Solving inverse problems using normalizing flow prior: Application to optical spectra

Jun H. Park ,[1,*] Juyeob Lee ,[2] and Jungseek Hwang [3,†]

[1] *School of Mechanical Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea*
[2] *Department of Applied Artificial Intelligence, Sungkyunkwan University, Seoul, 03063, Republic of Korea*
[3] *Department of Physics, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea*

(Received 15 January 2024; revised 1 March 2024; accepted 26 March 2024; published 16 April 2024)

We introduce a machine learning approach for solving ill-posed inverse problems, specifically addressing the Fredholm integral equation of the first kind. Harnessing the powerful capabilities of normalizing flows to approximate data distributions, combined with a robust probabilistic framework, our approach stands out by delivering robust solutions capable of handling high-level noises and out-of-distribution data and providing uncertainty estimation. A distinct feature lies in the unsupervised learning framework inherent in deep generative models, providing our approach with unparalleled flexibility across diverse experimental setups. This flexibility is exemplified through the successful application of our method to measured optical spectra.

DOI: 10.1103/PhysRevB.109.165130

I. INTRODUCTION

In the last decade, machine learning approaches have gained widespread acceptance across diverse scientific fields [1–4]. This is mainly driven by potential of deep neural networks to approximate almost any function based on a principle established by the universal approximation theorem ([5] and references therein). When training data are abundant, the theorem ensures that deep neural networks possess the desired expressive capacity, enabling them to deliver competitive results in applications ranging from materials discovery and quantum phase classification to genomic data mining.

This study focused on addressing the inversion of the Fredholm integral equation of the first kind using deep neural networks. This equation is expressed as

$$y(t) = \int_a^b d\tau\, x(\tau)\, k(t, \tau), \tag{1}$$

where $k(t, \tau)$ is called a kernel. The primary objective of our study was to recover a function $x(\cdot)$ from the observed data $y(\cdot)$ typically obtained experimentally. Notably, such inverse problems are well known to be ill posed [6] and frequently occur in many areas of physics [7–9]. Based on the universal approximation theorem, several methodologies, including those proposed in [10–12], employ supervised learning to address these challenging problems.

More specifically, this study focused on retrieving of the electron-boson spectral density (EBSD) function, denoted as $I^2\chi(\cdot)$, from the optical scattering rate spectra, represented by $1/\tau^{\mathrm{op}}(\cdot)$, acquired from experimental observations. The relationship between these two quantities is governed by the generalized Allen formula [7,13], which is expressed as

$$\frac{1}{\tau^{\mathrm{op}}(\omega; T)} = \int_0^\infty d\Omega\, I^2\chi(\Omega; T)\, K(\omega, \Omega; T), \tag{2}$$

where $K(\omega, \Omega; T)$, known as the Shulga's kernel [13], is expressed as

$$K(\omega, \Omega; T) = \frac{\pi}{\omega}\left[2\omega \coth\left(\frac{\Omega}{2T}\right) - (\omega + \Omega) \coth\left(\frac{\omega + \Omega}{2T}\right) + (\omega - \Omega) \coth\left(\frac{\omega - \Omega}{2T}\right)\right].$$

Temperature T is a parameter in this relationship. Direct comparison shows that Eq. (2) has the form of Eq. (1). Considering the presence of noise, which occurs commonly in experimental setups, the inverse problem of obtaining the EBSD function can be written as

$$\mathbf{y} = A\mathbf{x} + \boldsymbol{\eta}, \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ represent the discretized versions of $I^2\chi(\cdot)$ and $1/\tau(\cdot)$, respectively. The integral in Eq. (2) is discretized to obtain an $m \times n$ matrix A. The observation noise, $\boldsymbol{\eta}$, has a normal distribution, $\mathcal{N}(\mathbf{0}, \sigma^2 I)$ with zero mean and σ^2 variance.

Given the observation, $\mathbf{y}$, the above problem may be solved by finding an $\mathbf{x}$ that maximizes the likelihood, or equivalently, the log likelihood, $\log p(\mathbf{y}|\mathbf{x}) = \log \mathcal{N}(A\mathbf{x}, \sigma^2 I)$. However, because the problem is ill posed, identifying this unique $\mathbf{x}$ is difficult, which potentially leads to several comparable solutions. The maximum *a posteriori* (MAP) approach is an effective framework for solving inverse problems. It is expressed as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \{\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x})\}. \tag{4}$$

In the MAP, the log prior term, $\log p(\mathbf{x})$, enables solutions that align with the prior criteria, effectively narrowing the solution

[*] jun.park@skku.edu
[†] jungseek@skku.edu

space for the original ill-posed problem. Prior information is not explicitly considered in existing studies based on supervised learning [10–12]. It is primarily employed implicitly during the generation of training data $\mathbf{x}$ to ensure physically plausible shapes. During training, a neural network learns to minimize the loss, which is the distance between the predicted $\hat{\mathbf{x}}$ and the true $\mathbf{x}$ data. Although post training, these networks deliver competitive results; they remain blackbox models, lacking explanatory capabilities and being susceptible to the biases inherent in the training data.

The recurrent inference machine (RIM) approach [14–16] addresses inverse problems through an iterative application combining likelihood criteria and prior information. Although an RIM is still based on supervised learning, its implicit incorporation of a prior via iteration yields an improved performance. Moreover, its basis in the iterative Tikhonov regularization enhances its explainability and reliability [16].

The prior required in the MAP framework can be modeled using deep generative models (DGMs), which include generative adversarial networks (GANs) [17], variational autoencoders (VAEs) [18,19], diffusion models [20,21], and generative pretrained transformers [22]. DGMs have gained significant attention in the machine learning community and various fields such as economics, social sciences, and education. The key feature of these models is their ability to approximate data distributions, thereby enabling the generation of realistic samples through random drawings.

