jain, and Ukrit Watchareeruetai. Amortized variational inference: A systematic review. *Journal of Artificial Intelligence Research*, 78:167–215, October 2023. doi: 10.1613/jair.1.14258.

Omar Ghattas and Karen Willcox. Learning physics-based models from data: Perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554, 2021. doi: 10.1017/S0962492921000064.

Jinwoo Go and Peng Chen. Sequential infinite-dimensional Bayesian optimal experimental design with derivative-informed latent attention neural operator. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2409.09141.

Jinwoo Go and Peng Chen. Accurate, scalable, and efficient Bayesian optimal experimental design with derivative-informed neural operators. *Computer Methods in Applied Mechanics and Engineering*, 438:117845, 2025. ISSN 0045-7825. doi: 10.1016/j.cma.2025.117845.

Gene H. Golub and Qiang Ye. An inverse free preconditioned krylov subspace method for symmetric generalized eigenvalue problems. *SIAM Journal on Scientific Computing*, 24 (1):312–334, 2002. doi: 10.1137/S1064827500382579.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2404–2414. PMLR, 09–15 Jun 2019.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. doi: 10.1137/090771806.

J.S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018. doi: doi.org/10.1016/j.jcp.2018.02.037.

Xun Huan, Jayanth Jagalur, and Youssef Marzouk. Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840, 2024. doi: 10.1017/S0962492924000023.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2078–2087. PMLR, 10–15 Jul 2018.

Tobin Isaac, Noemi Petra, Georg Stadler, and Omar Ghattas. Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *Journal of Computational Physics*, 296:348–368, September 2015. doi: 10.1016/j.jcp.2015.04.047.

Priyank Jaini, Kira A. Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International*

*Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3009–3018. PMLR, 09–15 Jun 2019.

Michael G Kapteyn, Jacob VR Pretorius, and Karen E Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021. doi: 10.1038/s43588-021-00069-0.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2017. doi: 10.48550/arXiv.1412.6980.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint*, 2017. doi: 10.48550/arXiv.1606.04934.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016.

Herbert Knothe. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4:39–52, 1957. doi: 10.1307/mmj/1028990175.

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. doi: 10.1109/TPAMI.2020.2992934.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24 (89):1–97, 2023.

Remo Kretschmann. Are Minimizers of the Onsager–Machlup Functional Strong Posterior Modes? *SIAM/ASA Journal on Uncertainty Quantification*, 11(4):1105–1138, October 2023. doi: 10.1137/23m1546579.

Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. Nonlocality and nonlinearity implies universality in operator learning. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2304.13221.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf`.

Sen Li, Ke Li, Yu Liu, and Qifeng Liao. A conditional normalizing flow for domain decomposed uncertainty quantification. *Journal of Computational and Applied Mathematics*, 465:116571, 2025. doi: 10.1016/j.cam.2025.116571.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint*, 2021. doi: 10.48550/arXiv.2010.08895.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, feb 2024. doi: 10.1145/3648506.

Chad Lieberman, Karen Willcox, and Omar Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32 (5):2523–2542, 2010. doi: 10.1137/090775622.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Dingcheng Luo, Thomas O'Leary-Roseberry, Peng Chen, and Omar Ghattas. Efficient PDE-constrained optimization under high-dimensional uncertainty using derivative-informed neural operators. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2305.20053.

M. B. Lykkegaard, T. J. Dodwell, C. Fox, G. Mingas, and R. Scheichl. Multilevel delayed acceptance MCMC. *SIAM/ASA Journal on Uncertainty Quantification*, 11(1):1–30, 2023. doi: 10.1137/22M1476770.

Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification, R. Ghanem, D. Higdon, and H. Owhadi, editors*. Springer, 2016.

Youssef Marzouk and Dongbin Xiu. A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics*, 6(4):826–847, 2009.

Youssef M. Marzouk and Habib N. Najm. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862–1902, 2009. doi: 10.1016/j.jcp.2008.11.024.

Youssef M. Marzouk, Habib N. Najm, and Larry A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224 (2):560–586, 2007. doi: 10.1016/j.jcp.2006.10.010.

David Newton, Raghu Bollapragada, Raghu Pasupathy, and Nung Kwan Yip. A retrospective approximation approach for smooth stochastic optimization. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2103.04392.

David Nualart. *The Malliavin Calculus and Related Topics*. Probability and Its Applications. Springer, Berlin, 2 edition, 2006. ISBN 9783540283287. doi: 10.1007/3-540-28329-3.

Thomas O'Leary-Roseberry and Raghu Bollapragada. Fast unconstrained optimization via Hessian averaging and adaptive gradient sampling methods. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2408.07268.

Thomas O'Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning. *Journal of Computational Physics*, 496:112555, 2024. doi: 10.1016/j.jcp.2023.112555.

Thomas O'Leary-Roseberry, Xiaosong Du, Anirban Chaudhuri, Joaquim RRA Martins, Karen Willcox, and Omar Ghattas. Learning high-dimensional parametric maps via reduced basis adaptive residual networks. *Computer Methods in Applied Mechanics and Engineering*, 402:115730, 2022a. doi: 10.1016/j.cma.2022.115730.

Thomas O'Leary-Roseberry, Umberto Villa, Peng Chen, and Omar Ghattas. Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs. *Computer Methods in Applied Mechanics and Engineering*, 388:114199, 2022b. doi: 10.1016/j.cma.2021.114199.

George Papamakarios and Iain Murray. Fast $\epsilon$-free inference of simulation models with Bayesian conditional density estimation, 2018.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 837–848. PMLR, 16–18 Apr 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018. doi: 10.1137/16M1082469.

Noemi Petra, James Martin, Georg Stadler, and Omar Ghattas. A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1525–A1555, 2014. doi: 10.1137/130934805.

Allan Pinkus. *Ridge Functions*, volume 205 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2015. ISBN 9781107124394. doi: 10.1017/CBO9781316408124.

Felipe Maia Polo and Renato Vicente. Effective sample size, dimensionality, and generalization in covariate shift adaptation. *Neural Computing and Applications*, 35:18187–18199, 2020. doi: 10.1007/s00521-021-06615-1.

Yuan Qiu, Nolan Bridges, and Peng Chen. Derivative-enhanced deep operator network. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 20945–20981. Curran Associates, Inc., 2024.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470 – 472, 1952. doi: 10.1214/aoms/1177729394.

Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37(155):105–126, July 1981. doi: 10.1090/S0025-5718-1981-0616364-6.

Arvind K. Saibaba, Jonghyun Lee, and Peter K. Kitanidis. Randomized algorithms for generalized Hermitian eigenvalue problems with application to computing Karhunen–Loève expansion. *Numerical Linear Algebra with Applications*, 23(2):314–339, 2016. doi: 10.1002/nla.2026.

Daniel Sanz-Alonso and Zijian Wang. Bayesian update with importance sampling: Required sample size. *Entropy*, 23(1):22, December 2020. doi: 10.3390/e23010022.

Henrik Schopmans and Pascal Friederich. Conditional normalizing flows for active learning of coarse-grained molecular representations. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2402.01195.

Christian Soize and Roger Ghanem. Physical systems with random uncertainties: Chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26(2):395–410, 2004. doi: 10.1137/S1064827503424505.

D. C. Sorensen. Truncated QZ methods for large scale generalized eigenvalue problems. *Electron. Trans. Numer. Anal.*, 7:141–162, 1998.

Alessio Spantini, Daniele Bigoni, and Youssef Marzouk. Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, 19(1):2639–2709, 2018.

Ihab Sraj, Olivier P. Le Maître, Omar M. Knio, and Ibrahim Hoteit. Coordinate transformation and polynomial chaos for the Bayesian inference of a Gaussian process with parametrized prior covariance function. *Computer Methods in Applied Mechanics and Engineering*, 298:205–228, 2016. doi: 10.1016/j.cma.2015.10.002.

A M Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010. ISSN 09624929. doi: 10.1017/S0962492910000061.

Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013. doi: 10.1002/cpa.21423.

Jasper van den Eshof and Gerard L. G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):125–153, 2004. doi: 10.1137/S0895479802403459.

Umberto Villa and Thomas O'Leary-Roseberry. A note on the relationship between PDE-based precision operators and Matérn covariances. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2407.00471.

Umberto Villa, Noemi Petra, and Omar Ghattas. hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference. *ACM Transactions on Mathematical Software*, 47(2), April 2021. doi: 10.1145/3428447.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40):eabi8605, 2021. doi: 10.1126/sciadv.abi8605.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Josephine Westermann and Jakob Zech. Measure transport via polynomial density surrogates. *Foundations of Data Science*, 2025. doi: 10.3934/fods.2025001.

Hua Xiang and Jun Zou. Randomized algorithms for large-scale inverse problems with general Tikhonov regularizations. *Inverse Problems*, 31(8):085008, jul 2015. doi: 10.1088/0266-5611/31/8/085008.

Minglei Yang, Pengjun Wang, Ming Fan, Dan Lu, Yanzhao Cao, and Guannan Zhang. Conditional pseudo-reversible normalizing flow for surrogate modeling in quantifying uncertainty propagation. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2404.00502.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10665–10673, May 2021. doi: 10.1609/aaai.v35i12.17275.

Olivier Zahm, Paul G Constantine, Clémentine Prieur, and Youssef M Marzouk. Gradient-based dimension reduction of multivariate vector-valued functions. *SIAM Journal on Scientific Computing*, 42(1):A534–A558, 2020. doi: 10.1137/18M1221837.

Olivier Zahm, Tiangang Cui, Kody Law, Alessio Spantini, and Youssef Marzouk. Certified dimension reduction in nonlinear Bayesian inverse problems. *Mathematics of Computation*, 91(336):1789–1835, 2022. doi: 10.1090/mcom/3737.

Jakob Zech and Youssef Marzouk. Sparse approximation of triangular transports, Part I: The finite-dimensional case. *Constr. Approx.*, 55(3):919–986, 2022a. doi: 10.1007/s00365-022-09569-2.

Jakob Zech and Youssef Marzouk. Sparse approximation of triangular transports, Part II: The infinite-dimensional case. *Constr. Approx.*, 55(3):987–1036, 2022b. doi: 10.1007/s00365-022-09570-9.

Justine Zeghal, François Lanusse, Alexandre Boucaud, Benjamin Remy, and Eric Aubourg. Neural posterior estimation with differentiable simulators. *arXiv preprint*, 2022. doi: 10.48550/arXiv.2207.05636.

