DEQGAN: Learning the Loss Function for PINNs with Generative Adversarial Networks

Blake Bullwinkel, Dylan Randle, Pavlos Protopapas, David Sondak
Institute for Applied Computational Science (IACS), Harvard University

Abstract

Solutions to differential equations are of significant scientific and engineering relevance. Physics-Informed Neural Networks (PINNs) have emerged as a promising method for solving differential equations, but they lack a theoretical justification for the use of any particular loss function. This work presents Differential Equation GAN (DEQGAN), a novel method for solving differential equations using generative adversarial networks to "learn the loss function" for optimizing the neural network. We show that DEQGAN can obtain multiple orders of magnitude lower mean squared errors than PINNs that use L_2, L_1, and Huber loss functions and that DEQGAN achieves solution accuracies that are competitive with popular numerical methods. Finally, we present two methods to improve the robustness of DEQGAN to different hyperparameter settings.

Why Neural Networks?

In contrast to traditional numerical methods, PINNs:

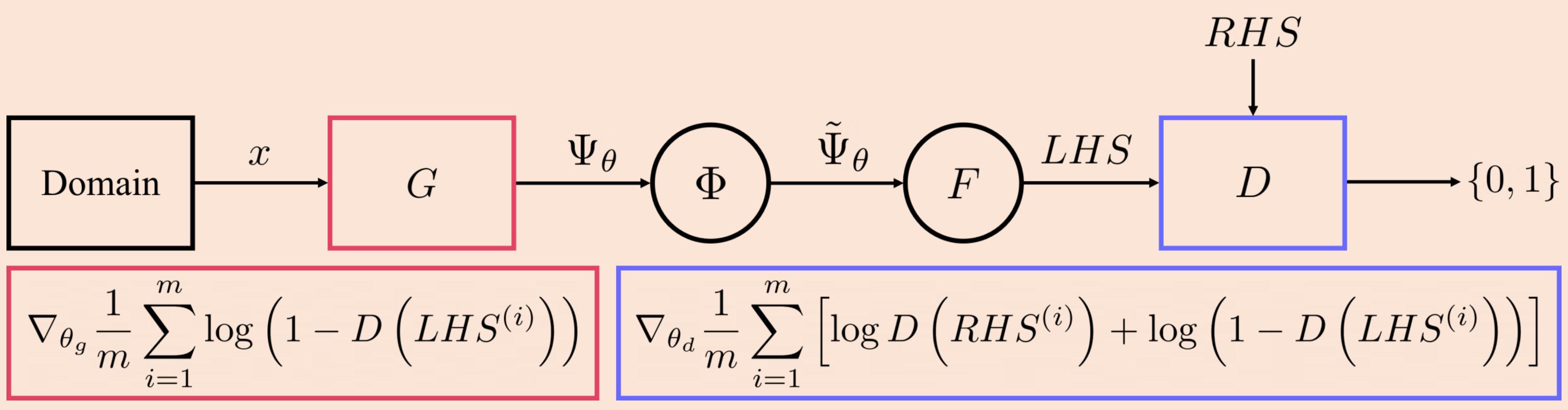
- Provide solutions that are closed-form
- Suffer less from the "curse of dimensionality"
- Provide a more accurate interpolation scheme
- Can leverage transfer learning for fast discovery of new solutions

Further, PINNs do not require an underlying grid and offer a meshless approach to solving differential equations. This makes it possible to use trained neural networks, which typically have small memory footprints, to generate solutions over arbitrary grids in a single forward pass.

Classical PINNs solve a differential equation $F = 0$ by minimizing some loss over the equation residuals, e.g.,

$$\min_{\theta} \sum_{(t,\mathbf{x})\in\mathcal{D}} F\left(t, \mathbf{x}, \Psi_\theta(t,x), \frac{d\Psi_\theta}{dt}, \frac{d^2\Psi_\theta}{dt^2}, \ldots, \Delta\Psi_\theta, \Delta^2\Psi_\theta, \ldots\right)^2$$

where Ψ_θ is a neural network pameterized by θ, $\mathcal{D}$ is the domain of the problem, and the derivatives are computed with automatic differentiation. PINNs have been successfully applied to a wide range of differential equations but provide no theoretical justification for the use of a particular loss function. To address this gap in the theory, we propose GANs for solving differential equations in a fully unsupervised manner.



