Recurrent Localization Networks applied to the Lippman-Schwinger Equation

Conlain Kelly, Surya Kalidindi

Mined Group, Georgia Institute of Technology

BACKGROUND:

- Designing new materials requires linkages between process, structure, and property
- Estimating and understanding these linkages involves testing and simulation
- Simulating physics over high-dimensional structures is expensive
- Machine learning models are faster than traditional simulations, but less accurate and interpretable
- Physics-informed learning is a rapidly-growing field

METHODS

1. Convert governing equation to equivalent Lippman-Schwinger (L-S) form
2. Approximate L-S operator with an ML model and solve iteratively
3. Train end-to-end on synthetic data
4. Result: learned L-S iteration!

RESULTS

- $\approx$1000X speedup over FEA baseline
- More accurate and efficient than standard deep learning models
- Iterative model was more powerful than a single feed-forward network with identical structure and number of parameters

CONCLUSIONS & FUTURE WORK

- Baking physics into a learning method creates more efficient and accurate models
- Recurrent, fully-convolutional networks are excellent for modeling physical systems
- Gradient-based formulation allows for property optimization through ML model

Acknowledgements:

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039655. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Embedding physics into machine learning creates better models.

These models can accelerate the development of new materials.

$$\boldsymbol{\sigma} = C\boldsymbol{\varepsilon}, \quad \nabla \cdot \boldsymbol{\sigma} = 0.$$