## Appendix A. Glossary of Terminology

**BIP**: Bayesian inverse problem or Bayesian inference.

**Amortized cost**: Discounted cost per BIP by spreading the total cost out over the solution of many BIPs.

**ANIS**: Auto-normalized importance sampling, also known as self-normalized importance sampling.

**A-SBI**: Amortized simulation-based inference.

**ESS**: Effective sample size, an estimation of the number of independent samples drawn from a proposal distribution to obtain an expectation estimator with equivalence variance as standard Monte Carlo.

**fKL**: Forward Kullback-Leibler of the approximate distribution $\mu$ from the target distribution $\nu$, $\mathcal{D}_{\mathrm{KL}}(\nu||\mu)$.

**LMVI**: Lazy map variational inference. Variational inference using structure-exploiting transport map with relatively low-dimensional non-linearity.

`LazyDINO`: Our proposed method for amortized Bayesian inversion via lazy map variational inference driven by derivative-informed neural operator (DINO) surrogates.

`LazyNO`: A competing method for amortized Bayesian inversion via lazy map variational inference driven by surrogates constructed using conventional supervised learning.

`LazyMap`: A transport map that acts only in a subspace of the input, by way of a linear projection-based dimension reduction. It is also used to refer to the algorithm to train it.

**MAP point**: Maximum A-Posteriori point, the parameter with the highest probability concentration of the posterior distribution in a Bayesian inverse problem.

**MC**: Monte Carlo, which will always refer specifically to an iid sampling-based approach to approximating an expectation.

**MCMC**: Markov chain Monte Carlo.

**PtO**: Parameter-to-observable. A PtO map is a function taking the parameter we wish to infer to the non-noisy *observable.* In contrast, *observations* are measurements of the observable corrupted via noise.

**rKL**: Reverse Kullback-Leibler of the target distribution $\nu$ from an approximate distribution $\mu$, $\mathcal{D}_{\mathrm{KL}}(\mu||\nu)$.

**TMVI**: Transport map variational inference. We use this term to distinguish from other forms of variational inference, i.e., inference within families of distributions that do not originate via the transport of a reference distribution.

## Appendix B. Stochastic Differentiation and Conditional Expectation

We provide a proof of the statement that the stochastic derivative of the optimal ridge function $D_H(\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}} \circ \mathcal{P})$ for $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M}; H_{\Gamma_n})$ is the optimal ridge function of the stochastic derivative $(D_H\boldsymbol{\mathcal{G}} \circ \mathcal{P})_{\mathrm{opt}}$, where the latter is defined via a conditional expectation as in (13). This result can be used to show that the single sample estimate of the $H^1_\mu(\mathscr{M}; H_{\mathcal{C}})$ operator learning error is the $H^1_\pi(\mathbb{R}^{d_r}; \mathbb{R}^{d_{\boldsymbol{y}}})$ surrogate error in the latent space in (42).

The exchangeability of stochastic differentiation and conditional expectation for $\boldsymbol{\mathcal{G}}$ in $H^1_\mu(\mathscr{M}; \mathbb{R})$ can be found in Proposition 1.2.8 of Nualart (2006). For completeness, we show an extension to $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M}; H_{\Gamma_n})$ based on the expansion of the observable given as follows. Let $\boldsymbol{V} = \{V_i\}_{i=1}^{d_{\boldsymbol{y}}}$ be an $H_\Gamma$-orthonormal basis. We consider the expansion $\boldsymbol{\mathcal{G}} = \sum_{i=1}^{d_{\boldsymbol{y}}} \boldsymbol{\mathcal{G}}_i V_i$, where $\boldsymbol{\mathcal{G}}_i(\cdot) = \langle \boldsymbol{\mathcal{G}}(\cdot), V_i \rangle_{\Gamma^{-1}}$, implying each component function $\boldsymbol{\mathcal{G}}_i$ is in $H^1_\mu(\mathscr{M}; \mathbb{R})$.

We now consider conditional expectation with respect to a sub-sigma-algebra generated by a projector $\mathcal{P}$. Let $\mathcal{P}$ be defined implicitly via a $H_{\mathcal{C}}$-orthonormal reduced basis $\Psi_r = \{\psi_j\}_{j=1}^{d_r}$ and $\mathrm{span}(\Psi_r) = \mathrm{Im}(\mathcal{P})$. Since $\mathcal{P}$ a bounded linear operator, we have $\sigma(\mathcal{P}) \subset \mathcal{B}(H_{\mathcal{C}})$, the Borel sigma-algebra of $H_{\mathcal{C}}$, and standard arguments for the exchanging of stochastic or Malliavin differentiation and conditional expectation can be employed: In particular, by Proposition 1.2.8 of Nualart (2006), we have

$$D_H \mathbb{E}[\boldsymbol{\mathcal{G}}_i | \sigma(\mathcal{P})](\mathcal{P}m) = \mathbb{E}[D_H \boldsymbol{\mathcal{G}}_i | \sigma(\mathcal{P})](\mathcal{P}m) \quad \text{a.e. } m \in H_{\mathcal{C}}. \tag{51}$$

To achieve this same order exchange between Malliavin Jacobian and conditional expectation, we define the conditional expectation of the vector-valued $\boldsymbol{\mathcal{G}}$:

$$\mathbb{E}[\boldsymbol{\mathcal{G}} | \sigma(\mathcal{P})] := \sum_{i=1}^{d_{\boldsymbol{y}}} \mathbb{E}[\boldsymbol{\mathcal{G}}_i | \sigma(\mathcal{P})] V_i, \tag{52}$$

and the Malliavin derivative:

$$D_H \boldsymbol{\mathcal{G}}(\cdot) := \sum_{i=1}^{d_{\boldsymbol{y}}} D_H \boldsymbol{\mathcal{G}}_i(\cdot) \otimes V_i, \tag{53}$$

both of which are well-defined for finite dimension $d_{\boldsymbol{y}}$.

The exchangeability of differentiation and conditional expectation order gives

$$\begin{aligned}\underbrace{D_H\mathbb{E}[\boldsymbol{\mathcal{G}}|\sigma(\mathcal{P})]}_{D_H(\widetilde{\boldsymbol{\mathcal{G}}}_{\text{opt}}\circ\mathcal{P})}(\mathcal{P}m) &= \sum_{i=1}^{d_{\boldsymbol{y}}} D_H\mathbb{E}[\boldsymbol{\mathcal{G}}_i|\sigma(\mathcal{P})](\mathcal{P}m)\otimes V_i && \text{(via (53))}\\ &= \sum_{i=1}^{d_{\boldsymbol{y}}} \mathbb{E}[D_H\boldsymbol{\mathcal{G}}_i|\sigma(\mathcal{P})](\mathcal{P}m)\otimes V_i && \text{(via (51))}\\ &= \mathbb{E}\left[\sum_{i=1}^{d_{\boldsymbol{y}}} D_H\boldsymbol{\mathcal{G}}_i(\mathcal{P}m)\otimes V_i|\sigma(\mathcal{P})\right] && \text{(linearity of sum)}\\ &= \mathbb{E}[D_H\boldsymbol{\mathcal{G}}|\sigma(\mathcal{P})](\mathcal{P}m) \quad \text{a.e. } m\in H_{\mathcal{C}} && \text{(via (52)).}\end{aligned} \tag{54}$$

Given the definitions $\mathcal{P}=\mathcal{D}_r\mathcal{E}_r$ and $\boldsymbol{z}:=\mathcal{E}_r m$, we further have that

$$\mathbb{E}[D_H\boldsymbol{\mathcal{G}}|\sigma(\mathcal{P})](\mathcal{P}m) = \mathbb{E}_{m_\perp\sim\mu_\perp}\left[D_H\boldsymbol{\mathcal{G}}(\mathcal{P}m+m_\perp)\right] = \mathbb{E}_{m_\perp\sim\pi_\perp}\left[D_H\boldsymbol{\mathcal{G}}(\mathcal{D}_r z+m_\perp)\right]. \tag{55}$$

Finally, to derive the derivative of the optimal latent representation, we first note that

$$\mathbf{g}_{\text{opt}}(\boldsymbol{z}) = \boldsymbol{V}^*\widetilde{\boldsymbol{\mathcal{G}}}_{\text{opt}}(\mathcal{P}m) = \boldsymbol{V}^*\widetilde{\boldsymbol{\mathcal{G}}}_{\text{opt}}(\mathcal{P}^2m) = \boldsymbol{V}^*\widetilde{\boldsymbol{\mathcal{G}}}_{\text{opt}}(\mathcal{P}\mathcal{D}_r\mathcal{E}_r m) = \boldsymbol{V}^*(\widetilde{\boldsymbol{\mathcal{G}}}_{\text{opt}}\circ\mathcal{P})(\mathcal{D}_r\boldsymbol{z}), \tag{56}$$

so that by combining (54), (55), and (56) we arrive at

$$\nabla_{\boldsymbol{z}}\mathbf{g}_{\text{opt}}(\boldsymbol{z}) = \boldsymbol{V}^*\circ\mathbb{E}_{m_\perp\sim\mu_\perp}\left[D_H\boldsymbol{\mathcal{G}}(\mathcal{D}_r\boldsymbol{z}+m_\perp)\right]\circ\mathcal{D}_r.$$

A single-sample Monte Carlo estimate of this produces the estimate in (43).

## Appendix C. Transport Map Variational Inference on Hilbert Spaces

Let us consider the rKL objective using a nonlinear transformation of the Gaussian measure $\mu=\mathcal{N}(0,\mathcal{C})$ on a separable Hilbert space $\mathscr{M}$. The derivation below follows Section 6.6 of Bogachev (1998). Let $\mathcal{T}=\text{Id}_{\mathscr{M}}+\mathcal{K}$ be the transport map where $\mathcal{K}:\mathscr{M}\to H_{\mathcal{C}}$ is nonlinear operator with a stochastic derivative $D_H\mathcal{K}:\mathscr{M}\to\text{HS}(H_{\mathcal{C}})$ such that $D_H\mathcal{T}\in\text{HS}(H_{\mathcal{C}})$ is invertible $\mu$-a.e. Then we have

$$\mathcal{D}_{\text{KL}}(\mu||\mathcal{T}^{\#}\mu^{\boldsymbol{y}}) = \int_{\mathscr{M}}\log\left(\frac{\mathrm{d}\mu}{\mathrm{d}\left(\mu^{\boldsymbol{y}}\circ\mathcal{T}\right)}(m)\right)\mathrm{d}\mu(m)\,.$$

Here, we derive the density between the pullback measure and the prior. Let $\mathscr{A}$ be any measurable subset of $\mathscr{M}$, we have

$$(\mu^{\boldsymbol{y}}\circ\mathcal{T})(\mathscr{A}) = \int_{\mathcal{T}(\mathscr{A})}\mathrm{d}\mu^{\boldsymbol{y}}(m) = \frac{1}{Z^{\boldsymbol{y}}}\int_{\mathscr{A}}\exp\left(-(\Phi^{\boldsymbol{y}}\circ\mathcal{T})(m)\right)\mathrm{d}(\mu\circ\mathcal{T})(m).$$

The existence and the formula of the Radon–Nikodym derivative between $\mu\circ\mathcal{T}$ and $\mu$ is non-trivial; see Section 6.6 of Bogachev (1998) for details. In the case where $D_H\mathcal{K}(m)$ is a trace class operator on $H_{\mathcal{C}}$ $\mu$-a.e., we have

$$\frac{\mathrm{d}(\mu\circ\mathcal{T})}{\mathrm{d}\mu}(m) = \det\nolimits_{H_{\mathcal{C}}}(D_H\mathcal{T})\exp(-\left\langle\mathcal{K}(m),m\right\rangle_{\mathcal{C}^{-1}} - \frac{1}{2}\left\|\mathcal{K}(m)\right\|^2_{\mathcal{C}^{-1}}),$$

where the determinant is taken as the product of eigenvalues with $H_{\mathcal{C}}$-orthonormal eigenbases.

