SIAM J. SCI. COMPUT.
Vol. 47, No. 4, pp. C932–C958

EFFICIENT PRIOR CALIBRATION FROM INDIRECT DATA*

O. DENIZ AKYILDIZ†, MARK GIROLAMI‡, ANDREW M. STUART§, AND ARNAUD VADEBONCOEUR¶

Abstract. Bayesian inversion is central to the quantification of uncertainty within problems arising from numerous applications in science and engineering. To formulate the approach, four ingredients are required: a *forward model* mapping the unknown parameter to an element of a solution space, often the solution space for a differential equation; an *observation operator* mapping an element of the solution space to the data space; a *noise model* describing how noise pollutes the observations; and a *prior model* describing knowledge about the unknown parameter before the data is acquired. This paper is concerned with learning the prior model from data, in particular, learning the prior from multiple realizations of indirect data obtained through the noisy observation process. The prior is represented, using a generative model, as the pushforward of a Gaussian in a latent space; the pushforward map is learned by minimizing an appropriate loss function. A metric that is well-defined under empirical approximation is used to define the loss function for the pushforward map to make an implementable methodology. Furthermore, an efficient residual-based neural operator approximation of the forward model is proposed and it is shown that this may be learned concurrently with the pushforward map, using a bilevel optimization formulation of the problem; this use of neural operator approximation has the potential to make prior learning from indirect data more computationally efficient, especially when the observation process is expensive, nonsmooth, or not known. The ideas are illustrated with the Darcy flow inverse problem of finding permeability from piezometric head measurements.

Key words. inverse problems, generative models, prior learning, operator learning, differential equations

MSC codes. 68T37, 65N30, 62F15, 35R30

DOI. 10.1137/24M166485X

1. Introduction.

1.1. Setup. This paper is concerned with learning a generative model for unobserved $\{z^{(n)}\}_{n=1}^N$ from indirect and noisy data $\{y^{(n)}\}_{n=1}^N$ given by

$$y^{(n)} = \mathcal{G}(z^{(n)}) + \varepsilon_\eta^{(n)}, \tag{1.1}$$

*Submitted to the journal's Machine Learning Methods for Scientific Computing section May 28, 2024; accepted for publication (in revised form) May 13, 2025; published electronically August 4, 2025.

https://doi.org/10.1137/24M166485X

Funding: The second author is supported by a Royal Academy of Engineering Research Chair and EPSRC grants EP/X037770/1, EP/Y028805/1, EP/W005816/1, EP/V056522/1, EP/V056441/1, EP/T000414/1, and EP/R034710/1. The third author is grateful for support through a Department of Defense Vannevar Bush Faculty Fellowship, from the Air Force Office of Scientific Research under MURI award FA9550-20-1-0358 (Machine Learning and Physics-Based Modeling and Simulation), and through the SciAI Center, funded by the Office of Naval Research (ONR), under grant N00014-23-1-2729. The fourth author is supported through EPSRC ROSEHIPS grant EP/W005816/1.

†Department of Mathematics, Imperial College London, London, SW7 2AZ UK (deniz.akyildiz@imperial.ac.uk).

‡Alan Turing Institute, London, NW1 2DB UK, and Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ UK (mag92@cam.ac.uk).

§Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125 USA (astuart@caltech.edu).

¶Corresponding author. Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ UK (av537@cam.ac.uk).

where the noise $\varepsilon_\eta^{(n)} \sim \eta$ i.i.d. The setting where $\mathcal{G}(\cdot)$ is the identity and the noise is zero is the standard problem of generative modeling and is well studied; in the context of this paper, Bayesian inversion, such a generative model may be used to construct a prior measure from prior samples $\{z^{(n)}\}_{n=1}^N$. However, in many science and engineering applications the prior samples $\{z^{(n)}\}_{n=1}^N$ are not directly observed, but the $\{y^{(n)}\}_{n=1}^N$, arising from (1.1) with choice $\mathcal{G} = g \circ F^\dagger$ originating from multiple instantiations of physical systems, are available. Thus, we are interested in the setting where

$$y^{(n)} = g \circ F^\dagger(z^{(n)}) + \varepsilon_\eta^{(n)}. \tag{1.2}$$

Here, $F^\dagger : Z \mapsto U$ maps between function spaces Z, U, representing a PDE parameter-to-solution operator, and $g : U \mapsto \mathbb{R}^{d_y}$ is a solution-to-data map. From the resulting finite-dimensional data we wish to construct a generative model for a prior measure on a function space, giving rise to the unobserved prior samples $\{z^{(n)}\}_{n=1}^N$.

To overview our approach to this problem we first describe it at a population loss level. Let ν denote the law of the $y^{(n)}$ and μ the desired law of the $z^{(n)}$. Letting $*$ denote convolution of measures and $\#$ denote pushforward, introducing divergence d_1 between probability measures in data space and $\mathcal{H}(\cdot)$ the regularization term on measures in the input space, we define population loss

$$\mathsf{J}_1(\mu) = \mathsf{d}_1\Big(\nu, \eta * (g \circ F^\dagger)_\# \mu\Big) + \mathcal{H}(\mu). \tag{Functional 1}$$

To develop algorithms to find μ we will represent μ as pushforward under map T^α of Gaussian measure μ_0 on a latent space.[1] Here $\alpha \in \mathbb{R}^{d_\alpha}$ represents a finite-dimensional parameterization of the pushforward from μ_0 to μ. We replace the regularization term $\mathcal{H}(\cdot)$ on measure μ by regularization term $h : \mathbb{R}^{d_\alpha} \to \mathbb{R}$ on α. We then consider the modified population loss

$$\mathsf{J}_2(\alpha) = \mathsf{d}_1\Big(\nu, \eta * (g \circ F^\dagger \circ T^\alpha)_\# \mu_0\Big) + h(\alpha). \tag{Functional 2}$$

We also observe that $F^\dagger$ may be expensive to compute and it may be desirable to replace it by a neural operator F^ϕ whose parameters $\phi \in \mathbb{R}^{d_\phi}$ need to be learned so that $F^\phi \approx F^\dagger$ in sets of high probability under μ. But we do not know μ, indeed we are trying to find it; thus the optimal parameters ϕ will depend on α and be defined by $\phi = \phi^*(\alpha)$. We thus introduce loss function

$$\mathsf{J}_3(\alpha) = \mathsf{d}_1\Big(\nu, \eta * (g \circ F^{\phi^*(\alpha)} \circ T^\alpha)_\# \mu_0\Big) + h(\alpha). \tag{Functional 3}$$

At the heart of all these loss functions is a matching of distributions. In practice both ν and the pushforward of μ_0 will only be available empirically and so it is necessary that the divergence d_1 can be readily evaluated on empirical measures. Empiricalized versions of the functionals $\mathsf{J}_2, \mathsf{J}_3$ will form the basis of the computational methodology proposed in this paper. The mapping $\phi^*(\alpha)$ will also be learned using minimization of a loss function, involving matching of distributions and evaluated empirically. The functional J_1 provides a theoretical underpinning of our approach and in the case $N = 1$ will be linked, in the empirical setting, to Bayes' theorem. The efficiency

[1] Other generative models replacing the Gaussian with different, but also straightforward to sample, measures can easily be accommodated; we choose a Gaussian in the latent space to make the presentation explicit.

of the proposed methodology is due to the following: (i) the replacement of costly PDE simulations with evaluations of a concurrently learned surrogate model trained through readily computable PDE residuals; (ii) as will be mentioned in Remark 3.3, the proposed method scales more favorably than alternative Bayesian approaches due to the challenges of high dimensional posterior sampling.

Subsection 1.2, which follows, summarizes our contributions and outlines the paper. In subsection 1.3 we review relevant literature in the area and in subsection 1.4 we overview the notation and model problem (Darcy) used in the paper.

1.2. Contributions and outline. The proposed novel methodology allows the construction of a calibrated measure over the parameters underlying a PDE model, given data from a *collection* of physical systems. Furthermore, the methodology may be combined with a novel concurrent neural operator approximation of the PDE. Together these ideas hold the potential to improve the accuracy and efficiency of Bayesian inversion and of generative modeling for physical systems. The work can be broken down into five primary contributions:

1. We introduce a suitable choice for divergence d_1 based on sliced-Wasserstein-2 distance and demonstrate that it leads to computationally feasible objective J_2 (Functional 2).
2. We introduce a residual-based probabilistic loss function to define choice of parameters $\phi^*(\alpha)$ in the neural operator approximation.
3. With this definition of $\phi^*(\alpha)$ we demonstrate a computationally feasible objective J_3 (Functional 3).
4. We show that, with our choice of d_1, minimization of Functional 1 may be linked to the Bayes' theorem when $N = 1$.
5. In order to be concrete we describe our methodology in the context of the Darcy flow model of porous medium flow which may be viewed as a mapping from the permeability field (z) to linear functionals of the piezometric head (y). Numerical experiments with Darcy flow, for two different choices of pushforward families T^α, are used to demonstrate feasibility and consistency of the proposed methodology.

In Section 2 we describe the efficient residual-based approach to operator learning that we adopt in this paper, addressing contribution 2. Section 3 introduces specific divergences for definition of Functionals 1–3 and, for the residual-based learning, addressing contributions 1 and 3; we also address contribution 4 in Theorem 3.4. Section 4 discusses algorithmic details, giving further detail on contributions 1 and 3, while Section 5 implements the algorithms on the Darcy flow problem, contribution 5. We conclude and discuss future works in Section 6.

1.3. Literature review. We overview relevant literature. First, we discuss the learning of priors from data. Second, we describe the surrogate modeling literature. And finally, we discuss related transport-based inference methods.

1.3.1. Learning priors. The task of selecting a *best prior* has received much attention in the Bayesian statistics community, with many objectives and ideals in mind [34]. Certain efforts concentrate on the careful formulation of uninformative priors [35, 5, 6]. Others focus on mathematical tractability through conjugacy [56] or eliciting priors from domain experts [53]. Some see the prior as an opportunity to share information from observations with possibly different underlying parameters, that are assumed to be drawn from the same distribution. Taking this point of view are methods related to hierarchical Bayes [29], empirical Bayes [59, 60], and

parametric empirical Bayes [15, 49]. In many ways, the idea of using a set of data to explicitly target an unconditional distribution is the basis of many modern generative modeling methods in the field of machine learning (ML). These include score-based models [69], diffusion models [67, 31], variational autoencoders [38], energy-based models [68, 72], normalizing flows [54], gradient flows [10], and more. Data-based unconditional distributions like these have been shown to be not only expressive generative models, but also powerful priors for Bayesian inversion [26, 11]. These ideas appear in a number of recent papers focused on inverse problems [55, 3, 2]. A key difference in our work and the basis for our contribution is that while the above works focus on learning (and using) priors from which unconditional samples are available, our methodology here extends this idea to learn priors from indirect data. This idea of indirect knowledge of priors and data is exploited in [20] and also in [28] for linear operators where the set of Bayesian inverse problems defined individually by (1.1), for each n, is solved in the case where $\mathcal{G}(\cdot) = A\cdot$ for some linear operator A; the collection of inverse problems is used to both learn prior information and learn the dependence of each individual posterior on data.

1.3.2. Surrogate modeling and operator learning. In many engineering applications related to design optimization, parameter inversion, and forward uncertainty quantification tasks, it is necessary to evaluate numerical models many times. When using a classical numerical scheme, there is no information carryover from one numerical solve operation to the next. Hence, there is room for improvement in the form of somehow interpolating information from the one numerical solution to the next. The idea of replacing computer code with a cheap-to-evaluate statistical interpolation model is well explored [61, 36]. Such methods have found their place as viable model order reduction techniques for multiquery problems. Modern advances have now begun to pose the task of surrogate modeling directly in function space, resulting in the field of operator learning [8, 47, 40, 44]. These methods are based on gathering input-output datasets of PDE parameters and PDE solutions obtain via classical numerical schemes, such as finite element models (FEMs) and spectral methods. Physics-informed surrogate modeling attempts to directly incorporate PDE information into the learning task [58, 78, 45]. The methods may then be dataless or semi-data-informed. Like physics-informed neural operators (PINO) [45] our optimization objective balances a data-driven loss with a PDE-enforcing regularizer. Unlike PINO, however, our data-driven loss is not related to operator learning; operator learning is introduced purely through the PDE-enforcing regularizer. Also, unlike PINO, which adds the loss and the regularizer, we adopt a bilevel optimization strategy because of issues related to balancing the two terms. Finally we use the variational form of the PDE to define the regularizer, demonstrating that the Fourier neural operator (FNO) interacts well with FEM-interpolation to enable derivative-based optimization. Certain classes of methods also pose prior distributions over PDE parameters and attempt to learn a surrogate trained on random draws from that prior [73, 74].