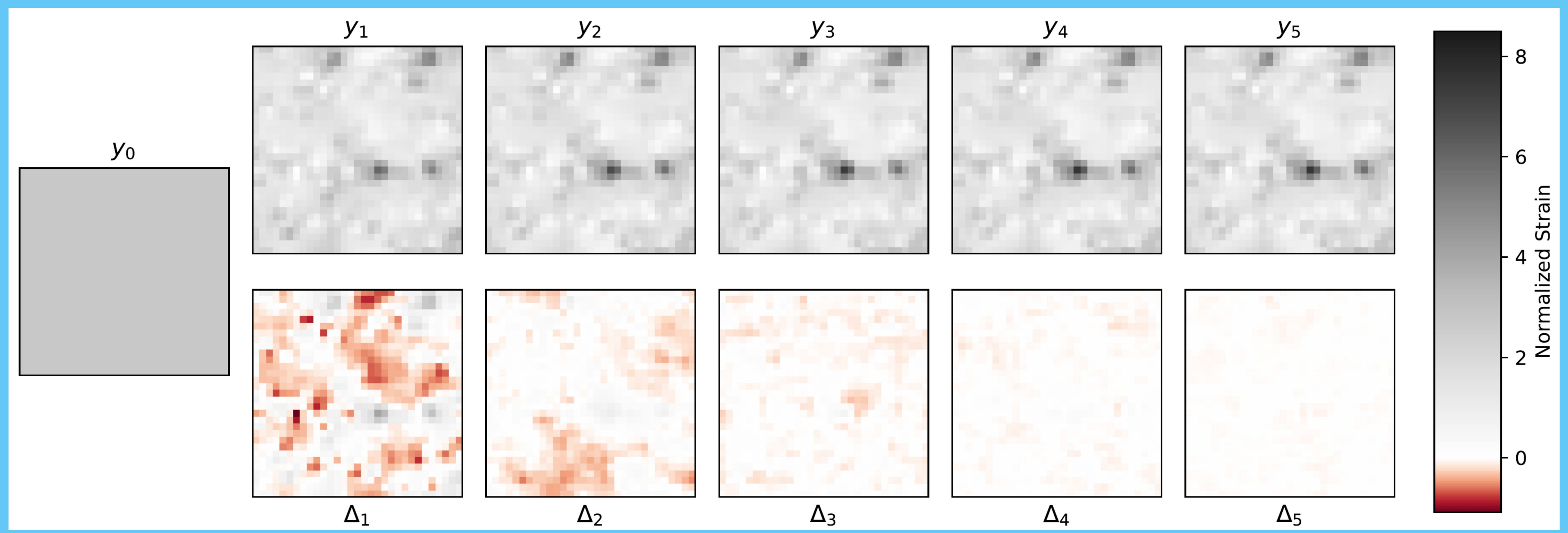
Original Physics

$$y(x) = y^R + \underbrace{\int_X G(x,s)B(y(s))ds}_{F(y)}$$

Lippmann-Schwinger Form

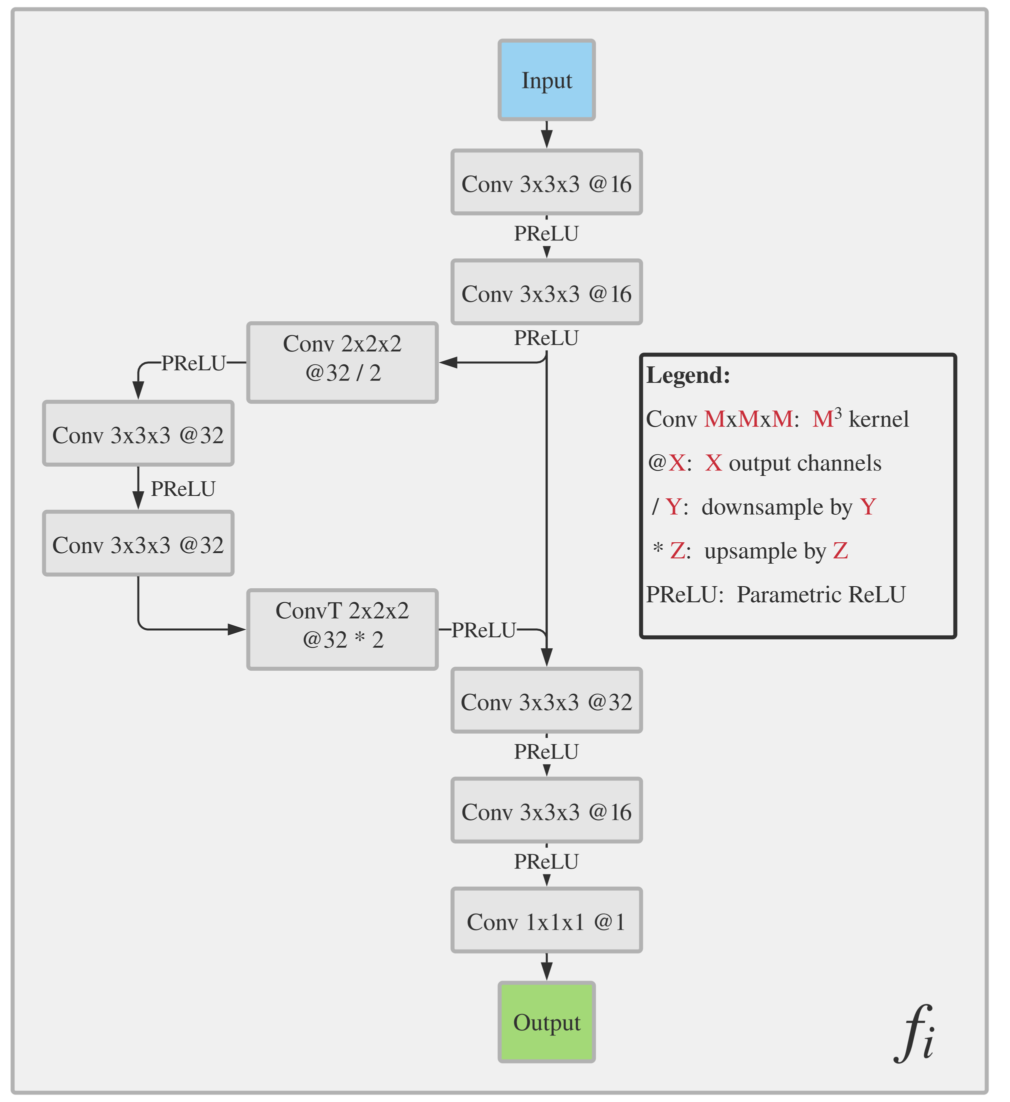
$$\begin{aligned}&\textbf{def } \hat{F}_\theta(y_0; m):\\ &\quad \textbf{for } i = 1, \dots, N:\\ &\qquad y_i = y^R + f_{\theta_i}(y_{i-1}, m)\\ &\quad \textbf{return } y_N\end{aligned}$$