Therefore, the transport objective is given by

$$\mathcal{D}_{\mathrm{KL}}(\mu||\mathcal{T}^{\#}\mu^{\boldsymbol{y}}) = \mathbb{E}_{m\sim\mu}\Big[(\Phi^{\boldsymbol{y}} \circ \mathcal{T})(m) - \log\det{}_{H_{\mathcal{C}}}(D_H\mathcal{T}) + \langle \mathcal{K}m, m\rangle_{\mathcal{C}^{-1}} + \frac{1}{2}\left\|\mathcal{K}(m)\right\|^2_{\mathcal{C}^{-1}}\Big] + C_1$$

Let us consider the perturbation of the identity $\mathcal{K}$ only acting on the latent space $\mathrm{Im}(\mathcal{E}_r) = \mathbb{R}^{d_r}$ of the projection $\mathcal{P} = \mathcal{D}_r \circ \mathcal{E}_r$ with $\mathcal{E}_r \circ \mathcal{D}_r = \mathrm{Id}_{\mathbb{R}^r}$. Then we have the following alternative definition of lazy map through $\mathcal{K}$:

$$\mathcal{K} = \underbrace{\mathcal{D}_r \circ (\mathbf{T} - \mathrm{Id}_{\mathbb{R}^{d_r}}) \circ \mathcal{E}_r}_{\substack{\text{Perturbation of the Identity} \\ \text{Transport in } \mathrm{Im}(\mathcal{P})}}, \quad \mathcal{T} = \mathrm{Id}_{\mathscr{M}} + \mathcal{K},$$

where $\mathcal{D}_r$ and $\mathcal{E}_r$ consists of $H_{\mathcal{C}}$-orthonormal reduced bases. In this case, we have

$$\begin{aligned}
\log\det(D_H\mathcal{T}(m)) &\implies \log\det(\nabla\mathbf{T}(\mathcal{E}_r m)),\\
\langle \mathcal{K}(m), m\rangle_{\mathcal{C}^{-1}} &\implies (\mathcal{E}_r m)^\top \mathbf{T}(\mathcal{E}_r m) - \left\|\mathcal{E}_r m\right\|^2,\\
\frac{1}{2}\left\|\mathcal{K}(m)\right\|^2_{\mathcal{C}^{-1}} &\implies \frac{1}{2}\left\|\mathbf{T}(\mathcal{E}_r m)\right\|^2 + \frac{1}{2}\left\|\mathcal{E}_r m\right\|^2 - (\mathcal{E}_r m)^\top \mathbf{T}(\mathcal{E}_r m).
\end{aligned}$$

Therefore we have

$$\begin{aligned}
&\mathcal{D}_{\mathrm{KL}}(\mu||\mathcal{T}^{\#}\mu^{\boldsymbol{y}})\\
=&\mathbb{E}_{m\sim\mu}\left[(\Phi^{\boldsymbol{y}} \circ \mathcal{T})(m) - \log\det(\nabla\mathbf{T}(\mathcal{E}_r m)) + \frac{1}{2}\left\|\mathbf{T}(\mathcal{E}_r m)\right\|^2\right] + C_2\\
=&\mathbb{E}_{(\boldsymbol{z},m_\perp)\sim\pi\otimes\mu_\perp}\Big[(\Phi^{\boldsymbol{y}}\left(\left(\mathcal{D}_r \circ \mathbf{T}\right)(\boldsymbol{z}) + m_\perp\right) - \log\det(\nabla\mathbf{T}(\boldsymbol{z})) + \frac{1}{2}\left\|\mathbf{T}(\boldsymbol{z})\right\|^2\Big] + C_2,
\end{aligned}$$

where $\mu_\perp = (\mathrm{Id}_{\mathscr{M}} - \mathcal{P})_{\#}\mu$ is the pushforward measure in the complimentary space.

## Appendix D. Proofs of Theorems 4 and 5 and Corollary 6

### D.1 Proof of Theorem 4

**Proof** Cui and Zahm (2021, Proposition 4.1) give the following equality:

$$\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\mathcal{D}_{\mathrm{KL}}(\mu^{\boldsymbol{y}}||\widetilde{\mu}^{\boldsymbol{y}})\right] = \frac{1}{2}\mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right] - \mathcal{D}_{\mathrm{KL}}(\gamma||\widetilde{\gamma}).$$

Using triangle and Cauchy–Schwarz inequalities, we have

$$\frac{1}{2}\mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right] \leq \mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m) - \widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right] + \mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\left\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z}) - \mathbf{g}(\boldsymbol{z})\right\|^2\right].$$

Furthermore, Cao et al. (2025, Proposition 7) give the following upper bound:

$$\mathbb{E}_{m\sim\mu}\left[\left\|\boldsymbol{\mathcal{G}}(m)-\widetilde{\boldsymbol{\mathcal{G}}}_{\mathrm{opt}}(\mathcal{P}m)\right\|^2_{\Gamma_n^{-1}}\right]\le \mathrm{Tr}_{H_{\mathcal{C}}}\left(\left(\mathrm{Id}_{H_{\mathcal{C}}}-\mathcal{P}\right)\mathcal{H}_A\left(\mathrm{Id}_{H_{\mathcal{C}}}-\mathcal{P}\right)\right),$$

which completes the proof. ■

### D.2 Proof of Theorem 5 and Corollary 6

**Proof** [Theorem 5] First, since the density between $\mathbf{\mathsf{T}}_{\boldsymbol{\theta}\sharp}\pi$ and $\pi$ is essentially bounded we have the following bound for any $f\in L^1(\pi)$ due to a change-of-variables formula and the Hölder inequality:

$$\mathbb{E}_{\boldsymbol{x}\sim\mathbf{\mathsf{T}}_{\boldsymbol{\theta}\sharp}\pi}\left[|f(\boldsymbol{x})|\right]\le C_1\mathbb{E}_{\boldsymbol{z}\sim\pi}\left[|f(\boldsymbol{z})|\right],$$

where $C_1$ is the essential supremum of the density. Next, we have

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}_1^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}_1^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta})&=\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top(\nabla\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})^\top\left((\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})-\boldsymbol{V}^*\boldsymbol{y}\right)\\&\quad-\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top(\nabla\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})^\top\left((\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})-\boldsymbol{V}^*\boldsymbol{y}\right)\\&=\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top\left((\nabla\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})-(\nabla\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})\right)^\top\left((\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})\right.\\&\qquad\qquad\left.-\boldsymbol{V}^*\boldsymbol{y}\right)\\&\quad+\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top(\nabla\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})^\top\left((\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})-(\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})\right).\end{aligned}$$

By Jensen's, triangle, and Hölder's inequalities, we have

$$\begin{aligned}\left(\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}^{\boldsymbol{y}}(\boldsymbol{\theta})\right\|\right]\right)^2&\le\left(\mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})\sim\gamma\otimes\pi}\left[\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_1^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}_1^{\boldsymbol{y}}(\boldsymbol{z},\boldsymbol{\theta})\right\|\right]\right)^2\\&\le\max\{C_2,C_3\}\mathbb{E}_{\boldsymbol{x}\sim\mathbf{\mathsf{T}}_{\boldsymbol{\theta}\sharp}\pi}\Big[\left\|\mathbf{\mathsf{g}}_{\mathrm{opt}}(\boldsymbol{x})-\mathbf{\mathsf{g}}(\boldsymbol{x})\right\|^2\\&\qquad+\left\|\nabla\mathbf{\mathsf{g}}_{\mathrm{opt}}(\boldsymbol{x})-\nabla\mathbf{\mathsf{g}}(\boldsymbol{x})\right\|_F^2\Big],\end{aligned}$$

where the constants are given by

$$\begin{aligned}C_2&=\mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\|(\nabla\mathbf{\mathsf{g}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})\partial_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2\right],\\C_3&=\mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})\sim\gamma\otimes\pi}\left[\left\|\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{z})^\top\left((\mathbf{\mathsf{g}}_{\mathrm{opt}}\circ\mathbf{\mathsf{T}}_{\boldsymbol{\theta}})(\boldsymbol{z})-\boldsymbol{V}^*\boldsymbol{y}\right)\right\|^2\right].\end{aligned}$$

Since $\boldsymbol{\mathcal{G}},\widetilde{\boldsymbol{\mathcal{G}}}\circ\mathcal{P}\in H^1_\mu(\mathscr{M};H_{\Gamma_n})$, we have $\mathbf{\mathsf{g}}_{\mathrm{opt}},\mathbf{\mathsf{g}}\in H^1_\pi(\mathbb{R}^{d_r};\mathbb{R}^{d_{\boldsymbol{y}}})$. Moreover, since $C_4=\operatorname{ess\,sup}_{\boldsymbol{z}\in\mathbb{R}^{d_r}}\|\nabla_{\boldsymbol{\theta}}\mathbf{\mathsf{T}}(\boldsymbol{z})\|<\infty$, we have

$$C_2\le C_1C_4\mathbb{E}_{\boldsymbol{z}\sim\pi}\left[\|\nabla\mathbf{\mathsf{g}}(\boldsymbol{z})\|^2\right]<\infty,\quad C_3\le C_1C_4\mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})\sim\gamma\otimes\pi}\left[\left\|\mathbf{\mathsf{g}}_{\mathrm{opt}}(\boldsymbol{z})-\boldsymbol{V}^*\boldsymbol{y}\right\|^2\right]<\infty.$$