1.3.3. Pushforwards and Wasserstein losses. Minimization of regularized loss functions over the space of probability measures, as exemplified by Functional 1, is central to modern computational statistics and ML: it lies at the heart of variational inference [76] and in many other emerging inference problems for probability measures [18, 46, 43, 41, 42]. In statistical inference and ML, there are many learning objective functions to make use of. A well-known method is maximizing the marginal likelihood. However, the task we are interested in for this work is distributional learning. Hence,

we must use a statistical divergence between measures. A common choice is the Kullback–Leibler (KL) divergence. However, this is not useful when both measures being compared are empirical, due to the nonoverlapping support of sampled Diracs, as is the case in this paper (the D_{KL} is either 0 or ∞).

Noting this restriction on suitable divergences we proceed to identify appropriate choices. The maximum mean divergence [66, 24] and (closely related) energy distances [70, 64] are one possible class of metrics that could be used. In this work, we focus on the use of computationally tractable optimal transport-based metrics on the space of measures; in particular we use the sliced-Wasserstein metric [9]. Many other works have explored the use of these optimal transport based metrics for ML inference tasks [39, 22, 50, 52, 46]. There is also work on combining pushforward measures and Bayesian inference in [14, 13, 48, 25]. Solving inverse problems with Wasserstein loss is also a partially explored topic in [1] and conditional flow matching [17]. Some works have also looked at Wasserstein metrics between pushforward measures [62]. Wasserstein and sliced-Wasserstein metrics have found use in approximate Bayesian computation [71, 4, 51] where the recovered measures have been shown to converge to the Bayesian posterior. Parameter estimation with Wasserstein metrics in a purer form is explored in [7]. In discussing an optimal transport based learning objective, it is important to mention entropy regularization, as in the Sinkhorn algorithm [19]. In these works, entropy regularization is put on the finite-dimensional observational space. We will see that entropy regularization comes up differently in our proposed methodology.

1.4. Notation. Let $D \subset \mathbb{R}^d$ be bounded and open. We denote the boundary of the set D by ∂D. We use the $L^p(D)$ classes of pth power Lebesgue integrable functions, $1 \le p < \infty$, extending to $p = \infty$ in the usual way via the essential supremum. We denote by $C^\infty(D)$ the set of infinitely differentiable functions and by $H_0^1(D)$ the Sobolev space of functions with one square-integrable weak derivative and homogeneous Dirichlet boundary conditions; we denote by $H^{-1}(D)$ the dual of $H_0^1(D)$ with respect to the canonical pairing through Lebesgue integration over D.

Let $\mathcal{P}(X)$ denote the space of probability measures on measurable space X. Divergences on the space of probability measures are denoted by d, sometimes with subscript $i \in \{1, 2\}$. We denote by $f_{\#}\mu$ the pushforward measure given by $f_{\#}\mu(A) = \mu(f^{-1}(A))$ for all μ measurable sets A. We denote indexing over different instances of variables in a collection (such as a dataset or set of randomly sampled variables) with superscript in parentheses, accessing elements of a vector are done through subscript in parentheses, and incrementing (such as in summations) is done with plain subscript. We denote by $\langle \cdot, \cdot \rangle_A = \langle \cdot, A^{-1} \cdot \rangle$ the covariance weighted inner-product, for any positive self-adjoint A, with induced norm $\| \cdot \|_A$.

Consider (1.2) and assume that $z^{(n)} \sim \mu^\dagger \in \mathcal{P}(Z)$ where $Z \subset L^\infty(D)$ is separable. Let $F^\dagger : Z \to H_0^1(D)$ and let $g : H_0^1(D) \to \mathbb{R}^{d_y}$ denote a set of functionals. If we assume that $\varepsilon_\eta^{(n)} \sim \eta$ for $n = 1, \dots, N$ are i.i.d. noise variables, then $y^{(n)} \sim \eta * (g \circ F^\dagger)_{\#}\mu^\dagger$. The problem of interest is to recover from the observations $\{y^{(n)}\}_{n=1}^N$ the law $\mu^\dagger$ of the parameter field.

To be concrete we will work with inverse problems defined by the Darcy equation

$$\nabla \cdot (z \nabla u) + f = 0 \quad \forall x \in D, \tag{1.3a}$$

$$u = 0 \quad \forall x \in \partial D. \tag{1.3b}$$

Here z denotes permeability and u the piezometric head. The mapping $z \mapsto u$ may be viewed as mapping $F^\dagger : Z \to H_0^1(D)$ for appropriately defined Z. To be concrete we

will focus on this setting and the problem of determining a prior on z from noisy linear functionals of u defined by mapping $g : H_0^1(D) \to \mathbb{R}^{d_y} \cong \left(H^{-1}(D)\right)^{d_y}$. The reader will readily see that the ideas in the paper apply more generally, and that consideration of the Darcy problem is simply for expository purposes.

2. Residual-based neural operator. This section is devoted to defining $\phi^*(\alpha)$, which appears in Functional 3. Recall that ϕ are the parameters of the neural network surrogate PDE model and that their optimization depends on the underlying input measure on which the surrogate needs to be accurate; for this reason they depend on α, i.e., the set of parameters characterizing the prior we want to learn. To define function $\phi^*(\alpha)$ we proceed as follows. Let $F^\phi : Z \to H_0^1(D)$ be a parametric family of maps approximating $F^\dagger$. Now define residual operator $R : Z \times H_0^1(D) \to H^{-1}(D)$. Intuitively, we can express our PDE as $R(z, u) = 0$. In the case of the Darcy equation (1.3) we may write

$$R(z, u) = \nabla \cdot (z \nabla u) + f,$$

and we have $u \in H_0^1(D)$, $z \in Z \subset L^\infty(D)$, and $f \in L^2(D)$. Note next that we can write $u = F^\dagger(z)$ by the definition of forward map, hence we have $R(z, F^\dagger(z)) = 0$ for all $z \in Z$. In order to incorporate this information into our loss functional, we define $R^\phi : Z \to H^{-1}(D)$ as

$$R^\phi(z) = R(z, F^\phi(z)), \tag{2.1}$$

where F^ϕ is the parametric family of maps. In order to obtain a computable loss, we introduce a *discretization operator* $\mathcal{O} : H^{-1}(D) \to \mathbb{R}^{d_o}$, where d_o is the dimension of the output of $\mathcal{O}$. In particular, given a set of basis functions $\{v_i\}_{i=1}^{d_o}$ for $H_0^1(D)$, we can write[2]

$$\mathcal{O}(R)_i = \langle v_i, R \rangle = \int_D v_i R(z, u)(x) \mathrm{d}x, \tag{2.2}$$

for $i = 1, \ldots, d_o$, for any given $v_i \in H_0^1(D)$, noting that integration by parts may be used to show well-definedness in the given function space setting. To compactly represent the discretization process with our emulator F^ϕ, we define $\mathcal{O}^\phi : X \to \mathbb{R}^{d_o}$ as

$$\mathcal{O}^\phi(z) = \mathcal{O}(R^\phi(z)). \tag{2.3}$$

We are now in a position to define our final loss functional using the constructions above. To learn the optimal parameter ϕ as α varies, we define the following coupled loss functional and associated minimization problem:

(Functional 4)
$$\mathsf{J}_4(\phi; \alpha) = \mathsf{d}_2 \left(\delta_0, (\mathcal{O}^\phi \circ T^\alpha)_{\#} \mu_0)\right),$$
$$\phi^\star(\alpha) = \operatorname*{argmin}_\phi \mathsf{J}_4(\phi; \alpha);$$

here d_2 is a divergence term between probability measures on $\mathbb{R}^{d_o}$ and δ_0 is the Dirac measure at zero. Minimizing $\mathsf{J}_4(\cdot; \alpha)$ for given α determines a residual-based approximation of $F^\dagger$, accurate with respect to $(T^\alpha)_{\#}\mu_0$. Using this expression shows that Functional 3 and Functional 4 define a bilevel optimization scheme [65, 32]. This scheme is the heart of our proposed methodology. Using the bilevel approach

[2]Here $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H_0^1(D)$ and $H^{-1}(D)$.

avoids balancing the contributions of Functional 3 and Functional 4 that arise from an additive approach. A similar bilevel optimization scheme is employed in [77], for similar reasons, to solve a different problem. In the next section we provide specific examples d_1 and d_2 and discuss some of their properties.

3. Choice of divergences. In order to instantiate the bilevel optimization scheme defined by Functional 3 and Functional 4 to obtain an implementable algorithm, we now define specific choices of the divergence terms d_1 and d_2 (subsection 3.1) and discuss empirical approximation of the input measures (subsection 3.2) required to evaluate these divergences in practice. And, to further establish a context for our work on learning priors, we make a connection to Bayesian inversion in the setting of (1.1) when $N = 1$ (subsection 3.3.)

3.1. Divergences. In this work, to effectively and efficiently compare empirical measures, we will use the sliced-Wasserstein distance to define the divergence term for d_1 and Wasserstein distance for d_2. To define precisely what we do, we first introduce the weighted Wasserstein distance,

$$\mathsf{W}_{2,\mathrm{B}}^2(\nu,\mu) = \inf_{\gamma\in\Pi(\nu,\mu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x-y\|_{\mathrm{B}}^2 \mathrm{d}\gamma(x,y), \tag{3.1}$$

where coupling $\Pi(\nu,\mu)$ is the set of all joint probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals ν and μ, and B is positive and self-adjoint. We let $\mathsf{W}_2 := \mathsf{W}_{2,\mathrm{I}}$. The following lemma shows that the weighted squared Wasserstein-2 distance can be seen as the squared Wasserstein-2 distance of pushforwards of the original measures.

LEMMA 3.1. *For* $P_{\mathrm{B}}(\cdot) = \mathrm{B}^{-1/2}\cdot$ *it follows that*

$$\mathsf{W}_{2,\mathrm{B}}^2(\nu,\mu) = \mathsf{W}_2^2(P_{\mathrm{B}\#}\nu, P_{\mathrm{B}\#}\mu). \tag{3.2}$$

Proof. See Appendix A.1. ∎

We now define the weighted and sliced-Wasserstein distance by

$$\mathsf{SW}_{2,\mathrm{B}}^2(\nu,\mu) = \int_{\mathbb{S}^{d-1}} \mathsf{W}_2^2(P_{\mathrm{B}\#}^\theta\nu, P_{\mathrm{B}\#}^\theta\mu)\mathrm{d}\theta, \tag{3.3}$$

where $P_{\mathrm{B}}^\theta(\cdot) = \langle \mathrm{B}^{-\frac{1}{2}}\cdot, \theta\rangle$. The sliced-Wasserstein distance leads to a computationally efficient alternative to the Wasserstein distance because it may be implemented by Monte Carlo approximation of integration over θ and then each slice, resulting from a randomly chosen θ, involves only evaluation of a Wasserstein distance between probabilities in $\mathbb{R}$.

Assuming that $\eta = \mathcal{N}(0,\Gamma)$ we then consider the losses Functional 3 and Functional 4 with $\mathsf{d}_1(\cdot,\cdot) = \frac{d_y}{2}\mathsf{SW}_{2,\Gamma}^2(\cdot,\cdot)$ and $\mathsf{d}_2(\cdot,\cdot) = \mathsf{W}_2^2(\cdot,\cdot)$, so that

$$\mathsf{J}_3(\alpha) = \frac{d_y}{2}\mathsf{SW}_{2,\Gamma}^2\Big(\nu, \eta * \big(g\circ F^{\phi^*(\alpha)}\circ T^\alpha\big)_{\#}\mu_0\Big) + h(\alpha), \tag{3.4a}$$

$$\mathsf{J}_4(\phi;\alpha) = \mathsf{W}_2^2\Big(\delta_0, \big(\mathcal{O}^\phi\circ T^\alpha\big)_{\#}\mu_0\Big), \tag{3.4b}$$

$$\phi^\star(\alpha) = \operatorname*{argmin}_\phi \mathsf{J}_4(\phi;\alpha). \tag{3.4c}$$

Remark 3.2. We highlight that using the squared Wasserstein-2 metric between a Dirac at zero and an abitrary measure reduces to computing an expected squared

2-norm of the samples drawn from that measure (Lemma 3.6). Hence, (3.4b) reduces to computing

$$\mathsf{J}_4(\phi;\alpha) = \mathbb{E}_{z\sim(T^\alpha)_{\#}\mu_0}\|\mathcal{O}^\phi(z)\|_2^2.$$

Framing (3.4b) this way gives a different interpretation to the commonly used physics-informed ML loss function.

3.2. Empiricalization. The optimization problem in (3.4) forms the basis of our computational methodology in this paper. However, to implement it, we need to use empirical approximations of the two input measures that define the loss function J_3. There are two ways in which evaluation of J_3 defined by (3.4) must be empiricalized to make a tractable algorithm:

- measure ν is replaced by measure
$$\nu^N = \frac{1}{N}\sum_{n=1}^{N}\delta_{y^{(n)}},$$
reflecting the fact that working with ν is not computationally tractable, but samples from ν are available;
- measures ν^N and $(g\circ F^{\phi^*(\alpha)}\circ T^\alpha)_{\#}\mu_0$ are replaced by their empirical approximations using, in each case, N_s independent samples, reflecting the fact that working with $(g\circ F^{\phi^*(\alpha)}\circ T^\alpha)_{\#}\mu_0$ is not computationally tractable, together with the computational simplicity arising from using the same number of samples in each argument of the divergence;
- measure $(\mathcal{O}^\phi\circ T^\alpha)_{\#}\mu_0$ is empiricalized using N_r independent samples from μ_0.

