Inverse Estimation of Elastic Modulus Using Physics-Informed Generative Adversarial Networks

James E. Warner
Langley Research Center, Hampton, Virginia

Julian Cuevas
University of Puerto Rico (Mayaguez)

Geoffrey F. Bomarito
Langley Research Center, Hampton, Virginia

Patrick E. Leser
Langley Research Center, Hampton, Virginia

William P. Leser
Langley Research Center, Hampton, Virginia

NASA STI Program...in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:
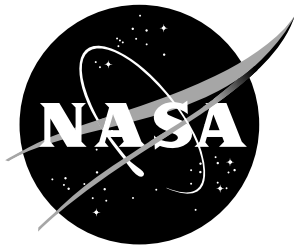
- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

Inverse Estimation of Elastic Modulus Using Physics-Informed Generative Adversarial Networks

James E. Warner
Langley Research Center, Hampton, Virginia

Julian Cuevas
University of Puerto Rico (Mayaguez)

Geoffrey F. Bomarito
Langley Research Center, Hampton, Virginia

Patrick E. Leser
Langley Research Center, Hampton, Virginia

William P. Leser
Langley Research Center, Hampton, Virginia

Abstract

While standard generative adversarial networks (GANs) rely solely on training data to learn unknown probability distributions, physics-informed GANs (PI-GANs) encode physical laws in the form of stochastic partial differential equations (PDEs) using auto differentiation. By relating observed data to unobserved quantities of interest through PDEs, PI-GANs allow for the estimation of underlying probability distributions without their direct measurement (i.e. inverse problems). The scalable nature of GANs allows high-dimensional, spatially-dependent probability distributions (i.e., random fields) to be inferred, while incorporating prior information through PDEs allows the training datasets to be relatively small.

In this work, PI-GANs are demonstrated for the application of elastic modulus estimation in mechanical testing. Given measured deformation data, the underlying probability distribution of spatially-varying elastic modulus (stiffness) is learned. Two feed-forward deep neural network generators are used to model the deformation and material stiffness across a two dimensional domain. Wasserstein GANs with gradient penalty are employed for enhanced stability. In the absence of explicit training data, it is demonstrated that the PI-GAN learns to generate realistic, physically-admissible realizations of material stiffness by incorporating the PDE that relates it to the measured deformation. It is shown that the statistics (mean, standard deviation, point-wise distributions, correlation length) of these generated stiffness samples have good agreement with the true distribution.

1 Introduction

The field of scientific machine learning has recently formed around the idea of infusion of current scientific knowledge into machine learning contexts [1]. This most commonly comes in the form of leveraging domain-specific knowledge in machine learning work flows [2]. For example, in physics-informed (PI) machine learning, known physical constraints are used as regularization in deep learning [3]. The physical constraints are encoded as partial differential equations (PDEs) using auto differentiation. Especially in the common case where data is limited but prior knowledge of the physical system is present, the regularization can result in more robust training and more accurate models [3–5]. Additionally, the resulting models can be easier to trust, based on their adherence to current scientific knowledge.

Generative adversarial networks (GANs) have seen tremendous success since their introduction in 2014, with the seminal work [6] having received more than 15,000 citations in the six years following its publication. Given the documented effectiveness of GANs for learning probability distributions from data [7–9], recent efforts have utlized GANs for uncertainty quantification in the context of scientific machine learning [10–12]. Particularly, the ability of GANs to scale to high stochastic dimensions makes them a promising alternative to common methods such as polynomial chaos expansion [13–16], stochastic collocation [17, 18], and stochastic reduced order models [19–21]. Early adoption of GANs was hindered by training instability, but recent contributions such as the Wasserstein GAN (WGAN) [22]

and the WGAN with gradient penalty (WGAN-GP) [23] have significantly improved convergence properties.

Earlier efforts to combine PI machine learning and GANs (PI-GANs) [10,12] have been partially motivated by their ability to perform inverse problems, i.e., inference of an unknown probability distribution based on its relation to another observed quantity. The formulations provided in these works generally define the relationship between the unknown and observed quantities as a vector-valued, stochastic PDE, and allow either or both quantities to be spatially or temporally varying. Importantly, however, the applications considered in these works either involved (i) a small, non-zero (e.g., as low as one) number of observations of the unknown quantity [10] or (ii) a direct functional dependence of the unknown quantity on the observed quantity (i.e., the unknown quantity is a function of the solution to the PDE) [12].

For many computational science and engineering inverse problems, the unknown probability distribution is specifically *unobservable* and may not be functionally dependent on the solution to the PDE. Solving this class of inverse problem was the focus of this work. Here, a PI-GAN was formulated which uses WGAN-GP and explicit penalization of a vector-valued PDE and arbitrarily many boundary conditions. Without direct observations of the quantity of interest, inference is strictly dependent on the PDE residual evaluated at collocation points in the problem domain. With this in mind, the proposed formulation follows previous work [3,12] and avoids hard limits (e.g., computer memory) on the number of collocation points by including a separate PDE loss term. The PI-GAN was applied to a solid mechanics problem in which an unobservable, spatially varying material property was inferred given observations of a two-dimensional material response. To the best of the authors' knowledge, this is the first example of PI-GANs in the field of solid mechanics.

The remainder of this paper is organized as follows. Section 2 formally defines the inverse problem and describes the PI-GAN formulation used herein. The problem of inverse material identification and its relevant physics are introduced in Section 3. The material identification problem is then performed in Section 4, wherein results on the accuracy and robustness of the method are discussed. Finally, Section 5 outlines contributions and conclusions.

2 PI-GANs for Inverse Problems

The PI-GAN framework integrates the deep learning concepts of GANs [6] and physics-informed neural networks [3] to learn probability distributions from data while adhering to relevant physical laws. This section provides a brief overview of the formulation in the context of solving PDE-constrained inverse problems.

