Generative Models for Inverse Design with Constraints

Ju Sun
McKnight Land-Grant Professor
Computer Science & Engineering
University of Minnesota

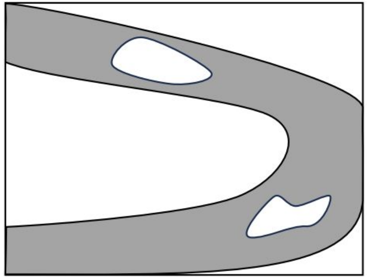
GLOVEX
https://glovex.umn.edu/

Topology optimization (TO)

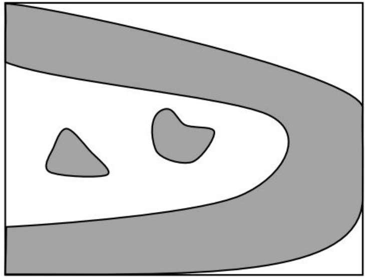
$$\min_{\boldsymbol{x}} \; c(\boldsymbol{x}) \doteq \boldsymbol{u}^\intercal \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u}$$

$$\text{s.t.} \quad \underbrace{\boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}}_{\text{physical constraint}} \;,\; \underbrace{V(\boldsymbol{x}) \leq V_0}_{\text{volume constraint}} \;,\; \underbrace{\boldsymbol{x} \in \{0,1\}^N}_{\text{discrete constraint}} \;,$$

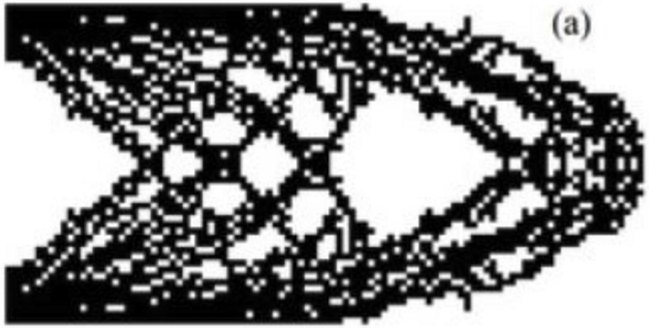
Implicit Constraints



Multiply connected
(void islands)



Non-connected
(solid islands)



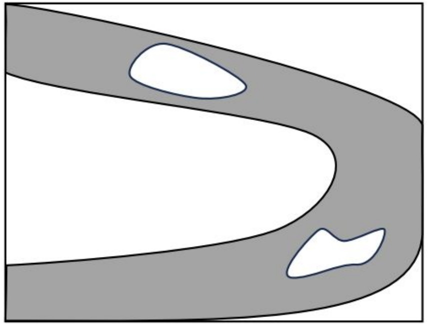
SOTA of TO : lack of feasibility & generalizability

Eliminating **u** $\boldsymbol{u} = \boldsymbol{K}^{-1}(\boldsymbol{x})\boldsymbol{f}$

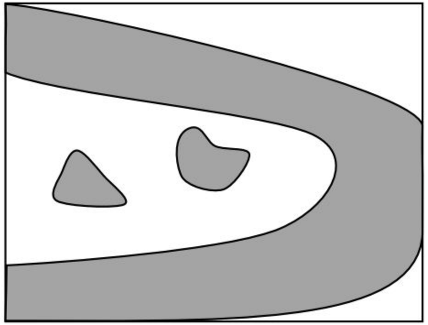
But in multi-material design

$$\boldsymbol{G}(\boldsymbol{X}, \boldsymbol{u}) = \boldsymbol{f}$$

$$\boldsymbol{x}_i \in \{\boldsymbol{e}_j\}_j$$



Multiply connected
(void islands)



Non-connected
(solid islands)

Neural TO with principled constrained optimization

$$\min_{\boldsymbol{x},\boldsymbol{u}} \ c(\boldsymbol{x}) \doteq \boldsymbol{u}^\intercal \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u}$$
$$\text{s.t. } \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}; \quad V(\boldsymbol{x}) \le V_0; \quad \boldsymbol{x} \odot (1 - \boldsymbol{x}) = \mathbf{0}$$