$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(LHS^{(i)}\right)\right)$$

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(RHS^{(i)}\right) + \log\left(1 - D\left(LHS^{(i)}\right)\right)\right]$$

Figure 1. Schematic representation of DEQGAN. We pass input points x to a generator G, which produces candidate solutions Ψ_θ. We analytically adjust these solutions according to Φ and apply automatic differentiation to construct LHS from the differential equation F. RHS and LHS are passed to a discriminator D, which is trained to classify them as "real" and "fake," respectively.

DEQGAN

To solve a differential equation, we move all terms to the left-hand side (LHS) so that the right-hand side (RHS) is zero. During training, we sample points from the domain $(t, \mathbf{x}) \sim \mathcal{D}$ and use them as input to a generator network $G(x)$, which produces candidate solutions Ψ_θ. We then adjust Ψ_θ for initial and boundary conditions to obtain $\widehat{\Psi}_\theta$ and construct the LHS from the differential equation F

$$LHS = F\left(t, \mathbf{x}, \widehat{\Psi}_\theta(t,x), \frac{d\widehat{\Psi}_\theta}{dt}, \frac{d^2\widehat{\Psi}_\theta}{dt^2}, \ldots, \Delta\widehat{\Psi}_\theta, \Delta^2\widehat{\Psi}_\theta, \ldots\right)$$

We update the weights of the generator G and the discriminator D according to the gradients

$$g_G = \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(LHS^{(i)}\right)\right),$$

$$g_D = \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(RHS^{(i)}\right) + \log\left(1 - D\left(LHS^{(i)}\right)\right)\right]$$

Where $LHS^{(i)}$ is the output of $G\left(x^{(i)}\right)$ after adjusting for initial or boundary conditions. Informally, our algorithm trains a GAN by setting the "fake" component to be the LHS and the "real" to be the RHS. This results in a GAN that learns to produce solutions that make LHS indistinguishable from RHS, thereby approximately solving the equation.