2.1 Inverse Problem Definition

Consider the following general form of a time-independent stochastic PDE

$$\mathcal{N}_x[\mathbf{u}(\mathbf{x},\omega); E(\mathbf{x},\omega)] = \mathbf{f}(\mathbf{x}), \ \mathbf{x} \in \mathcal{D}, \ \omega \in \Omega \tag{1}$$

$$\mathcal{B}_x^{(k)}[\mathbf{u}(\mathbf{x},\omega); E(\mathbf{x},\omega)] = \mathbf{b}^{(k)}(\mathbf{x}), \ \mathbf{x} \in \Gamma^{(k)}, \tag{2}$$

where $\mathcal{N}_x$ is an arbitrary differential operator, $\mathcal{D} \subset \mathbb{R}^d$ is the physical domain, Ω is the probability space, and $\{\mathcal{B}_x^{(k)}\}_{k=1}^{N_k}$ are the N_k boundary condition operators applied to the respective portions of the boundary, $\Gamma^{(k)}$. With a proper specification of the coefficient $E(\mathbf{x}, \omega)$, forcing function $\mathbf{f}(\mathbf{x})$, and boundary conditions $\{\mathbf{b}^{(k)}(\mathbf{x})\}_{k=1}^{N_k}$, the *forward problem* can be solved to approximate the solution, $\mathbf{u}(\mathbf{x}, \omega)$. Here, $\mathbf{u} : \mathbb{R}^d \times \Omega \to \mathbb{R}^d$ is a vector-valued random field, i.e., a function of both the spatial coordinate, $\mathbf{x}$, and random event, ω. Note that it has been assumed that uncertainty in $\mathbf{u}$ is solely due to randomness in E (the forcing function and boundary condition are deterministic) for simplicity, but this need not be the case.

This work focuses on the corresponding *inverse problem* associated with Equations (1) and (2). That is, to determine $E(\mathbf{x}, \omega)$ using incomplete observations of the solution $\mathbf{u}(\mathbf{x}, \omega)$, again with known functions, $\mathbf{f}(\mathbf{x})$ and $\mathbf{b}^{(k)}(\mathbf{x})$[1]. The observations of $\mathbf{u}$ are assumed to have been collected from N_u independent measurements at $N_{\hat{\mathbf{x}}_u}$ sensor locations, $\{\hat{\mathbf{x}}_u^{(i)}\}_{i=1}^{N_{\hat{\mathbf{x}}_u}}$, resulting in the dataset,

$$\mathcal{U} = \{\hat{u}^{(i,j)}\}, \; i = 1, ..., N_{\hat{\mathbf{x}}_u}, \; j = 1, ..., N_u, \tag{3}$$

where $\hat{u}^{(i,j)} \equiv \hat{u}(\hat{\mathbf{x}}_u^{(i)}, \omega^{(j)})$. An ideal solution to this inverse problem is one that provides a full spatial and probabilistic description of E capable of producing realistic random field realizations that are physically admissible according to Equations (1) and (2) (as opposed to only low order statistics of E).

2.2 Generative Adversarial Networks (GANs)

Standard generative adversarial networks (GANs) solve the problem of learning a probability distribution given sample data. For example, using a training dataset like the one in Equation (3), GANs can learn to produce samples that closely mimic those from the true underlying $\mathbf{u}$ distribution, $\mathbb{P}_\mathbf{u} \in \mathbb{R}^d$. Training is done through a game between two competing networks: a generator network that learns to map random noise to realistic samples of $\mathbf{u}$ and a discriminator network that learns to distinguish between generated (fake) samples and true samples from $\mathcal{U}$.

Let the generator, $G_\theta(\cdot)$, and discriminator, $D_\phi(\cdot)$, be two feed-forward deep neural networks parameterized by θ and ϕ, respectively. Here, the generator accepts the random variable, $\mathbf{z} \in \mathbb{R}^m$ (with assumed distribution $\mathbb{P}_z$) as an input and outputs a sample $G_\theta(\mathbf{z}) \in \mathbb{R}^d$ (with learned distribution $\mathbb{P}_g$). The goal of the generator is to approximate $\mathbb{P}_\mathbf{u}$ with $\mathbb{P}_g$. The discriminator, on the other hand, takes a sample $\mathbf{v} \in \mathbb{R}^d$ and learns to classify it as being fake (from $\mathbb{P}_g$) or real (from $\mathbb{P}_\mathbf{u}$).

The GAN training procedure can be formally stated as:

$$\min_G \max_D \mathbb{E}_{\mathbf{v} \sim \mathbb{P}_\mathbf{u}}[\log(D_\phi(\mathbf{v}))] + \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z}[\log(1 - D_\phi(G_\theta(\mathbf{z})))], \tag{4}$$

where the loss functions for the generator and discriminator are

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_z}[\log(1 - D_\phi(G_\theta(\mathbf{z})))] \tag{5}$$

---

[1] When the solution to the inverse problem is the coefficient of a PDE, it is often referred to as parameter estimation.

and

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{v}\sim\mathbb{P}_{\mathbf{u}}}[\log(D_\phi(\mathbf{v})] - \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_z}\left[\log(1 - D_\phi(G_\theta(\mathbf{z})))\right], \tag{6}$$

respectively. It can be shown that if the discriminator is optimal, then minimizing Equation (5) is equivalent to minimizing the Jensen-Shannon (JS) divergence between $\mathbb{P}_{\mathbf{u}}$ and $\mathbb{P}_g$. However, training GANs through the solution of Equation (4) is a delicate process and can be plagued with vanishing gradients and instabilities.

Following the developments in [10], Wasserstein GANs (WGANs) with gradient penalty [23] are adopted in this work for improved training stability and approximation of $\mathbb{P}_{\mathbf{u}}$. The original GAN generator and discriminator loss functions are modified in this case as follows:

$$\mathcal{L}_G^w = -\mathbb{E}_{\mathbf{z}\sim\mathbb{P}_z}\left[\bar{D}_\phi(G_\theta(\mathbf{z}))\right] \tag{7}$$

$$\mathcal{L}_D^w = \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_z}\left[\bar{D}_\phi(G_\theta(\mathbf{z}))\right] - \mathbb{E}_{\mathbf{v}\sim\mathbb{P}_{\mathbf{u}}}[\bar{D}_\phi(\mathbf{v})] + \lambda\mathbb{E}_{\hat{\mathbf{v}}\sim\mathbb{P}_{\hat{\mathbf{v}}}}\left[(\|\nabla_{\hat{\mathbf{v}}}\bar{D}_\phi(\hat{\mathbf{v}})\|_2 - 1)^2\right], \tag{8}$$

where the WGAN discriminator (typically called the *critic* in this context), $\bar{D}_\phi$, is constrained to be a 1-Lipschitz function using the last term in Equation (8) to penalize discriminator gradients that deviate from unity. Here, λ is the gradient penalty coefficient and samples from $\mathbb{P}_{\hat{\mathbf{v}}}$ are generated by sampling uniformly along straight lines between pairs of points from $\mathbb{P}_{\mathbf{u}}$ and $\mathbb{P}_g$.

