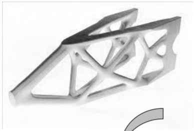
DNN based topology optimization: spatial invariance and neural tangent kernel

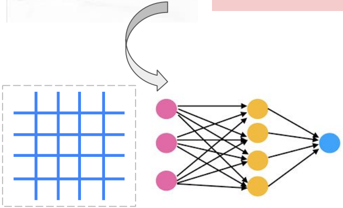
Benjamin Dupuis, Arthur Jacot
Chair of Statistical Field Theory - EPFL

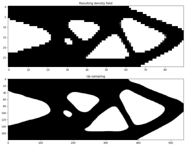
October 17, 2021

Overview



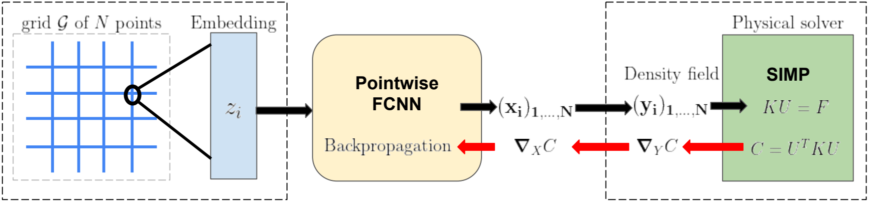
Topology optimization aims to find optimal shapes with respect to some forces and boundary conditions





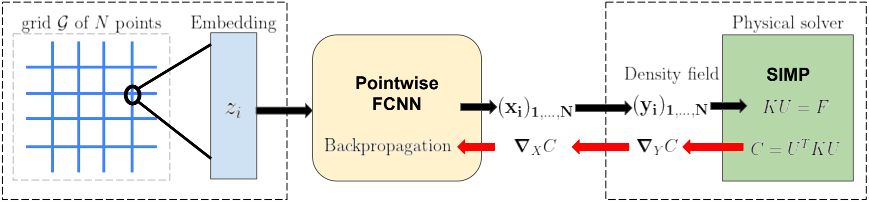
- **Our goal is to use DNNs as an implicit representation of the shape**
- **We analyse it through NTK theory**
- **We suggest tools to improve coordinates-based generative models**
- **We analize an analogy between the NTK and a filter**

Algorithm



Loss function C is called the compliance.

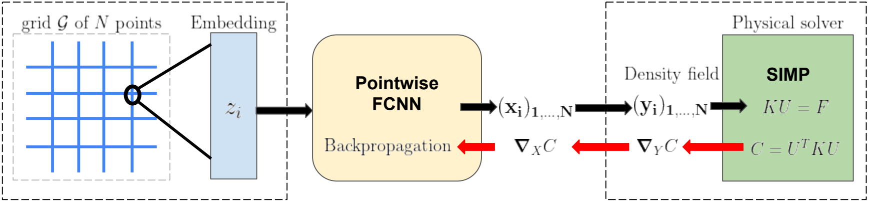
Algorithm



Loss function C is called the compliance.

- Embedding $z_i = \varphi(p_i)$, $\varphi : \mathbb{R}^2 \longrightarrow \mathbb{R}^{n_0}$
- Fully-connected DNN: $x_i = f_\theta(z_i)$, $f_\theta : \mathbb{R}^{n_0} \longrightarrow \mathbb{R}$

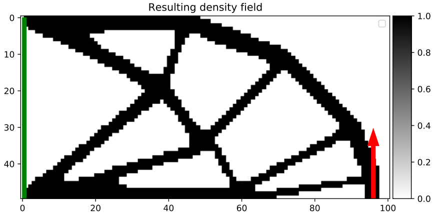
Algorithm

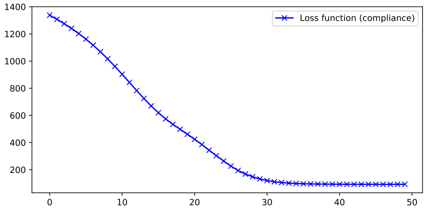


Loss function C is called the compliance.