Remark 3.3. In contrast to empirical/hierarchical Bayesian approaches for similar problems, the proposed methodology side-steps the sampling of a challenging posterior distribution with dimension that grows like $O(NM)$, where N is the number of physical systems from which we have data, and M is the dimensionality of the parameters, $z^{(n)}$, of the individual physical systems. Instead, this is replaced with the straightforward empiricalization of a pushforward measure with N_s samples of dimension M.

In the numerical examples explored in Section 5, for one-dimensional (1D) Darcy $M = 20$ and for 2D Darcy $M = 400$, these will be the number of bases parametrizing the permeability field in the two setups, respectively.

3.3. Connection to Bayes' theorem. Consider the inverse problem defined by (1.1) in setting $N = 1$:

$$y = \mathcal{G}(z) + \varepsilon, \tag{3.5}$$

where $\varepsilon \sim \eta := \mathcal{N}(0,\Gamma)$ and $\mathcal{G} : Z \to \mathbb{R}^d$. Consider Functional 1 with $\nu = \delta_y$, $\mathcal{G}(\cdot)$ replacing $(g\circ F^\dagger)(\cdot)$[3] and choice of d_1 as in subsection 3.1:

$$\mathsf{J}(\mu) = \frac{d}{2}\mathsf{SW}_{2,\Gamma}^2\Big(\delta_y, \eta * (\mathcal{G})_{\#}\mu\Big) + \mathcal{H}(\mu). \tag{3.6}$$

[3]There is nothing intrinsic to the factorization $\mathcal{G} = g\circ F^\dagger$ in this subsection; results are expressed purely in terms of $\mathcal{G}$.

Thus we have returned to the population loss description of the problem in the setting where optimization to determine the prior is over all probability measures, not the parameterized family that we will use for computation. Thus the regularization is on the space of probability measures. The message of the following theorem is that, with this problem formulation on the space of measures, in the case $N = 1$ and with appropriate choice of regularization, Bayes' theorem is recovered.

THEOREM 3.4. *Define* $\mathcal{H}(\mu) := D_{\mathrm{KL}}(\mu||\mu^{\mathrm{prior}})$ *for some probability measure* $\mu^{\mathrm{prior}} \in \mathcal{P}(Z)$. *Consider the Bayesian inverse problem defined by* (3.5) *with* $z \sim \mu^{\mathrm{prior}}$ *independent of* $\varepsilon \sim \eta := \mathcal{N}(0, \Gamma)$. *Then the minimizer of* (3.6) *over the set of probability measures* $\{\mu \in \mathcal{P}(Z) : \mathbb{E}_{y' \sim \eta * (\mathcal{G}_{\#})\mu} \|y'\|_\Gamma^2 < \infty\}$ *is the Bayesian posterior given by*

$$\mu^y(A) = \frac{1}{\mathcal{Z}} \int_A \exp\left(-\frac{1}{2}\|y - \mathcal{G}(z)\|_\Gamma^2\right) \mathrm{d}\mu^{\mathrm{prior}}(z), \tag{3.7a}$$

$$\mathcal{Z} = \int_Z \exp\left(-\frac{1}{2}\|y - \mathcal{G}(z)\|_\Gamma^2\right) \mathrm{d}\mu^{\mathrm{prior}}(z). \tag{3.7b}$$

Remark 3.5. As discussed in the introduction to this paper, our method learns a *data-informed probability measure*—a generative model—for z, which may be used as a *prior* for downstream inference tasks. The theorem shows that, in the special case of $N = 1$, the learned prior actually coincides with the Bayesian posterior for the inverse problem defined by (1.1) if regularizer $\mathcal{H}$ is chosen appropriately. This is entirely consistent with our broader agenda when $N > 1$ as the posterior distribution is the natural prior for downstream tasks in Bayesian inference.

To prove this theorem we establish a sequence of lemmas. The first shows that the weighted Wasserstein-2 distance may be simplified when one of its argument is a Dirac.

LEMMA 3.6. *For any* $y \in \mathbb{R}^d$ *and* $\mu \in \mathcal{P}(\mathbb{R}^d)$

$$\mathsf{W}_{2,\Gamma}^2(\delta_y, \mu) = \mathbb{E}_{x \sim \mu} \|y - x\|_\Gamma^2. \tag{3.8}$$

Proof. See Appendix A.2. ∎

Using the definition of pushforward, it follows from Lemma 3.6 that

$$\mathsf{W}_{2,\Gamma}^2(\delta_y, (\mathcal{G})_{\#}\mu) = \mathbb{E}_{z \sim \mu} \|y - \mathcal{G}(z)\|_\Gamma^2 = \int_{\mathbb{R}^d} \|y - \mathcal{G}(z)\|_\Gamma^2 \mathrm{d}\mu(z). \tag{3.9}$$

The following lemma shows that the sliced-Wasserstein metric also simplifies when one of its argument is a Dirac.

LEMMA 3.7. *Let* $y \in \mathbb{R}^d$ *and* $\mu \in \mathcal{P}(\mathbb{R}^d)$, *and assume* $\mathbb{E}_{z \sim \mu} \|z\|_\Gamma^2 < \infty$. *Then*

$$\mathsf{SW}_{2,\Gamma}^2(\delta_y, \mu) = \frac{1}{d} \mathsf{W}_{2,\Gamma}^2(\delta_y, \mu). \tag{3.10}$$

Proof. See Appendix A.3. ∎

The final lemma concerns convolution.

LEMMA 3.8. *Let* $\mathsf{I} : \mathcal{P}(Z) \to \mathbb{R}$ *be defined by* $\mathsf{I}(\cdot) = \mathsf{W}_{2,\Gamma}^2(\delta_y, \cdot)$. *Let* η *be a centered random variable and finite second moment. Then*

$$\mathsf{I}(\eta * \mu) = \mathsf{I}(\mu) + \mathbb{E}_{\varepsilon \sim \eta} \|\varepsilon\|_\Gamma^2.$$

Proof. See Appendix A.4. ∎

Proof of Theorem 3.4. By (3.9) and Lemmas 3.7 and 3.8 it follows that the loss function (3.6) can be written as

$$\mathsf{J}(\mu) = \frac{1}{2}\mathbb{E}_{z\sim\mu}\|y - \mathcal{G}(z)\|_\Gamma^2 + \mathcal{H}(\mu) + C_1, \tag{3.11}$$

where C_1 is independent of μ. With the choice $\mathcal{H}(\mu) := D_{\mathrm{KL}}(\mu||\mu^{\mathrm{prior}})$ we see that $\mathsf{J}(\mu) = D_{\mathrm{KL}}(\mu||\mu^y) + C_2$, where C_2 is independent of μ, and is determined by C_1 and $\mathcal{Z}$. The result follows since $D_{\mathrm{KL}}(\cdot||\mu^y)$ is minimized at μ^y. ∎

Remark 3.9. In order to use Lemmas 3.7 and 3.8 in the proof of Theorem 3.4, we assume the pushforward measure $\eta * (\mathcal{G}_\# \mu)$ has $\|\cdot\|_\Gamma^2$-moment finite as stated in Theorem 3.4. This will follow from second moments of $\mathcal{G}_\#\mu$. Such results can often be established using the Fernique theorem; see, for example, [16, 21].

Remark 3.10. Now consider functional $\mathsf{J}(\cdot)$ given by (3.6), replacing $\mathcal{G}$ by the specific choice $g \circ F^\dagger$ arising from the inverse problem defined by (1.2) in the case where $N = 1$. If we now parameterize target measure μ by $\mu = T^\alpha_\# \mu_0$, make the choice of KL divergence for $\mathcal{H}$, and view $\mathsf{J}(\cdot)$ as parameterized by $\alpha \in \mathbb{R}^{d_\alpha}$ rather than by $\mu \in \mathbb{P}(Z)$, we obtain

$$\mathsf{J}(\alpha) = \frac{d}{2}\mathsf{SW}_{2,\Gamma}^2\Big(\delta_y, \eta * (g \circ F^\dagger \circ T^\alpha)_\# \mu_0\Big) + D_{\mathrm{KL}}((T^\alpha)_\#\mu_0||\mu^{\mathrm{prior}}).$$

By the same reasoning used in the proof of Theorem 3.4 we may rewrite this as

$$\mathsf{J}(\alpha) = \frac{1}{2}\mathbb{E}_{z\sim(T^\alpha)_\#\mu_0}\|y - (g \circ F^\dagger)(z)\|_\Gamma^2 + D_{\mathrm{KL}}((T^\alpha)_\#\mu_0||\mu^{\mathrm{prior}}).$$

From this form it is clear that minimizing $\mathsf{J}(\alpha)$ over $\alpha \in \mathbb{R}^{d_\alpha}$ is simply the variational Bayes methodology applied to approximate the Bayesian posterior μ^y given by (3.7). Furthermore, this provides motivation for the consideration of Functional 2 to determine the prior on z, in the case where $N > 1$, noting that it reduces to variational Bayes when $N = 1$ with appropriate choices of $\mathsf{d}_1(\cdot,\cdot)$ and $h(\cdot)$.

4. Algorithms. In this section we discuss practical aspects pertaining to the implementation of the proposed methodologies. First, although Remark 3.10 suggests a specific choice for regularization $h(\alpha)$ in Functional 2 or Functional 3, in practice it is not computationally straightforward to work with this choice and simpler choices are made. Second, the evaluation of the objective functions is carried out using the empiricalization described in subsection 3.2. Third, the physics-based residual can be evaluated using Lemma 3.6 and the considerations of subsection 3.2. Fourth, once loss functions are evaluated we obtain gradients with respect to prior (and operator) parameters through back-propagation with standard ML library tools, such as JAX [12].

In Algorithm 4.1 we show how to implement the proposed inference methodology of Functional 2 for the task of prior calibration given a forward model $F^\dagger$. In Algorithm 4.2 we show how to implement Functional 3 for joint prior calibration and operator learning. We note that to correctly implement the bilevel optimization scheme proposed to minimize Functional 3 we must differentiate through the lower-level optimization steps given by Functional 4 to take into account the dependence of $\phi^\star(\alpha)$ on α. This incurs additional computational costs; efficient approximation methods for this task will be explored in future works.

Algorithm 4.1. Prior calibration.

Initialize α_0, T (number of iterations), N_s (number of samples for J_2), $F^\dagger$ (forward model)
for $t = 1, \dots, T$ **do**
  for $i = 1, \dots, N_s$ **do**
    Sample $\varepsilon_0^{(i)} \sim \mu_0$
    Sample $z^{(i)} \sim T^{\alpha_{t-1}}(\varepsilon_0^{(i)})$
    Sample $\varepsilon_\eta^{(i)} \sim \eta$
    Compute $y'^{(i)} = g \circ F^\dagger(z^{(i)}) + \varepsilon_\eta^{(i)}$
    Sample $y^{(i)} \sim \nu^N$
  end for
  Compute $\mathsf{J}_2(\alpha_{t-1})$ from the N_s samples $\{y^{(i)}, y'^{(i)}\}_{i=1}^{N_s}$.
  $\alpha_t = \mathrm{OPTIMISER}(\alpha_{t-1}, \mathsf{J}_2(\alpha_{t-1}))$
end for
return $\alpha^\star \leftarrow \alpha_T$.

We note that only the outermost parameter update loops must be computed sequentially. All other loops can be efficiently computed in parallel in a GPU-efficient manner. In all experiments we make use of the Adam optimizer [37]. The sliced-Wasserstein implementation is based on that of [27]. If one is interested in applying this methodology to nonlinear PDEs, computing Functional 2 requires the solution of N_s nonlinear systems of equations at every parameter update step. However, this is not the case for the method using Functional 3 and Functional 4 as we only need to compute N_r residuals, which is done in the same manner for linear or nonlinear PDEs. (See also subsection 3.2 for definitions of N_s, N_r.)

5. Numerical results for Darcy flow. We now illustrate the performance of our methodology to learn generative models for priors, based on indirect observations. All our numerical examples are in the setting of (1.3). The presented PDE is to be interpreted in the weak sense; see Appendix B for a short discussion on the weak form and approximate computational methods exploiting GPU-efficient array shifting operations. We consider two classes of coefficient function z: the first is a class of piecewise constant functions with discontinuity sets defined as level sets of a smooth field, in subsection 5.1; the second is a class of functions defined as the pointwise exponential of a smooth field, in subsection 5.2. In both cases we will write the prior as the pushforward of a Gaussian measure, on the underlying smooth field, under a parameter-dependent map T^α; we refer to the first class as *level set priors* and the second as *lognormal priors.* We attempt to learn the parameters α through minimization of either Functional 2 (using a numerical PDE solver) or Functional 3 (using residual-based operator learning); in all cases we use the framework of Section 3. We denote the true measure which we wish to recover by $\mu^\dagger$.