Training WGANs corresponds to minimizing the Wasserstein-1 (or *Earth Mover*) distance between $\mathbb{P}_{\mathbf{u}}$ and $\mathbb{P}_g$, rather than the JS divergence, and has been shown to make optimization of the generator easier [22]. Additionally, it was shown [10] that WGANs with gradient penalty are more suitable for learning random functions that are deterministic at discrete points (on boundaries).

2.3   Physics-Informed GANs (PI-GANs)

While the WGAN with gradient penalty formulation described in the previous section can be used to learn $\mathbf{u}(\mathbf{x}, \omega)$ from a measurement dataset $\mathcal{U}$, the primary focus of this work is the recovery of $E(\mathbf{x}, \omega)$ using the same data. In this section, an approach to the inverse problem described in Section 2.1 is formulated using physics-informed GANs (PI-GANs) [10] that combines measurement data from $\mathcal{U}$ and the prior knowledge of the governing stochastic PDEs in Equations (1) and (2).

With this approach, the random fields $\mathbf{u}(\mathbf{x}, \omega)$ and $E(\mathbf{x}, \omega)$ are modeled using two independent deep neural networks

$$\tilde{\mathbf{u}}_{\theta_u}(\mathbf{x}, \xi) : \mathbb{R}^{d+m} \to \mathbb{R}^d \tag{9}$$

$$\tilde{E}_{\theta_E}(\mathbf{x}, \xi) : \mathbb{R}^{d+m} \to \mathbb{R}^1 \tag{10}$$

with parameters θ_u and θ_E, respectively. Here, $\tilde{\mathbf{u}}_{\theta_u}$ and $\tilde{E}_{\theta_E}$ are generators that operate on a single spatial coordinate $\mathbf{x} \in \mathbb{R}^d$ and random noise input $\xi \in \mathbb{R}^m$ to produce a realization of the random field at that point. It is important to emphasize that the generator $\tilde{E}_{\theta_E}$ represents the solution to the inverse problem defined in Section 2.1.

In order to use measurement data on $\mathbf{u}$ to train the generator for E, the neural networks in Equations (9) and (10) must be related through the governing equations

(1) and (2) during training. Motivated by the approach of physics-informed neural networks [3], this is accomplished by applying the operators $\mathcal{N}_x$ and $\mathcal{B}_x^{(k)}$ to the networks $\tilde{\mathbf{u}}_{\theta_u}$ and $\tilde{E}_{\theta_E}$ to construct "induced" neural networks for the PDE residual:

$$\tilde{\mathbf{r}}_{\theta_{u,E}}(\mathbf{x}, \xi) = \mathcal{N}_x[\tilde{\mathbf{u}}_{\theta_u}(\mathbf{x}, \xi); \tilde{E}_{\theta_E}(\mathbf{x}, \xi)] - \mathbf{f}(\mathbf{x}) \tag{11}$$

and boundary condition discrepancy functions

$$\tilde{\mathbf{b}}_{\theta_{u,E}}^{(k)}(\mathbf{x}, \xi) = \mathcal{B}_x^{(k)}[\tilde{\mathbf{u}}_{\theta_u}(\mathbf{x}, \xi); \tilde{E}_{\theta_E}(\mathbf{x}, \xi)] - \mathbf{b}^{(k)}(\mathbf{x}) \tag{12}$$

using auto differentiation [24]. Here, the shorthand notation $\theta_{u,E} \equiv [\theta_u, \theta_E]$ has been used to indicate the explicit dependence of the induced networks $\tilde{\mathbf{r}}$ and $\tilde{\mathbf{b}}^{(k)}$ on the parameters of both of the original generators, θ_u and θ_E.

Now, the generator loss function can be augmented to encourage jointly generated samples from $\tilde{\mathbf{u}}$ and $\tilde{E}_{\theta_E}$ to satisfy the PDE and boundary conditions during training. For instance, the contribution to the loss for the PDE residual is given by

$$\mathcal{L}_{PDE}(\theta_{u,E}) = \frac{1}{N_r N_{\hat{\mathbf{x}}_r}} \sum_{j=1}^{N_r} \sum_{i=1}^{N_{\hat{\mathbf{x}}_r}} \|\hat{\mathbf{r}}_{\theta_{u,E}}^{(i,j)}\|^2, \tag{13}$$

where $\hat{\mathbf{r}}_{\theta_{u,E}}^{(i,j)} = \tilde{\mathbf{r}}_{\theta_{u,E}}(\hat{\mathbf{x}}_r^{(i)}, \xi^{(j)})$ and $\{\mathbf{x}_r^{(i)}\}_{i=1}^{N_{\hat{\mathbf{x}}_r}} \in \mathcal{D}$ are collocation points where the PDE is enforced. Similarly, the boundary condition loss is given by

$$\mathcal{L}_{BC}(\theta_{u,E}) = \sum_{k=1}^{N_k} \frac{1}{N_b N_{\hat{\mathbf{x}}_{bk}}} \sum_{j=1}^{N_b} \sum_{i=1}^{N_{\hat{\mathbf{x}}_{bk}}} \|\hat{\mathbf{b}}_{\theta_{u,E}}^{(i,j,k)}\|^2, \tag{14}$$

where $\hat{\mathbf{b}}_{\theta_{u,E}}^{(i,j,k)} = \tilde{\mathbf{b}}_{\theta_{u,E}}^{(k)}(\hat{\mathbf{x}}_{bk}^{(i)}, \xi^{(j)})$ and $\{\mathbf{x}_{bk}^{(i)}\}_{i=1}^{N_{\hat{\mathbf{x}}_{bk}}} \in \Gamma^{(k)}$ are the boundary condition collocation points for boundary condition k. Here, N_r and N_b are the number of randomly generated samples used to evaluate the PDE and boundary condition loss, respectively.