Therefore, we have

$$\begin{aligned}\left(\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta})-\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}^{\boldsymbol{y}}(\boldsymbol{\theta})\right\|\right]\right)^2\le\max\{C_2,C_3\}C_1C_4\mathbb{E}_{\boldsymbol{z}\sim\pi}\Big[&\left\|\mathbf{\mathsf{g}}_{\mathrm{opt}}(\boldsymbol{z})-\mathbf{\mathsf{g}}(\boldsymbol{z})\right\|^2\\&+\left\|\nabla\mathbf{\mathsf{g}}_{\mathrm{opt}}(\boldsymbol{z})-\nabla\mathbf{\mathsf{g}}(\boldsymbol{z})\right\|_F^2\Big].\end{aligned}$$

■

**Proof** [Corollary 6] By the Polyak–Lojasiewicz inequality and the results in Part I, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\sqrt{\mathcal{L}^{\boldsymbol{y}}(\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger}) - \mathcal{L}^{\boldsymbol{y}}(\boldsymbol{\theta}^{\boldsymbol{y},\dagger})}\right] &\le \mathbb{E}_{\boldsymbol{y}\sim\gamma}\left[\frac{1}{\sqrt{2\lambda^{\boldsymbol{y}}}}\left\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\widetilde{\boldsymbol{\theta}}^{\boldsymbol{y},\dagger})\right\|\right] \\ &\le \frac{1}{C_5}\max\{C_2,C_3\}C_1C_4\Big(\mathbb{E}_{\boldsymbol{z}\sim\pi}\Big[\left\|\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z})-\mathbf{g}(\boldsymbol{z})\right\|^2 \\ &\qquad\qquad + \left\|\nabla\mathbf{g}_{\mathrm{opt}}(\boldsymbol{z})-\nabla\mathbf{g}(\boldsymbol{z})\right\|_F^2\Big]\Big)^{1/2},\end{aligned}$$

where $C_5 = \text{ess inf}_{\boldsymbol{y}\sim\gamma}\sqrt{2\lambda^{\boldsymbol{y}}} > 0$. ■

## Appendix E. `LazyDINO` Implementation

In this section, we describe the algorithm for performing posterior approximation using `LazyDINO`. In Appendix E.1, we describe the *offline* surrogate construction phase. Next, in Appendix E.2, we describe the *online* solving of a BIP for a given observational data $\boldsymbol{y}$.

### E.1 Offline Phase: Surrogate Construction

**Solving the Generalized Eigenvalue Problem** We first describe the eigenvalue problem, referred to as `eigenvalue_problen` in Algorithm 1, for finding the reduced basis $\Psi_r = \{\psi_j\}_{j=1}^{d_r}$, which is used to form the encoder and decoder that maps between the full parameter space and the latent space. Motivated by Theorem 4, we take $\Psi_r$ to be composed of the leading $H_{\mathcal{C}}$-orthonormal eigenbasis functions of an MC approximation of the generalized eigenvalue problem defined in (34):

$$\mathcal{C}^{-1}\mathcal{H}_A \approx \frac{1}{N_L}\sum_{j=1}^{N_L} D\boldsymbol{\mathcal{G}}(m^{(j)})^*\Gamma_n^{-1}D\boldsymbol{\mathcal{G}}(m^{(j)}), \quad m^{(j)} \overset{\text{iid}}{\sim} \mu. \tag{57}$$

Following (35), the latent parameter space dimension $d_r \le \dim(\mathscr{M})$ is chosen to capture the dominant information in $\mathcal{H}_{\mathcal{A}}$; specifically, it is desirable to ensure that the eigenvalue tail sum is small. For many high- to infinite-dimensional BIPs, $d_r$ is expected to be small due to, e.g., Saint–Venant's principle for coercive elliptic PDEs, concentration of measure, and the low-rankness of sparse observations extracted from a PDE state.

When the discretization dimension of the problem is large, the generalized eigenvalue decomposition must be computed matrix-free. To this end, there are many suitable computational tools, such as randomized methods (Halko et al., 2011; Xiang and Zou, 2015; Saibaba et al., 2016) and Krylov subspace methods (Golub and Ye, 2002; Saad, 1981; Sorensen, 1998; van den Eshof and Sleijpen, 2004; Chowdhury, 1976). In this case, since the computation of the eigenvalue tail is intractable, one can adaptively find a sufficiently large dimension, $d_r$, such that the eigenvalues have adequately decayed, i.e., $\lambda_{d_r}/\lambda_1$ is small.

The samples needed to compute the reduced basis can be reused as part of the training data set; far fewer samples are usually needed to compute the reduced basis than are needed

```
Algorithm 1: LazyDINO: Identify Latent Space and Embed Dataset
Input:
  (i) prior distribution sampler: $\mu$, prior precision operator: $\mathcal{C}^{-1}$
  (ii) noise precision matrix: $\Gamma_n^{-1}$, observable basis $\boldsymbol{V}$
  (iii) PtO map: $m \mapsto \boldsymbol{\mathcal{G}}(m)$, Jacobian action: $D\boldsymbol{\mathcal{G}}(m)$
  (iv) # desired training dataset samples: $N \in \mathbb{N}$
  (v) # samples to compute encoder/decoder: $N_L \leq N$
  (vi) embedding dimension: $d_r$ or eigenvalue tail sum tolerance: $\epsilon_L$
Output:
  (i) encoder/decoder pair: $\mathcal{E}_r, \mathcal{D}_r$
  (ii) latent space training dataset inputs: $\{\boldsymbol{z}^{(j)}\}$, outputs: $\{\boldsymbol{g}^{(j)}\}$, $\{\boldsymbol{J}_r^{(j)}\}$, $j = 1, \dots, N$
begin
    1. $m^{(j)} \sim \mu, \quad j = 1, \dots, N$                              ▷ Sample prior
    2. $\boldsymbol{\mathcal{G}}(m^{(j)}), D\boldsymbol{\mathcal{G}}(m^{(j)}), \quad j = 1, \dots, N$        ▷ Evaluate PtO map/Jacobian
    3. Create encoder/decoder:
        $\{\psi_k \in \mathscr{M}\}_{k=1}^{d_r} \leftarrow$ eigenvalue_problem($\big\{D\boldsymbol{\mathcal{G}}(m^{(j)})\big\}_{i=1}^{N_L}, \Gamma_n^{-1}, \mathcal{C}^{-1}, \epsilon_L$ or $d_r$)
      ▷ (34) and (57)
        $\mathcal{D}_r \boldsymbol{z} := \sum_{k=1}^{d_r} \boldsymbol{z}_k \psi_k, \ \mathcal{E}_r := \mathcal{D}_r^\top \mathcal{C}^{-1}$                  ▷ (16) and (18)
    4. Embed dataset:                                                  ▷ (40) and (44)
        $\boldsymbol{z}^{(j)} \leftarrow \mathcal{E}_r m^{(j)}, \ \boldsymbol{g}^{(j)} \leftarrow \boldsymbol{V}^\top \Gamma_n^{-1} \boldsymbol{\mathcal{G}}(m^{(j)}), \ \boldsymbol{J}_r^{(j)} \leftarrow \boldsymbol{V}^\top \Gamma_n^{-1} D\boldsymbol{\mathcal{G}}(m^{(j)}) \mathcal{D}_r, \quad j =$
      $1, \dots, N$
end
```

to train RB-DINO to the required low error tolerances; see Cao et al. (2025). In this case, the additional latent Jacobian samples can be directly formed via its action or adjoint action, as in (44). Specifically, once the PtO evaluation at $m^{(j)}$ is available, only $\min(d_{\boldsymbol{y}}, d_r)$ evaluations of the PtO map derivative or its adjoint action are needed for a latent Jacobian evaluation. The evaluation cost can often be reduced to a fraction of that for evaluating the PtO map for linear or highly nonlinear PDEs. For details on efficient means to form latent Jacobian matrices for PDE-constrained PtO maps, see Cao et al. (2025); Ghattas and Willcox (2021). Empirical evidence of the low relative cost of forming the latent Jacobian matrices can be seen in our numerical results in Tables 3 and 4.

Using the encoder and decoder, defined previously in terms of the reduced basis $\Psi_r$ and prior precision $\mathcal{C}^{-1}$ in (16) and (18), we embed the training data into the latent space, resulting in the whitened latent inputs $\boldsymbol{z}^{(j)}$, whitened PtO samples $\boldsymbol{g}^{(j)}$, and whitened latent Jacobian samples $\boldsymbol{J}_r^{(j)}$. A summary of these procedures is given in Algorithm 1.

RB-DINO **Surrogate Training** Next, in Algorithm 2, we train RB-DINO using the embedded data set. This involves a straightforward empirical risk minimization arising from MC estimate of (43). Any method for stochastic unconstrained optimization, e.g., stochastic gradient descent, Adam (Kingma and Ba, 2017), or second-order methods (Yao et al., 2021; O'Leary-Roseberry and Bollapragada, 2024) can be used.

```
Algorithm 2: LazyDINO: Train Surrogate Latent Representation
  Input:
    (i) training dataset inputs: $\{z^{(j)}\}$, outputs: $\{g^{(j)}\}, \{J_r^{(j)}\}$, $j = 1, \dots, N$
    (ii) untrained neural network: $\mathbf{g}_w : \mathbb{R}^{d_r} \times \mathbb{R}^{d_W} \to \mathbb{R}^{d_y}$

  Output:
    (i) trained neural network: $\mathbf{g}_{w^*}$
  begin
      1. Train $\mathbf{g}_w$ by minimizing an empirical risk:
      $w^* = \underset{w \in \mathbb{R}^{d_W}}{\operatorname{argmin}} \frac{1}{N} \sum_{j=1}^{N} \left( \left\| g^{(j)} - \mathbf{g}_w\left(z^{(j)}\right) \right\|^2 + \underbrace{\left\| J_r^{(j)} - \nabla_z \mathbf{g}_w\left(z^{(j)}\right) \right\|_F^2} \right)$
  end
```

### E.2 Online Phase: Surrogate-Driven Lazy Map Variational Inference

We perform LMVI using a stochastic approximation of the surrogate rKL objective and its gradient defined in (45a). Algorithm 3 summarizes the LazyDINO training procedure when using first order methods.

Since our latent PtO surrogate is a neural network, an MC estimate of the gradient of the rKL objective with respect to transport map parameters, i.e., $\mathbb{E}_{z \sim \pi}\left[\widetilde{\mathcal{L}}_{1,r}^{y}(z, \theta; w^*)\right]$, can be computed rapidly on GPUs, especially when the surrogate objective function gradient is a compiled batch-vectorized expression. In practice, the evaluation time of the surrogate objective function gradient is almost negligible compared to the evaluation time of the model-constrained LazyMap objective gradient.