- Embedding $z_i = \varphi(p_i)$, $\varphi : \mathbb{R}^2 \longrightarrow \mathbb{R}^{n_0}$
- Fully-connected DNN: $x_i = f_\theta(z_i)$, $f_\theta : \mathbb{R}^{n_0} \longrightarrow \mathbb{R}$
- Mass control: we find $\bar{b}$ such that $\sum_i \sigma(x_i + \bar{b}) = V_0$
- Implicit differenciation: $\nabla_X C = D_X \nabla_Y C$

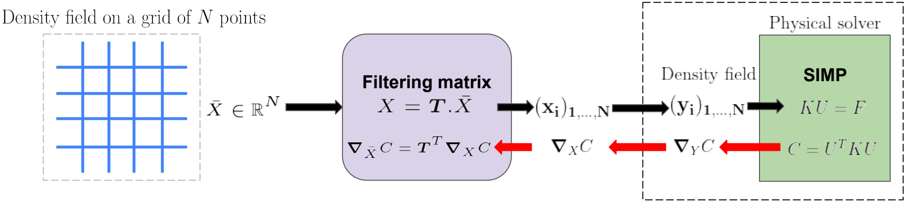
Example



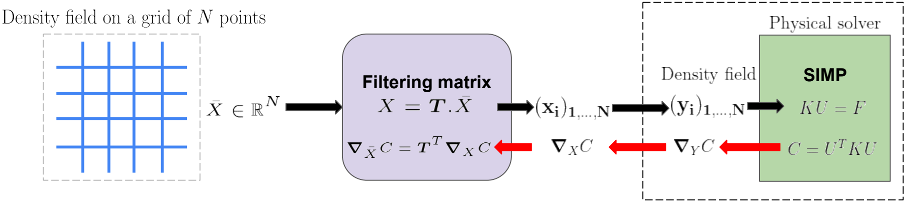


- Our method achieves excellent numerical results in a small number of iterations

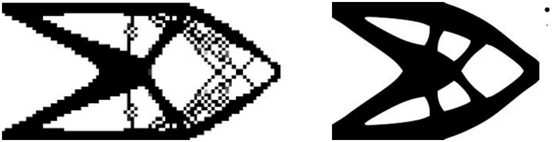
Comparison with traditional filtering in SIMP



Comparison with traditional filtering in SIMP



- Filtering aims to remove checkerboards but it has drawbacks



Neural Networks

- NTK parametrization of FCNNs:
  $a^0(x) = x, \quad \tilde{a}^{l+1}(x) = \frac{\alpha}{\sqrt{n_l}} W^l a^l(x) + \beta b^l, \quad a^{l+1}(x) = \mu\big(\tilde{a}^{l+1}(x)\big),$
- $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\mu(X)^2] = 1$

Neural Networks

- NTK parametrization of FCNNs:
  $a^0(x) = x, \quad \tilde{a}^{l+1}(x) = \frac{\alpha}{\sqrt{n_l}} W^l a^l(x) + \beta b^l, \quad a^{l+1}(x) = \mu\big(\tilde{a}^{l+1}(x)\big),$
- $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\mu(X)^2] = 1$
- NTK: $\Theta^L_\theta(z, z') = \sum_p \frac{\partial f_\theta}{\partial \theta_p}(z) \frac{\partial f_\theta}{\partial \theta_p}(z') = (\nabla_\theta f_\theta(z) | \nabla_\theta f_\theta(z'))$
- Gradient flow $\partial_t X(\theta(t)) = -\tilde{\Theta}^L_{\theta(t)} \nabla_X C$

Neural Networks

- NTK parametrization of FCNNs:
  $a^0(x) = x, \quad \tilde{a}^{l+1}(x) = \frac{\alpha}{\sqrt{n_l}} W^l a^l(x) + \beta b^l, \quad a^{l+1}(x) = \mu\big(\tilde{a}^{l+1}(x)\big),$
- $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\mu(X)^2] = 1$
- NTK: $\Theta_\theta^L(z, z') = \sum_p \frac{\partial f_\theta}{\partial \theta_p}(z) \frac{\partial f_\theta}{\partial \theta_p}(z') = (\nabla_\theta f_\theta(z) | \nabla_\theta f_\theta(z'))$
- Gradient flow $\partial_t X(\theta(t)) = -\tilde{\Theta}_{\theta(t)}^L \nabla_X C$

Gradient flow in our method

$$\partial_t Y^{\mathrm{NN}}(\theta(t)) = -D_X(t) \tilde{\Theta}_\infty^L D_X(t) \nabla_Y C(Y^{\mathrm{NN}}(\theta(t)))$$

Neural Networks

- NTK parametrization of FCNNs:
  $a^0(x) = x, \quad \tilde{a}^{l+1}(x) = \frac{\alpha}{\sqrt{n_l}} W^l a^l(x) + \beta b^l, \quad a^{l+1}(x) = \mu\left(\tilde{a}^{l+1}(x)\right),$
- $\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\mu(X)^2] = 1$
- NTK: $\Theta^L_\theta(z,z') = \sum_p \frac{\partial f_\theta}{\partial \theta_p}(z) \frac{\partial f_\theta}{\partial \theta_p}(z') = (\nabla_\theta f_\theta(z) | \nabla_\theta f_\theta(z'))$
- Gradient flow $\partial_t X(\theta(t)) = -\tilde{\Theta}^L_{\theta(t)} \nabla_X C$