Inverse problems were solved for the first time using DGMs, such as GANs or VAEs, in a seminal study by Bora *et al.* [23], inspired by the concept of compressed sensing. In this approach, a generator G learns a mapping from a randomly drawn $\mathbf{z}$ in a latent space, $\mathcal{Z} \subset \mathbb{R}^k$ to $\mathbf{x}$ in the data space, $\mathcal{X} \subset \mathbb{R}^n$. The generator produces $\mathbf{x} = G(\mathbf{z})$ that closely resembles the given data. The inverse problem is then addressed by finding a $\hat{\mathbf{z}}$ that minimizes $||\mathbf{y} - AG(\mathbf{z})||$ and then substituting it into the generator to obtain $\hat{\mathbf{x}} = G(\hat{\mathbf{z}})$. However, because the latent space in these models is typically smaller than the data space, i.e., $(k \ll n)$, the range of G does not cover the entire data space, $\mathcal{X}$, leading to an intractable convergence problem, as observed in [23]. Although Shah and Hedge [24] addressed this issue by refining the GAN algorithm with a projective gradient descent and a GAN prior, GANs still suffer from mode collapse and are unsuitable for probabilistic frameworks [25].

In this study, we developed a method for addressing ill-posed inverse problems based on the MAP framework by leveraging an explicit prior modeled using a DGM called a normalizing flow (NF). NFs establish an invertible mapping from $\mathcal{Z} \subset \mathbb{R}^n$ to $\mathcal{X} \subset \mathbb{R}^n$, eliminating the coverage issue of GANs and VAEs owing to their smaller latent spaces. The computational burden arising from the invertibility of NFs is addressed using various methods, such as creative network architectures (elaborated in the Methods section). The method developed in this study is similar to that in [26], among the numerous NF-based approaches for solving inverse problems [26–28]. It focuses on uncertainty estimation through variational inference (VI) [29] in the context of solving inverse problems.

Our study deviates from previous work that primarily focused on image-related tasks like denoising and inpainting, as we address the inversion of the Fredholm integral equation of the first kind. Leveraging NFs' ability to estimate probabilities and generate out-of-distribution (OOD) data [30], our approach proves advantageous for addressing ill-posed inverse problems across scientific disciplines. The explicit integration of prior and likelihood in our method establishes an ideal platform for tackling such problems effectively. Our approach also demonstrated robustness to noise, estimated uncertainty estimation for solution reliability, and yielded results comparable to those of maximum entropy methods (MEMs) when obtaining EBSD functions from optical spectra. More importantly, the unsupervised learning framework inherent in deep generative models imparts enhanced flexibility to our approach. A single training of our model proves sufficient to obtain EBSD functions from various experiments with different temperature setups, showing another advantage over supervised machine learning approaches.

II. METHODS

An NF approximates a data distribution $p_X(\mathbf{x})$ using a sequence of invertible transformations. The fundamental concept is that complex distributions can be constructed by applying simple transformations sequentially. Let $\mathbf{z} \in \mathbb{R}^n$ be a random vector with a standard normal distribution, i.e., $p_Z(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$, where I is an $n \times n$ identity matrix. An invertible function $g : \mathbf{z} \to \mathbf{x}$ modifies the distribution of $\mathbf{z}$ according to the following formula:

$$p_X(\mathbf{x}) = p_Z(\mathbf{z})\left|\det\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right| = p_Z(g^{-1}(\mathbf{x}))\left|\det\frac{\partial g^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right|.$$

The composition of these transformations is represented as

$$G := g_K \circ g_{K-1} \cdot \circ g_2 \circ g_1,$$

and the change in the variable formula (in a logarithmic form) can be expressed as

$$\log p_X(\mathbf{x}) = \log p_Z(G^{-1}(\mathbf{x})) + \log\left|\det\frac{\partial G^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right|, \tag{5}$$

where $G^{-1}(\mathbf{x}) = (g_1^{-1} \circ g_2^{-1} \circ \cdots \circ g_{K-1}^{-1} \circ g_K^{-1})(\mathbf{x})$. The log of the Jacobian determinant (second term) on the right-hand side can be expressed as

$$\log\left|\det\frac{\partial G^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right| = \log\prod_{k=1}^{K}\left|\det\frac{\partial g_k^{-1}(\mathbf{x})}{\partial \mathbf{x}_k}\right| = \sum_{k=1}^{K}\log\left|\det\frac{\partial g_k^{-1}(\mathbf{x})}{\partial \mathbf{x}_k}\right|.$$

The first equality originates from the chain rule and the factoring property of the determinant. Assuming a training set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with an independent and identical distribution, a parametric model can be established for the data distribution and finding the optimal parameter that maximizes the log likelihood,

$$\hat{\psi} = \arg\max_{\psi} \sum_{n=1}^{N} \log p_X(\mathbf{x}_n; \psi), \tag{6}$$

using any suitable optimization algorithm.

To implement this algorithm using a deep neural network, two design criteria must be satisfied: the network must be invertible, and the computation of the determinant of the Jacobian term should be computationally feasible. In general, the calculation requires $O(D^3)$ operations, which are prohibitively expensive for a reasonable data dimension D. Since the inception of NFs, various procedures have been derived for fulfilling these criteria. Examples include planar and radial flows [31,32], nonlinear independent component estimation [33], real-valued nonvolume-preserving flows [34], glow [35], neural spline flows [36], and others (see [37]). In this study, we adopted a neural spline flow [36] for modeling the data distribution because it demonstrates competitive flexibility while being analytically invertible.

Owing to the invertibility of this network, we could evaluate the probability of any given datum $\mathbf{x}$ using the parametrized version of Eq. (5),

$$\log p_X(\mathbf{x};\psi) = \log p_Z\big(G_\psi^{-1}(\mathbf{x})\big) + \log\left|\det\frac{\partial G_\psi^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right|, \tag{7}$$

and generate a new sample using

$$\mathbf{x} = G_\psi(\mathbf{z}),$$

where $\mathbf{z} \sim p_Z(\mathbf{z})$, which is typically assumed to be a multivariate Gaussian. These two capabilities of an NF play crucial roles in solving inverse problems.

