On the Benefits of Convolutional Models: a Kernel Perspective

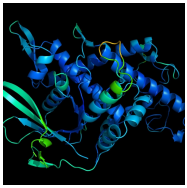
Alberto Bietti

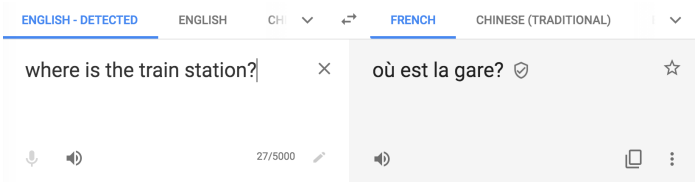
NYU

Meta. April 21, 2022.

Success of Deep Learning

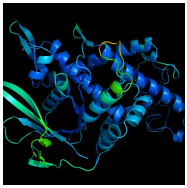
State-of-the-art models in various domains (images, speech, language, biology, ...)

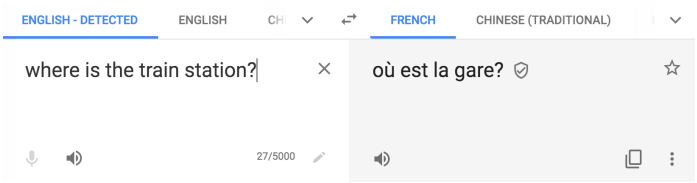




Success of Deep Learning

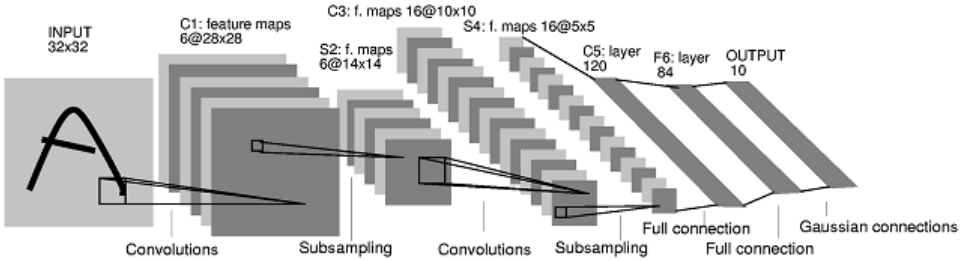
State-of-the-art models in various domains (images, speech, language, biology, ...)





Recipe: **huge models** + **lots of data** + **compute** + **simple algorithms**

Convolutional Networks

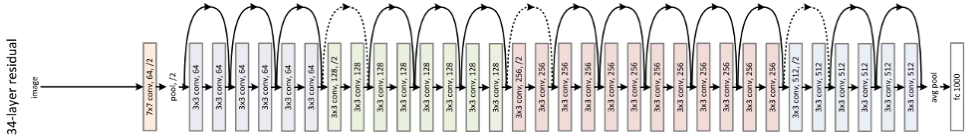


(LeCun et al., 1998)

Exploiting the structure of natural images

- Model local information at different scales, hierarchically
- Provide some invariance through pooling
- Useful **inductive biases** for learning efficiently on natural signals

Convolutional Networks



(He et al., 2016)

Exploiting the structure of natural images

- Model local information at different scales, hierarchically
- Provide some invariance through pooling
- Useful **inductive biases** for learning efficiently on natural signals

Understanding Deep Learning

Understanding Deep Learning

The challenge of deep learning theory

- **Over-parameterized** (millions/billions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** to zero training error with (stochastic) gradient descent!

Understanding Deep Learning

The challenge of deep learning theory

- **Over-parameterized** (millions/billions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** to zero training error with (stochastic) gradient descent!

A functional viewpoint

- View deep networks as functions in some functional space
- Non-parametric models, natural measures of complexity (*e.g.*, norms)
- Optimization performs **implicit regularization** towards

$$\min_f \Omega(f) \quad \text{s.t.} \quad y_i = f(x_i), \quad i = 1, \dots, n$$

Understanding Deep Learning

The challenge of deep learning theory

- **Over-parameterized** (millions/billions of parameters)
- **Expressive** (can approximate any function)
- Complex **architectures** for exploiting problem structure
- Yet, **easy to optimize** to zero training error with (stochastic) gradient descent!

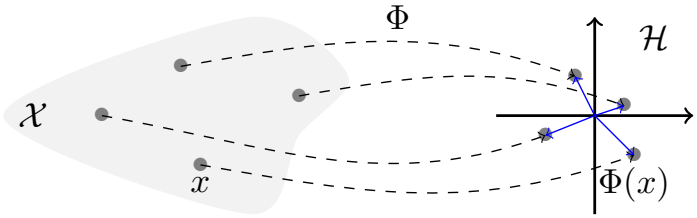
A functional viewpoint

- View deep networks as functions in some functional space
- Non-parametric models, natural measures of complexity (*e.g.*, norms)
- Optimization performs **implicit regularization** towards

$$\min_f \Omega(f) \quad \text{s.t.} \quad y_i = f(x_i), \quad i = 1, \dots, n$$

Q: What is an appropriate functional space / norm Ω?