Gradient flow in our method

$$\partial_t Y^{\mathrm{NN}}(\theta(t)) = -D_X(t)\tilde{\Theta}^L_\infty D_X(t) \nabla_Y C(Y^{\mathrm{NN}}(\theta(t)))$$

Gradient flow without neural network

$$\partial_t Y^{\mathrm{MF}}(t) = -D_X(t) T T^T D_X(t) \nabla_Y C(Y^{\mathrm{MF}}(t))$$

Ensuring spatial invariance

- analogy between $\tilde{\Theta}_\infty$ and TT^T
- Ensuring spatial invariance of the filter is crucial
- Idea: introduce spatial invariance of the NTK via embeddings.

Ensuring spatial invariance

- analogy between $\tilde{\Theta}_\infty$ and TT^T
- Ensuring spatial invariance of the filter is crucial
- Idea: introduce spatial invariance of the NTK via embeddings.

Proposition

Let $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_0}$ for $d > 2$ and any finite n_0. If φ satisfies $\varphi(x)^T\varphi(x') = K(\|x - x'\|)$ for some continuous function K then both φ and K are constant.

Ensuring spatial invariance

- analogy between $\tilde{\Theta}_\infty$ and TT^T
- Ensuring spatial invariance of the filter is crucial
- Idea: introduce spatial invariance of the NTK via embeddings.

Proposition

Let $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_0}$ for $d > 2$ and any finite n_0. If φ satisfies $\varphi(x)^T\varphi(x') = K(\|x - x'\|)$ for some continuous function K then both φ and K are constant.

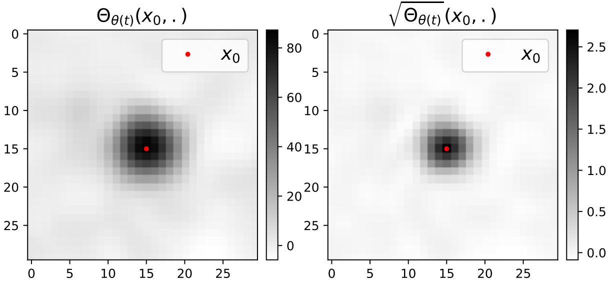
- We propose embeddings ensuring spatial invariance of the NTK

Torus embedding

- The NTK is invariant under rotation (function of $z^T z'$, $\|z\|$, $\|z'\|$)
- We transfer this property to translation invariance
- $\mathbb{R}^2 \ni p = (p_1, p_2) \longmapsto \varphi(p) = r(\cos(\delta p_1), \sin(\delta p_1), \cos(\delta p_2), \sin(\delta p_2))$

Torus embedding

- The NTK is invariant under rotation (function of $z^T z'$, $\|z\|$, $\|z'\|$)
- We transfer this property to translation invariance
- $\mathbb{R}^2 \ni p = (p_1, p_2) \longmapsto \varphi(p) = r(\cos(\delta p_1), \sin(\delta p_1), \cos(\delta p_2), \sin(\delta p_2))$



proposition

The square root of the Gram Matrix $\sqrt{\tilde{\Theta}_\infty}$ is a discrete convolution matrix

Fourier Features embeddings

- Bochner theorem, for a positive kernel: $k(x) = k(0)\mathbb{E}_{\omega\sim\mathbb{Q}}\left[e^{i\omega . x}\right]$

Fourier Features embeddings

- Bochner theorem, for a positive kernel: $k(x) = k(0)\mathbb{E}_{\omega \sim \mathbb{Q}}\left[e^{i\omega . x}\right]$
- we can formulate embeddings invariant by rotation and translation:

$$\varphi(p)_i = \sqrt{2k(0)}\sin(w_i^T p + \frac{\pi}{4} + b_i)$$

- With $w \sim \mathbb{Q}$ and b_i i.i.d. random variable from a symmetric distribution.
- Gaussian embedding: $w \sim \mathcal{N}(0, \frac{1}{\ell^2})$