Since iterations can be performed rapidly, we use rounds of stochastic approximation (SA) optimization, increasing the batch sample size (Bollapragada et al., 2018) and decreasing the learning rate each time for several computationally tractable iterations. We found that using such a strategy was more successful than using either small or large batch sizes alone. This strategy is similar to *retrospective approximation* (RA) (Newton et al., 2024).

## Appendix F. Laplace Approximation

We define the Laplace Approximation as the Gaussian distribution centered at the unique strong minimizer, the *maximum-a-posteriori* (MAP) estimate $m_{\text{MAP}}$, of the Onsager–Machlup functional $I_{\mu^y} : \mathscr{M} \to \mathbb{R}^+$ of $\mu^y$, assuming it exists, with covariance defined as the inverse of the Hessian operator of the functional at the MAP estimate. For the posteriors considered in this work, the Onsager–Machlup functional is the sum of the potential function $\Phi^y$ and the Onsager–Machlup functional of the Gaussian prior distribution, i.e.

$$\mu_{\text{LA}} = \mathcal{N}(m_{\text{MAP}}, \mathcal{C}_{\text{LA}}), \qquad \begin{cases} m_{\text{MAP}} = \underset{m \in \mathscr{M}}{\operatorname{argmin}}\, I_{\mu^y}(m), \\ I_{\mu^y}(m) = \Phi^y(m) + \frac{1}{2}\|m\|_{\mathcal{C}^{-1}}^2, \\ \mathcal{C}_{\text{LA}} = (D^2 I(m_{\text{MAP}}))^{-1}, \end{cases} \tag{58}$$

**Algorithm 3:** `LazyDINO`: Surrogate-Driven Training of Latent Space Transport

```
Input:
  (i) whitened latent prior sampler: $\pi$
  (ii) single-sample surrogate latent space rKL objective for observation $\boldsymbol{y}$: $\widetilde{\mathcal{L}}^{\boldsymbol{y}}_{1,r}(\cdot, \cdot\,; \boldsymbol{w}^*)$
  (iii) untrained transport map with random initial weights: $\mathbf{T}_{\boldsymbol{\theta}} : \mathbb{R}^{d_r} \to \mathbb{R}^{d_r}$, $\boldsymbol{\theta}^0$
  (iv) $J$ batch sizes, learning rates, # iterations: $(B_j, a_j, I_j), j = 1, \dots, J$
Output:
  (i) trained transport map with pushforward density: $\mathbf{T}_{\boldsymbol{\theta}^*}$, $(\mathbf{T}_{\boldsymbol{\theta}^*})_\sharp\pi$
begin
    for $j = 1, \dots, J$ do
        for $i = 1, \dots, I_j$ do
            $\{\boldsymbol{z}^{(k)}\}_{k=1}^{B_j} \sim \pi$    ▷ Sample a new stochastic batch
            $\Delta\boldsymbol{\theta}^i \leftarrow \frac{1}{B_j}\sum_{k=1}^{B_j} \nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}^{\boldsymbol{y}}_{1,r}(\boldsymbol{z}^{(k)}, \boldsymbol{\theta}^{i-1}; \boldsymbol{w}^*)$    ▷ Estimate gradient of objective function
            $\boldsymbol{\theta}^i$ ←stochastic_gradient_based_iteration($\Delta\boldsymbol{\theta}^i, (\boldsymbol{\theta}^0, \dots, \boldsymbol{\theta}^{i-1}), \alpha_j$)    ▷ e.g., Adamax
        end
        $\boldsymbol{\theta}^0 \leftarrow \boldsymbol{\theta}^{I_j}$
    end
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{I_J}$    ▷ Last parameter is approximately optimal
end
```

so long as the potential function is Lipschitz continuous. The Onsager–Machlup functional $I_{\mu^{\boldsymbol{y}}}$ generalizes the commonly known *negative log-posterior density* with respect to Lebesgue measure, $\log \pi^{\boldsymbol{y}}$, to posterior probability distributions; see, e.g., Kretschmann (2023) for a more detailed discussion. In our numerical examples, we use an efficient Inexact Newton–Conjugate Gradients numerical optimization algorithm (Dembo et al., 1982; Eisenstat and Walker, 1996; Villa et al., 2021) to find the MAP estimate, $m_{\mathrm{MAP}}$, which usually converged within $O(10) - O(100)$ inexact Newton iterations.

## Appendix G. Density-Based Diagnostics

The key to computing density-based diagnostics is to evaluate the Radon–Nikodym derivative between the posterior approximation of interest and the prior, i.e., the approximate likelihood evaluations. Here, we provide the formula for this Radon–Nikodym derivative for Laplace approximation (LA) and transport map pushforward distributions.

### G.1 Laplace Approximation

We consider the following decomposition of the LA covariance

$$\mathcal{C}_{\mathrm{LA}} = \mathcal{C} - \mathcal{D}_{\mathrm{LA}}\left(\frac{\lambda_j}{\lambda_j + 1}\delta_{jk}\right)\mathcal{E}_{\mathrm{LA}}\mathcal{C}, \quad \mathcal{C}_{\mathrm{LA}}^{-1} = \mathcal{C}^{-1} + \mathcal{C}^{-1}\mathcal{D}_{\mathrm{LA}}(\lambda_j\delta_{jk})\mathcal{E}_{\mathrm{LA}},$$

where $\mathcal{D}_{\mathrm{LA}}$ and $\mathcal{E}_{\mathrm{LA}}$ are the linear encoder and decoder based on the eigendecomposition of the prior-preconditioned Hessian of the potential at the MAP point $m_{\mathrm{MAP}}$, and $\mathbf{\Lambda}_{\mathrm{LA}} = \lambda_j\delta_{ij}$ is a diagonal matrix consisting of eigenvalues. The Radon–Nikodym derivative between $\mu_{\mathrm{LA}}$ and the prior $\mu$ is given by

$$\begin{aligned}\frac{\mathrm{d}\mu_{\mathrm{LA}}}{\mathrm{d}\mu}(m) &= \frac{\mathrm{d}\mu_{\mathrm{LA}}}{\mathrm{d}\mathcal{N}(0,\mathcal{C}_{\mathrm{LA}})} \times \frac{\mathrm{d}\mathcal{N}(0,\mathcal{C}_{\mathrm{LA}})}{\mathrm{d}\mu}\\ &= \exp\Big( - \frac{1}{2}\left\|m_{\mathrm{MAP}}\right\|^2_{\mathcal{C}^{-1}} - \frac{1}{2}\left\|\mathcal{E}_{\mathrm{LA}} m_{\mathrm{MAP}}\right\|^2_{\mathbf{\Lambda}_{\mathrm{LA}}} + (\mathcal{E}_{\mathrm{LA}} m_{\mathrm{MAP}})^\top \mathbf{\Lambda}_{\mathrm{LA}} (\mathcal{E}_{\mathrm{LA}} m) \qquad (59)\\ &\quad + \langle m_{\mathrm{MAP}}, m\rangle_{\mathcal{C}^{-1}} + \frac{1}{2}\sum_j \log(1+\lambda_j) - \frac{1}{2}\left\|\mathcal{E}_{\mathrm{LA}} m\right\|^2_{\mathbf{\Lambda}_{\mathrm{LA}}}\Big).\end{aligned}$$

For example, the rKL between the LA and the true posterior is given by

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}(\mu_{\mathrm{LA}}||\mu^{\boldsymbol{y}}) &= \mathbb{E}_{m\sim\mu_{\mathrm{LA}}}\left[\log\left(\frac{\mathrm{d}\mu_{\mathrm{LA}}}{\mathrm{d}\mu}(m)\frac{\mathrm{d}\mu}{\mathrm{d}\mu^{\boldsymbol{y}}}(m)\right)\right]\\ &= \mathbb{E}_{m\sim\mu_{\mathrm{LA}}}\left[\Phi^{\boldsymbol{y}}(m) + \log\left(\frac{\mathrm{d}\mu_{\mathrm{LA}}}{\mathrm{d}\mu}(m)\right)\right] + \log Z^{\boldsymbol{y}},\end{aligned}$$

where the Radon–Nikodym derivative at parameters samples can be computed using (59).

### G.2 Lazy Map Pushforward Posterior

Let $\mathcal{T}$ be a lazy map, then we have

$$\frac{\mathrm{d}\mathcal{T}_\sharp\mu}{\mathrm{d}\mu}(m) = \frac{\mathbf{T}_\sharp\pi(\mathcal{E}_r m)}{\pi(\mathcal{E}_r m)} = \frac{(\pi\circ\mathbf{T}^{-1})(\mathcal{E}_r m)|\det\nabla\mathbf{T}^{-1}(\mathcal{E}_r m)|}{\pi(\mathcal{E}_r m)}, \qquad (60)$$

where $\mathbf{T}$ is the latent space transport map. For example, the rKL between the lazy map pushforward and the posterior is given by:

$$\begin{aligned}D_{\mathrm{KL}}(\mathcal{T}_\sharp\mu||\mu^{\boldsymbol{y}}) &= \mathbb{E}_{m\sim\mathcal{T}_\sharp\mu}\left[\log\left(\frac{\mathrm{d}\mathcal{T}_\sharp\mu}{\mathrm{d}\mu}(m)\frac{\mathrm{d}\mu}{\mathrm{d}\mu^{\boldsymbol{y}}}(m)\right)\right]\\ &= \mathbb{E}_{m\sim\mathcal{T}_\sharp\mu}\left[\Phi^{\boldsymbol{y}}(m) + \log\left(\frac{\mathrm{d}\mathcal{T}_\sharp\mu}{\mathrm{d}\mu}(m)\right)\right] + \log Z^{\boldsymbol{y}},\end{aligned}$$

where the Radon–Nikodym derivative at parameters samples can be computed using (60).

## Appendix H. Details of the Training Procedures

**Surrogate Training.** For `RB-DINO`, we used a learning rate of $1\times10^{-3}$ and decreased it to $3\times10^{-4}$ for the final 375 epochs. For `RB-NO` in Example I, the learning rate $\alpha^j$ and epoch number $E^j$ for each training data size $N^j$, reported here as ($\alpha^j$, $E^j$, $N^j$) are $\Big\{(2\times10^{-4}, 1500, 125), (2\times10^{-4}, 1500, 250), \ldots (2\times10^{-4}, 1500, 8k), (2\times10^{-4}, 500, 16k)\Big\}$. For `RB-NO` in Example II, we used $\Big\{((1\times10^{-4}, 3000, 125), (1\times10^{-4}, 3000, 250), \ldots (1\times10^{-4}, 3000, 4k), (1\times10^{-4}, 5000, 8k), (5\times10^{-5}, 5000, 16k)\Big\}$. We used the validation set to ensure that training with these parameters yielded similar training and generalization errors.