The complete loss functions for PI-GANs are then as follows

$$\mathcal{L}_G^{PI}(\theta_{u,E}) = \mathcal{L}_G^w(\theta_u) + \mathcal{L}_{PDE}(\theta_{u,E}) + \mathcal{L}_{BC}(\theta_{u,E}) \tag{15}$$

$$\mathcal{L}_D^{PI}(\theta_{u,E}, \phi) = \mathcal{L}_D^w(\theta_{u,E}, \phi). \tag{16}$$

Note that the explicit dependence of the PI-GAN generator and discriminator loss functions on the parameters θ_E is what makes the solution of the inverse problem possible. In this way, $\tilde{E}_{\theta_E}$ is encouraged to learn from $\tilde{\mathbf{u}}_{\theta_u}$ through the governing equations, while $\tilde{\mathbf{u}}_{\theta_u}$ learns from the measurement data. The terms $\mathcal{L}_{PDE}$ and $\mathcal{L}_{BC}$ also act as a regularization mechanism for learning $\mathbf{u}$ by substantially restricting the space of admissible functions to only those that satisfy the governing equations.

3   Application

The problem of material identification is common to the disciplines of solid mechanics and mechanical testing. Specifically, the focus is on the estimation of spatially varying

material properties – in this case, the elastic modulus $E(\mathbf{x}, \omega)$ – from measured deformations $\mathbf{u}(\mathbf{x}, \omega)$ under load. The underlying probability distribution of material properties is assumed to be inherent to the material, and once identified, can aid in the quantification of uncertainty in mechanical behavior of all structures made of that material.

In this application, the PDE represented by Equation (1) is:

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, E) = \mathbf{0}, \ \mathbf{x} \in \mathcal{D}, \ \omega \in \Omega, \tag{17}$$

where $\boldsymbol{\sigma}$ is the second order tensor for stress. Assuming a 2-D plane-stress formulation, isotropic elasticity, and small strain theory [25], the stress can be written as a function of E and $\mathbf{u}$ as follows:

$$\boldsymbol{\sigma} = \frac{2E}{1+\nu} \left[\nabla \mathbf{u} + \nabla \mathbf{u}^T + \frac{2\nu}{1-\nu} \operatorname{Tr}(\nabla \mathbf{u})\mathbf{I} \right], \tag{18}$$

where ν is the Poisson ratio, $\operatorname{Tr}(\cdot)$ is the trace function, and $\mathbf{I}$ is the identity tensor. In all cases in this work, ν is assumed to be constant ($\nu = 0.3$), though the more general case of $\nu = \nu(\mathbf{x}, \omega)$ could be considered as well.

The boundary conditions are divided into two types: Dirichlet boundary conditions with known deformation $\mathbf{u}_D$ and Neuman boundary conditions with applied tractions $\boldsymbol{\tau}$,

$$\mathbf{u} = \mathbf{u}_D, \ \mathbf{x} \in \Gamma_D \tag{19}$$

and

$$\boldsymbol{\sigma}(\mathbf{u}, E) \cdot \mathbf{n} = \boldsymbol{\tau}, \ \mathbf{x} \in \Gamma_N, \tag{20}$$

where $\mathbf{n}$ is the outward normal for the surface.

Note that the explicit dependence of $\mathbf{u}(\mathbf{x}, \omega)$ and $E(\mathbf{x}, \omega)$ on $\mathbf{x}$ and ω in the above equations has been dropped to simplify notation. See the Appendix for the explicit, expanded forms of the PDE and boundary conditions that are enforced through auto differentiation for PI-GAN training.

4   Numerical Example

The PI-GAN approach for inverse problems is now demonstrated for the identification of a spatially and randomly varying elastic modulus, motivated by the application described in the previous section. The framework was implemented in Tensorflow v2.0 and executed on one NVIDIA Tesla V100 GPU. All data and code associated with this work will be made available at https://github.com/NASA/pigans-material-ID.

4.1   Physical Problem Description

The problem domain can be seen in Figure 1(a), assumed to be a unit square ($d = 2$) in this example ($L = w = 1.0$). Dimensionless quantities are assumed throughout. Dirichlet boundary conditions (Equation (19)) are applied to the left boundary as follows

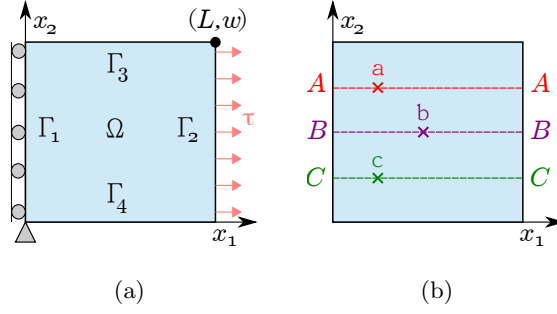$$u_1 = 0, \ \mathbf{x} \in \Gamma_1 \tag{21}$$

Figure 1. Domain for the numerical example along with (a) the boundary conditions and (b) the points and lines of interest for evaluating the accuracy of the PI-GAN.

and

$$u_2 = 0, \ \mathbf{x} = (0,0), \tag{22}$$

where u_1 and u_2 are the components of deformation in the x_1 and x_2 directions, respectively (i.e., $\mathbf{u} = [u_1, u_2]$). Neumann boundary conditions (Equation (20)) are applied to the remaining boundaries, where $\boldsymbol{\tau} = [0,0]$ on Γ_3 and Γ_4 and $\boldsymbol{\tau} = [1.5, 0]$ on Γ_4. See the Appendix for explicit forms of the implemented boundary conditions.