Fourier Features embeddings

- Bochner theorem, for a positive kernel: $k(x) = k(0)\mathbb{E}_{\omega\sim\mathbb{Q}}\left[e^{i\omega.x}\right]$
- we can formulate embeddings invariant by rotation and translation:

$$\varphi(p)_i = \sqrt{2k(0)}\sin(w_i^T p + \frac{\pi}{4} + b_i)$$

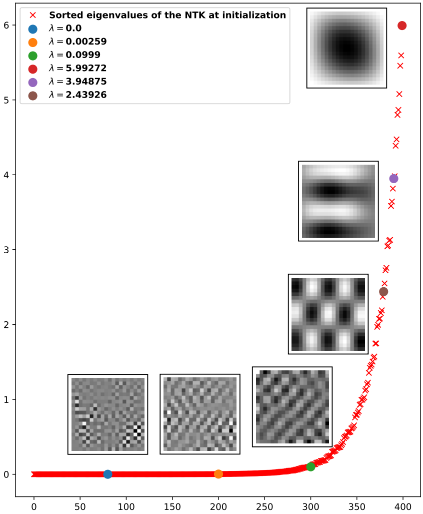
- With $w \sim \mathbb{Q}$ and b_i i.i.d. random variable from a symmetric distribution.
- Gaussian embedding: $w \sim \mathcal{N}(0, \frac{1}{\ell^2})$

Proposition

Let φ be an embedding as described above for a positive radial kernel $k \in L^1(\mathbb{R}^d)$ with $k(0) = 1$, $k \geq 0$. There is a filter function $g : \mathbb{R} \to \mathbb{R}$ and a constant C such that for all p, p', in probability:

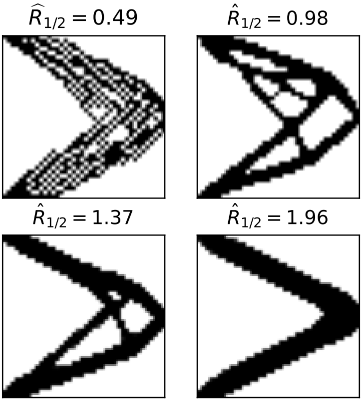
$$\lim_{n_0\to\infty} \Theta_\infty(\varphi(p), \varphi(p')) = C + (g \star g)(p - p'),$$

Experimental results - spectral decomposition of the NTK

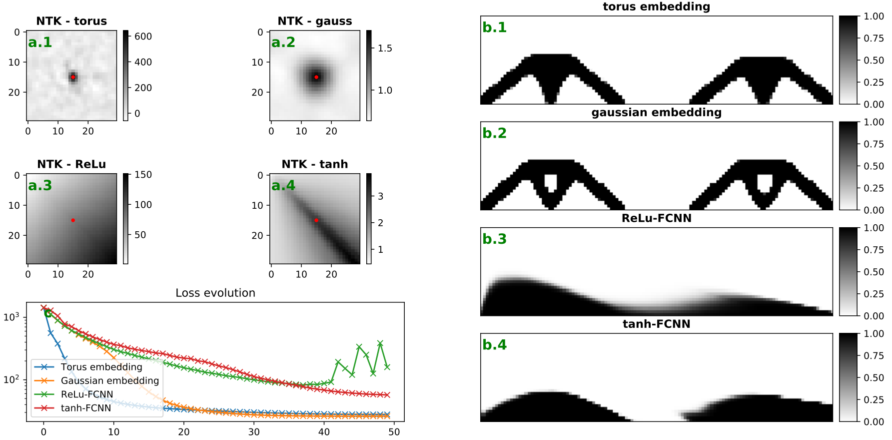


Experimental results - Filter radius control

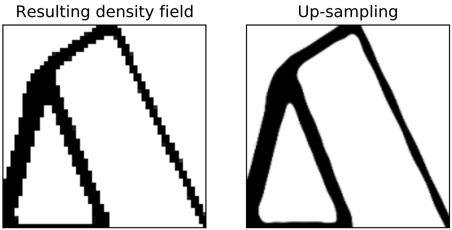
- We are able to define and control a "filtering radius"



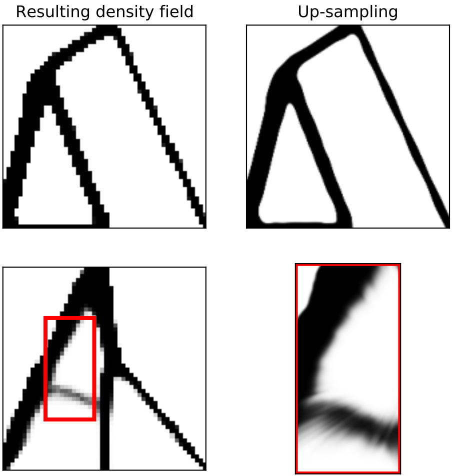
Experimental results - Filter radius control



up sampling



up sampling



Thank you