For simple problems regularization may not be needed. For more challenging problems, regularization on α may help with numerical stability, particulariliy when the surrogate model $F^{\phi^\star(\alpha)}$ is concurrently learned. Hence, for the 1D Darcy problems we omit the regularizer, and for the 2D Darcy examples we use regularization of the form $h(\alpha) = 1/(2\sigma_h^2)\|\log(\alpha) - m_h\|_2^2$ in which the log is applied componentwise; we refer to σ_h and m_h as the standard deviation and mean of the regularizer term. We

Algorithm 4.2. Prior calibration and operator learning.

Initialize α_0, ϕ_0, T (number of outer-loop α updates), L (number of inner-loop ϕ updates), N_s (number of samples for J_3), N_r (number of samples for J_4)
for $t = 1, \dots, T$ **do**
  for $l = 1, \dots, L$ **do**
    for $j = 1, \dots, N_r$ **do**
      Sample $\varepsilon_0^{(j)} \sim \mu_0$
      Sample $z^{(j)} \sim T^{\alpha_{t-1}}(\varepsilon_0^{(i)})$
      Compute $r^{(j)} = \mathcal{O}^{\phi_{l-1}(\alpha_{t-1})}(z^{(j)})$
    end for
    Compute $\mathsf{J}_4(\alpha_{t-1}, \phi_{l-1}(\alpha_{t-1}))$ from the N_r samples $\{r^{(j)}\}_{j=1}^{N_r}$.
    $\phi_l(\alpha_{t-1}) = \text{OPTIMISER}(\phi_{l-1}(\alpha_{t-1}), \mathsf{J}_4(\alpha_{t-1}, \phi_{l-1}(\alpha_{t-1}))$
  end for
  $\phi^\star(\alpha_{t-1}) \leftarrow \phi_L(\alpha_{t-1})$
  for $i = 1, \dots, N_s$ **do**
    Sample $\varepsilon_0^{(i)} \sim \mu_0$
    Sample $z^{(i)} \sim T^{\alpha_{t-1}}(\varepsilon_0^{(i)})$
    Sample $\varepsilon_\eta^{(i)} \sim \eta$
    Compute $y'^{(i)} = g \circ F^{\phi^\star(\alpha_{t-1})}(z^{(i)}) + \varepsilon_\eta^{(i)}$
    Sample $y^{(i)} \sim \nu^N$
  end for
  Compute $\mathsf{J}_3(\alpha_{t-1}, \phi^\star(\alpha_{t-1}))$ from the N_s samples $\{y^{(i)}, y'^{(i)}\}_{i=1}^{N_s}$.
  $\alpha_t = \text{OPTIMISER}(\alpha_{t-1}, \mathsf{J}_3(\alpha_{t-1}, \phi^\star(\alpha_{t-1})))$
end for
return $\{\alpha^\star \leftarrow \alpha_T, \phi^\star(\alpha^\star) \leftarrow \phi^\star(\alpha_T)\}$.

have not found it necessary to regularize the neural operator parameters as we are not interested in recovering a specific value for $\phi^\star(\alpha)$; rather any value which achieves a small error when evaluating the PDE residual of the output is sought.

For all experiments, we use 1000 slicing directions, θ, to evaluate the sliced-Wasserstein term, and we minimize the relevant loss functions using Adam [37] with a learning rate of 10^{-2} decayed by half four times on the parameters α, other than where explicitly stated. All experiments are run on a 24 GB NVIDIA RTX 4090 GPU.

The numerical results we present substantiate the following conclusions:

- The proposed methodology using Functional 2 can effectively recover the true parameters of a level set prior and a lognormal prior.
- The proposed methodology using Functional 2 can effectively recover the true parameters of a level set prior despite using a smoothening of the level set formulation, so that there is prior misspecification.
- We highlight the impact of dataset size (N), and the number of samples used to estimate the loss (N_s), on the quality of the prior parameter estimation; this demonstrates the strength of distributional inference for prior calibration.
- The proposed methodology using Functional 3 and Functional 4 can achieve comparable parameter estimation accuracy to that under Functional 2; thus jointly estimating the operator approximation and the parameters of the prior both is feasible and, for reasons of efficiency, will be desirable in settings where $F^\dagger$ is expensive to evaluate, is not differentiable, or is not available.

- We show how, under certain circumstances, prior parameters may be unidentifiable, and we show limitations of the proposed methodology in this setting.

5.1. Level set priors. Prior construction here is based on the methodology introduced in [23, 33]. We define $\alpha = \{\kappa^-, \kappa^+, \lambda\}$ where $\kappa^\pm \in (0, \infty)$ and $\lambda > 0$. We now define a function $z : C^\infty(D) \times \mathbb{R}^2 \to L^\infty(D)$ which will take a λ-dependent function in $C^\infty(D)$ and the two values $\kappa^\pm \in \mathbb{R}^2$ to create a function in $L^\infty(D)$. We introduce parameterized family of measures $(T^\alpha)_\# \mu_0$ on $L^\infty(D)$ by pushing forward a Gaussian measure on $a \in C^\infty(D)$ under map T^α to define a measure on z. To this end we define z by

$$z(x) = \kappa^- \mathbb{1}_{D^-(a)}(x) + \kappa^+ \mathbb{1}_{D^+(a)}(x), \tag{5.1}$$

$$D^-(a) = \{x \in D | a(x) < 0\}, \quad D^+(a) = \{x \in D | a(x) \geq 0\}. \tag{5.2}$$

To complete the description of T^α we construct a as a λ-dependent Gaussian random field (GRF). To do this we fix a scalar $\beta > 0$. The construction of a differs in details between dimensions one and two. For a 1D physical domain D, we define the λ-dependent Gaussian measure on a through the Karhunen–Loève expansion

$$a(x; \lambda, \beta) = \sum_{j=1}^{J} \left(j^2\pi^2 + \lambda^2\right)^{-\beta/2} \varepsilon_j \varphi_j(x), \quad \varepsilon_j \sim \mathcal{N}(0,1) \text{ i.i.d.},$$

where $\varphi_j(x) = \cos(j\pi x)$. In dimension two we generalize to obtain

$$a(x; \lambda, \beta) = \sum_{j=1,k=1}^{J,K} \left((j^2 + k^2)\pi^2 + \lambda^2\right)^{-\beta/2} \varepsilon_{j,k} \varphi_{j,k}(x), \quad \varepsilon_{j,k} \sim \mathcal{N}(0,1) \text{ i.i.d.},$$

with $\varphi_{jk}(x) = \cos(j\pi x_{(1)}) \cos(k\pi x_{(2)})$. For all experiments, we fix $\beta = 4$.

Note that, for $J = \infty$ in dimension one (resp., $(J, K) = \infty$ in dimension two) $a \sim \mathcal{N}(0, C_{\lambda,\beta})$ where

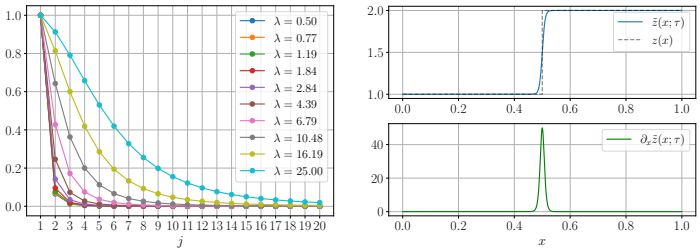
$$C_{\lambda,\beta}^{-1} = \left(-\Delta + \lambda^2 \mathrm{I}\right)^\beta, \tag{5.3}$$

and $-\Delta$ is the Laplacian equipped with homogeneous Neumann boundary conditions. Draws from this measure have Hölder regularity up to exponent $\beta - d/2$ where d is dimension of domain d. This statement about the probability measure from which a is drawn is approximate when J is (resp., (J, K) are) finite. Because of the countable nature of the construction of the level set prior, it lies in a separable subspace Z of $L^\infty(D)$. The derivative of the objective functional with respect to λ is not well behaved for the *sharp* level set setup as $\partial_a z$ is zero almost everywhere. Hence we introduce a *smoothened* level set parametrization of $z(\alpha)$ as

$$\tilde{z}(x; \tau) = \frac{1}{2} \tanh(\tau \bar{a})(\kappa^+ - \kappa^-) + \frac{1}{2}(\kappa^+ - \kappa^-) + \kappa^-, \tag{5.4}$$

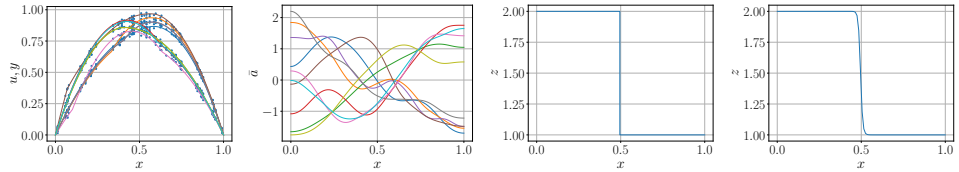
$$\bar{a} = a / \|a\|_{L^2(D)}, \tag{5.5}$$

where the parameter τ controls the sharpness of transition from κ^- to κ^+. In Figure 1(a) we show the spectrum decay of the GRF generated from the covariance operator (5.3) in dimension one. Figure 1(b) shows the smoothing of the sharp level set function z and its spatial derivative. In Figure 2 we show 10 sampled random fields $\bar{a}$ and the associated PDE solutions for the 1D Darcy problem. We only show one



(a) Normalized Spectrum, $\beta = 4$ (b) Smoothed level set function

FIG. 1. (a) *Spectrum decay of the square root of the eigenvalues for the covariance operator* (5.3) *(which corresponds to the standard deviation of the basis expansion coefficient of each mode* $\varphi_j(x)$*) for* $\beta = 4$ *and different choices of length-scale parameter* λ*. As* λ *increases, more modes play a significant role.* (b) *A comparison of smooth level set function* $\tilde{z}$ *and sharp level set function* z *transitioning at* $x = 0.5$ *for* $\tau = 100$*.*



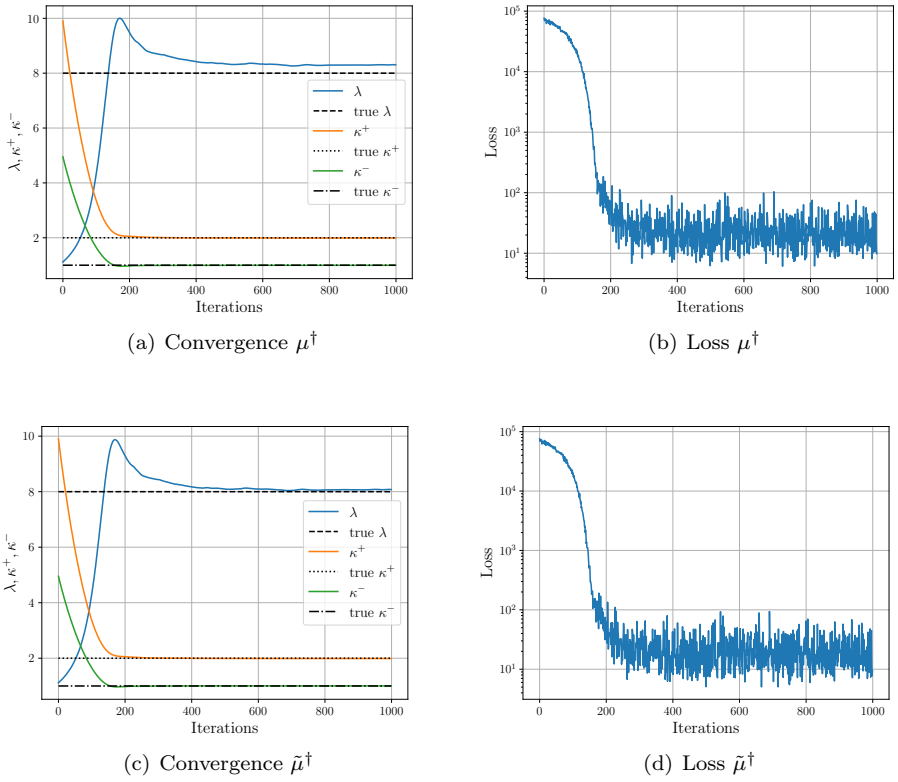
(a) several u and y (b) Random fields $\bar{a}$ (c) Sharp z (d) Smooth z, $\tau = 10$

FIG. 2. (a)–(c) *Data generated from sharp prior* $\mu^\dagger$*, which are the PDE solution fields with the observational data at points, the normalized GRF* $\bar{a}$ *samples, and one example of a level set function resulting from one of the* $\bar{a}$ *fields.* (d) *The same diffusivity field* $\tilde{z}$ *as in* (c)*, but smoothed with* (5.4) *and* $\tau = 10$*.*

diffusion field z for clarity. We note that there is a symmetry in inference between κ^+ and κ^-, so after convergence we sort the $\kappa^\pm$ to associate κ^+ with the larger value and vice versa for κ^-. The number of physical systems from which we have data, N, is set to 1000 for the subsequent examples as this is shown to give very accurate results while highlighting the main benefit of the methodology—to efficiently and accurately estimate underlying distributional parameters from large amounts of data.