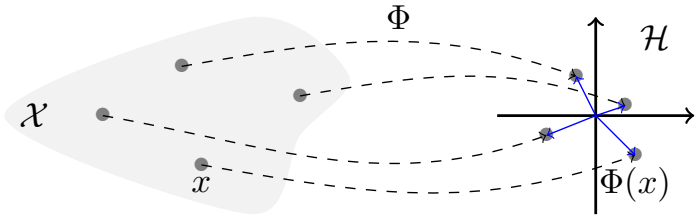
Kernels



Kernels?

- Map data $x \in \mathcal{X}$ to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Functions $f \in \mathcal{H}$ are linear in features: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$ (f can be non-linear in x!)

Kernels



Kernels?

- Map data $x \in \mathcal{X}$ to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Functions $f \in \mathcal{H}$ are linear in features: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$ (f can be non-linear in x!)
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$
  - $\mathcal{H}$ can be infinite-dimensional! (*kernel trick*)
  - Use a kernel matrix $K = [K(x_i, x_j)]_{ij} \in \mathbb{R}^{N \times N}$ or its approximations

Kernels for Convolutional Models

My work (B. and Mairal, 2019a,b; B. et al., 2021b; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Kernels for Convolutional Models

My work (B. and Mairal, 2019a,b; B. et al., 2021b; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Invariance

Kernels for Convolutional Models

My work (B. and Mairal, 2019a,b; B. et al., 2021b; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Invariance

Locality
Long-range interactions

Kernels for Convolutional Models

My work (B. and Mairal, 2019a,b; B. et al., 2021b; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Invariance

Locality
Long-range interactions

Deformation stability
Multi-scale structure

Why Kernels?

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems (*e.g.*, smooth functions)

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems (*e.g.*, smooth functions)

We rarely have *all three* in deep learning theory, *e.g.*:

- Benefits of depth (*e.g.*, Eldan and Shamir, 2016; Mhaskar and Poggio, 2016): no algorithms
- Optimization landscape (*e.g.*, Soltanolkotabi et al., 2018): no universal approximation

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems (*e.g.*, smooth functions)

We rarely have *all three* in deep learning theory, *e.g.*:

- Benefits of depth (*e.g.*, Eldan and Shamir, 2016; Mhaskar and Poggio, 2016): no algorithms
- Optimization landscape (*e.g.*, Soltanolkotabi et al., 2018): no universal approximation

A starting point to understand CNNs

- Understand the **features** $\Phi(x)$ provided by architectures
- $\approx$ study least squares before the Lasso

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems (*e.g.*, smooth functions)

We rarely have *all three* in deep learning theory, *e.g.*:

- Benefits of depth (*e.g.*, Eldan and Shamir, 2016; Mhaskar and Poggio, 2016): no algorithms
- Optimization landscape (*e.g.*, Soltanolkotabi et al., 2018): no universal approximation

A starting point to understand CNNs

- Understand the **features** $\Phi(x)$ provided by architectures
- $\approx$ study least squares before the Lasso
- Good performance on standard vision datasets (Mairal, 2016; Shankar et al., 2020; B., 2022)

Outline

1. Invariance and Stability (B., Venturi, and Bruna, 2021b)
2. Locality and Depth (B., 2022)
3. Multi-Scale Structure and Stability (B. and Mairal, 2019a,b)
4. Concluding Remarks and Research Directions

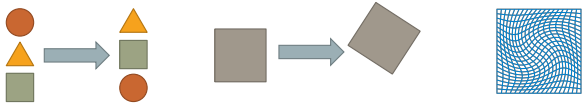
Outline

Invariance and Geometric Stability

Invariance and Geometric Stability

Q: Does invariance improve learning efficiency?