In this study, the inverse problem considered was solved using the following procedure. First, we trained the generator using Eqs. (6) and (7) to obtain $G_{\hat\psi}$, which is called the pretrained generator hereafter. Second, we solved the following optimization problem:

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \|AG_{\hat\psi}(\mathbf{z}) - \mathbf{y}\|_2^2. \tag{8}$$

Finally, we substituted its solution $\hat{\mathbf{z}}$ into the pretrained generator as follows:

$$\hat{\mathbf{x}} = G_{\hat\psi}(\hat{\mathbf{z}}), \tag{9}$$

which yielded the solution of the inverse problem. Notably, the solution, $\hat{\mathbf{x}}$, satisfies the likelihood maximization as well as the prior because it was in the range of the generator while fulfilling Eq. (8). Note also that Eq. (8) is an optimization problem over $\mathbf{z}$ with a fixed network parameter $\hat\psi$. Empirically, it can be solved using the gradient descent approach, although it is nonconvex owing to the generator, $G_{\hat\psi}$. The pseudocode outlining this approach is presented in Algorithm 1.

ALGORITHM 1. Inverse problem.

```
Input: $G_{\hat\psi}$, $\mathbf{y}$, $A$, $T$
Output: $\hat{\mathbf{x}}$
  Initialize $\mathbf{z}_0$
  while $t \leqslant T$ do
      $\mathbf{z}_t \leftarrow \mathbf{z}_t - \gamma(\frac{\partial G}{\partial \mathbf{z}}(\mathbf{z}_t))^T A^T (AG_{\hat\psi}(\mathbf{z}_t) - \mathbf{y})$
      $t \leftarrow t + 1$
  end while
  $\hat{\mathbf{x}} \leftarrow G_{\hat\psi}(\mathbf{z}_T)$
```

Although Algorithm 1 provides solutions to inverse problems, it does not quantify uncertainties of these solutions. Quantifying uncertainties is crucial for the reliability of obtained solutions and for serving as the starting point for various downstream tasks. Uncertainty estimation can be performed using the posterior density as follows:

$$p(\mathbf{x}|\mathbf{y};\hat\psi) = \frac{p(\mathbf{y}|\mathbf{x})p_X(\mathbf{x};\hat\psi)}{\int p(\mathbf{y}|\mathbf{x})p_X(\mathbf{x};\hat\psi)d\mathbf{x}}.$$

In this study, we used the prior obtained from the pretrained generator. Obtaining the posterior density is typically challenging owing to the high dimensional integration in the denominator. Therefore, we approximated the posterior using VI [38], where the variational density, $q_X(\mathbf{x};\phi)$, was obtained to minimize the difference between $p(\mathbf{x}|\mathbf{y};\hat\psi)$ and $q_X(\mathbf{x};\phi)$. The goal of VI can be expressed as

$$\hat\phi = \arg\min_{\phi} \mathrm{KL}(q_X(\mathbf{x};\phi)||p(\mathbf{x}|\mathbf{y};\hat\psi)), \tag{10}$$

where KL is the Kullback-Leibler divergence defined as

$$\mathrm{KL}(f||g) := \mathbb{E}_{f(\mathbf{x})}\left[\log\frac{f(\mathbf{x})}{g(\mathbf{x})}\right] = \int f(\mathbf{x})\log\frac{f(\mathbf{x})}{g(\mathbf{x})}d\mathbf{x}.$$

Minimizing the KL divergence is equivalent to maximizing the evidence lower bound (ELBO) (see Appendix A), i.e.,

$$\mathcal{L}(\phi) = \mathbb{E}_{q_X(\mathbf{x};\phi)}[\log p(\mathbf{y}|\mathbf{x})] - \mathrm{KL}(q_X(\mathbf{x};\phi)||p_X(\mathbf{x};\hat\psi)). \tag{11}$$

Maximizing the ELBO is identical to finding a $q_X(\mathbf{x};\phi)$ that maximizes the expected log likelihood (first term) and minimizes the KL (second term). This is equivalent to finding a variational density that is close to the prior, $p_X(\mathbf{x};\hat\psi)$.

The performance of VI depends on the expressiveness of the variational density, $q_X(\mathbf{x};\phi)$; therefore, we adopted another NF based on affine coupling layers [34]. We combined the new NF, denoted as $G_\phi : \boldsymbol{\epsilon} \to \mathbf{z}$, with the pretrained generator, $G_{\hat\psi}$. The VI framework is represented as a mapping sequence as follows:

$$\boldsymbol{\epsilon} \overset{G_\phi}{\longmapsto} \mathbf{z} \overset{G_{\hat\psi}}{\longmapsto} \mathbf{x}.$$

The implementation of this procedure was a type of the blackbox VI [39]. However, owning to its ill-posed nature, a simple implementation did not produce moderate results. To address this problem, we adopted a concept from the iterative Tikhonov regularization. Specifically, we utilized the following gradient, derived from the preconditioned Landweber iteration [40]:

$$\nabla \log p(\mathbf{y}|\mathbf{x}) = (A^TA + h^2I)^{-1}A^T(\mathbf{y} - A\mathbf{x}), \tag{12}$$

where h is the regularization parameter. Combining all results, we developed Algorithm 2 (see Appendix B for the derivation).