With this experimental setup established, we can test the proposed methodology. Subsections 5.1.1 and 5.1.2 concern the setting where we use a PDE solve for the forward model, and hence minimize Functional 2; we consider dimensions $d = 1$ and $d = 2$, respectively. In subsections 5.1.3 and 5.1.4 we combine learning of α with operator learning to replace the PDE solve, and hence minimize Functional 3; we again consider dimensions $d = 1$ and $d = 2$, respectively.

5.1.1. Prior calibration: 1D Darcy. In this section we focus our attention on using Functional 2 to infer parameters of a prior $(T^\alpha)_{\#}\mu_0$ given an empirical measure ν^N and a forward operator $F^\dagger$. We use $N = 1000$ data y each of $d_y = 50$ noisy pointwise observations of the solution with $\Gamma = 0.01^2\mathrm{I}$ observational noise covariance. The prior KL expansion is truncated at 20 terms. We use $N_s = 1000$ samples to evaluate Functional 2. The true parameters for data generation are $\lambda = 8, \kappa^+ = 2, \kappa^- = 1$ and we set the level set smoothing parameter $\tau = 10$. We use a finite element mesh of 100 nodes with a weak form residual computed through array shifting and solved with conjugate gradients and the forcing function f is set to a constant value of 10 for all examples. No regularizer $h(\alpha)$ is used for the 1D example. We minimize using



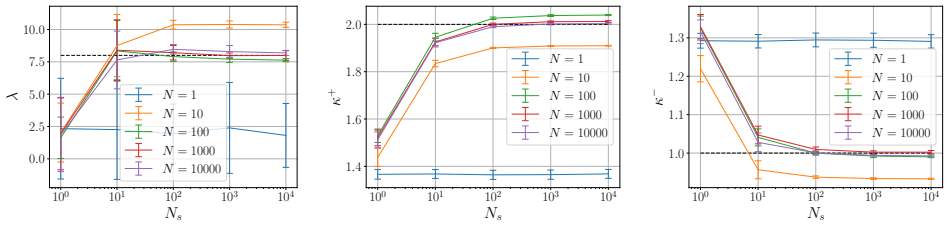
(a) Convergence $\mu^\dagger$ (b) Loss $\mu^\dagger$

(c) Convergence $\tilde{\mu}^\dagger$ (d) Loss $\tilde{\mu}^\dagger$

FIG. 3. *Comparison of convergence of* $\alpha = \{\kappa^\pm, \lambda\}$ *for data generated from the physically realistic sharp* $\mu^\dagger$ *and the smoothed* $\tilde{\mu}^\dagger$ *for* 1*D Darcy with Functional* 2. *We plot in* (a) *and* (c) *the convergence of the parameters themselves, and in* (b) *and* (d) *the loss function values.*

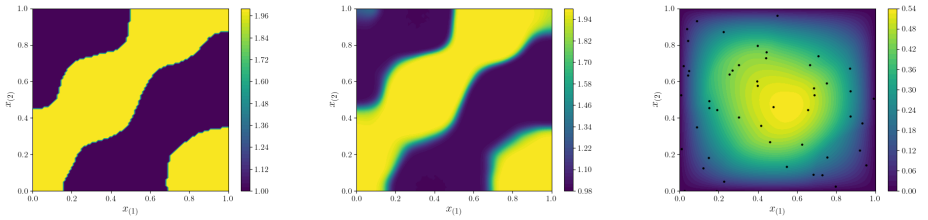
the smoothed model without sharp interfaces but we derive data both with the sharp and smoothed interfaces. In both cases we are able to accurately recover all three hyperparameters in α; see Figure 3. The runtime for these experiments is 41 s for the full 1k iterations. The relative error of the predicted parameters for the smoothed level set prior are 0.56%, 0.28%, and 0.96% for κ^+, κ^-, and λ, respectively. We observe that the misspecification between $\mu^\dagger$ (draws are discontinuous) and $(T^\alpha)_{\#}\mu_0$ (draws are smooth) is reflected in a small inference bias, as would be expected. In Figure 4 we plot the prior parameter estimation mean and standard deviations for 100 random initializations, varying the number of samples N_s used for estimating the loss functions and the size of the dataset (N). The α parameters are initialized from $\log\lambda \sim \text{Unif}(\log 0.5, \log 4)$, $\log\kappa^- \sim \text{Unif}(\log 0.5, \log 4)$, and $\log\kappa^- \sim \text{Unif}(\log 6, \log 10)$. We observe that the mean of the recovered parameters roughly converges for this problem setup after a dataset size $N = 100$ and number of samples $N_s = 100$.

5.1.2. Prior calibration: 2D Darcy. In this section we generalize the setting of the previous section to the 2D domain $D = [0,1] \times [0,1]$. As in the 1D example, we use $N = 1000$ data y each of $d_y = 50$ noisy pointwise observations of the solution with $\Gamma = 0.01^2\text{I}$ observational noise covariance. The prior KL expansion is truncated at 20 terms in each dimension, for a total of 400 terms. We use $N_s = 100$ samples to evaluate Functional 2. The true parameters for data generation are $\lambda = 5, \kappa^+ = 2, \kappa^- = 1$ and we set the level set smoothing parameter $\tau = 5$. We use a finite element mesh of 100×100 nodes. The regularizer $h(\alpha)$ in Functional 2 has means $m_{h,\lambda} = \log(10)$



(a) Convergence for λ (b) Convergence for κ^+ (c) Convergence for κ^-

FIG. 4. *Converged parameter estimation for $\lambda, \kappa^+, \kappa^-$ individually, for different dataset sizes N, and number of samples N_s for Functional 2 on the 1D Darcy problem.*



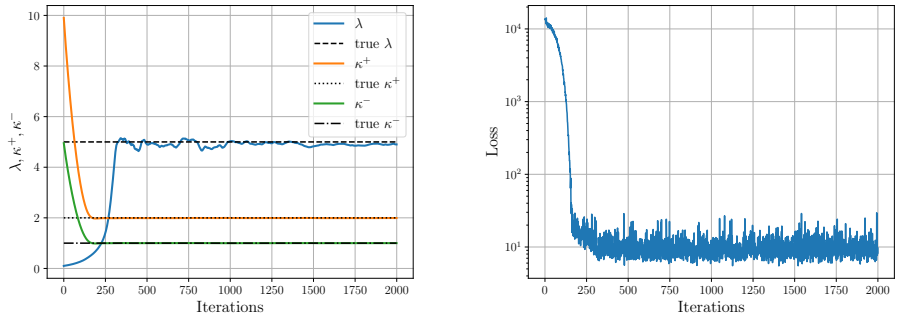
(a) z field (b) Smoothed z field (c) u and 50 observations

FIG. 5. *Plots of a sharp permeability field and the smoothed field and PDE solutions with observation locations in the dataset for the 2D Darcy problems. We note that the solutions are computed from the sharp level sets for the dataset, whereas $(T^\alpha)_{\#}\mu_0$ in the loss function uses the smoothed level sets to make use of gradient-based optimization.*

and $m_{h,\kappa^\pm} = \log(3)$, with standard deviations $\sigma_h = 2$. In Figure 5, two permeability fields associated with their sets of observations from the dataset are shown. Figure 6 shows the convergence plots for prior parameter calibration. Inference time was 1.52 minutes for the 2k iterations. The relative error of the predicted parameters is 0.39%, 0.03%, and 1.96% for κ^+, κ^-, and λ, respectively.

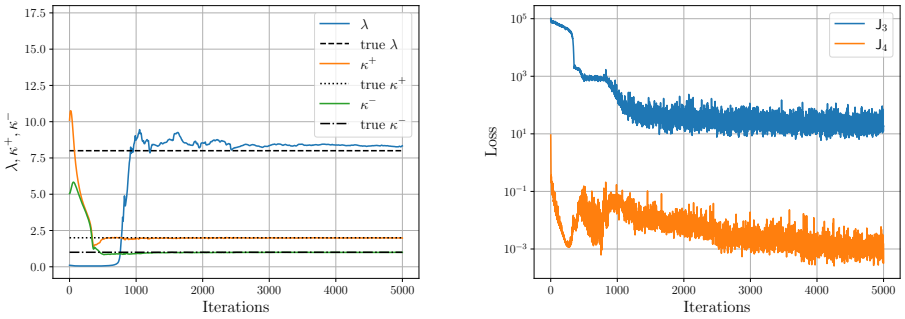
5.1.3. Prior calibration and operator learning: 1D Darcy. We now turn our attention to the task of using Functional 3 and Functional 4 to jointly estimate $(T^\alpha)_{\#}\mu_0$ and learn a parametrized operator F^ϕ. We choose this operator to be a four-layer Fourier neural operator [44] with 64-dimensional channel space and eight Fourier modes and Silu activation functions [30] ($\sigma(x) = x\,\text{sigmoid}(x)$). The last layer performs a pointwise multiplication with a function that is zero on the boundary ($\sin(\pi x)$). Functional 4 is evaluated with $N_r = 20$ samples and Functional 3 is evaluated with $N_s = 1000$ samples. We perform 10 update steps on ϕ per α update in the lower-level optimization routine. We use the same data setup, prior, and regularizer as in subsection 5.1.1. In Figure 7 we show that we are able to jointly learn the neural operator approximation of the forward model and the unknown parameters. The runtime is 2.44 minutes. The relative error for the FNO predictions against the solver for the data samples $z \sim \mu^\dagger$ from the true sharp level set prior is 0.40%. The relative error of the predicted parameters is 1.13%, 0.71%, and 4.18% for κ^+, κ^-, and λ, respectively.

5.1.4. Prior calibration and operator learning: 2D Darcy. We now apply the joint prior-operator learning methodology to the 2D setting, keeping the same



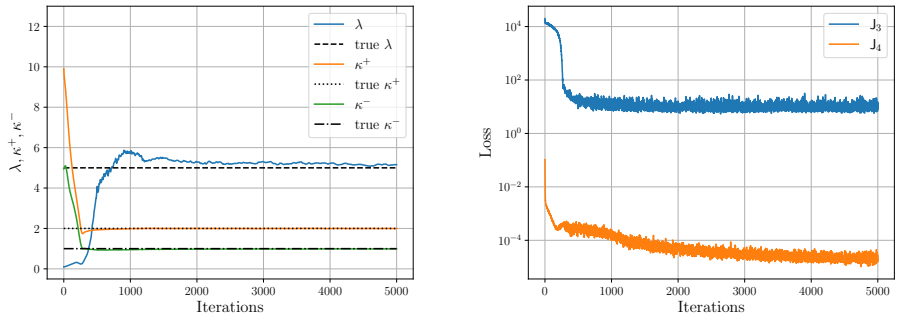
(a) α convergence (b) Losses

FIG. 6. *Convergence of prior parameters α for 2D Darcy trained with Functional 2.*



(a) α convergence (b) Losses

FIG. 7. *Convergence of prior parameters α, with learned operator F^ϕ on 1D Darcy with Functional 3 and Functional 4. We take 10 parameter updates steps on ϕ in the lower objective of the bilevel optimization for every step on α.*



(a) α convergence with $\phi^\star(\alpha)$ (b) Losses

FIG. 8. *Convergence of prior parameters α for the level set prior on the 2D Darcy problem with jointly learned operator F^ϕ. For every step of prior update α, 10 parameter updates are performed on ϕ.*

FNO parameters as in the 1D case (subsection 5.1.3) but for 2D input fields. The last layer of the FNO is now $\sin(\pi x_{(1)})\sin(\pi x_{(2)})$. We use the same data setup, prior, and regularizer as in subsection 5.1.2 and $N_r = 20$, $N_s = 100$. Figure 8 shows the convergence for the learning of both $(T^\alpha)_{\#}\mu_0$ and F^ϕ. The model converged

accurately after a runtime of 56.46 minutes. The relative error for the FNO predictions against the solver for the data samples $z \sim \mu^\dagger$ from the true sharp level set prior is of 0.73%. The relative error of the predicted parameters are 0.10%, 0.98%, and 3.19% for κ^+, κ^-, and λ, respectively.