The true elastic modulus is modeled as a lognormal-distributed random field

$$E(\mathbf{x}, \omega) = \alpha + \beta \exp(g(\mathbf{x}, \omega)) \tag{23}$$

$$g(\mathbf{x}, \omega) \sim \mathcal{GP}\left(0, \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right)\right) \tag{24}$$

i.e., a zero-mean Gaussian process transformed using an exponential function and scaled using $\alpha = 1.0$ and $\beta = 0.1$. The correlation length of the Gaussian process is chosen to be $l = 1.0$. Samples of elastic modulus are generated by computing a truncated Karhunen-Loeve (KL) expansion [26] for $g(\mathbf{x})$ and then applying the transformation in Equation (23). In this example, five terms are used for the KL expansion which retains 99.9% of the variance of $g(\mathbf{x})$.

The training dataset (Equation (3)) is generated by solving Equation (17) with the boundary conditions listed above using the finite element method [27] as implemented in `FEniCS` Python library [28]. Here, a displacement solution, $\mathbf{u}$, is computed for N_u randomly sampled E fields and then interpolated to $N_{\hat{\mathbf{x}}_u}$ sensor locations. In all results shown, $N_{\hat{\mathbf{x}}_u} = 90$, and the sensor locations are chosen by creating an equidistant 10×10 grid throughout the domain and then removing the sensors along $x_1 = 0$. The boundary condition loss in Equation (14) is used to enforce the displacement boundary conditions (Equations (21) and (21)) on boundary Γ_1 instead.

For illustration, Figure 2 shows three representative random samples of E generated using Equations (23), (24), and the KL expansion along with the corresponding solutions for u_1 and u_2 using `FEniCS`.
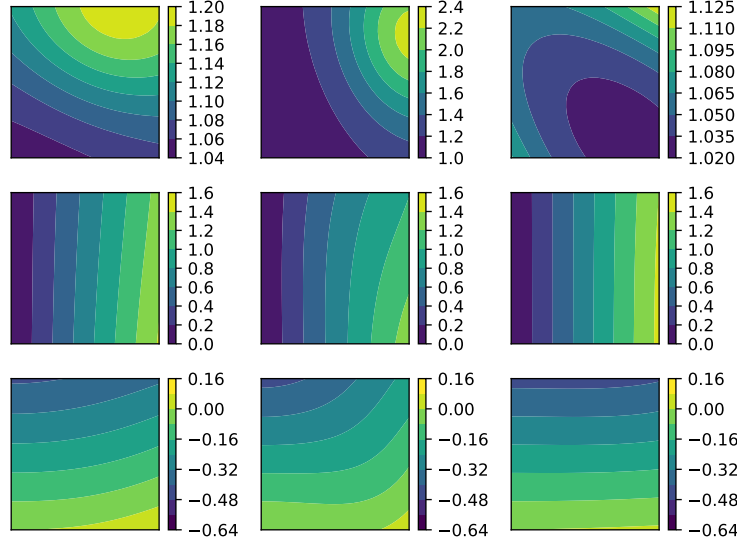
Figure 2. Three randomly drawn samples of elastic modulus E (row 1) and corresponding computed deformations, u_1 (row 2) and u_2 (row 3), from the test dataset.

4.2 Neural Network Specification & Hyperparameters

For all results to be shown, the generators and discriminators are implemented using feed-forward deep neural networks with four hidden layers of width 128. The hyperbolic tangent (*tanh*) activation function is used for all networks, which is shown to be effective in PI-GAN approaches where higher order derivatives are necessary [10]. The random noise input $\xi \in \mathbb{R}^m$ to the generators is distributed according to a standard multivariate Gaussian distribution with $m = 5$, so that the total input dimension to the generator is seven.

The Adam optimizer [29] is used for the minimization of the PI-GAN loss functions in Equations (15) and (16) with hyper-parameters $\beta_1 = 0$, $\beta_2 = 0.9$, and learning rate of 10^{-4}. Five generator training steps are performed for every one discriminator step during training. A gradient penalty coefficient of $\lambda = 0.1$ is used and 10^5 steps are taken overall. The batch size is equal to the total number of measurements in each case.

4.3 PI-GAN Material Identification Accuracy

First, PI-GANs were trained using $N_u = 1000$ measurements and $N_{\hat{\mathbf{x}}_r} = 100$ collocation points defined on an equidistant 10×10 grid throughout the domain. One hundred generated samples were used to enforce both the PDE and boundary condition loss ($N_r = N_b = 100$) in Equations (13) and (14), while ten equidistant points were used on each boundary ($N_{\hat{\mathbf{x}}_{bk}} = 10$). Accuracy in the recovery of elastic modulus was assessed by generating 1000 samples from $\tilde{E}_{\theta_E}$ after training was complete and comparing with 10000 samples from an independent test set. The average time per training step in this example was 2.5×10^{-1} seconds, resulting in a
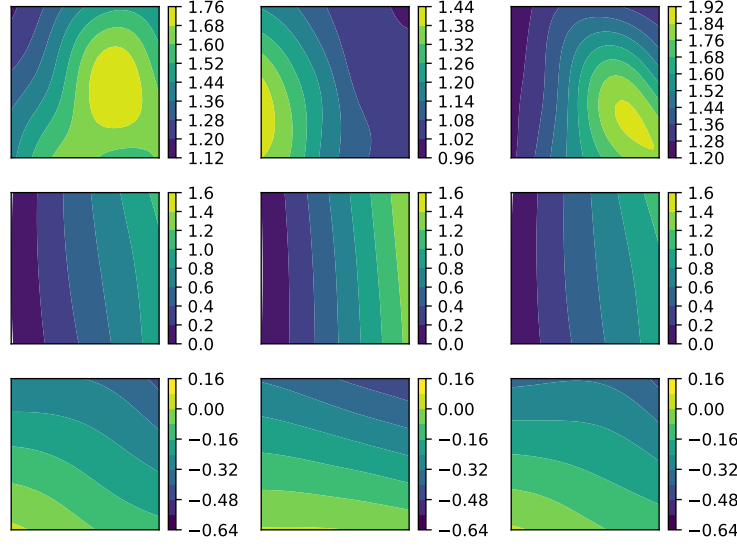
Figure 3. Three randomly generated samples of elastic modulus E (row 1) and corresponding deformations, u_1 (row 2) and u_2 (row 3), from the trained generators in Section 4.3.

total training time of about 7 hours.