III. RESULTS AND DISCUSSIONS

We first trained the NF, G_ψ, to approximate the data distribution, $p_X(\mathbf{x})$, which was the prior in the MAP estimate [Eq. (4)]. The generator G_ψ is modeled using a sequence of neural spline flows [36] with a total sequence length of three. The specific configuration includes 64 rational quadratic functions, and the boundary value is set to five. The training

ALGORITHM 2. Variational inference.

```
Input: $G_{\hat{\psi}}, \mathbf{y}, A, L, T$
Output: $q_X(\mathbf{x}; \hat{\phi})$
  Initialize $\phi_0$
  while $t \leqslant T$ do
      for $l = 1, \dots, L$ do
          $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, I)$
          $\mathbf{z}^{(l)} \leftarrow G_{\phi_t}(\boldsymbol{\epsilon}^{(l)})$
          $\mathbf{x}^{(l)} \leftarrow G_{\hat{\psi}}(\mathbf{z}^{(l)})$
          $\mathbf{p}^{(l)} \leftarrow -\frac{1}{2}(A^TA + h^2 I)^{-1}\|\mathbf{y} - A\mathbf{x}^{(l)}\|_2^2$
      end for
      $\mathcal{L}(\phi_t) \leftarrow \frac{1}{L}\sum_{l=1}^{L}\mathbf{p}^{(l)} - \frac{1}{2}\|\mathbf{z}^{(l)}\|_2^2 + \log|\det\frac{\partial \mathbf{z}^{(l)}}{\partial \boldsymbol{\epsilon}}|$
      $\phi_t \leftarrow \phi_t + \gamma\nabla\mathcal{L}(\phi_t)$
  end while
  $\hat{\phi} \leftarrow \phi_T$
```

aimed to maximize the log likelihood expressed in Eq. (6) using the Adam optimizer [41] with a learning rate $\gamma = 1 \times 10^{-3}$. We utilized 100 000 data that were mixtures of Gaussians resembling $I^2\chi$. When the number of training data was insufficient, for example, 1000 in this study, nonphysical artifacts like intermittent abrupt jumps were prevalent in the generated samples. As the number of training data increased, these artifacts gradually diminished. The dataset was partitioned into training, validation, and test sets with ratios of 0.8, 0.1, and 0.1, respectively. Once the training was completed, various samples were generated from the prior simply using $\mathbf{x} = G_{\hat{\psi}}(\mathbf{z})$, where $\mathbf{z} \sim p_Z(\mathbf{z})$. Here, $p_Z(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$. The mapping from $\mathbf{z}$ to $\mathbf{x}$ was deterministic, and the randomness originated from sampling only. Figures 1(a) and (b) show 100 training data and 100 generated samples from $p_X(\mathbf{x}; \hat{\psi})$, respectively. Notably, the generated samples exhibit more diversity than the training data, and some of the generated samples are not physically plausible, i.e., they become negative. The former shows the NF's capability of generating OOD samples, whereas the latter indicates that the NF does not make assumptions about the inherent structure of the data, such as smoothness. We acknowledge that applying ReLU activation at the end could prevent nonnegative samples. However, we chose not to apply this, as it might risk accepting incorrect solutions to the inverse problem by modifying unphysical ones into physical ones. In this section, we present the use of functions $I^2\chi$ and $1/\tau^{\text{op}}$ interchangeably with their discretized versions $\mathbf{x}$ and $\mathbf{y}$, respectively.

Given observation $\mathbf{y}$, the solution to the inverse problem was obtained using Algorithm 1. As mentioned earlier, optimization was performed with respect to the $\mathbf{z}$ variable using the Adam optimizer. Figures 1(c) and 1(d) show solutions $\hat{\mathbf{x}}$'s for certain $\mathbf{y}$'s from a noiseless data set that is not used for the training of G_ψ. To ensure numerical stability, the data are scaled by dividing $\mathbf{y}$'s by 300. For comparison, the true $\mathbf{x}$'s and their reconstructions $A\hat{\mathbf{x}}$'s are also plotted.

To assess the robustness of our approach to noise, we selected an arbitrary data pair $(\mathbf{x}, \mathbf{y})$ from the test dataset that was not used during the NF training. We then added different levels of Gaussian noise to $\mathbf{y}$ to simulate observation noises. The noise levels were determined from the standard deviations, set to σ times at the maximum value of each $\mathbf{y}$.

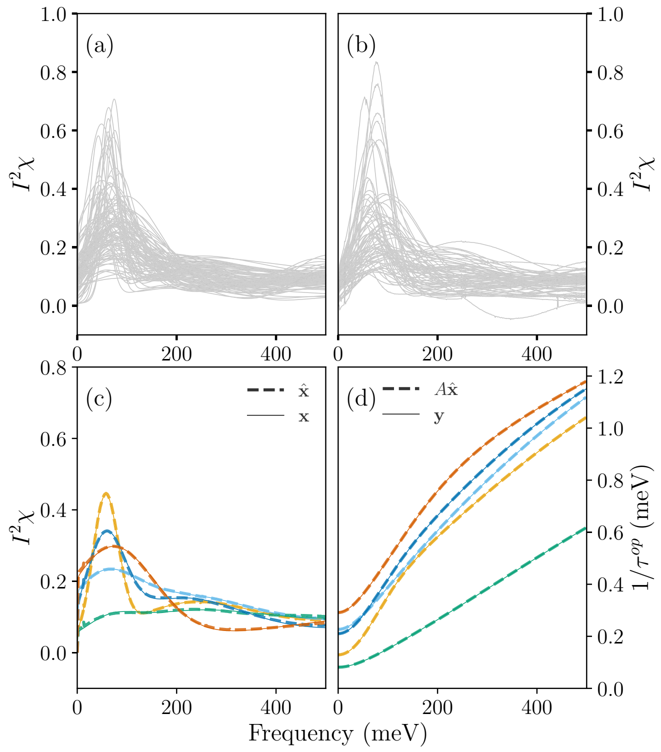

FIG. 1. Hundred random samples from (a) training data and (b) data distribution $p_X(\mathbf{x}; \hat{\psi})$. (c) Some solutions $\hat{\mathbf{x}}$ of inverse problem with true $\mathbf{x}$'s using Algorithm 1 and (d) corresponding reconstructions $A\hat{\mathbf{x}}$'s and true $\mathbf{y}$'s. (All $1/\tau^{\text{op}}$'s are scaled down by 300.)