5.2. Lognormal priors. In this subsection we focus on learning parameters of a lognormal prior. We introduce a Matérn-like field a with regularity ν, amplitude σ, and length-scale ℓ; to be specific we generate a centered GRF with covariance operator

$$C_{\sigma,\ell,\nu} = \gamma \ell^d (-\ell^2 \Delta + \mathrm{I})^{-\nu - d/2},$$

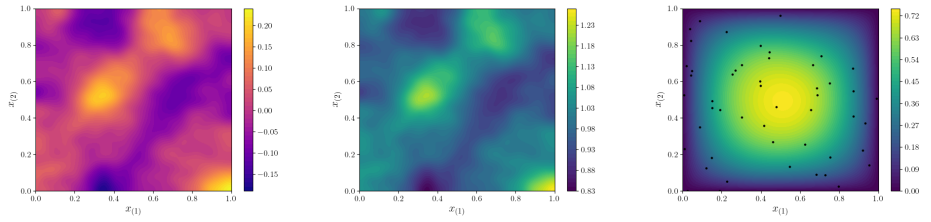
setting

$$\gamma = \sigma^2 \frac{2^d \pi^{d/2} \Gamma(\nu + d/2)}{\Gamma(\nu)},$$

where $\Gamma(\cdot)$ is the Gamma function (not to be confused with our use of Γ as the observational noise covariance). In practice we truncate a Karhunen–Loève expansion in the form

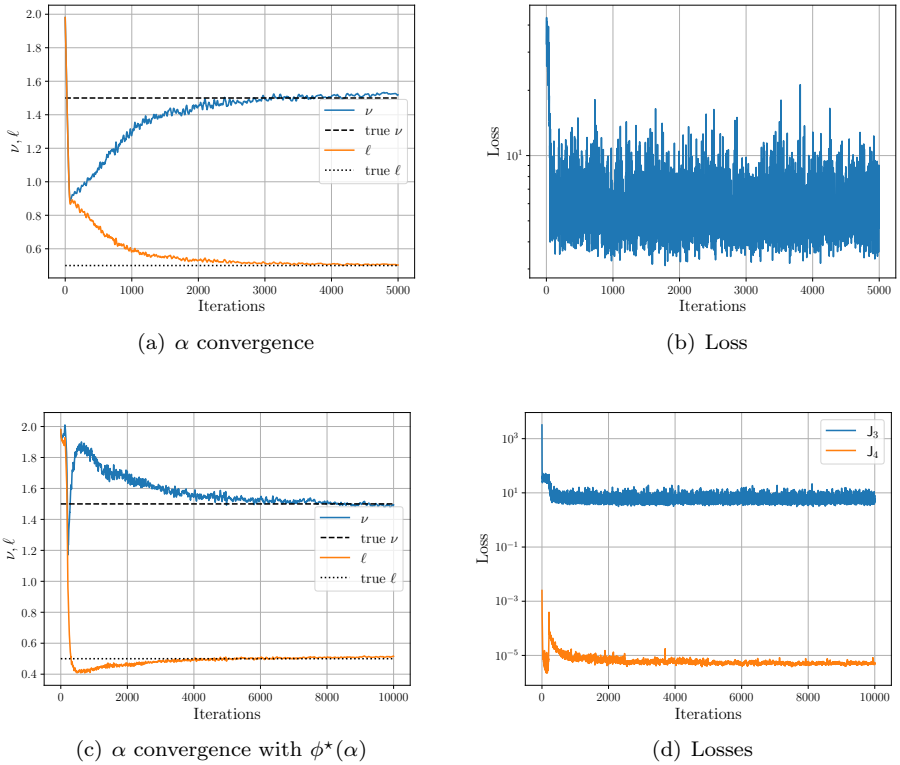
$$a(x;\sigma,\ell,\nu) = \sum_{j=1,k=1}^{J,K} \sqrt{\gamma \ell^d \left(\ell^2 (j^2+k^2)\pi^2 + 1\right)^{-\nu - d/2}} \varepsilon_{j,k} \varphi_{j,k}(x), \ \varepsilon_{j,k} \sim \mathcal{N}(0,1) \text{ i.i.d.}$$

We then set $z = \exp(a)$. Again, because of the countable nature of the construction of the lognormal prior, it lies in a separable subspace Z of $L^\infty(D)$. We focus on estimating $\alpha = \{\nu, \ell\}$ which represent the regularity and length-scale of the lognormal permeability field z. We do not attempt to jointly infer the amplitude σ as this induces a lack of identifiability of parameters at the level of the Karhunen–Loève expansion spectrum. We note that as the amplitude parameter does not contain much spatially dependent information, it can be accurately estimated by other means, hence here it is set to $\sigma = 1$. We attempt to jointly learn $\alpha = \{\nu, \ell\}$ and ϕ the parameters of a residual-based neural operator approximation. Figure 9 shows a randomly sampled function from this expansion, the corresponding permeability field, and the associated Darcy flow solution. In subsection 5.2.1 we consider a setting where the regularity and length-scale parameters are identifiable and show successful joint learning of (α, ϕ). However, it is intuitive that the regularity and length-scale may not be separately identifiable; we show in subsection 5.2.2 that in such situations the entanglement of jointly learning (α, ϕ) can cause convergence problems with the proposed methodology.



(a) a (b) z (c) u and 50 observations

FIG. 9. (a) *A sampled GRF* a, (b) *the exponentiated random field* z, (c) *the solution field associated to* z *with the* 50 *observation locations.*



(a) α convergence (b) Loss

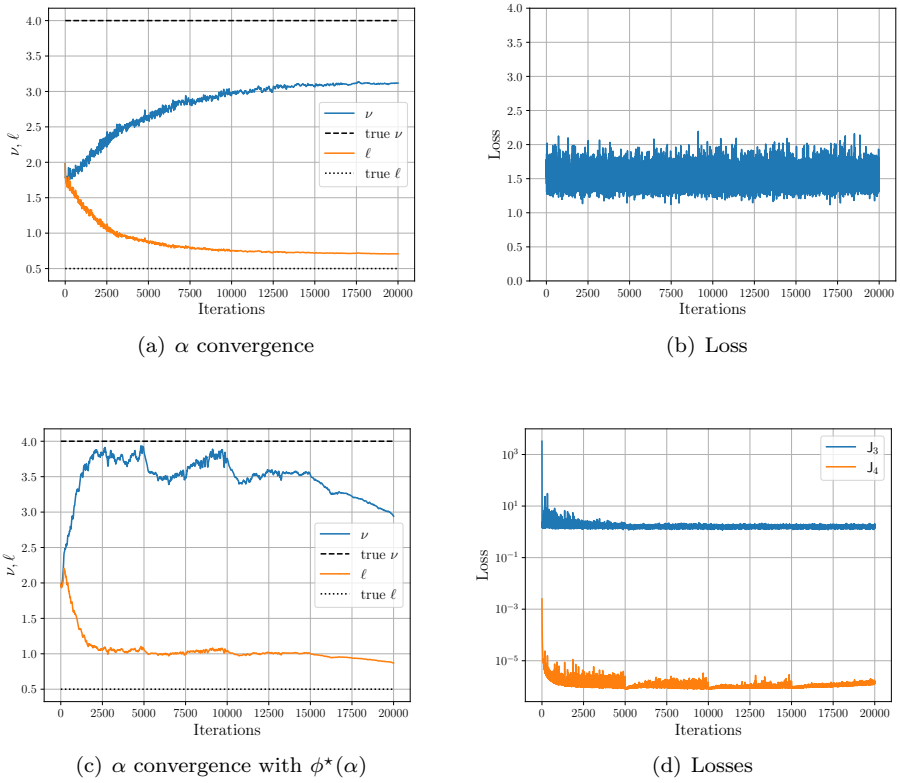
(c) α convergence with $\phi^\star(\alpha)$ (d) Losses

FIG. 10. (a)–(b) *Convergence and loss for* $\alpha = \{\nu, \ell\}$ *on the* 2*D Darcy problem for the lognormal prior.* (b)–(c) *Convergence of combinations of prior parameters* $\alpha = \{\nu, \ell\}$ *estimated jointly with* ϕ *for* 2*D Darcy on the lognormal prior with* 10 ϕ *updates in the lower-level optimization for every* α *step.*

5.2.1. Identifiable setting. We use $N = 1000$ data y each of $d_y = 50$ noisy pointwise observations of the solution with $\Gamma = 0.01^2\mathrm{I}$ observational noise covariance. The prior KL expansion is truncated at 20 terms in each dimension for a total of 400 expansion terms. We set $N_s = 100$. The true parameters for data generation are $\nu = 1.5, \ell = 0.5$. We use a finite element mesh of 100×100 nodes. The regularizer $h(\alpha)$ in Functional 3 has means $m_{h,\nu} = \log(3.5)$ and $m_{h,\ell} = \log(1)$, with standard deviations $\sigma_h = 2$. In Figure 10(a), (b) we show the convergence plots for the prior only learning. The relative error on parameter estimation is 1.26% and 0.66% on ν and ℓ, respectively. The runtime is 4.26 min.

We then test the joint estimation of α and ϕ with Functional 3 and Functional 4. The data setup, prior, and regularizers are the same as in the α-only case described in subsection 5.1.2 with $N_r = 20$, $N_s = 100$. The operator setup is the same as in subsection 5.1.4. In Figure 10(c), (d) we show the joint prior learning with operator learning. The achieved relative error for the learned F^ϕ on $z \sim \mu^\dagger$ is 0.128%. The relative error on parameter estimation is 0.44% and 3.13% on ν and ℓ, respectively. The runtime is 112 mins for the 10k iterations.

5.2.2. Unidentifiable setting. For certain parameter values of α, from which the true data is generated, there can be a lack of identifiability. In the previous section, the data was generated from a relatively rough random field with a relatively



(a) α convergence

(b) Loss

(c) α convergence with $\phi^\star(\alpha)$

(d) Losses

FIG. 11. (a)–(b) *Convergence and loss for α learning.* (c)–(d) *Convergence and losses for α and ϕ learning. Both sets of figures are for the 2D Darcy on log normal prior for unidentifiable α parameter values.*

long length-scale. Hence the smoothness and length-scale parameters each have a distinct impact on the prior's spectrum. We will now look at a smooth random field setting $\nu = 4$, keeping $\ell = 0.5$. Now, both parameters will have a similar influence on the spectrum of the prior. Other parameters for the data, solver/operator, and regularizer are kept the same as in subsection 5.2.1. In Figure 11 we show the convergence of the α parameters for the unidentifiable parameter regime for the α only learning and the joint α, ϕ prior and operator learning tasks. We can see the lack of identifiability causes more issues in the operator learning as it struggles to gain any inferential traction. For these examples, it was necessary to run the learning models for more iterations (20k) and use a 6-time decaying learning rate on α (as opposed to the usual 4 in previous examples). Furthermore, due to training instability, the ϕ parameters had to be trained with AMSGrad [57] (with the same settings) instead of Adam. As we can see from Figure 11, the proposed methodology applied to this particular unidentifiable setup does not fail silently, the α only learning has a very long convergence time, and the α and ϕ learning does not reach equilibrium. Practitioners may also find the method is more sensitive to initializations in such unidentifiable settings. Other possible avenues to identifying nonuniqueness in prior parameter estimation include assessing the sensitivity of the estimated parameters on the choice of regularizer mean and standard deviation, or the use of hierarchical parametric priors where a distribution on α is learned. There, a wide (or multimodal) inferred

distribution on α could indicate nonidentifiability of the prior parameters as well as the covariance between entangled prior parameters.

6. Conclusions. In this work we propose a novel methodology for learning a generative model for a prior on function space, based on indirect and noisy observations defined through solution of a PDE. The learning scheme is based on the minimization of a loss functional computing divergences at a distributional level. We prove our methodology recovers the Bayesian posterior when observations originate from a single physical system ($N = 1$). We demonstrate the accuracy of our methodology on a series of examples pertinent for practitioners. These are 1D and 2D steady-state Darcy flow problems for two different parametric priors, namely, a level set prior and a log-normal prior. Furthermore, we show how the proposed framework can be augmented to jointly learn an operator, mapping samples of PDE parameter fields drawn from the estimated prior, and the solution space of the PDE, through a bilevel optimization strategy. The operator learning takes place on-the-fly during the optimization of the prior parameters and is residual-based. Finally, we indicate the possible pitfalls of unidentifiable priors and show the emergent behavior of the methodology under this setting. A selection of avenues for future work includes the use of different metrics on the space of measures, testing on different PDE systems (nonlinear, coupled, time-evolving, etc.), other choices of parametrized prior measures, studying identifiability and nonuniqueness of inferred prior parameters, and testing the downstream use of such learned priors and operators as in Bayesian inversion and generative modeling of physical systems.

Appendix A. Proofs.

A.1. Proof of Lemma 3.1.