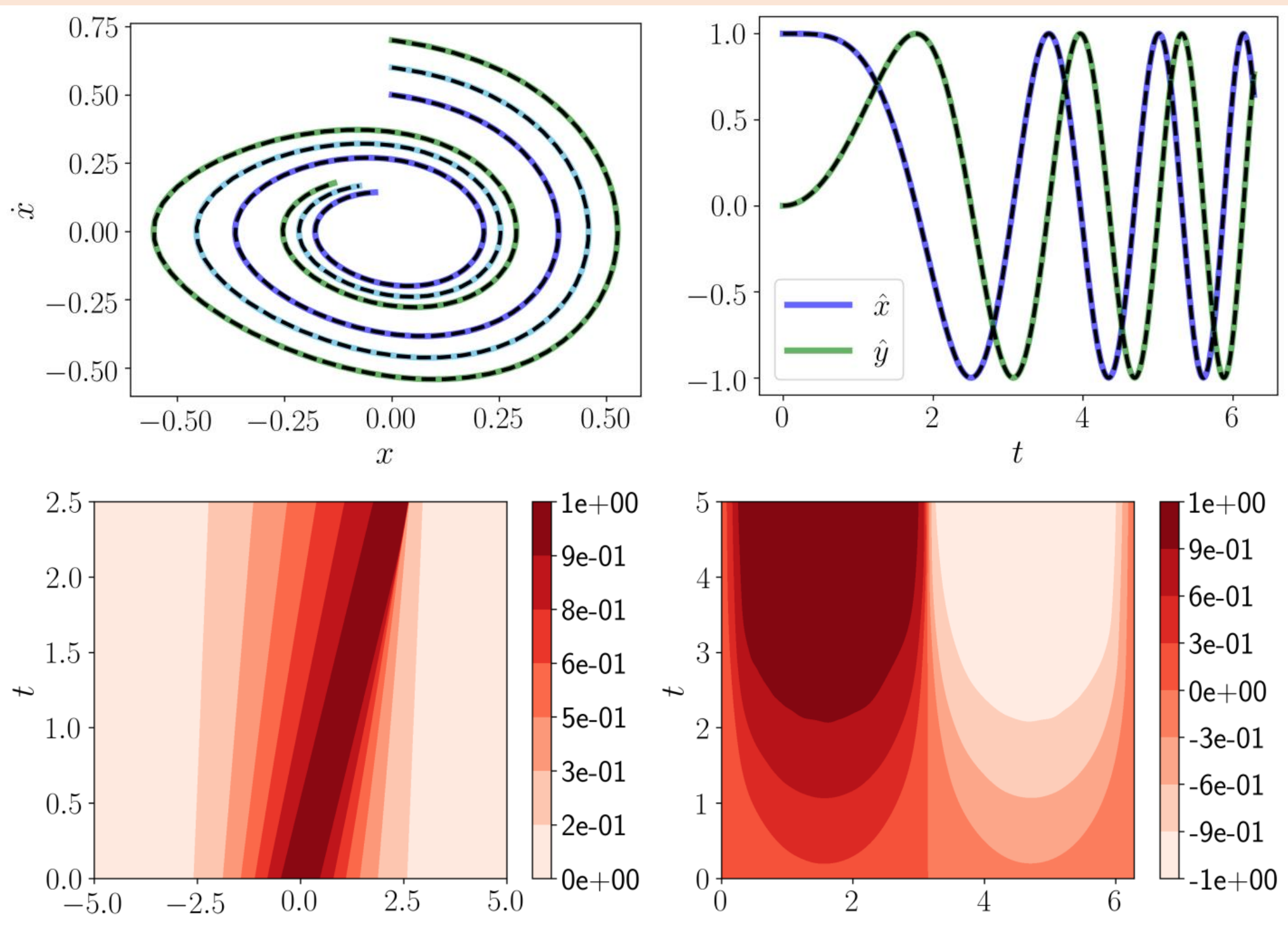


Figure 2. Visualization of DEQGAN solutions to four example equations. Solutions to the nonlinear oscillator (top left) and coupled oscillator (top right) ODEs are plotted in color with numerical solutions in dashed black lines for comparison. DEQGAN solutions to the Burgers' (bottom left) and Allen-Cahn (bottom right) PDEs are shown using contour plots.

Experimental Results

We conducted experiments on a suite of twelve differential equations and found that DEQGAN consistently obtained lower mean squared errors than classical PINNs that use L_2, L_1 and Huber loss functions, often by several orders of magnitude. Figure 3 plots the mean squared error vs. training iteration for DEQGAN and classical PINNs on six challenging equations. DEQGAN solutions to four problems are visualized in Figure 2.