Figure 2(a) shows resulting $\mathbf{x}$'s obtained from $\mathbf{y}$'s for the different noise levels as shown in Figs. 2(b)–2(e). Interestingly, as the noise level varies from minimal to very strong values, the corresponding solutions of the inverse problem do not vary significantly, even with substantial noise (e.g., $\sigma = 0.1$), as shown in Fig. 2(a).

The results of the uncertainty estimation are presented in Fig. 3 for a randomly selected observation data $\mathbf{y}$ from the test dataset with two different noise levels: $\sigma = 0.01$ and $\sigma = 0.1$. The variational densities were obtained for each case using Algorithm 2. We employ a sequential application of affine coupling layers [34] with a length of six and hidden

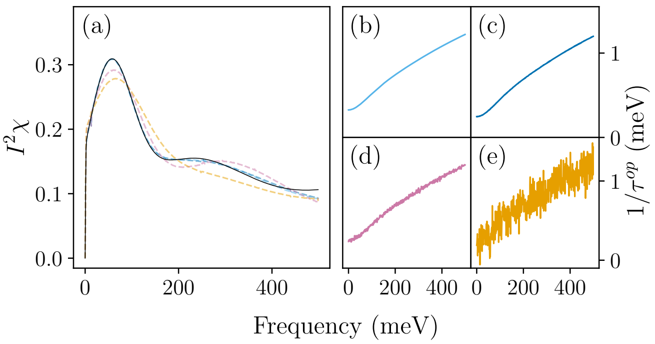

FIG. 2. (a) Solutions of the inverse problem using Algorithm 1 from observations with different noise levels: (b) $\sigma = 0.0001$, (c) $\sigma = 0.001$, (d) $\sigma = 0.01$, and (e) $\sigma = 0.1$. (All $1/\tau^{\text{op}}$'s are scaled down by 300.)

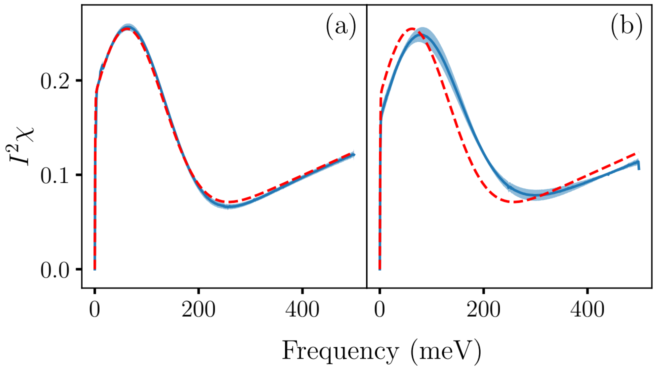

FIG. 3. Sample means (blue solid) and error bars of two standard deviations (filled region) are shown with true $\mathbf{x}$'s (red dashed) for noise levels (a) $\sigma = 0.01$ and (b) $\sigma = 0.1$.

dimensions of 1024 to model G_ϕ. The total number of Monte Carlo samples was set to $L = 20$, with a regularization parameter of $h = 0.001$. For optimization, we used RMSprop with a learning rate of 5×10^{-6} and a momentum value of 0.9. The figures depict the sample means and error bars (filled areas) based on 100 samples generated from the variational densities. For comparison, the true $\mathbf{x}$'s are also plotted. The relatively narrow filled regions in both figures indicates the reliability of our approach, even in the presence of uncertainties denoted by the relative thickness. The error bars represent two sample standard deviations above and below the sample mean.

Finally, we applied Algorithm 1 to real experimental data consisting of measured optical spectra, $1/\tau^{\mathrm{op}}(\omega, T)$: one optimally doped sample ($T_c = 96$ K) and two overdoped samples ($T_c = 82$ and 60 K) of $\mathrm{Bi_2Sr_2CaCu_2O_{8+\delta}}$ (Bi-2212). T_c is the superconducting transition temperature. The three samples are denoted as OPT96, OD82, and OD60, respectively, and represented in different colors in Fig. 4. These experiments were performed at three different temperature setups: $T = 100$ K, 200 K, and 300 K.

These experimental spectra were used as input for Algorithm 1 for inference determination. The results for each temperature setup are shown in Fig. 4 in the top ($T = 100$ K), middle ($T = 200$ K), and bottom ($T = 300$ K) panels. We denote the result of Algorithm 1 as "flow" in the figure. In the left column of Fig. 4, the results of the NF (dashed lines) are compared with those of the MEM (dotted lines) reported in [42]. In the right column, the reconstructed optical spectra obtained using the NF (dashed lines) are compared with those using the MEM (dotted lines) and the experimental results (solid curves). The resulting $I^2\chi(\omega)$'s obtained using NF and MEM are comparable to each other.

The coupling constant (λ) is defined as $\lambda(T) \equiv 2\int_0^{\omega_c}[I^2\chi(\omega, T)/\omega]d\omega$, where ω_c is the cutoff frequency, which was 500 meV in our study. The coupling constant is closely related to the strength of the interactions between electrons in a material. The coupling constants derived using NF (solid symbol) and MEM (open symbol) from the resulting $I^2\chi(\omega)$'s are shown in Fig. 5. Both coupling constants exhibit similar doping (or T_c) and temperature dependencies. Therefore, the interactions may be associated with the antiferromagnetic fluctuations, based on the phase diagram of the Bi-2212 cuprate system.