**Transport Map Training** For `LazyDINO` and `LazyNo`, we employ Adamax (Kingma and Ba, 2017) with 5 batch sizes, as defined in Algorithm 3. For minimization $j$, we use $I_j$ iterations using a $B_j$–sample MC gradient estimator and learning rate $\alpha_j$, labeled here as $(I_j, B_k, \alpha_j)$: $\Big\{(5k, 200, 5\times 10^{-3}), (1k, 500, 5\times 10^{-3}), (1k, 2,000, 5\times 10^{-3}), (1k, 5,000, 5\times 10^{-3}), (1k, 7,500, 5\times 10^{-4})\Big\}$, where we decrease the learning rate slightly when we reach the final stochastic approximation batch sample size $7,500$. For `LazyMap`, since each Adamax iteration involves PtO map evaluations (referred to as training samples in our results), we use only one batch size, $(I_0, B_0, \alpha_0) = (200, 640, 5\times 10^{-3})$ for Example I and $(I_0, B_0, \alpha_0) = (200, 100, 5\times 10^{-3})$ for Example II, for a total of 128,000 and 20,000 PtO map evaluations, respectively. For comparison in the numerical result sections, error measures are recorded after steps $\#(5, 10, \ldots, 320, 640)$, which equal training sample sets of size $(1,000, 2,000, \ldots, 64,000, 128,000)$, respectively. For A-SBI, we train in batches of size 100, sampled without replacement, over epochs with a fixed learning rate of $5\times 10^{-4}$. We terminated optimization when the validation error did not decrease for 10 epochs.

## Appendix I. Additional Numerical Results

We first visualize discrepancies in the mean, MAP point, and pointwise variances across different methods and training sample sizes for BIP #2 of Example I in Figures 22 to 24 and for BIP #4 of Example II in Figures 25 to 27. These comparisons allow the methods to be visually differentiated in their ability to resolve features in the inferred parameter fields. The figures show that `LazyDINO` can capture these posterior predictives somewhat faithfully for 250 samples, while other methods struggle with orders of magnitude more samples.

In Figure 28, we visualize the eigenvalues used for latent space identification in (34). The resulting decoder and encoder are visualized in Figures 29 and 30 for Examples I and II, respectively. Notably, the location of the pointwise output observation is visible in the plots of encoder columns, which reflects the fact that the derivative-based dimension reduction seeks to explain the variance in the observables as opposed to the parameter only.

Lastly, we showcase the capability of `LazyDINO` for accurate posterior approximation through expanded posterior marginal plots in Figures 31 and 32 for BIP #2 of Example I and BIP #4 of Example II, respectively. These plots show that in the ten leading latent space coordinates, the true posterior marginals are matched remarkably well by `LazyDINO` at 16k training samples. To understand the difficulty of these BIPs, we also visualize these posterior marginals along with the prior contours in Figures 33 and 34. These plots show that the difference between the prior and posterior is indeed significant.

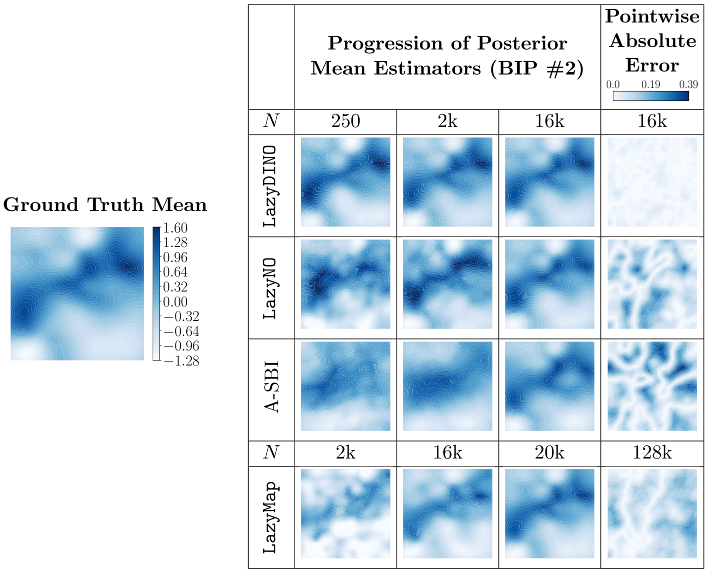

Figure 22: **Example I: Progression of Posterior Mean Estimators.**

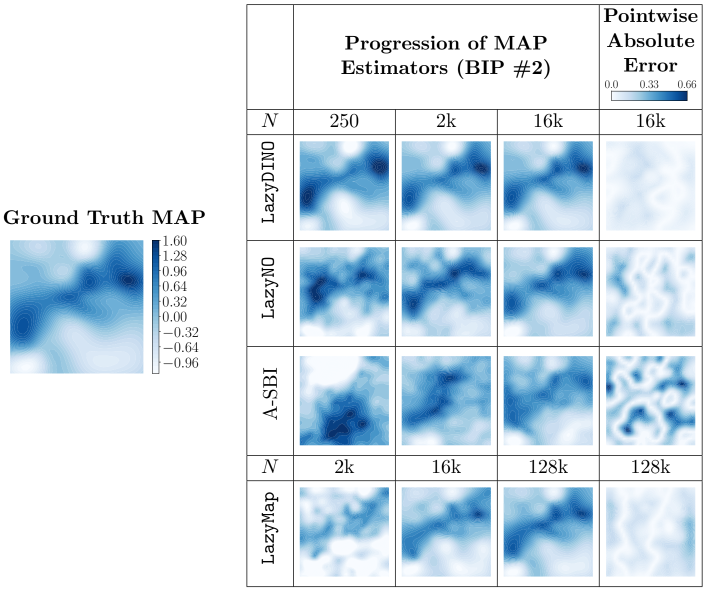

Figure 23: **Example I: Progression of MAP Estimators.**

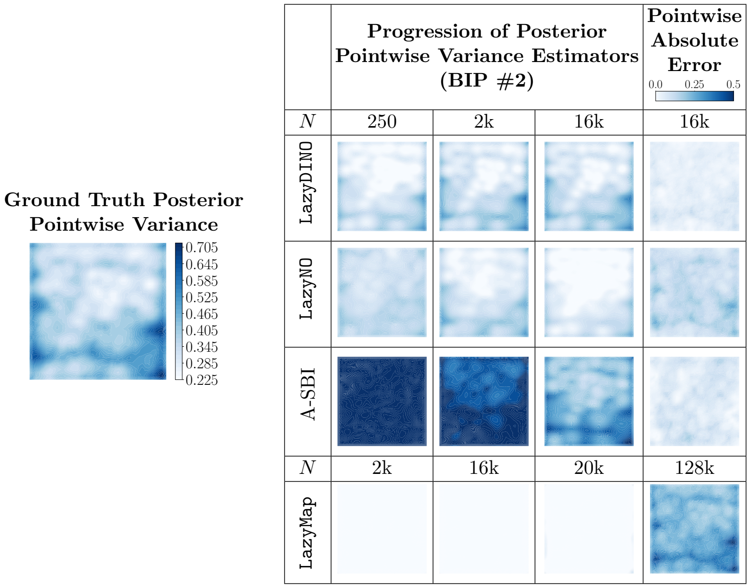

Figure 24: **Example I: Progression of Posterior Pointwise Variance Estimators.**