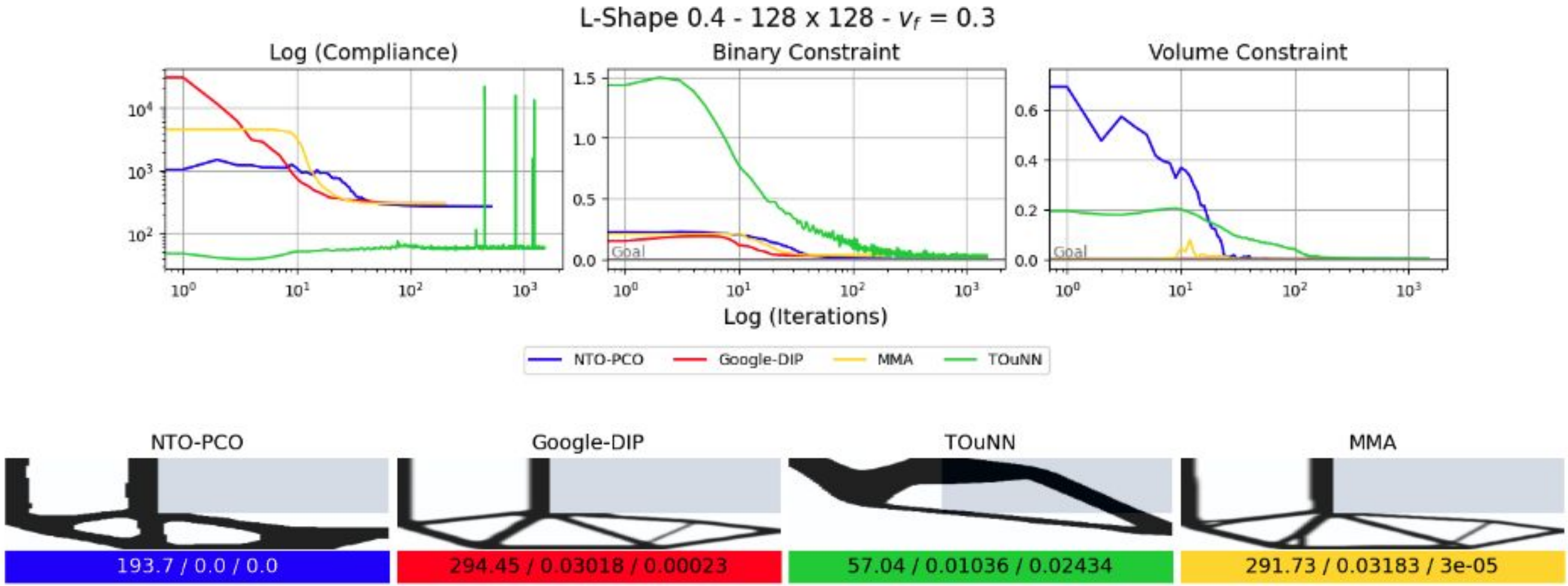
➕ Spatial smoothing via reparametrization $\boldsymbol{x} = \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})$

⬇

$$\min_{\boldsymbol{\theta},\boldsymbol{u}} \ c \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z}) \doteq \boldsymbol{u}^\intercal \boldsymbol{K} \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{u}$$
$$\text{s.t. } \boldsymbol{K} \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})\boldsymbol{u} = \boldsymbol{f}; \quad V \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z}) \le V_0; \quad \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z}) \odot (1 - \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})) = \mathbf{0}$$

Finally: Solved by principled solvers for deep learning with constraints

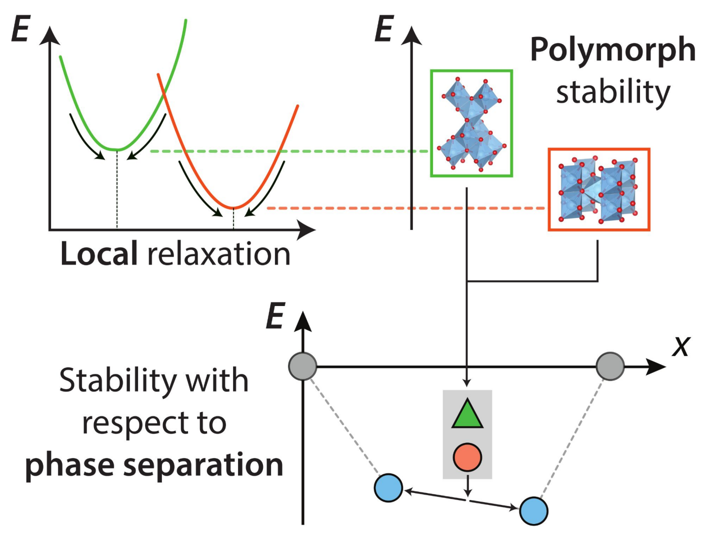
Result: TO with no-design region



Values: [Compliance, Binary Constraint Violation, Volume Constraint Violation]

Deep learning with nontrivial **hard** constraints

Material Science



Robustness evaluation

$$\max_{\boldsymbol{x}'} \ell(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}'))$$

$$\text{s.t. } \boldsymbol{x}' \in \Delta(\boldsymbol{x}) = \{\boldsymbol{x}' \in [0,1]^n : d(\boldsymbol{x}, \boldsymbol{x}') \leq \varepsilon\}$$

$$\min_{\boldsymbol{x}' \in [0,1]^n} d(\boldsymbol{x}, \boldsymbol{x}') \quad \text{s.t. } \max_{\ell \neq y} f_{\boldsymbol{\theta}}^{\ell}(\boldsymbol{x}') \geq f_{\boldsymbol{\theta}}^{y}(\boldsymbol{x}')$$