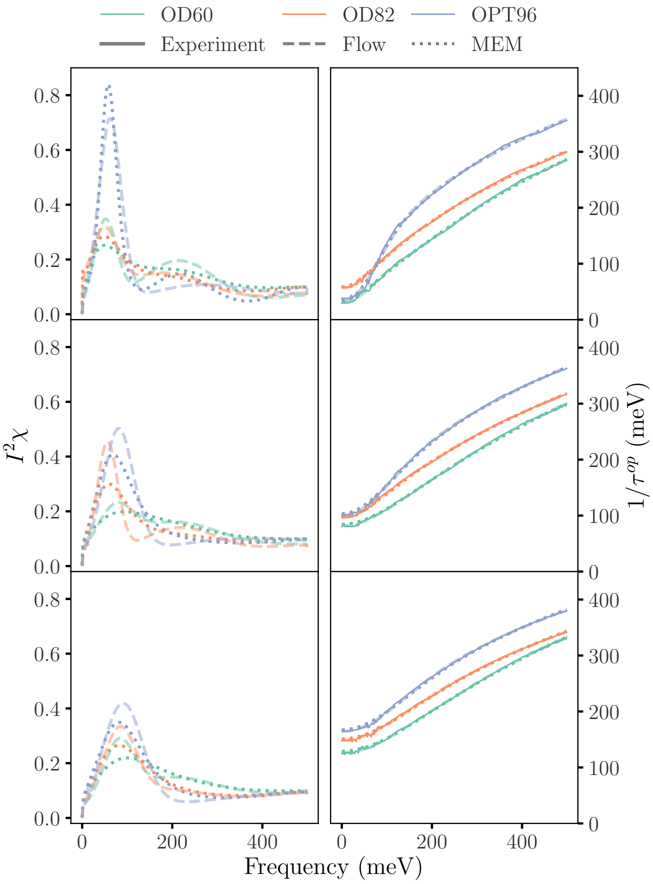

FIG. 4. Left column: comparison of inference results of $I^2\chi$ obtained using NF (dashed lines) and MEM (dotted lines). Right column: experimental observations (solid curves) and reconstructions using results of NF (dashed lines), and MEM (dotted) of $1/\tau^{\mathrm{op}}$. Each experiment is performed in different temperature setups: top ($T = 100$ K), middle ($T = 200$ K), and bottom ($T = 300$ K). Samples (OD60, D82, and OPT96) are differentiated using the same color code in all figures.

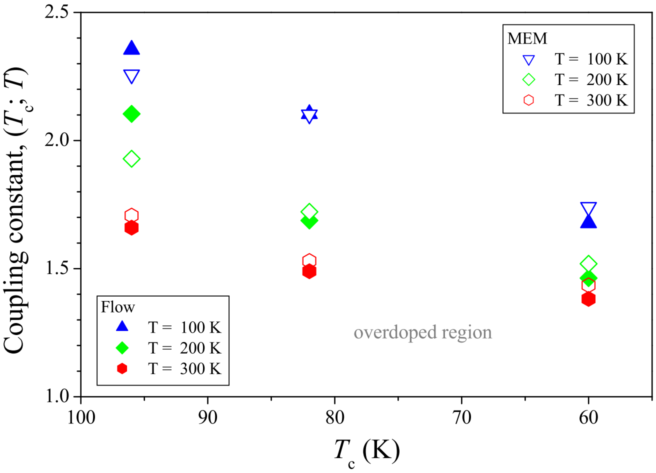

FIG. 5. Comparison of coupling constants obtained from inference results of $I^2\chi$ using NF and MEM are compared at different temperature setups: $T = 100$ K, 200 K, and 300 K.

IV. CONCLUSIONS

We have investigated a machine learning approach to solve the inverse problem of the Fredholm integral equation of the first kind. In contrast to previous approaches based on supervised learning, our method leverages deep generative models and offers a more transparent perspective on the solution procedures. This enhances the reliability and explainability of the solution within an appropriate probabilistic framework.

While delivering results comparable to those achieved by MEM, our approach shows significant robustness to noise. The prior distribution obtained from the NF provides more diverse samples than the trained data. This diversity is advantageous for handling OOD data and offers a degree of flexibility beyond the inductive bias of domain experts. The reliability of the solutions, as assessed through uncertainty estimation, offers an additional insights for solving inverse problems. Importantly, our method is versatile across various experimental temperature setups, distinguishing it from supervised learning-based approaches, which are frequently constrained to specific temperature setups.

Although we have demonstrated the capacity of NFs to generate OOD samples, a comprehensive quantification of this capability has not been provided in the current work. Recognizing the importance of NFs' generalization capability for handling OOD data, we plan to explore and address this aspect in future work.

ACKNOWLEDGMENTS

This study was supported by the National Research Foundation of Korea (NRFK Grants No. 2017R1D1A1B03029534 and No. 2021R1A2C101109811).

APPENDIX A: VARIATIONAL INFERENCE (VI) AND EVIDENCE LOWER BOUND (ELBO)

In variational inference (VI) [38], the posterior, $p(\mathbf{x}|\mathbf{y})$, is approximated by a variational density $q(\mathbf{x};\phi)$ that minimizes the KL divergence from $q(\mathbf{x};\phi)$ to $p(\mathbf{x}|\mathbf{y})$ is minimized, i.e.,

$$\hat{\phi} = \arg\min_{\phi} \mathrm{KL}(q(\mathbf{x};\phi)||p(\mathbf{x}|\mathbf{y})).$$

Notably, the density estimation of $p_X(\mathbf{x}|\mathbf{y})$ becomes an optimization problem through VI. From the definition of the KL divergence, we obtain

$$\begin{aligned}\mathrm{KL}(q(\mathbf{x};\phi)||p(\mathbf{x}|\mathbf{y})) &= \int q(\mathbf{x};\phi)\log\frac{q(\mathbf{x};\phi)}{p(\mathbf{x}|\mathbf{y})}d\mathbf{x}\\ &= \int q(\mathbf{x};\phi)\log\frac{q(\mathbf{x};\phi)p(\mathbf{y})}{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}d\mathbf{x},\end{aligned}$$

from which, one can show that

$$\log p(\mathbf{y}) = \mathcal{L}(\phi) + \mathrm{KL}(q(\mathbf{x};\phi)||p(\mathbf{x}|\mathbf{y})), \tag{A1}$$

where

$$\mathcal{L}(\phi) = \mathbb{E}_{q(\mathbf{x};\phi)}[\log p(\mathbf{y}|\mathbf{x})] - \mathbb{E}_{q(\mathbf{x};\phi)}\left[\log\frac{q(\mathbf{x};\phi)}{p(\mathbf{x})}\right]. \tag{A2}$$

Given that the KL divergence is nonnegative, Eq. (A1) indicates that minimizing it is equivalent to maximizing $\mathcal{L}(\phi)$. Note that $\mathcal{L}(\phi)$ is frequently referred to as the negative variational free energy or evidence lower bound (ELBO).