Three representative samples of E, u_1, and u_2 from the trained generators, $\tilde{E}_{\theta_E}$ and $\tilde{\mathbf{u}}_{\theta_u}$, are shown in Figure 3. Qualitatively speaking, the generated samples have similar overall appearance and characteristics (e.g., spatial variation and magnitude) compared to the reference samples from the test set shown in Figure 2. Note that since the sets of samples in Figures 2 and 3 are drawn randomly and independently of one another, exact agreement is not to be expected. However, there will be more similarity between random realizations of u_1 and u_2 compared to E due to the imposed boundary conditions.

To assess the accuracy more quantitatively, the pointwise error in the estimated mean and standard deviation of E with respect to the samples from the test set can be seen in Figure 4, . The errors for both are relatively low (E has a magnitude that is $O(1)$). Better accuracy is observed for estimating the mean compared to the standard deviation, as expected.

Since one primary motivation behind adopting a PI-GAN approach is to be able to capture higher order statistics beyond just mean and standard deviation, the ability of $\tilde{E}_{\theta_E}$ to recover pointwise distributions and pairwise correlations was also assessed. These comparisons were done for the points and lines of interest shown in Figure 1(b). First, the estimated probability density function of E using generated samples at the three reference points is shown in Figure 5(a). Good agreement is observed with the reference distribution computed from the samples in the test dataset. Note that the distribution of E according to Equation (23) is homogenous in space, so the distribution at all points should coincide[2].

---

[2]The reference distribution in Figure 5(a) was computed at $\mathbf{x} = [0.5, 0.5]$.
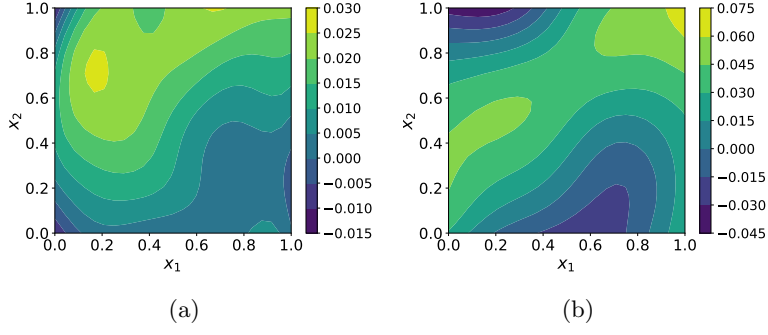
Figure 4. Point-wise error (absolute difference) between the reference and generated (a) mean and (b) standard deviation for elastic modulus.
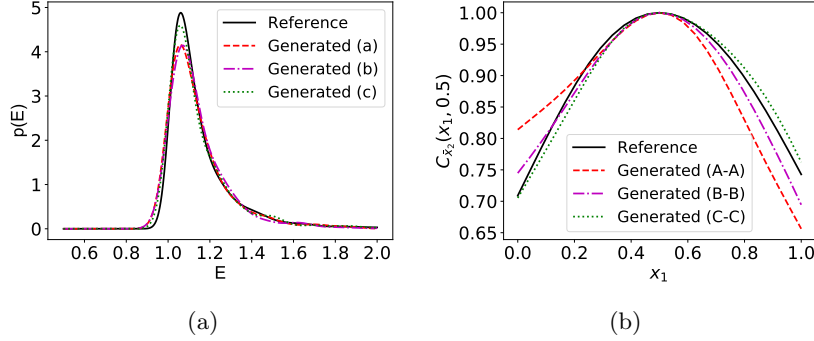

Figure 5. (a) Comparison of the generated probability density function for E at the points of interest in Figure 1(b) and (b) Comparison of the 1D correlation function (Equation (25)) across the lines of interest in Figure 1(b).

Next, the accuracy of the estimated spatial correlation is shown in Figure 5(b). Here, the 1D correlation at fixed x_2 coordinates,

$$C_{\bar{x}_2}(x_1, x_1') = \text{Correlation}\left[E(x_1, \bar{x}_2), E(x_1', \bar{x}_2)\right], \tag{25}$$

is computed and displayed as a function of x_1 and compared to the reference correlation from test samples. Note that $x_1' = 0.5$ and $\bar{x}_2 = 0.75, 0.5, 0.25$ for sections A-A, B-B, and C-C in Figure 1(b), respectively. The generated correlations decay from $C_{\bar{x}_2} = 1.0$ at $x_1 = 0.5$ towards the boundaries in good agreement with the reference solution, displaying the ability of the PI-GAN to accurately infer the spatial variation of the elastic modulus random field.

4.4 Effect of Number of Measurements and Collocation Points

The accuracy of the PI-GAN approach for varying numbers of measurements (N_u) and collocation points (N_r) is now illustrated. Note that all other problem specifications remain unchanged from the previous section. Three random trials were performed for each case and the average relative $L2$ error in the estimated mean and standard
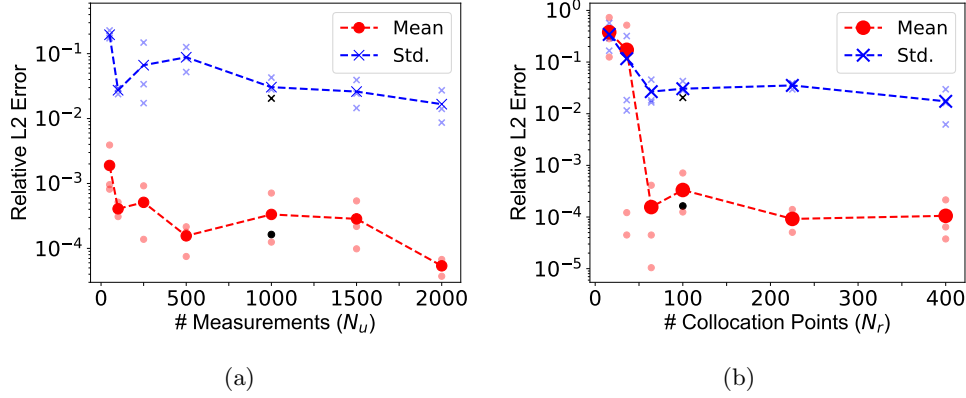
Figure 6. Relative L2 error in the estimated mean and standard deviation of E versus (a) number of measurements and (b) number of collocation points. The curves show the average of errors from three random trials, represented individually by the smaller markers. The black symbols at $N_u = 1000$ and $N_r = 100$ correspond to errors from the trial used to produce the results in Section 4.3.

deviation of E was computed. Here, the $L2$ norms were calculated using numerical integration on a uniform 25×25 grid.

