Proposed Papers for AMPD UP: Applied Math Presentations & Discourse for Understanding Papers

Brandon Burkhardt, Kimball Johnston, Jimmie Adriazola

School of Mathematical and Statistical Sciences, Arizona State University

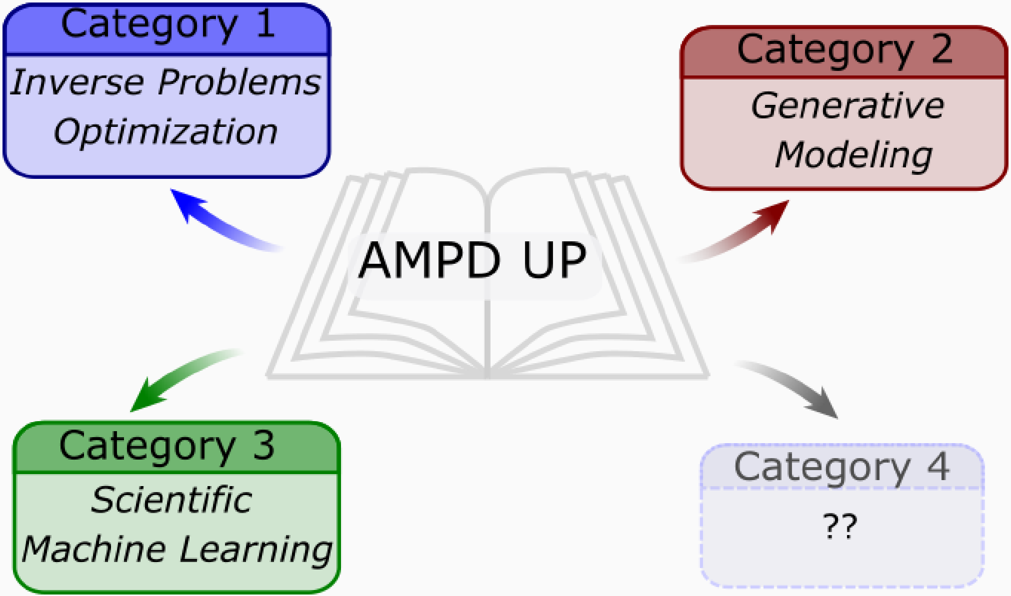
August 27, 2025

What is AMPD UP?

- We are a reading group that meets Wednesdays at 9AM (go to coffee break first and bring your bagels!)
- Intention is to have a 40-50 minute board talk explaining the big ideas of a paper
- Bonus points if you do a computer demo at the end (think open source code that the authors released on github)
- Anyone here that's willing (independent of career stage) can take a presentation
- We intend to maintain a website with several resources related to the group.
- This is meant to be casual!

Why a Journal Club?

- Improve reading and comprehension
- Improve presentation and communication
- Find your area of interest
- Broaden your research vision
- Practice technical writing
- Have fun challenging your peers while they struggle presenting a tough paper :)


Category 1
Inverse Problems
Optimization
Category 2
Generative
Modeling
AMPD UP
Category 3
Scientific
Machine Learning
Category 4
??

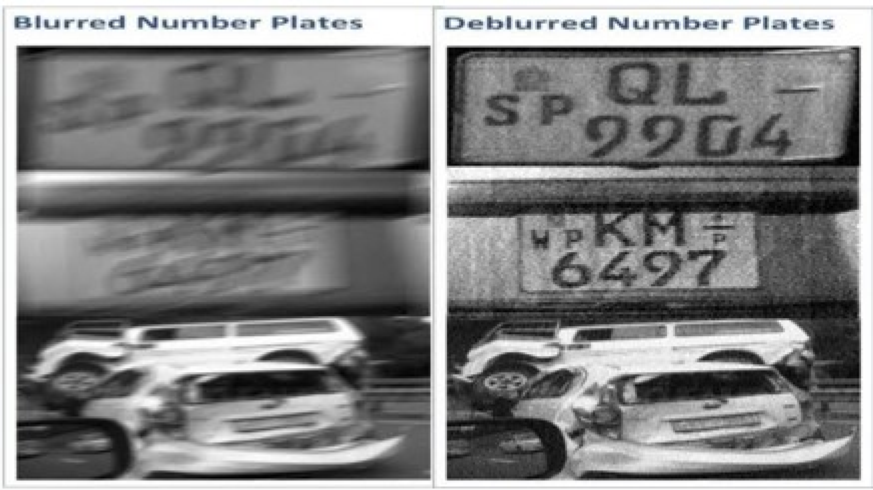
Paper Examples and Suggestions

Category 1:

Inverse Problems, Linear Algebra, and Uncertainty Quantification

The Problem

Fugitives! (Or just blurry images)



Solution 1: Variable Projection Method

Paper: Variable projection methods for separable nonlinear inverse problems with general-form Tikhonov regularization (Espanol, Pasha, 2023)

- Utilizes the variable projection (VarPro) method (c. 1975) to solve the general-form Tikhonov regularization problem
- Applies iterative solution techniques

IOP Publishing — Inverse Problems
Inverse Problems **39** (2023) 084002 (25pp) https://doi.org/10.1088/1361-6420/acdd1b

Variable projection methods for separable nonlinear inverse problems with general-form Tikhonov regularization

Malena I Español[1,*] and Mirjeta Pasha[2]

[1] School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, United States of America
[2] Department of Mathematics, Tufts University, Medford, MA, United States of America

E-mail: malena.espanol@asu.edu

Received 1 October 2022; revised 21 May 2023
Accepted for publication 9 June 2023
Published 28 June 2023

CrossMark

Abstract
The variable projection (VarPro) method is an efficient method to solve separable nonlinear least squares problems. In this paper, we propose a modified VarPro method for solving separable nonlinear least squares problems with general-form Tikhonov regularization. In particular, we apply the Gauss–Newton method to the corresponding reduced problem and investigate its convergence when different approximations of the Jacobian matrix are used. For special cases when computing the generalized singular value decomposition is feasible or a joint spectral decomposition of both forward and regularization operators exists, we provide efficient ways to compute the Jacobians and the solution of the linear subproblems. For large-scale problems, where matrix decompositions are not an option, we compute a reduced Jacobian and apply projection-based iterative methods and generalized Krylov subspace methods to solve the linear subproblems. In all cases, the regularization parameter can be computed automatically at each iteration using generalized cross validation. Several numerical examples highlight the proposed approach's performance in the quality of the reconstructed image and the reconstructed forward operator, including large-scale two-dimensional imaging problems arising from semi-blind deblurring.

Keywords: variable projection method, separable nonlinear, Tikhonov regularization, semi-blind deconvolution

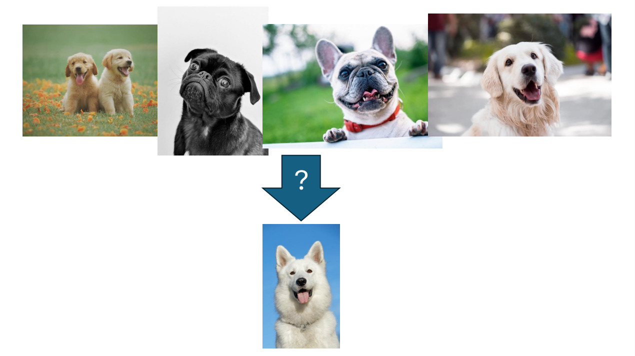
(Some figures may appear in colour only in the online journal)

[*] Author to whom any correspondence should be addressed.