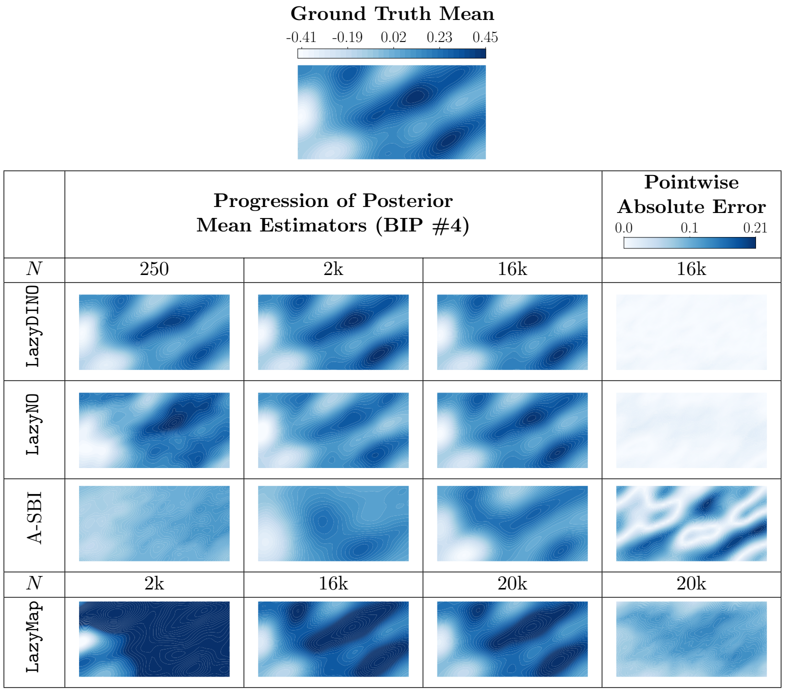

Figure 25: **Example II: Progression of Posterior Mean Estimators.**

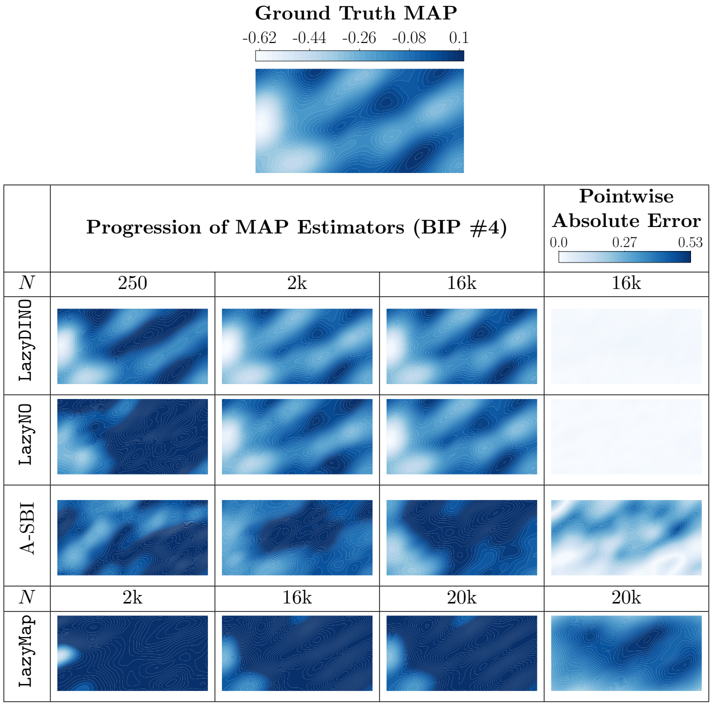

Figure 26: **Example II: Progression of MAP Estimators.**

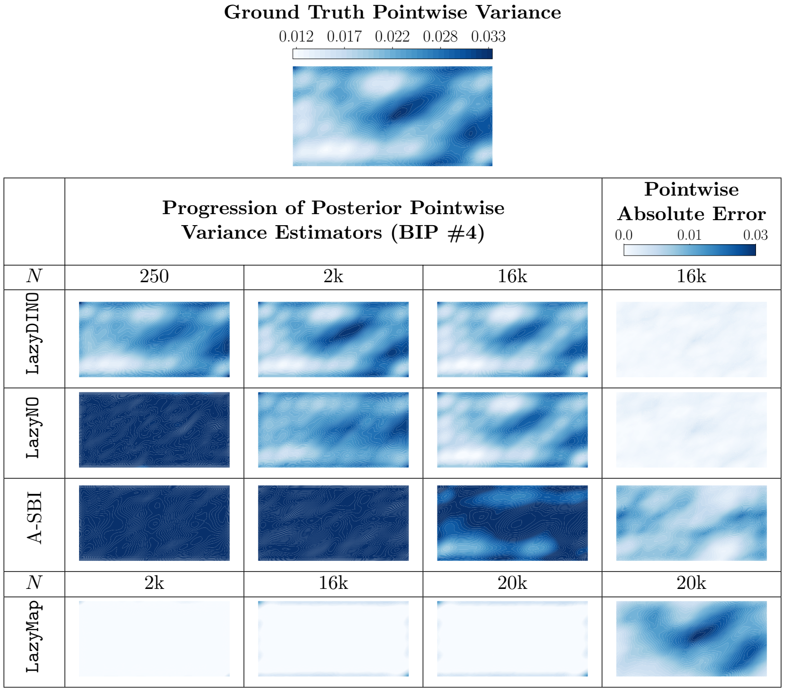

Figure 27: **Example II: Progression of Posterior Pointwise Variance Estimators.**

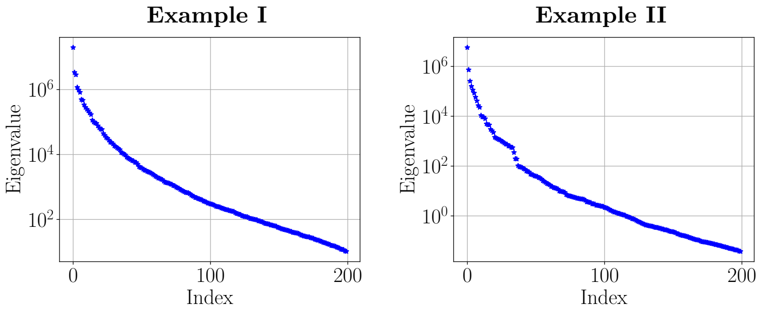

Figure 28: Visualization of eigenvalue decay for the generalized eigenvalue problem in (34) for subspace identification in the two numerical examples.

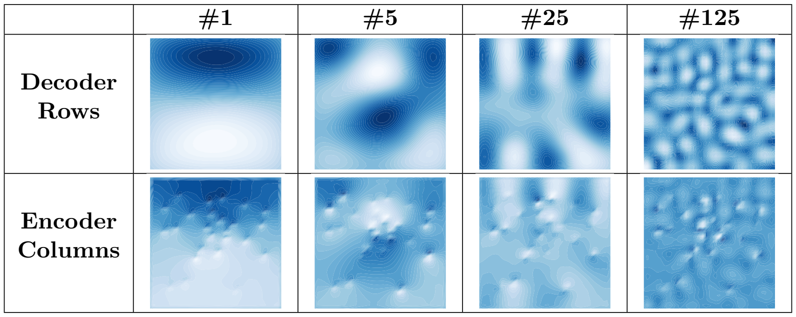

Figure 29: **Example I: Selected Decoder Rows and Encoder Columns.** The encoder columns are computed by applying the prior precision operator to the decoder rows. The encoder action on input is given by the vector space inner product of the encoder columns (discretized) on the input (discretized).

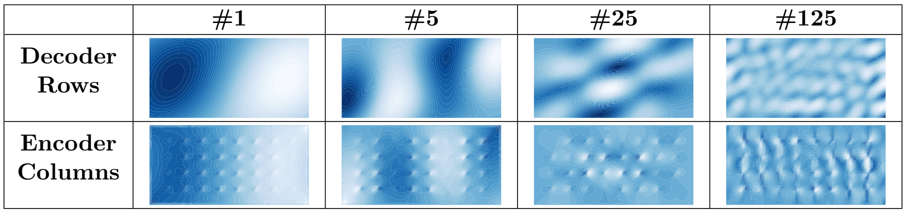

Figure 30: **Example II: Selected Decoder Rows and Encoder Columns.** See comments in Figure 29 for interpretation.

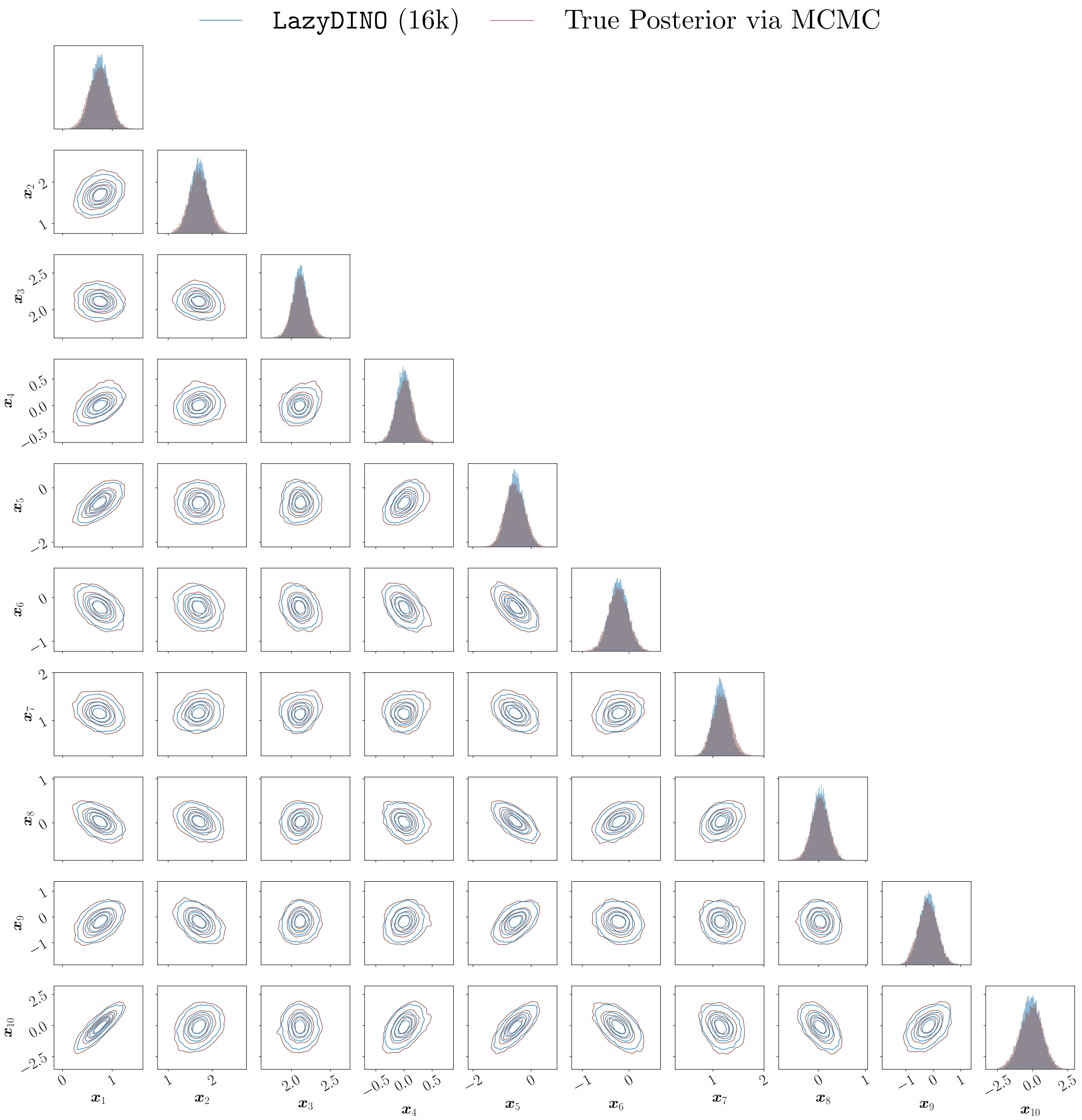

Figure 31: **Example I and BIP #2:** `LazyDINO` **Posterior Marginals.** These coordinates correspond to the leading coordinates parameter latent space with the largest eigenvalues of (34).