First, using $N_r = 100$, the errors for varying numbers of measurements is shown in Figure 6(a). The errors generally decrease with increasing N_u but it can be seen that relatively accurate estimates are possible with $O(100)$ measurements. Note that the black symbols at $N_u = 1000$ on the plot correspond to errors associated with the results shown in the previous section for illustration purposes.

Next, for a fixed number of measurements, $N_u = 1000$, the errors are shown versus number of collocations points in Figure 6(b). Similar to the previous section, the collocation points are specified on equidistant grids across the domain: 4×4, 6×6, 8×8, 10×10, 15×15, and 20×20. As expected, the accuracy of the estimated elastic modulus field generally increases with the number of collocation points used. Here, it is important to reiterate that the primary source of information the PI-GAN has to infer E is through the PDE loss in Equation (13), and hence a sufficient number of collocation points must be used ($\gtrsim 50$ in this example), after which diminishing returns are observed.

Finally, it is noted that relatively significant random variation is observed in the accuracy among independent trials, as evidenced by the spread in the small markers for individual cases in Figures 6(a) and 6(b). The fact that the relative $L2$ errors are not strictly decreasing as a function of N_u and N_r is likely a result of this variability in training. A more thorough study on the impact of the various PI-GAN parameters (discriminator/generator architectures, number of displacement sensors, Adam optimizer hyper-parameters, WGAN gradient penalty coefficient, etc.) on the accuracy and stability of training will be pursued in future work.

11

5   Conclusion

Physics-informed generative adversarial networks (PI-GANs) are an emerging tool for scientific machine learning that enable the solution of complex, stochastic partial differential equations. In this work, a PI-GAN is formulated for the purpose of solving inverse problems where an unknown probability distribution is estimated based on its relation to another, observed quantity. Specifically, this approach enables solutions to the class of inverse problem in which the unknown quantity is strictly unobservable and is not a function of the PDE solution. As a result, the only information used to train the feed-forward neural network representing the unknown probability distribution is indirectly obtained through the governing PDE and boundary conditions. To address this, separate PDE and boundary condition loss terms are included in the classic generator-discriminator GAN formulation. Additionally, recent improvements regarding training stability are incorporated by using the Wasserstein GAN with gradient penalty (WGAN-GP). The formulation allows for vector-valued PDEs with unknown and spatially varying quantities of interest and for arbitrary boundary conditions.

The PI-GAN was demonstrated on a solid mechanics example in which the unknown, spatially varying stiffness distribution was estimated given two-dimensional observations of material deformation in response to an applied load. In the context of this numerical example, it was shown that statistics (mean, standard deviation, point-wise distributions and correlation length) of the generated stiffness samples were in good agreement with those of the true distribution. Additionally, the effect on accuracy of varying the number of observations and collocation points (i.e., the number of points at which the PDE constraint was enforced in the problem domain during training) was studied. It is found that $O(100)$ measurements were enough to produce accurate estimates given 100 collocation points. Fixing the number of measurements at 1000, accurate estimates were achieved with $O(10)$ collocation points ($\gtrsim 50$ in the specific example presented herein). To the best of the authors' knowledge, this is both the first application of a PI-GAN in the field of solid mechanics and to this specific class of inverse problem (i.e., no direct observations of the quantity of interest and no dependence of that quantity on the PDE solution).

6   Appendix

PDE Residual Expressions

A more explicit form of the PDE residual implemented in this work can be obtained by substituting Equation (18) into Equation (17) and expressing the result in index notation, i.e.,

$$\frac{(1-\nu)}{2}\left[E_{,2}(u_{1,2}+u_{2,1})+E(u_{1,22}+u_{2,12})\right]+$$
$$E_{,1}\left(u_{1,1}+\nu u_{2,2}\right)+E\left(u_{1,11}+\nu u_{2,21}\right)=0 \quad (26)$$

and

$$\frac{(1-\nu)}{2}\left[E_{,1}(u_{1,2}+u_{2,1})+E(u_{1,21}+u_{2,11})\right]+$$
$$E_{,2}\left(\nu u_{1,1}+u_{2,2}\right)+E\left(\nu u_{1,12}+u_{2,22}\right)=0,\quad(27)$$

where the notation $f_{i,j}\equiv\frac{\partial f_i}{\partial x_j}$ has been adopted.

Boundary Condition Expressions

A more explicit form of the implemented Neumann boundary conditions from the numerical example can be derived by using $\mathbf{n}=[1,0]$ on Γ_2, $\mathbf{n}=[0,1]$ on Γ_3, and $\mathbf{n}=[0,-1]$ on Γ_4 with Equation (20) and expressing the result in index notation, i.e.,

$$\frac{E}{1-\nu^2}\left(u_{1,1}+\nu u_{2,2}\right)=1.5\ \ \mathbf{x}\in\Gamma_2,\tag{28}$$

$$\frac{E}{2(1+\nu)}\left(u_{1,2}+u_{2,1}\right)=0\ \ \mathbf{x}\in\Gamma_2,\tag{29}$$

$$\frac{E}{1-\nu^2}\left(\nu u_{1,1}+u_{2,2}\right)=0\ \ \mathbf{x}\in\Gamma_3,\Gamma_4,\ \text{and}\tag{30}$$

$$\frac{E}{2(1+\nu)}\left(u_{1,2}+u_{2,1}\right)=0\ \ \mathbf{x}\in\Gamma_3,\Gamma_4.\tag{31}$$

For more implementation details, the interested reader can find the code and data needed to reproduce the results of this report at https://github.com/NASA/pigans-material-ID.