Category 2:
Generative Modeling

The Problem

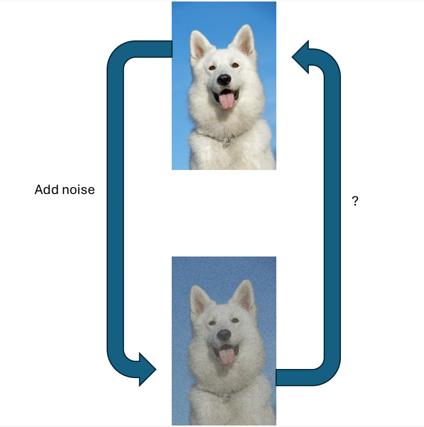
- Say you have some collection of data points
  - E.g., dog images
- How can you generate a new data point that "looks like" the original data?
  - E.g., a new dog image



Solution 1: DDPM

Paper: Denoising Diffusion Probabilistic Models (Ho, Jain, Abbeel, 2020)

- One can easily add small increments of noise to one's data points
- Idea: try to train a neural network to reverse this process and "denoise" it
- If one starts with a random noise sample and then denoises it with this neural net, what will happen?
    - Hopefully, it will produce a new data point that "looks like" the training data



Solution 2: Score-based generative modeling

Paper: Score-Based Generative Modeling through Stochastic Differential Equations (Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole, 2021)

- Like DDPM, based on adding noise to the data while trying to learn to “denoise” it
- Instead of modeling the problem as many discrete steps (like DDPM), models it as a continuous-time stochastic differential equation (SDE)
- At the heart of this SDE is an object called the “score” that one tries to learn


Published as a conference paper at ICLR 2021

SCORE-B SED GENER TIVE MODELING THROUGH STOCH STIC DIFFERENTI L EQU TIONS

Yang Song[*]
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

bhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

BSTR CT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in score-based generative modeling and diffusion probabilistic modeling, allowing for new sampling procedures and new modeling capabilities. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, but additionally enables exact likelihood computation, and improved sampling efficiency. In addition, we provide a new way to solve inverse problems with score-based models, as demonstrated with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIF R-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 2.99 bits/dim, and demonstrate high fidelity generation of 1024×1024 images for the first time from a score-based generative model.

1 INTRODUCTION

Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density with respect to data) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption, using knowledge of the functional form of the reverse distributions to make training tractable. For continuous state spaces, the DDPM training objective implicitly computes scores at each noise scale. We therefore refer to these two model classes together as *score-based generative models*.

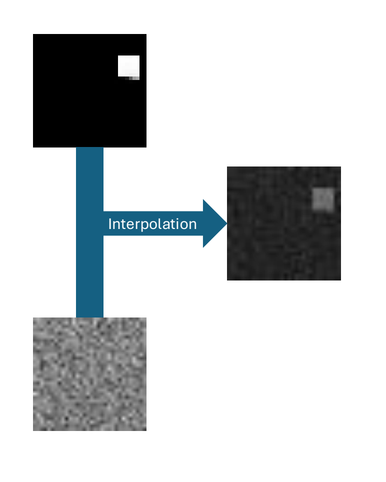
Score-based generative models, and related techniques (Bordes et al., 2017; Goyal et al., 2017; Du & Mordatch, 2019), have proven effective at generation of images (Song & Ermon, 2019; 2020; Ho et al., 2020), audio (Chen et al., 2020; Kong et al., 2020), graphs (Niu et al., 2020), and shapes (Cai

[*]Work partially done during an internship at Google Brain.

Solution 3: Stochastic interpolants

Paper: Building Normalizing Flows with Stochastic Interpolants (Albergo, Vanden-Eijnden, 2023)

- Like the score-based approach but based on a simple ordinary differential equation instead of an SDE
- Unlike DDPM and the score-based approach, no diffusion
- Called “stochastic interpolants” because it “interpolates” between a data point and a noise sample (instead of repeatedly adding random noise to the data point)



Solution 4: Solution 2 plus solution 3

Paper: Stochastic Interpolants: A Unifying Framework for Flows and Diffusions (Albergo, Boffi, Vanden-Eijnden, 2023)

- Introduces a framework that lets you mix-and-match score-based methods with stochastic-interpolant-based methods
- Brings diffusion into the stochastic interpolant approach

arXiv:2303.08797v3 [cs.LG] 6 Nov 2023

Stochastic Interpolants:
A Unifying Framework for Flows and Diffusions

Michael S. Albergo*[1], Nicholas M. Boffi*[2], and Eric Vanden-Eijnden[2]