Proof. Let ν, μ denote two probability measures defined on a common measure space (X, Σ). Let $\Pi(\nu, \mu)$ denote the set of all coupling probablity measures γ, on the product space $X \times X$, such that $\gamma(A \times X) = \nu(A)$, $\gamma(X \times A) = \mu(A)$ for all $A \in \Sigma$. Remembering $P_{\mathrm{B}}(\cdot) = \mathrm{B}^{-1/2}\cdot$, now recall the definition

$$\mathsf{W}^2_{2,\mathrm{B}}(\nu, \mu) = \inf_{\gamma \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2_{\mathrm{B}} \mathrm{d}\gamma(x, y). \tag{A.1}$$

The set of all $\gamma' \in \Pi(P_{\mathrm{B}\#}\nu, P_{\mathrm{B}\#}\mu)$ is equivalent to the set $(P_{\mathrm{B}} \otimes P_{\mathrm{B}})_{\#}\gamma$ defined over all $\gamma \in \Pi(\nu, \mu)$. Thus we have

$$\begin{aligned}
\mathsf{W}^2_2(P_{\mathrm{B}\#}\nu, P_{\mathrm{B}\#}\mu) &= \inf_{\gamma' \in \Pi(P_{\mathrm{B}\#}\nu, P_{\mathrm{B}\#}\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x' - y'\|^2_2 \,\mathrm{d}\gamma'(x', y') \\
&= \inf_{\gamma \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2_2 \,\mathrm{d}((P_{\mathrm{B}} \otimes P_{\mathrm{B}})_{\#}\gamma(x, y)) \\
&= \inf_{\gamma \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\mathrm{B}^{-\frac{1}{2}}x - \mathrm{B}^{-\frac{1}{2}}y\|^2_2 \,\mathrm{d}\gamma(x, y) \\
&= \inf_{\gamma \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2_{\mathrm{B}} \,\mathrm{d}\gamma(x, y) \\
&= \mathsf{W}^2_{2,\mathrm{B}}(\nu, \mu). \qquad \square
\end{aligned} \tag{A.2}$$

(A.3)

A.2. Proof of Lemma 3.6.

Proof. To show the desired result we determine a point at which the infimum over couplings $\Pi(\delta_y, \mu)$ is achieved. It is known from [63, section 1.4] that when one of the

measures in the argument of a Kantorovich problem (of which Wasserstein metrics are a special case) is a Dirac, i.e., δ_y for any $y \in \mathbb{R}^d$, then the set of couplings $\Pi(\delta_y, \mu)$ with marginals δ_y, μ contains a single element, namely $\delta_y \otimes \mu$. It follows that

$$\begin{aligned} \mathsf{W}_{2,\Gamma}^2(\delta_y, \mu) &= \inf_{\gamma \in \Pi(\delta_y, \mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|y' - x\|_\Gamma^2 \mathrm{d}\gamma(y', x) \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \|y' - x\|_\Gamma^2 (\mathrm{d}\delta_y(y') \otimes \mathrm{d}\mu(x)) \\ &= \mathbb{E}_{x \sim \mu} \|y - x\|_\Gamma^2. \end{aligned} \tag{A.4}$$

∎

A.3. Proof of Lemma 3.7.

Proof. We want to show, for $y \in \mathbb{R}^d$ and $\mu \in \mathcal{P}(\mathbb{R}^d)$,

$$\mathsf{SW}_{2,\Gamma}^2(\delta_y, \mu) = \frac{1}{d} \mathsf{W}_{2,\Gamma}^2(\delta_y, \mu).$$

Thus, we have

$$\begin{aligned} \mathsf{SW}_{2,\Gamma}^2(\delta_y, \mu) &= \int_{\mathbb{S}^{d-1}} \mathsf{W}_2^2(P_{\Gamma\#}^\theta \delta_y, P_{\Gamma\#}^\theta \mu) \mathrm{d}\theta \\ &= \int_{\mathbb{S}^{d-1}} \mathsf{W}_2^2(\delta_{\langle \Gamma^{-1/2} y, \theta \rangle}, P_{\Gamma\#}^\theta \mu) \mathrm{d}\theta \\ &= \int_{\mathbb{S}^{d-1}} \mathbb{E}_{z \sim \mu} (\langle \Gamma^{-1/2} y, \theta \rangle - \langle \Gamma^{-1/2} z, \theta \rangle)^2 \mathrm{d}\theta \\ &= \mathbb{E}_{z \sim \mu} \int_{\mathbb{S}^{d-1}} (\langle \Gamma^{-1/2} y, \theta \rangle - \langle \Gamma^{-1/2} z, \theta \rangle)^2 \mathrm{d}\theta \qquad \text{(A.5b)} \\ &= \mathbb{E}_{z \sim \mu} \int_{\mathbb{S}^{d-1}} \big(\langle \Gamma^{-1/2} (y - z), \theta \rangle \big)^2 \mathrm{d}\theta, \end{aligned} \tag{A.5a}$$

where, in order to move from (A.5a) to (A.5b), we use the fact that $\mathbb{E}_{z \sim \mu} \|z\|_\Gamma^2 < \infty$. Furthermore,

$$\langle \Gamma^{-1/2} a, \theta \rangle \langle \Gamma^{-1/2} b, \theta \rangle = (\Gamma^{-1/2} a)^\top \theta \theta^\top (\Gamma^{-1/2} b),$$

and noting, from [75, section 3.3.1], the covariance of the uniform on the sphere, $\operatorname{Cov} \operatorname{Unif}(\mathbb{S}^{d-1}) = \int_{\mathbb{S}^{d-1}} \theta \theta^\top \mathrm{d}\theta = \frac{1}{d} \mathrm{I}$. Then

$$\begin{aligned} \int_{\mathbb{S}^{d-1}} \langle \Gamma^{-1/2} a, \theta \rangle \langle \Gamma^{-1/2} b, \theta \rangle \mathrm{d}\theta &= \frac{1}{d} (\Gamma^{-1/2} a)^\top (\Gamma^{-1/2} b) \\ &= \frac{1}{d} \langle a, b \rangle_\Gamma. \end{aligned}$$

In particular

$$\int_{\mathbb{S}^{d-1}} \big(\langle \Gamma^{-1/2} (y - z), \theta \rangle \big)^2 \mathrm{d}\theta = \frac{1}{d} \|y - z\|_\Gamma^2.$$

And so, from (A.5b) and Lemma 3.6 we obtain

$$\begin{aligned} \mathsf{SW}_{2,\Gamma}^2(\delta_y, \mu) &= \frac{1}{d} \mathbb{E}_{z \sim \mu} \|y - z\|_\Gamma^2 \\ &= \frac{1}{d} \mathsf{W}_{2,\Gamma}^2(\delta_y, \mu). \end{aligned}$$

∎

A.4. Proof of Lemma 3.8.

Proof. We have

$$\mathsf{I}(\eta * \mu) = \mathbb{E}_{x' \sim \eta * \mu} \|y - x'\|_\Gamma^2,$$
$$\mathsf{I}(\mu) = \mathbb{E}_{x \sim \mu} \|y - x\|_\Gamma^2.$$

Thus

$$\begin{aligned}
\mathsf{I}(\eta * \mu) &= \mathbb{E}_{(x,\varepsilon) \sim \mu \otimes \eta} \|y - x - \varepsilon\|_\Gamma^2, \\
&= \mathbb{E}_{(x,\varepsilon) \sim \mu \otimes \eta} \Big(\|y - x\|_\Gamma^2 - 2\langle y - x, \varepsilon \rangle_\Gamma + \|\varepsilon\|_\Gamma^2 \Big), \\
&= \mathbb{E}_{x \sim \mu} \|y - x\|_\Gamma^2 + \mathbb{E}_{\varepsilon \sim \eta} \|\varepsilon\|_\Gamma^2, \\
&= \mathsf{I}(\mu) + \mathbb{E}_{\varepsilon \sim \eta} \|\varepsilon\|_\Gamma^2,
\end{aligned}$$

using that η is centered, is independent of μ, and has finite second moments. ∎

Appendix B. Weak form residuals with array shifting. We now show how to compute a residual in the weak form for a Darcy problem on domain D, with homogeneous boundary conditions using array shifting. Taking (1.3) and testing the domain residual against a set of test functions $\{v_i\}_{i=1}^{d_o}$, we obtain the discretized variational formulation

$$\mathcal{O}(R(z,u))_i = \langle v_i, R(z,u) \rangle = \int_D v_i (\nabla \cdot (z \nabla u)) \mathrm{d}x + \int_D v_i f \, \mathrm{d}x = 0 \, \forall v_i \in V.$$

Through integration by parts we obtain the weak form. Here, we use as shorthand $\mathsf{r}_i = \mathcal{O}(R(z,u))_i$. In two dimensions,

$$\mathsf{r}_i = -\int_D z \partial_{x_{(1)}} v_i \, \partial_{x_{(1)}} u \, \mathrm{d}x - \int_D z \partial_{x_{(2)}} v_i \, \partial_{x_{(2)}} u \, \mathrm{d}x + \int_D v_i f \, \mathrm{d}x. \tag{B.1}$$

Traditional finite element solvers would assemble a system of sparse linear equations of the form $\mathsf{A}\mathsf{u} = \mathsf{f}$ where $\mathsf{A} \in \mathbb{R}^{d_o \times d_o}$ is sparse, and $\mathsf{u}, \mathsf{f} \in \mathbb{R}^{d_o}$. The matrix vector product $\mathsf{A}\mathsf{u}$ is comprised of the first two terms in (B.1) and f is the third term. To compute residuals using array shifting in two dimensions we use double indexing, denoted by jk, to represent the mesh nodes as shown in Figure 12. We denote the set of indices $\{i\}_{i=1}^{d_o} = \mathsf{ravel}(\{j,k\}_{j,k=1}^{j,k=\sqrt{d_o}})$. Then

$$\mathsf{r}_{jk} = -\mathsf{r}_{x_{(1)},jk} - \mathsf{r}_{x_{(2)},jk} + \mathsf{r}_{f,jk},$$

where each of the three terms corresponds to the ones in (B.1). Assuming z is given at the nodes and is piecewise constant from the top left of an element, we have

$$\mathsf{r}_{x_{(1)},jk} = \frac{1}{2}((z_{j_{-1},k_{-1}} + z_{j,k_{-1}})(u_{j,k} - u_{j,k_{-1}}) - (z_{j,k} + z_{j_{-1},k})(u_{j,k_{+1}} - u_{j,k})),$$
$$\mathsf{r}_{x_{(2)},jk} = \frac{1}{2}((z_{j,k_{-1}} + z_{j,k})(u_{j,k} - u_{j_{+1},k}) - (z_{j_{-1},k} + z_{j_{-1},k_{-1}})(u_{j_{-1},k} - u_{j,k})).$$

We compute the tested inhomogeneous term as $\int_D v f \, \mathrm{d}x = \sum_{e=1}^{6} \int_{D_e} v f \, \mathrm{d}x$ integrating using 1 point Gauss integration at $(1/3h, 1/3h)$. Thus

$$\mathsf{r}_{f,jk} = \frac{h^2}{9} \left(3 f_{j,k} + f_{j_{-1},k_{-1}} + f_{j,k_{-1}} + f_{j_{+1},k} + f_{j_{+1},k_{+1}} + f_{j,k_{+1}} + f_{j_{-1},k} \right).$$

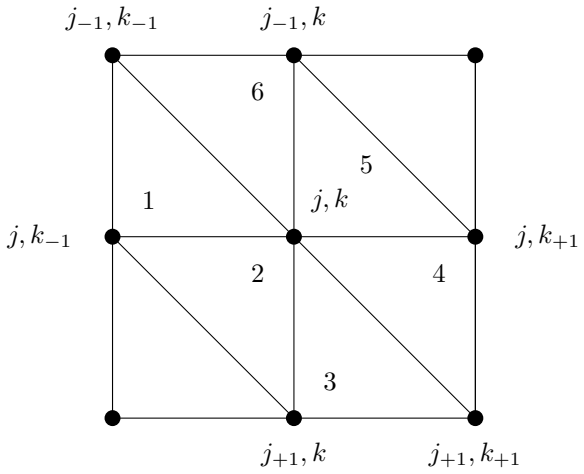


FIG. 12. *2D FEM mesh centered at node jk. We assume equal spacing, h, of the nodes.*

All operations can be rapidly computed using array shifting. To solve the linear differential equation we use an iterative linear solver such as conjugate gradient for positive definite matrices A or GMRES for more general problems. To solve nonlinear differential equations, one would use Newton's method. The Newton update step can be seen as solving a linear system with a Jacobian-vector product $\mathsf{u}_{\mathsf{n}+1} = \mathsf{u}_{\mathsf{n}} - \mathrm{J}_{\mathcal{O}}(\mathsf{u}_{\mathsf{n}})^{-1}\mathcal{O}(\mathsf{u}_{\mathsf{n}})$. Deep learning libraries are designed to compute Jacobian-vector products with high efficiency, resulting in a GPU efficient Newton–Krylov method.

REFERENCES

[1] J. Adler, A. Ringh, O. Öktem, and J. Karlsson, *Learning to Solve Inverse Problems Using Wasserstein Loss*, preprint, arXiv:1710.10898, 2017.

[2] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174, https://doi.org/10.1017/S0962492919000059.

[3] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, *Invertible generative models for inverse problems: Mitigating representation error and dataset bias*, in Proceedings of ICML, 2020, pp. 399–409.

[4] M. A. Beaumont, W. Zhang, and D. J. Balding, *Approximate Bayesian computation in population genetics*, Genetics, 162 (2002), pp. 2025–2035, https://doi.org/10.1093/genetics/162.4.2025.

[5] J. M. Bernardo, *Reference posterior distributions for bayesian inference*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 41 (1979), pp. 113–128, https://doi.org/10.1111/j.2517-6161.1979.tb01066.x.

[6] J. M. Bernardo and A. F. Smith, *Bayesian Theory*, Wiley Ser. Probab. Stat. 405, John Wiley & Sons, New York, 2009.