Traditionally, Eq. (A2) is solved using a mean-field assumption, where $q(\mathbf{x};\phi)$ is factored into $\prod q_i(\mathbf{x}_i;\phi)$ and solved for each $q_i(\mathbf{x}_i;\phi)$ while keeping other $q_{j\neq i}(\mathbf{x}_j;\phi)$'s fixed. Mean-field-based approaches are advantageous because they are analytically solvable. However, they are limited in dealing with complex posterior distributions owing to the incorrect independence assumption of $q_i(\mathbf{x}_i;\phi)$'s. In recent decades, numerous studies have explored methods for generating more expressive variational densities than the mean-field approach, making them scalable for dealing with large datasets [39,43–45].

APPENDIX B: UNCERTAINTY ESTIMATION USING VARIATIONAL INFERENCE (VI)

For uncertainty estimation, VI was utilized to obtain a variational density to approximate the hard-to-obtain posterior density, $p(\mathbf{x}|\mathbf{y})$. We achieved this by combining the pretrained generator, $G_{\hat{\psi}}$, with an additional normalizing flow, G_ϕ such that

$$\boldsymbol{\epsilon} \overset{G_\phi}{\longmapsto} \mathbf{z} \overset{G_{\hat{\psi}}}{\longmapsto} \mathbf{x},$$

where $\boldsymbol{\epsilon} \sim q_\epsilon(\boldsymbol{\epsilon};\theta)$. On changing the variable formula, we obtain

$$q_Z(\mathbf{z};\theta,\phi) = q_\epsilon(\boldsymbol{\epsilon};\theta)\left|\det\frac{\partial G_\phi(\boldsymbol{\epsilon})}{\partial\boldsymbol{\epsilon}}\right|^{-1} \tag{B1}$$

and the variational density can be written as

$$q_X(\mathbf{x};\theta,\phi,\hat{\psi}) = q_Z(\mathbf{z};\theta,\phi)\left|\det\frac{\partial G_{\hat{\psi}}(\mathbf{z})}{\partial\mathbf{z}}\right|^{-1}. \tag{B2}$$

The corresponding ELBO becomes

$$\begin{aligned}\mathcal{L}(\omega) &= \mathbb{E}_{q_X(\mathbf{x};\omega,\hat{\psi})}[\log p(\mathbf{y}|\mathbf{x})]\\ &\quad - \mathbb{E}_{q_X(\mathbf{x};\omega,\hat{\psi})}\left[\log\frac{q_X(\mathbf{x};\omega,\hat{\psi})}{p_X(\mathbf{x};\hat{\psi})}\right],\end{aligned} \tag{B3}$$

where $\omega := \{\theta,\phi\}$ represents the trainable parameters.

From Eq. (B2) and the pretrained prior,

$$p_X(\mathbf{x};\hat{\psi}) = p_Z(\mathbf{z})\left|\det\frac{\partial G_{\hat{\psi}}(\mathbf{z})}{\partial\mathbf{z}}\right|^{-1},$$

the above ELBO [Eq. (B3)] can be simplified to

$$\begin{aligned}\mathcal{L}(\omega) &= \mathbb{E}_{q_Z(\mathbf{z};\omega)}[\log p(\mathbf{y}|G_{\hat{\psi}}(\mathbf{z}))] - \mathbb{E}_{q_Z(\mathbf{z};\omega)}\left[\log\frac{q_Z(\mathbf{z};\omega)}{p_Z(\mathbf{z})}\right]\\ &= \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon};\theta)}[\log p(\mathbf{y}|G_{\hat{\psi}}(G_\phi(\boldsymbol{\epsilon}))]\\ &\quad - \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon};\theta)}\left[\log\frac{q_Z(G_\phi(\boldsymbol{\epsilon});\omega)}{p_Z(G_\phi(\boldsymbol{\epsilon}))}\right].\end{aligned} \tag{B4}$$

In this study, we set $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)$; therefore, q was independent of the parameters, making Eq. (B4)

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}[\log p(\mathbf{y}|G_{\hat{\psi}}(G_\phi(\boldsymbol{\epsilon}))] \\ -\mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}\left[\log \frac{q_Z(G_\phi(\boldsymbol{\epsilon});\phi)}{p_Z(G_\phi(\boldsymbol{\epsilon}))}\right]. \tag{B5}$$

Based on the forward model [Eq. (3) in the main text], the first term in the right-hand side becomes

$$-\mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}\left[\frac{1}{2\sigma^2}\|\mathbf{y} - AG_{\hat{\psi}}(G_\phi(\boldsymbol{\epsilon}))\|_2^2 + C\right],$$

where C represents all the terms that are independent of the trainable parameter, ϕ. Given that $q_\epsilon(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, I)$ and Eq. (B1), the second term can be written as

$$\mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}[\log p_Z(G_\phi(\boldsymbol{\epsilon})) - \log q_Z(G_\phi(\boldsymbol{\epsilon});\phi)] \\ = \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}\left[-\frac{1}{2}\|G_\phi(\boldsymbol{\epsilon})\|_2^2 - \frac{n}{2}\log 2\pi\right] \\ - \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}[\log q_\epsilon(\boldsymbol{\epsilon})] + \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}\left[\log\left|\det\frac{\partial G_\phi(\boldsymbol{\epsilon})}{\partial\boldsymbol{\epsilon}}\right|\right].$$