[1]Center for Cosmology and Particle Physics, New York University
[2]Courant Institute of Mathematical Sciences, New York University

November 7, 2023

Abstract

A class of generative models that unifies flow-based and diffusion-based methods is introduced. These models extend the framework proposed in [2], enabling the use of a broad class of continuous-time stochastic processes called 'stochastic interpolants' to bridge any two arbitrary probability density functions exactly in finite time. These interpolants are built by combining data from the two prescribed densities with an additional latent variable that shapes the bridge in a flexible way. The time-dependent probability density function of the stochastic interpolant is shown to satisfy a first-order transport equation as well as a family of forward and backward Fokker-Planck equations with tunable diffusion coefficient. Upon consideration of the time evolution of an individual sample, this viewpoint immediately leads to both deterministic and stochastic generative models based on probability flow equations or stochastic differential equations with an adjustable level of noise. The drift coefficients entering these models are time-dependent velocity fields characterized as the unique minimizers of simple quadratic objective functions, one of which is a new objective for the score of the interpolant density. We show that minimization of these quadratic objectives leads to control of the likelihood for generative models built upon stochastic dynamics, while likelihood control for deterministic dynamics is more stringent. We also construct estimators for the likelihood and the cross-entropy of interpolant-based generative models, and we discuss connections with other methods such as score-based diffusion models, stochastic localization processes, probabilistic denoising techniques, and rectifying flows. In addition, we demonstrate that stochastic interpolants recover the Schrödinger bridge between the two target densities when explicitly optimizing over the interpolant. Finally, algorithmic aspects are discussed and the approach is illustrated on numerical examples.

*Author ordering alphabetical; authors contributed equally.

1

Solution 5: Mean field games

Paper: A mean-field games laboratory for generative modeling (Zhang, Katsoulakis, 2023)

- Also gives us a framework to unify solution 2 and solution 3, but now extends itself a class of methods called “Wasserstein gradient flows” as well
- Formulates the problem as a so-called mean-field game (MFG)
- MFGs help give another PDE perspective for generative modeling to understand well-posedness of strategies

A Mean-Field Games Laboratory for Generative Modeling

Benjamin J. Zhang BJZHANG@UMASS.EDU
Department of Mathematics and Statistics
University of Massachusetts Amherst
Amherst, MA 01003-9305, USA

Markos A. Katsoulakis MARKOS@UMASS.EDU
Department of Mathematics and Statistics
University of Massachusetts Amherst
Amherst, MA 01003-9305, USA

Abstract

We demonstrate the versatility of mean-field games (MFGs) as a mathematical framework for explaining, enhancing, and designing generative models. In generative flows, a Lagrangian formulation is used where each particle (generated sample) aims to minimize a loss function over its simulated path. The loss, however, is dependent on the paths of other particles, which leads to a competition among the population of particles. The asymptotic behavior of this competition yields a mean-field game. We establish connections between MFGs and major classes of generative flows and diffusions including continuous-time normalizing flows, score-based generative models (SGM), and Wasserstein gradient flows. Furthermore, we study the mathematical properties of each generative model by studying their associated MFG's optimality condition, which is a set of coupled forward-backward nonlinear partial differential equations. The mathematical structure described by the MFG optimality conditions identifies the inductive biases of generative flows. We investigate the well-posedness and structure of normalizing flows, unravel the mathematical structure of SGMs, and derive a MFG formulation of Wasserstein gradient flows. From an algorithmic perspective, the optimality conditions yields Hamilton-Jacobi-Bellman (HJB) regularizers for enhanced training of generative models. In particular, we propose and demonstrate an HJB-regularized SGM with improved performance over standard SGMs. We present this framework as an MFG laboratory which serves as a platform for revealing new avenues of experimentation and invention of generative models.