Imbalance Learning

$$\max_{\boldsymbol{\theta}, t} \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\} \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^{N} \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}$$

$$\text{s.t. } \frac{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\} \mathbb{1}\{f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) > t\}}{\sum_{i=1}^{N} \mathbb{1}\{y_i = +1\}} \geq \alpha$$

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \ell(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \Omega(\boldsymbol{\theta}) \quad \text{s.t. } \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}_p) \geq f_{\boldsymbol{\theta}}(\boldsymbol{x}_q) \ \forall\, (p, q) \in \mathcal{O}_{\text{W-CSTR-T}},$$

NCVX: A principled solver for constrained DL

PyGRANSO

NCVX PyGRANSO Documentation

Search the docs ...

Introduction

Installation

Sett

Exa

Home

https://ncvx.org/

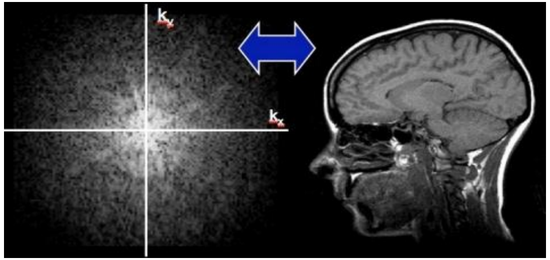
NCVX

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \;\; \text{s.t. } c_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{I}; \;\; c_i(\mathbf{x}) = 0, \forall i \in \mathcal{E}$$

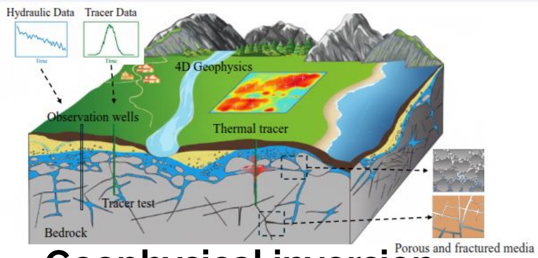
- **First general-purpose solver for <u>hard-constrained</u> deep learning problems**
- **Recently updated to be compatible with PyTorch 2.8**

Generative models for **inverse modeling**

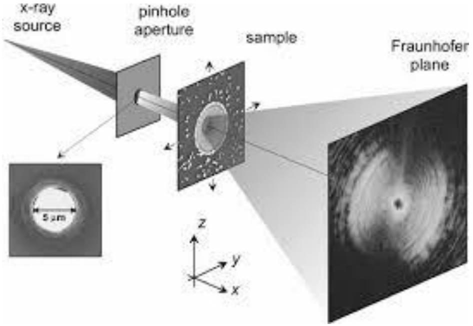
(inverse problems, reconstruction, data assimilation, **inverse design/control, conditional generation**, …)



MRI reconstruction



Geophysical inversion



Coherent diffraction imaging

Inverse shape control

Given $\boldsymbol{y} \approx f(\boldsymbol{x})$, recover $\boldsymbol{x}$

$$\min_{\boldsymbol{x}} \ \ell(\boldsymbol{y}, f(\boldsymbol{x})) + \Omega(\boldsymbol{x})$$

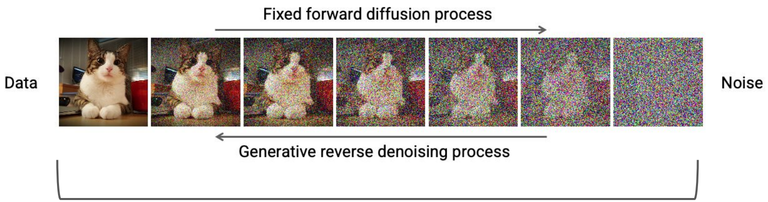
Challenges:

- Linear vs. **nonlinear** f
- Unconstrained vs. **constrained** (e.g., PDE (x, y) = 0)
- Explicit vs. **implicit** f

A **plug-in principle** for leveraging **pretrained** deep generative models to solve IM

object-only datasets $\{\boldsymbol{x}_i\}_{i=1,\dots,N}$

Distribution learning via deep generative models



$\mathcal{G}_{\boldsymbol{\theta}}$

Plug in pretrained deep generative priors

$$\min_{\boldsymbol{z}} \ \mathcal{L}(\boldsymbol{z}) \doteq \ell(\boldsymbol{y}, \mathcal{A} \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})) + \Omega \circ \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{z})$$

DMPlug: A Plug-in Method for Solving Inverse Problems with Diffusion Models

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, Ju Sun

NeurIPS'24; with **domain-specific** priors

Saving Foundation Flow-Matching Priors for Inverse Problems

Yuxiang Wan, Ryan Devera, Wenjie Zhang, Ju Sun

ICML'26; with **foundation** priors (e.g., stable diffusion models)

Challenge: Foundation generative models are powerful because **they're not specific**