The gradient of the ELBO becomes

$$\nabla_\phi\mathcal{L}(\phi) = \mathbb{E}_{q_\epsilon(\boldsymbol{\epsilon})}\left[\nabla_\phi\left\{-\frac{1}{2\sigma^2}\|\mathbf{y} - AG_{\hat{\psi}}(G_\phi(\boldsymbol{\epsilon}))\|_2^2 \right.\right. \\ \left.\left. -\frac{1}{2}\|G_\phi(\boldsymbol{\epsilon})\|_2^2 + \log\left|\det\frac{\partial G_\phi(\boldsymbol{\epsilon})}{\partial\boldsymbol{\epsilon}}\right|\right\}\right],$$

and its Monte Carlo approximation can be expressed as

$$\nabla_\phi\left[\frac{1}{L}\sum_{l=1}^{L}\left\{-\frac{1}{2}(A^TA + h^2I)^{-1}\|\mathbf{y} - AG_{\hat{\psi}}(G_\phi(\boldsymbol{\epsilon}^{(l)}))\|_2^2 \right.\right. \\ \left.\left. -\frac{1}{2}\|G_\phi(\boldsymbol{\epsilon}^{(l)})\|_2^2 + \log\left|\det\frac{\partial G_\phi(\boldsymbol{\epsilon}^{(l)})}{\partial\boldsymbol{\epsilon}}\right|\right\}\right],$$

where $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, I)$ and L is the number of samples. We adopted a preconditioning factor [40] to address inherently ill-posed problems. The regularization parameter, h, includes σ. Algorithm 2 in the main text was derived by drawing L samples with sequential substitutions as follows:

$$\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, I)\ \mathbf{z}^{(l)} = G_\phi(\boldsymbol{\epsilon}^{(l)})\ \mathbf{x}^{(l)} = G_{\hat{\psi}}(\mathbf{z}^{(l)}).$$

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova, Rev. Mod. Phys. **91**, 045002 (2019).

[2] D. Piccinotti, K. F. MacDonald, S. A Gregory, I. Youngs, and N. I. Zheludev, Rep. Prog. Phys. **84**, 012401 (2021).

[3] A. S. Fuhr and B. G. Sumpter, Front. Mater. Sci. **9**, 865270 (2022).

[4] I. H. Sarker, SN Computer Science **2**, 160 (2021).

[5] S. Park, C. Yun, J. Lee, and J. Shin, arXiv:2006.08859v1.

[6] V. N. Vapnik, *Statistical Learning Theory* (John Wiley & Sons, New York, 1998).

[7] E. Schachinger, D. Neuber, and J. P. Carbotte, Phys. Rev. B **73**, 184507 (2006).

[8] C. W. Groetsch, J. Phys.: Conf. Ser. **73**, 012001 (2007).

[9] A.-M. Wazwaz, Comput. Math. Appl. **61**, 2981 (2011).

[10] R. Fournier, L. Wang, O. V. Yazyev, and Q. S. Wu, Phys. Rev. Lett. **124**, 056401 (2020).

[11] H. Yoon, J. H. Sim, and M. J. Han, Phys. Rev. B **98**, 245101 (2018).

[12] H. Park, J. H. Park, and J. Hwang, Phys. Rev. B **104**, 235154 (2021).

[13] S. V. Shulga, O. V. Dolgov, and E. G. Maksimov, Phys. C: Supercond. **178**, 266 (1991).

[14] P. Putzky and M. Welling, arXiv:1706.04008v1.

[15] W. R. Morningstar, L. P. Levasseur, Y. D. Hezaveh, R. Blandford, P. Marshall, P. Putzky, T. D. Rueter, R. Wechsler, and M. Welling, Astrophys. J. **883**, 14 (2019).

[16] J. Hwang, J. H. Park, and H. Park, under review, arXiv:2404.02387v1.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Curran Associates, Inc., 2014), Vol. 27.

[18] D. P. Kingma and M. Welling, arXiv:1312.6114v1.

[19] D. J. Rezende, S. Mohamed, and D. Wierstra, arXiv:1401.4082v1.

[20] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, arXiv:1503.03585.

[21] J. Ho, A. Jain, and P. Abbeel, arXiv:2006.11239v1.

[22] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, arXiv:2005.14165.

[23] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, arXiv:1703.03208v1.

[24] V. Shah and C. Hegde, arXiv:1802.08406v1.

[25] Y. Saatchi and A. G. Wilson, arXiv:1705.09558v1.

[26] J. Whang, E. Lindgren, and A. Dimakis, arXiv:2002.11743v1.

[27] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, arXiv:1905.11672v1.

[28] J. Whang, Q. Lei, and A. Dimakis, arXiv:2003.08089v1.

[29] D. J. Rezende and S. Mohamed, arXiv:1505.05770v1.

[30] P. Kirichenko, P. Izmailov, and A. G. Wilson, arXiv:2006.08545v1.

[31] E. G. Tabak and E. Vanden-Eijnden, Commun. Math. Sci. **8**, 217 (2010).

[32] E. G. Tabak and C. V. Turner, Commun. Pure Appl. Math. **66**, 145 (2013).

[33] L. Dinh, D. Krueger, and Y. Bengio, arXiv:1410.8516v1.

[34] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv:1605.08803.

[35] D. P. Kingma and P. Dhariwal, arXiv:1807.03039v1.

[36] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, arXiv:1906.04032v1.

[37] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, arXiv:1912.02762v1.

[38] D. M. Blei, A. Kucukelbir, and J. D. Mcauliffe, J. Am. Stat. Assoc. **112**, 859 (2017).

[39] R. Ranganath, S. Gerrish, and D. Blei, arXiv:1401.0118v1.

[40] A. Neumaier, SIAM Rev. **40**, 636 (1998).

[41] D. P. Kingma and J. Ba, arXiv:1412.6980v1.

[42] J. Hwang, T. Timusk, E. Schachinger, and J. P. Carbotte, Phys. Rev. B **75**, 144508 (2007).

[43] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, arXiv:1206.7051v1.

[44] D. Wingate and T. Weber, arXiv:1301.1299v1.

[45] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, arXiv:1603.00788v1.