Keywords: Generative modeling, mean-field games, Hamilton-Jacobi-Bellman equation, normalizing flows, score-based generative models, Wasserstein gradient flow, inductive bias

Contents

1 Introduction **3**
1.1 A preview of results 6

2 Background on mean-field games **8**
2.1 Optimality conditions 9

Category 3:

Scientific Machine Learning for PDEs

Physics-Informed Neural Networks (PINNs): Big Idea

- Learn a *function* $u_\theta(x,t)$ (neural net with parameters θ) that fits data *and* satisfies the governing PDE.
- Use the PDE itself as a training signal: penalize the residual of the differential operator evaluated on u_θ via automatic differentiation (AD).
- Train on mixed batches:
  - **Data points:** observed (x_i, t_i, y_i)
  - **Physics points:** collocation (x_j, t_j) for enforcing the PDE
  - **BC/IC points:** boundary/initial constraints
- Outcome: a single "surrogate" model u_θ that is consistent with measurements *and* the model physics.

PINNs: Core Loss Idea

- Network: $u_\theta(x,t)$ with AD-computed derivatives.
- **PDE residual at collocation points:** evaluate the governing operator on the network output, e.g.

$$\mathcal{R}_\theta(x,t) = \partial_t u_\theta(x,t) - \nu\,\partial_{xx} u_\theta(x,t) - f(x,t),$$

  which should vanish if u_θ satisfies the PDE.
- Loss blends three parts:
  - **Data fit:** match observed values
  - **Physics:** penalize $\|\mathcal{R}_\theta\|^2$
  - **BC/IC:** enforce boundary and initial conditions
- Training minimizes

$$\mathcal{L} = \lambda_d\,(\text{data}) + \lambda_p\,(\text{physics}) + \lambda_b\,(\text{BC/IC})$$

- Big picture: PINNs combine measurements + PDE constraints in a single loss.

GFINNs: GENERIC Formalism Informed Neural Networks

Paper: GFINNs: GENERIC Formalism Informed Neural Networks for Deterministic and Stochastic Dynamical Systems (Zhang, Shin, Karniadakis, 2022)

- Embeds the GENERIC (General Equation for Non-Equilibrium Reversible–Irreversible Coupling) structure into neural nets.
- Splits dynamics: reversible (Hamiltonian) vs. irreversible (dissipative); thermodynamically consistent.
- Works for deterministic & stochastic systems; strong results on benchmark problems.

PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

CrossMark
click for updates

Article submitted to journal

Subject Areas:
deep learning, applied mathematics, thermodynamics

Keywords:
data-driven discovery, physics-informed neural networks, GENERIC formalism, interpretable scientific machine learning

Author for correspondence:
Yeonjong Shin
e-mail: yeonjong_shin@brown.edu

GFINNs: GENERIC Formalism Informed Neural Networks for Deterministic and Stochastic Dynamical Systems

Zhen Zhang[1], Yeonjong Shin[1] and George Em Karniadakis[1,2]

[1] Division of Applied Mathematics, and [2] School of Engineering, Brown University, Providence, RI, 02912, USA

We propose the GENERIC formalism informed neural networks (GFINNs) that obey the symmetric degeneracy conditions of the GENERIC formalism. GFINNs comprise two modules, each of which contains two components. We model each component using a neural network whose architecture is designed to satisfy the required conditions. The component-wise architecture design provides flexible ways of leveraging available physics information into neural networks. We prove theoretically that GFINNs are sufficiently expressive to learn the underlying equations, hence establishing the universal approximation theorem. We demonstrate the performance of GFINNs in three simulation problems: gas containers exchanging heat and volume, thermoelastic double pendulum and the Langevin dynamics. In all the examples, GFINNs outperform existing methods, hence demonstrating good accuracy in predictions for both deterministic and stochastic systems.

Operator Learning

- Learns *maps between function spaces*, i.e. operators

$$\mathcal{G} : f(x) \mapsto u(x)$$

  rather than pointwise functions.
- Examples: DeepONets, Fourier Neural Operators (FNOs).
- Advantages over function learning (e.g. PINNs):
  - Generalizes across different initial/boundary conditions and parameters.
  - Learns solution *families*, not just one instance.
- Outlook: operator learning is emerging as a new paradigm for PDE surrogates, superseding function learning by treating the PDE as a mapping rather than a single trajectory.