Learned L-S iteration

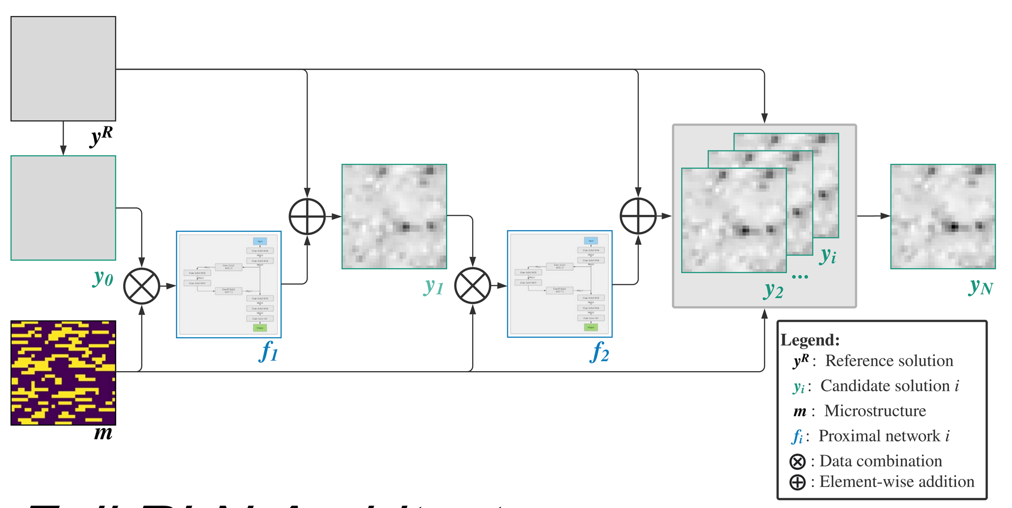


Fast, approximate physics solver

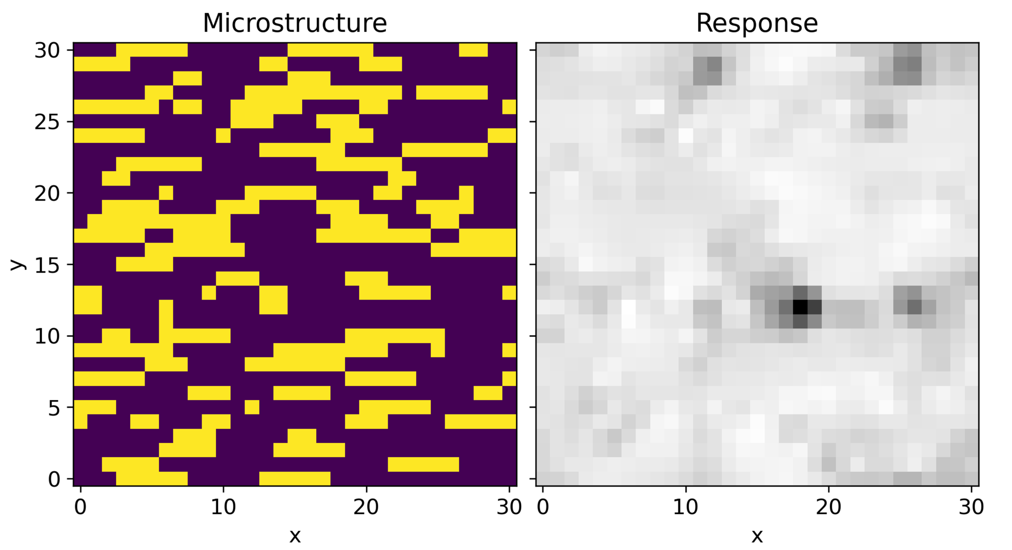
Scan me to download the full paper!



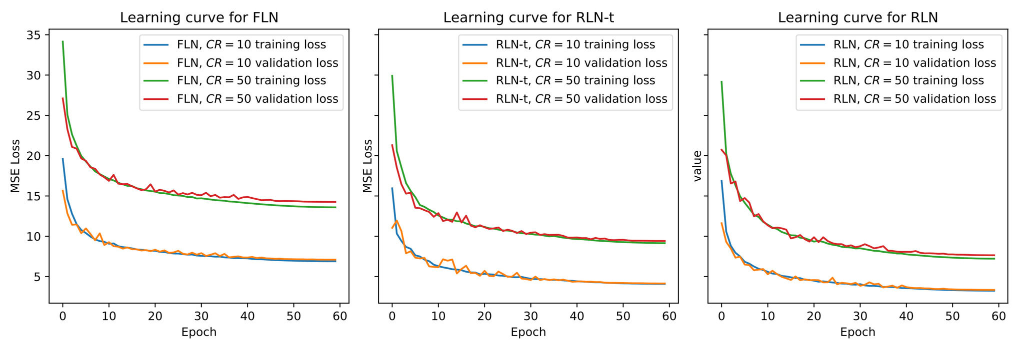
Proximal Network Architecture



Full RLN Architecture



Example input/output pair



Learning curves for each configuration

Model	MASE (mean $\pm$ std. dev.)
Contrast-10	
Comparison DL model [11]	3.07%±1.22%
FLN	4.98%±1.49%
RLN-t	1.81%±0.58%
RLN	1.21%±0.37%
Contrast-50	
Comparison DL model	5.71%±2.46%
FLN	9.23%±3.29%
RLN-t	4.26%±1.65%
RLN	2.92%±1.17%

Results for each configuration

Georgia Tech

NSF

MINED