[7] E. Bernton, P. E. Jacob, M. Gerber, and C. P. Robert, *On parameter estimation with the Wasserstein distance*, Inf. Inference, 8 (2019), pp. 657–676, https://doi.org/10.1093/imaiai/iaz003.

[8] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model reduction and neural networks for parametric pdes*, SMAI J. Comput. Math., 7 (2021), pp. 121–157, https://doi.org/10.5802/smai-jcm.74.

[9] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, *Sliced and radon Wasserstein barycenters of measures*, J. Math. Imaging Vision, 51 (2015), pp. 22–45, https://doi.org/10.1007/s10851-014-0506-3.

[10] N. Bonnotte, *Unidimensional and Evolution Methods for Optimal Transportation*, Ph.D. thesis, Université Paris Sud-Paris XI, Scuola normale superiore (Pise, Italie), 2013.

[11] B. Boys, M. Girolami, J. Pidstrigach, S. Reich, A. Mosca, and O. D. Akyildiz, *Tweedie moment projected diffusions for inverse problems*, Transact. Mach. Learn. Res., (2024), https://openreview.net/forum?id=4unJi0qrTE.

[12] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, *JAX: Composable Transformations of Python+NumPy Programs*, 2018, http://github.com/jax-ml/jax.

[13] T. Butler, J. Jakeman, and T. Wildey, *A Consistent Bayesian Formulation for Stochastic Inverse Problems Based on Push-Forward Measures*, preprint, arXiv:1704.00680, 2017.

[14] T. Butler, J. Jakeman, and T. Wildey, *Combining push-forward measures and Bayes' rule to construct consistent solutions to stochastic inverse problems*, SIAM J. Sci. Comput., 40 (2018), pp. A984–A1011, https://doi.org/10.1137/16M1087229.

[15] G. Casella, *An introduction to empirical Bayes data analysis*, Amer. Statist., 39 (1985), pp. 83–87, https://doi.org/10.1080/00031305.1985.10479400.

[16] J. Charrier, *Strong and weak error estimates for elliptic partial differential equations with random coefficients*, SIAM J. Numer. Anal., 50 (2012), pp. 216–246, https://doi.org/10.1137/100800531.

[17] J. Chemseddine, P. Hagemann, C. Wald, and G. Steidl, *Conditional Wasserstein Distances with Applications in Bayesian OT Flow Matching*, preprint, arXiv:2403.18705, 2024.

[18] F. R. Crucinio, V. De Bortoli, A. Doucet, and A. M. Johansen, *Solving a class of Fredholm integral equations of the first kind via Wasserstein gradient flows*, Stochastic Process. Appl., 173 (2024), 104374.

[19] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, Adv. Neural Inf. Process. Syst., 26 (2013).

[20] G. Daras, K. Shah, Y. Dagan, A. Gollakota, A. Dimakis, and A. Klivans, *Ambient diffusion: Learning clean distributions from corrupted data*, Adv. Neural Inf. Process. Syst, 36 (2023), pp. 288–313.

[21] M. Dashti and A. M. Stuart, *Uncertainty quantification and weak approximation of an elliptic inverse problem*, SIAM J. Numer. Anal., 49 (2011), pp. 2524–2542, https://doi.org/10.1137/100814664.

[22] I. Deshpande, Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing, *Max-sliced Wasserstein distance and its use for GANs*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10648–10656.

[23] M. M. Dunlop, M. A. Iglesias, and A. M. Stuart, *Hierarchical bayesian level set inversion*, Stat. Comput., 27 (2017), pp. 1555–1584, https://doi.org/10.1007/s11222-016-9704-8.

[24] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, *Training generative neural networks via maximum mean discrepancy optimization*, in Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI'15, Amsterdam, Netherlands, pp. 258–267.

[25] T. A. El Moselhy and Y. M. Marzouk, *Bayesian inference with optimal maps*, J. Comput. Phys., 231 (2012), pp. 7815–7850, https://doi.org/10.1016/j.jcp.2012.07.022.

[26] B. T. Feng, J. Smith, M. Rubinstein, H. Chang, K. L. Bouman, and W. T. Freeman, *Score-based diffusion models as principled priors for inverse imaging*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10520–10531.

[27] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, *POT: Python optimal transport*, J. Mach. Learn. Res., 22 (2021), pp. 1–8, http://jmlr.org/papers/v22/20-451.html.

[28] A. F. Gao, O. Leong, H. Sun, and K. L. Bouman, *Image reconstruction without explicit priors*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5.

[29] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, Boca Raton, FL, 1995, https://doi.org/10.1201/9780429258411.

[30] D. Hendrycks and K. Gimpel, *Gaussian Error Linear Units (GELUs)*, preprint, arXiv:1606.08415, 2016.

[31] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 6840–6851.

[32] G. Holler, K. Kunisch, and R. C. Barnard, *A bilevel approach for parameter learning in inverse problems*, Inverse Problems, 34 (2018), 115012, https://doi.org/10.1088/1361-6420/aade77.

[33] M. A. Iglesias, K. Lin, and A. M. Stuart, *Well-posed bayesian geometric inverse problems arising in subsurface flow*, Inverse Problems, 30 (2014), 114001, https://doi.org/10.1088/0266-5611/30/11/114001.
[34] E. T. Jaynes, *Prior probabilities*, IEEE Trans. Syst. Sci. Cybernet., 4 (1968), pp. 227–241, https://doi.org/10.1109/TSSC.1968.300117.
[35] H. Jeffreys, *An invariant form for the prior probability in estimation problems*, Proc. A, 186 (1946), pp. 453–461, https://doi.org/10.1098/rspa.1946.0056.
[36] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. Roy. Statist. Soc. Ser. A, 63 (2001), pp. 425–464, https://doi.org/10.1111/1467-9868.00294.
[37] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, preprint, arXiv:1412.6980, 2014.
[38] D. P. Kingma and M. Welling, *Auto-encoding Variational Bayes*, preprint, arXiv:1312.6114, 2013.
[39] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, *Generalized sliced Wasserstein distances*, Adv. Neural Inf. Process. Syst., 32 (2019).
[40] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Neural operator: Learning maps between function spaces with applications to pdes*, JMLR, 24 (2023), pp. 1–97.
[41] Q. Li, M. Oprea, L. Wang, and Y. Yang, *Stochastic Inverse Problem: Stability, Regularization and Wasserstein Gradient Flow*, preprint, arXiv:2410.00229, 2024.
[42] Q. Li, M. Oprea, L. Wang, and Y. Yang, *Inverse Problems Over Probability Measure Space*, preprint, arXiv:2504.18999, 2025.
[43] Q. Li, L. Wang, and Y. Yang, *Differential equation–constrained optimization with stochasticity*, SIAM/ASA J. Uncertain. Quantif., 12 (2024), pp. 549–578, https://doi.org/10.1137/23M1571162.
[44] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Fourier Neural Operator for Parametric Partial Differential Equations*, preprint, arXiv:2010.08895, 2020.
[45] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar, *Physics-informed neural operator for learning partial differential equations*, ACM/JMS J. Data Sci., 1 (2021), 9, https://dl.acm.org/doi/10.1145/3648506.
[46] A. Liutkus, U. Simsekli, S. Majewski, A. Durmus, and F.-R. Stöter, *Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions*, in Proceedings of ICML, 2019, pp. 4104–4113.
[47] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*, Nature Machine Intell., 3 (2021), pp. 218–229, https://doi.org/10.1038/s42256-021-00302-5.
[48] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini, *An Introduction to Sampling via Measure Transport*, preprint, arXiv:1602.05023, 2016.
[49] C. N. Morris, *Parametric empirical bayes inference: Theory and applications*, J. Amer. Statist. Assoc., 78 (1983), pp. 47–55, https://doi.org/10.1080/01621459.1983.10477920.
[50] K. Nadjahi, *Sliced-Wasserstein Distance for Large-Scale Machine Learning: Theory, Methodology and Extensions*. Institut polytechnique de, Paris, 2021.
[51] K. Nadjahi, V. De Bortoli, A. Durmus, R. Badeau, and U. Şimşekli, *Approximate Bayesian computation with the sliced-Wasserstein distance*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 5470–5474.
[52] K. Nguyen and N. Ho, *Energy-based sliced Wasserstein distance*, Adv. Neural Inf. Process. Syst., 36 (2024).
[53] A. O'Hagan, *Expert knowledge elicitation: Subjective but scientific*, Amer. Statist., 73 (2019), pp. 69–81, https://doi.org/10.1080/00031305.2018.1518265.
[54] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, JMLR, 22 (2021), pp. 1–64.
[55] D. V. Patel and A. A. Oberai, *GAN-based priors for quantifying uncertainty in supervised learning*, SIAM/ASA J. Uncertain. Quantif., 9 (2021), pp. 1314–1343, https://doi.org/10.1137/20M1354210.
[56] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory*, John Wiley & Sons, New York, 2000.
[57] S. J. Reddi, S. Kale, and S. Kumar, *On the convergence of Adam and beyond*, in International Conference on Learning Representations, 2018, https://openreview.net/forum?id=ryQu7f-RZ.

[58] M. Rixner and P.-S. Koutsourelakis, *A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables*, J. Comput. Phys., 434 (2021), 110218, https://doi.org/10.1016/j.jcp.2021.110218.

[59] H. Robbins, *The empirical Bayes approach to statistical decision problems*, Ann. Math. Stat., 35 (1964), pp. 1–20, https://doi.org/10.1214/aoms/1177703729.

[60] H. E. Robbins, *An empirical Bayes approach to statistics,*, in Breakthroughs in Statistics: Foundations and Basic Theory, Springer, Berlin, 1992, pp. 388–394.

[61] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–423, https://doi.org/10.1214/ss/1177012413.

[62] A. Sagiv, *The Wasserstein distances between pushed-forward measures with applications to uncertainty quantification*, Commun. Math. Sci., 18 (2020), pp. 707–724, https://doi.org/10.4310/CMS.2020.v18.n3.a6.

[63] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Springer, Cham, 2015.

[64] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, *Equivalence of distance-based and RKHS-based statistics in hypothesis testing*, Ann. Statist., 41 (2013), pp. 2263–2291, https://doi.org/10.1214/13-AOS1140.

[65] A. Sinha, P. Malo, and K. Deb, *A review on bilevel optimization: From classical to evolutionary approaches and applications*, IEEE Trans. Evol. Comput., 22 (2017), pp. 276–295, https://doi.org/10.1109/TEVC.2017.2712906.

[66] A. Smola, A. Gretton, L. Song, and B. Schölkopf, *A Hilbert space embedding for distributions*, in Algorithmic Learning Theory, Lecture Notes in Comput. Sci. 4754, Springer, Cham, 2007, pp. 13–31, https://doi.org/10.1007/978-3-540-75225-7.

[67] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, Adv. Neural Inf. Process. Syst., 32 (2019).

[68] Y. Song and D. P. Kingma, *How to Train Your Energy-Based Models*, preprint, arXiv:2101.03288, 2021.

[69] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-based generative modeling through stochastic differential equations*, in Proceedings of 9th International Conference on Learning Representations, ICLR, 2021, Austria, https://openreview.net/forum?id=PxTIG12RRHS.

[70] G. J. Székely and M. L. Rizzo, *Energy statistics: A class of statistics based on distances*, J. Statist. Plann. Inference, 143 (2013), pp. 1249–1272, https://doi.org/10.1016/j.jspi.2013.03.018.

[71] S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly, *Inferring coalescence times from DNA sequence data*, Genetics, 145 (1997), pp. 505–518, https://doi.org/10.1093/genetics/145.2.505.

[72] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton, *Energy-based models for sparse overcomplete representations*, JMLR, 4 (2003), pp. 1235–1260.

[73] A. Vadeboncoeur, Ö. D. Akyildiz, I. Kazlauskaite, M. Girolami, and F. Cirak, *Fully probabilistic deep models for forward and inverse problems in parametric PDEs*, J. Comput. Phys., 491 (2023), 112369, https://doi.org/10.1016/j.jcp.2023.112369.

[74] A. Vadeboncoeur, I. Kazlauskaite, Y. Papandreou, F. Cirak, M. Girolami, and O. D. Akyildiz, *Random grid neural processes for parametric partial differential equations,* in Proceedings of ICML, 2023.

[75] R. Vershynin, *High-dimensional Probability: An Introduction with Applications in Data Science*, Cambridge Ser. Stat. Probab. Math. 47, Cambridge University Press, Cambridge, UK, 2018.

[76] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference*, Found. Trends Mach. Learn., 1 (2008), pp. 1–305, https://doi.org/10.1561/2200000001.

[77] R. Z. Zhang, X. Xie, and J. Lowengrub, *BiLO: Bilevel Local Operator Learning for PDE Inverse Problems*, preprint, arXiv:2404.17789, 2024.

[78] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data*, J. Comput. Phys., 394 (2019), pp. 56–81, https://doi.org/10.1016/j.jcp.2019.05.024.