Fourier Neural Operator (FNO)

Paper: Fourier Neural Operator for Parametric Partial Differential Equations (Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, Stuart, Anandkumar, 2020/2021)

- Introduces Fourier Neural Operator (FNO), a framework for operator learning.
- Learns mappings between function spaces by parameterizing kernels in Fourier space.
- Provides mesh-independent generalization and efficient evaluation.
- Demonstrated on PDE families such as Darcy flow and Navier–Stokes.

Published as a conference paper at ICLR 2021

FOURIER NEURAL OPERATOR FOR PARAMETRIC PARTIAL DIFFERENTIAL EQUATIONS

Zongyi Li zongyili@caltech.edu
Nikola Kovachki nkovachki@caltech.edu
Kamyar Azizzadenesheli kamyar@purdue.edu
Burigede Liu bgl@caltech.edu
Kaushik Bhattacharya bhatta@caltech.edu
Andrew Stuart astuart@caltech.edu
Anima Anandkumar anima@caltech.edu

ABSTRACT

The classical development of neural networks has primarily focused on learning mappings between finite-dimensional Euclidean spaces. Recently, this has been generalized to neural operators that learn mappings between function spaces. For partial differential equations (PDEs), neural operators directly learn the mapping from any functional parametric dependence to the solution. Thus, they learn an entire family of PDEs, in contrast to classical methods which solve one instance of the equation. In this work, we formulate a new neural operator by parameterizing the integral kernel directly in Fourier space, allowing for an expressive and efficient architecture. We perform experiments on Burgers' equation, Darcy flow, and Navier-Stokes equation. The Fourier neural operator is the first ML-based method to successfully model turbulent flows with zero-shot super-resolution. It is up to three orders of magnitude faster compared to traditional PDE solvers. Additionally, it achieves superior accuracy compared to previous learning-based solvers under fixed resolution.

1 INTRODUCTION

Many problems in science and engineering involve solving complex partial differential equation (PDE) systems repeatedly for different values of some parameters. Examples arise in molecular dynamics, micro-mechanics, and turbulent flows. Often such systems require fine discretization in order to capture the phenomenon being modeled. As a consequence, traditional numerical solvers are slow and sometimes inefficient. For example, when designing materials such as airfoils, one needs to solve the associated inverse problem where thousands of evaluations of the forward model are needed. A fast method can make such problems feasible.

Conventional solvers vs. Data-driven methods. Traditional solvers such as finite element methods (FEM) and finite difference methods (FDM) solve the equation by discretizing the space. Therefore, they impose a trade-off on the resolution: coarse grids are fast but less accurate; fine grids are accurate but slow. Complex PDE systems, as described above, usually require a very fine discretization, and therefore very challenging and time-consuming for traditional solvers. On the other hand, data-driven methods can directly learn the trajectory of the family of equations from the data. As a result, the learning-based method can be orders of magnitude faster than the conventional solvers.

Machine learning methods may hold the key to revolutionizing scientific disciplines by providing fast solvers that approximate or enhance traditional ones (Raissi et al., 2019; Jiang et al., 2020; Greenfeld et al., 2019; Kochkov et al., 2021). However, classical neural networks map between finite-dimensional spaces and can therefore only learn solutions tied to a specific discretization. This is often a limitation for practical applications and therefore the development of mesh-invariant neural networks is required. We first outline two mainstream neural network-based approaches for PDEs – the finite-dimensional operators and Neural-FEM.

Finite-dimensional operators. These approaches parameterize the solution operator as a deep convolutional neural network between finite-dimensional Euclidean spaces Guo et al. (2016); Zhu

Solving High-Dimensional PDEs Using Deep Learning

Paper: Solving High-Dimensional Partial Differential Equations Using Deep Learning (Han, Jentzen, E, 2017)

- Reformulates parabolic PDEs via backward SDEs (BSDEs); learns the solution's gradient with a NN (Deep BSDE method).
- Trains by simulating forward SDE paths and enforcing terminal/initial conditions in expectation.
- Demonstrates scalability to very high dimensions (e.g., HJB, Black–Scholes).
- Monte Carlo–friendly; avoids spatial meshes and explicit discretization of the PDE operator.