Acknowledgement

References

1. Baker, N.; et al.: Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. , USDOE Office of Science (SC), Washington, DC (United States), 2019.

2. Roscher, R.; Bohn, B.; Duarte, M. F.; and Garcke, J.: Explainable machine learning for scientific insights and discoveries. *arXiv preprint arXiv:1905.08883*, 2019.

3. Raissi, M.; Perdikaris, P.; and Karniadakis, G.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, vol. 378, 2019, pp. 686–707.

4. Tartakovsky, A. M.; Marrero, C. O.; Perdikaris, P.; Tartakovsky, G. D.; and Barajas-Solano, D.: Learning parameters and constitutive relationships with physics informed deep neural networks. *arXiv preprint arXiv:1808.03398*, 2018.

5. Ling, J.; Kurzawski, A.; and Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, vol. 807, 2016, pp. 155–166.

6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems*, 2014, pp. 2672–2680.

7. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; and Shi, W.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2016.

8. Zhang, Y.; Gan, Z.; Fan, K.; Chen, Z.; Henao, R.; Shen, D.; and Carin, L.: Adversarial Feature Matching for Text Generation. 2017.

9. Yang, L.-C.; Chou, S.-Y.; and Yang, Y.-H.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. 2017.

10. Yang, L.; Zhang, D.; and Karniadakis, G. E.: Physics-Informed Generative Adversarial Networks for Stochastic Differential Equations. 2018.

11. Yang, L.; et al.: Highly-Scalable, Physics-Informed GANs for Learning Solutions of Stochastic PDEs. *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, IEEE, 2019, pp. 1–11.

12. Yang, Y.; and Perdikaris, P.: Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, vol. 394, 2019, pp. 136–152.

13. Ghanem, R.; and Spanos, P. D.: Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, vol. 57, no. 1, 1990, pp. 197–202.

14. Xiu, D.; and Karniadakis, G. E.: The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, vol. 24, no. 2, 2002, pp. 619–644.

15. Xiu, D.: *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.

16. Oladyshkin, S.; and Nowak, W.: Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, vol. 106, 2012, pp. 179–190.

17. Babuška, I.; Nobile, F.; and Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, vol. 45, no. 3, 2007, pp. 1005–1034.

18. Nobile, F.; Tempone, R.; and Webster, C. G.: A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, 2008, pp. 2309–2345.

19. Grigoriu, M. D.: Reduced order models for random functions. Application to stochastic problems. *Applied Mathematical Modelling*, vol. 33, no. 1, 2009, pp. 161–175.

20. Warner, J. E.; Grigoriu, M. D.; and Aquino, W.: Stochastic reduced order models for random vectors: Application to random eigenvalue problems. *Probabilistic Engineering Mechanics*, vol. 31, 2013, pp. 1–11.

21. Warner, J. E.; Aquino, W.; and Grigoriu, M. D.: Stochastic reduced order models for inverse problems under uncertainty. *Computer Methods in Applied Mechanics and Engineering*, vol. 285, 2015, pp. 488–514.

22. Arjovsky, M.; Chintala, S.; and Bottou, L.: Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

23. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C.: Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

24. Baydin, A. G.; Pearlmutter, B. A.; and Radul, A. A.: Automatic differentiation in machine learning: a survey. *CoRR*, vol. abs/1502.05767, 2015. URL `http://arxiv.org/abs/1502.05767`.

25. Reddy, J. N.: *An Introduction to Continuum Mechanics*. Cambridge University Press, 2 ed., 2013.

26. Kirby, M.; and Sirovich, L.: Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, Jan. 1990, p. 103?108. URL `https://doi.org/10.1109/34.41390`.

27. Hughes, T.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Civil and Mechanical Engineering, Dover Publications, 2000. URL `https://books.google.com/books?id=yarmSc7ULRsC`.

28. Alnæs, M. S.; Blechta, J.; Hake, J.; Johansson, A.; Kehlet, B.; Logg, A.; Richardson, C.; Ring, J.; Rognes, M. E.; and Wells, G. N.: The FEniCS Project Version 1.5. *Archive of Numerical Software*, vol. 3, no. 100, 2015.

29. Kingma, D. P.; and Ba, J.: Adam: A Method for Stochastic Optimization. 2014. URL `http://arxiv.org/abs/1412.6980`, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-04-2020	Technical Memorandum	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Inverse Estimation of Elastic Modulus Using Physics-Informed Generative Adversarial Networks	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
James E. Warner and Julian Cuevas and Geoffrey F. Bomarito and Patrick E. Leser and William P. Leser	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
NASA Langley Research Center Hampton, Virginia 23681-2199	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
National Aeronautics and Space Administration Washington, DC 20546-0001	NASA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/ TM–2020–5001300

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited
Subject Category 61
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

While standard generative adversarial networks (GANs) rely solely on training data to learn unknown probability distributions, physics-informed GANs (PI-GANs) encode physical laws in the form of stochastic partial differential equations (PDEs) using auto differentiation. By relating observed data to unobserved quantities of interest through PDEs, PI-GANs allow for the estimation of underlying probability distributions without their direct measurement (i.e. inverse problems). The scalable nature of GANs allows high-dimensional, spatially-dependent probability distributions (i.e., random fields) to be inferred, while incorporating prior information through PDEs allows the training datasets to be relatively small.

In this work, PI-GANs are demonstrated for the application of elastic modulus estimation in mechanical testing. Given measured deformation data, the underlying probability distribution of spatially-varying elastic modulus (stiffness) is learned. Two feed-forward deep neural network generators are used to model the deformation and material stiffness across a two dimensional domain. Wasserstein GANs with gradient penalty are employed for enhanced stability. In the absence of explicit training data, it is demonstrated that the PI-GAN learns to generate realistic, physically-admissible realizations of material stiffness by incorporating the PDE that relates it to the measured deformation. It is shown that the statistics (mean, standard deviation, point-wise distributions, correlation length) of these generated stiffness samples have good agreement with the true distribution.

15. SUBJECT TERMS

Uncertainty Quantification, Machine Learning, Inverse Problems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Information Desk (help@sti.nasa.gov)
U	U	U	UU	20	19b. TELEPHONE NUMBER *(Include area code)* (757) 864-9658