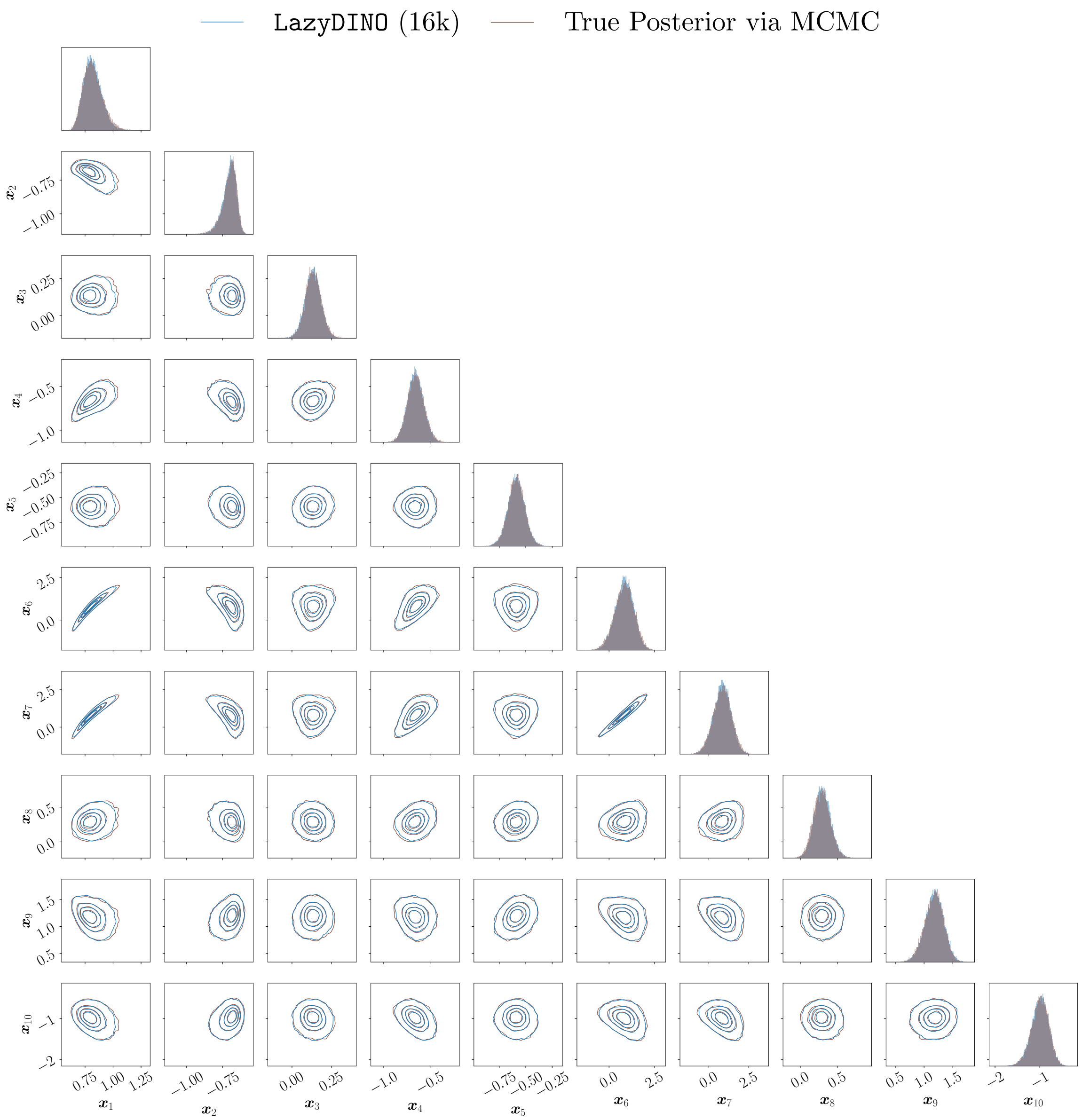

Figure 32: **Example II and BIP #4: LazyDINO Posterior Marginals.** These coordinates correspond to the leading coordinates parameter latent space with the largest eigenvalues of (34).

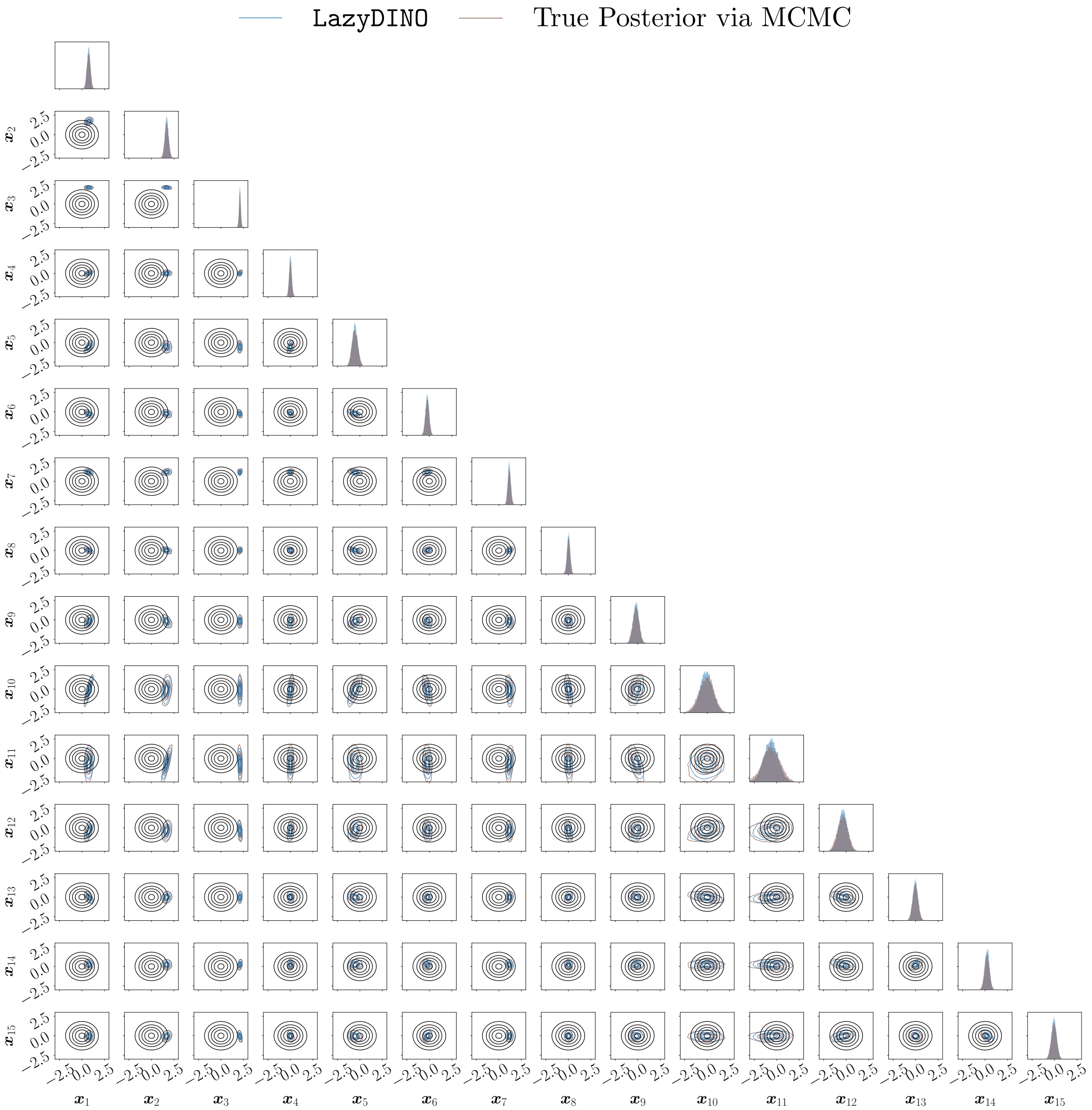

Figure 33: **Example I and BIP #2: `LazyDINO` Posterior Marginals vs. Prior Marginals.** These coordinates correspond to the leading coordinates parameter latent space with the largest eigenvalues of (34). The black contours represent the prior, which is standard normal in the latent space coordinates.

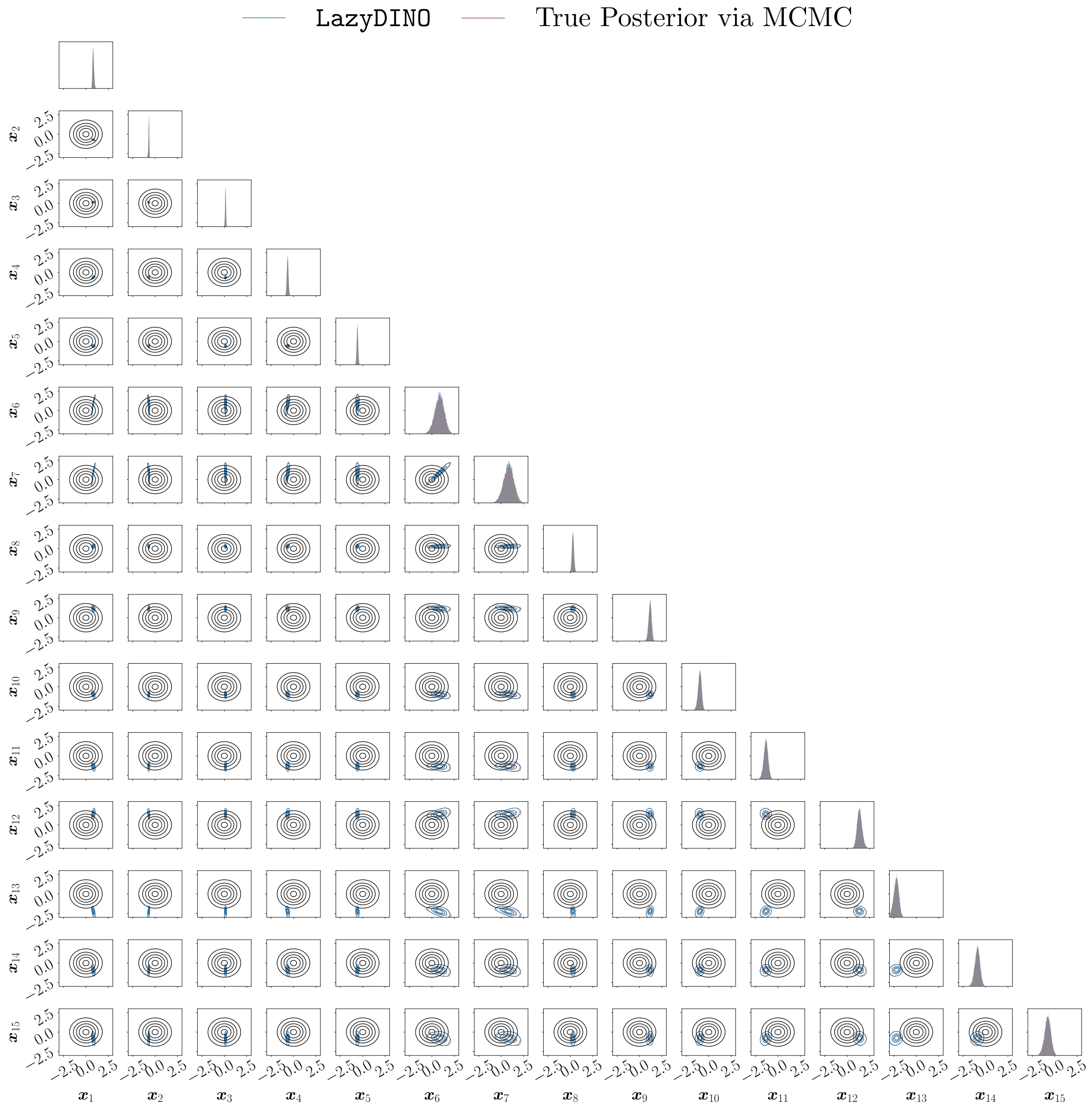

Figure 34: **Example II BIP #4:** `LazyDINO` **Posterior Marginals vs. Prior Marginals.** These coordinates correspond to the leading coordinates parameter latent space with the largest eigenvalues of (34). The black contours represent the prior, which is standard normal in the latent space coordinates.