Solving High-Dimensional Partial Differential Equations Using Deep Learning

Jiequn Han[1], Arnulf Jentzen[2], and Weinan E[*4,3,1]

[1]Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA
[2]Department of Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland
[3]Department of Mathematics, Princeton University, Princeton, NJ 08544, USA
[4]Beijing Institute of Big Data Research, Beijing, 100871, China

Abstract

Developing algorithms for solving high-dimensional partial differential equations (PDEs) has been an exceedingly difficult task for a long time, due to the notoriously difficult problem known as the "curse of dimensionality". This paper introduces a deep learning-based approach that can handle general high-dimensional parabolic PDEs. To this end, the PDEs are reformulated using backward stochastic differential equations and the gradient of the unknown solution is approximated by neural networks, very much in the spirit of deep reinforcement learning with the gradient acting as the policy function. Numerical results on examples including the nonlinear Black-Scholes equation, the Hamilton-Jacobi-Bellman equation, and the Allen-Cahn equation suggest that the proposed algorithm is quite effective in high dimensions, in terms of both accuracy and cost. This opens up new possibilities in economics, finance, operational research, and physics, by considering all participating agents, assets, resources, or particles together at the same time, instead of making ad hoc assumptions on their inter-relationships.

1 Introduction

Partial differential equations (PDEs) are among the most ubiquitous tools used in modeling problems in nature. Some of the most important ones are naturally formulated as PDEs in high dimensions. Well-known examples include:

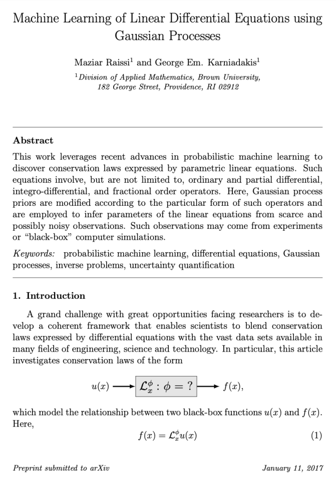
1. The Schrödinger equation in quantum many-body problem. In this case the dimensionality of the PDE is roughly three times the number of electrons or quantum particles in the system.
2. The nonlinear Black-Scholes equation for pricing financial derivatives, in which the dimensionality of the PDE is the number of underlying financial assets under consideration.

*weinan@math.princeton.edu

Machine Learning of Linear DEs using Gaussian Processes

Paper: Machine Learning of Linear Differential Equations using Gaussian Processes (Raissi and Karniadakis, 2017)

- Leverages Gaussian process priors tailored to linear differential operators.
- Learns parameters of ODEs, PDEs, integro-differential, and fractional operators directly from noisy data.
- Provides a probabilistic framework with uncertainty quantification.
- Bridges Gaussian process regression with inverse modeling for scientific discovery.


Machine Learning of Linear Differential Equations using Gaussian Processes

Maziar Raissi[1] and George Em. Karniadakis[1]

[1]*Division of Applied Mathematics, Brown University, 182 George Street, Providence, RI 02912*

Abstract

This work leverages recent advances in probabilistic machine learning to discover conservation laws expressed by parametric linear equations. Such equations involve, but are not limited to, ordinary and partial differential, integro-differential, and fractional order operators. Here, Gaussian process priors are modified according to the particular form of such operators and are employed to infer parameters of the linear equations from scarce and possibly noisy observations. Such observations may come from experiments or "black-box" computer simulations.