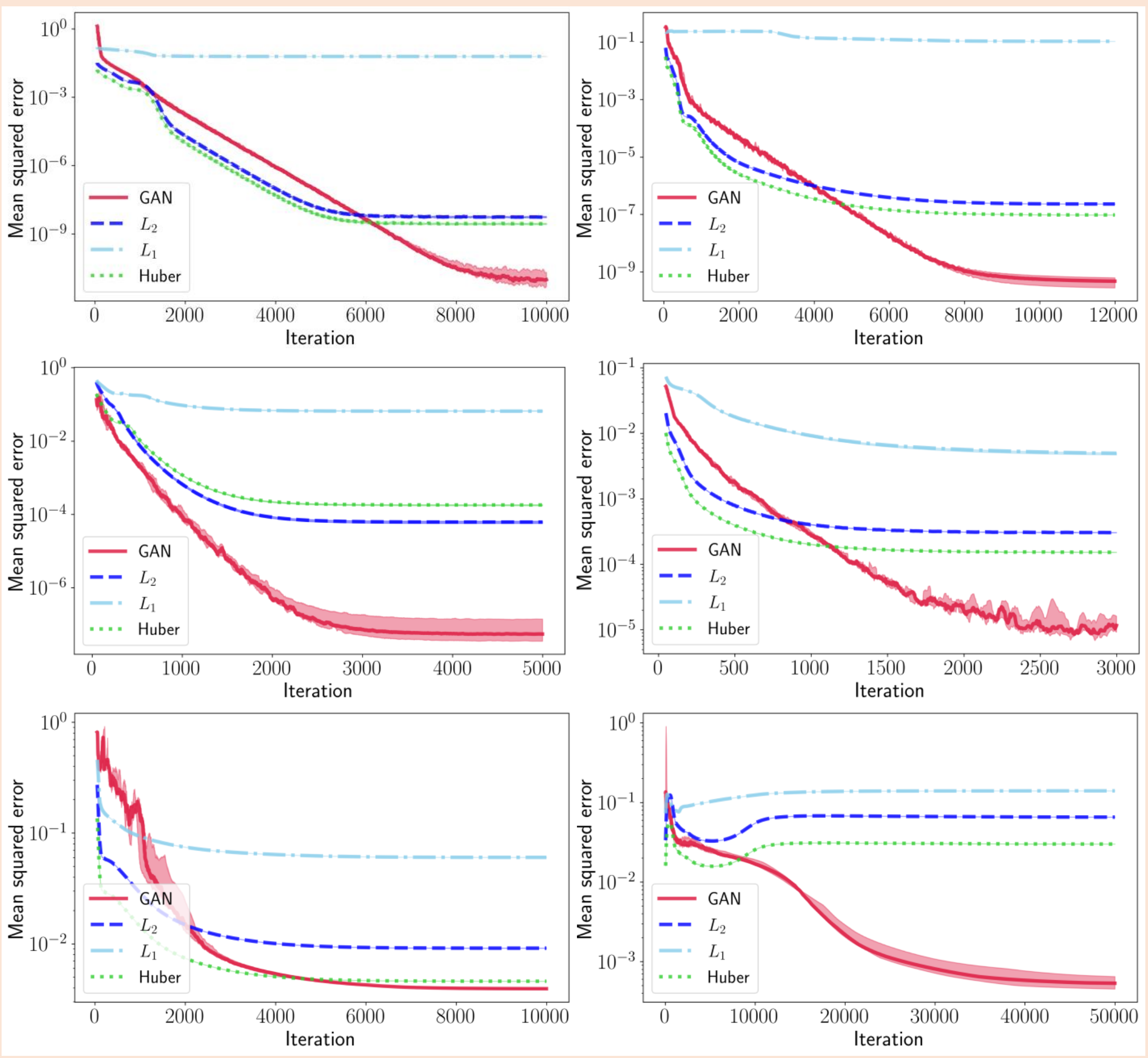


Figure 3. Mean squared errors vs. iteration for DEQGAN, L_2, L_1, and Huber loss for the (left to right, top to bottom): nonlinear oscillator, Hamilton, wave, Burgers', Allen-Cahn, and modified Einstein's gravity equations. We perform ten randomized trials and plot the median (bold) and (25, 75) percentile range (shaded). We smooth the values using a simple moving average with window size 50.

Conclusion

In this work, we presented DEQGAN, a novel method that leverages GAN-based adversarial training to "learn" the loss function for solving differential equations with PINNs. We demonstrated that our method can obtain multiple orders of magnitude lower mean squared errors than PINNs that use L_2, L_1, and Huber loss functions, including on highly nonlinear PDEs and systems of ODEs. While our results evidence the advantage of "learning the loss function" with a GAN, future work could focus on characterizing the differences between classical losses and the loss functions learned by DEQGAN, which could deepen our understanding of PINN optimization more generally.