Keywords: probabilistic machine learning, differential equations, Gaussian processes, inverse problems, uncertainty quantification

1. Introduction

A grand challenge with great opportunities facing researchers is to develop a coherent framework that enables scientists to blend conservation laws expressed by differential equations with the vast data sets available in many fields of engineering, science and technology. In particular, this article investigates conservation laws of the form

$$u(x) \longrightarrow \boxed{\mathcal{L}_x^{\phi} : \phi = ?} \longrightarrow f(x),$$

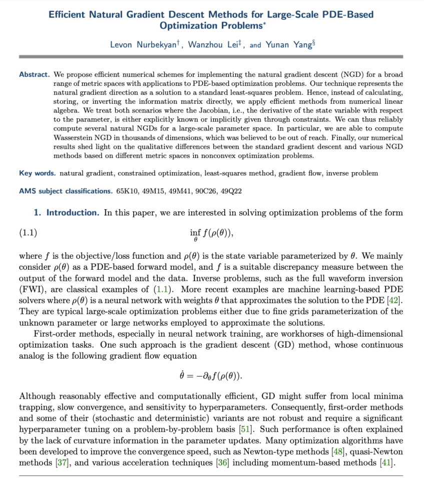
which model the relationship between two black-box functions $u(x)$ and $f(x)$. Here,

$$f(x) = \mathcal{L}_x^{\phi} u(x) \quad (1)$$

Efficient Natural Gradient Descent for PDE Optimization

Paper: Efficient Natural Gradient Descent Methods for Large-Scale PDE-Based Optimization Problems (Nurbekyan, Lei, Yang, 2023)

- Develops scalable algorithms for natural gradient descent in PDE-constrained optimization.
- Reformulates natural gradient computation as a least-squares problem.
- Avoids explicit formation/inversion of Fisher information matrices.
- Enables applications such as Wasserstein natural gradient descent in high dimensions.


Efficient Natural Gradient Descent Methods for Large-Scale PDE-Based Optimization Problems*

Levon Nurbekyan†, Wanzhou Lei‡, and Yunan Yang§

Abstract. We propose efficient numerical schemes for implementing the natural gradient descent (NGD) for a broad range of metric spaces with applications to PDE-based optimization problems. Our technique represents the natural gradient direction as a solution to a standard least-squares problem. Hence, instead of calculating, storing, or inverting the information matrix directly, we apply efficient methods from numerical linear algebra. We treat both scenarios where the Jacobian, i.e., the derivative of the state variable with respect to the parameter, is either explicitly known or implicitly given through constraints. We can thus reliably compute several natural NGDs for a large-scale parameter space. In particular, we are able to compute Wasserstein NGD in thousands of dimensions, which was believed to be out of reach. Finally, our numerical results shed light on the qualitative differences between the standard gradient descent and various NGD methods based on different metric spaces in nonconvex optimization problems.

Key words. natural gradient, constrained optimization, least-squares method, gradient flow, inverse problem

AMS subject classifications. 65K10, 49M15, 49M41, 90C26, 49Q22

1. Introduction. In this paper, we are interested in solving optimization problems of the form

$$\inf_\theta f(\rho(\theta)), \tag{1.1}$$

where f is the objective/loss function and $\rho(\theta)$ is the state variable parameterized by θ. We mainly consider $\rho(\theta)$ as a PDE-based forward model, and f is a suitable discrepancy measure between the output of the forward model and the data. Inverse problems, such as the full waveform inversion (FWI), are classical examples of (1.1). More recent examples are machine learning-based PDE solvers where $\rho(\theta)$ is a neural network with weights θ that approximates the solution to the PDE [42]. They are typical large-scale optimization problems either due to fine grids parameterization of the unknown parameter or large networks employed to approximate the solutions.

First-order methods, especially in neural network training, are workhorses of high-dimensional optimization tasks. One such approach is the gradient descent (GD) method, whose continuous analog is the following gradient flow equation

$$\dot\theta = -\partial_\theta f(\rho(\theta)).$$

Although reasonably effective and computationally efficient, GD might suffer from local minima trapping, slow convergence, and sensitivity to hyperparameters. Consequently, first-order methods and some of their (stochastic and deterministic) variants are not robust and require a significant hyperparameter tuning on a problem-by-problem basis [51]. Such performance is often explained by the lack of curvature information in the parameter updates. Many optimization algorithms have been developed to improve the convergence speed, such as Newton-type methods [48], quasi-Newton methods [37], and various acceleration techniques [36] including momentum-based methods [41].

Category 4:

You Pick!