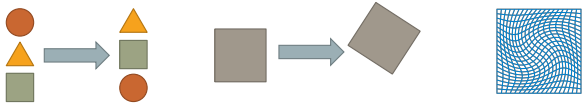
Invariance and Geometric Stability: Definitions



Functions $f : \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations

Invariance and Geometric Stability: Definitions

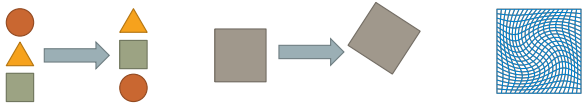


Functions $f : \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

Invariance and Geometric Stability: Definitions



Functions $f : \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

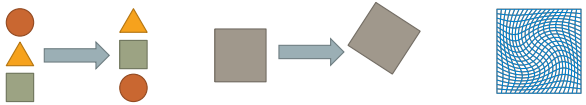
- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

Group invariance: If G is a group (*e.g.*, cyclic shifts, all permutations), we want

$$f(\sigma \cdot x) = f(x), \quad \sigma \in G$$

Invariance and Geometric Stability: Definitions



Functions $f : \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

Group invariance: If G is a group (*e.g.*, cyclic shifts, all permutations), we want

$$f(\sigma \cdot x) = f(x), \quad \sigma \in G$$

Geometric stability: For other sets G (*e.g.*, local shifts, deformations), we want

$$f(\sigma \cdot x) \approx f(x), \quad \sigma \in G$$

Interlude: Kernels for Infinite Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$

Interlude: Kernels for Infinite Shallow Networks

$$
\begin{aligned}
f(x) &= \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle) \\
&= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m
\end{aligned}
$$

Interlude: Kernels for Infinite Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$
$$= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m$$

- **Random Features** (RF, Neal, 1996; Rahimi and Recht, 2007): $w_i \sim \mathcal{N}(0, I)$, learn v

$$K_{RF}(x, x') = \lim_{m \to \infty} \langle \varphi(x), \varphi(x') \rangle$$
$$= \mathbb{E}_w[\rho(\langle w, x \rangle)\rho(\langle w, x' \rangle)] = \kappa_\rho(\langle x, x' \rangle) \text{ when } \|x\| = \|x'\| = 1$$

Interlude: Kernels for Infinite Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$
$$= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m$$

- **Random Features** (RF, Neal, 1996; Rahimi and Recht, 2007): $w_i \sim \mathcal{N}(0, I)$, learn v

$$K_{RF}(x, x') = \lim_{m \to \infty} \langle \varphi(x), \varphi(x') \rangle$$
$$= \mathbb{E}_w[\rho(\langle w, x \rangle)\rho(\langle w, x' \rangle)] = \kappa_\rho(\langle x, x' \rangle) \text{ when } \|x\| = \|x'\| = 1$$

- Related to **Neural Tangent Kernel** (NTK, Jacot et al., 2018): train both w_i and v_i near random initialization

Group-Invariant Models through Pooling

$$\varphi(x) = \frac{1}{\sqrt{m}}\rho(Wx)$$

Convolutional network with pooling (group averaging)

$$f_G(x) = \langle v, \underbrace{\frac{1}{|G|}\sum_{\sigma \in G} \varphi(\sigma \cdot x)}_{\Phi(x)} \rangle$$

Group-Invariant Models through Pooling

$$\varphi(x) = \frac{1}{\sqrt{m}}\rho(Wx)$$

Convolutional network with pooling (group averaging)

$$f_G(x) = \langle v, \underbrace{\frac{1}{|G|}\sum_{\sigma \in G} \varphi(\sigma \cdot x)}_{\Phi(x)} \rangle$$

Invariant Random Features kernel

$$K_G(x, x') = \lim_{m \to \infty} \langle \Phi(x), \Phi(x') \rangle = \frac{1}{|G|}\sum_{\sigma \in G} \kappa_\rho(\langle \sigma \cdot x, x' \rangle), \quad \text{when } \|x\| = \|x'\| = 1$$

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**
- **Goal**: minimize risk $R(f) := \mathbb{E}(y - f(x))^2$

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**
- **Goal**: minimize risk $R(f) := \mathbb{E}(y - f(x))^2$
- **Learn** using

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**
- **Goal**: minimize risk $R(f) := \mathbb{E}(y - f(x))^2$
- **Learn** using

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021b))

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \lesssim \left(\frac{1}{|G|n}\right)^{\beta} \qquad \textit{vs.} \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^{\beta}$$

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**
- **Goal**: minimize risk $R(f) := \mathbb{E}(y - f(x))^2$
- **Learn** using

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021b))

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \lesssim \left(\frac{1}{|G|n}\right)^{\beta} \qquad \textit{vs.} \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^{\beta}$$

$\implies$ **gains by a factor $|G|$ in sample complexity.**

Generalization Benefits of Group Invariance

- **Data**: x uniform on the sphere $\mathbb{S}^{d-1}$, labels $y = f^*(x) + \text{noise}$, with f^* G-**invariant**
- **Goal**: minimize risk $R(f) := \mathbb{E}(y - f(x))^2$
- **Learn** using

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

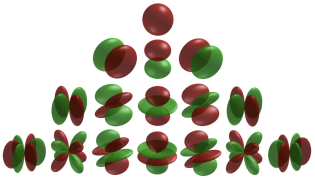
Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021b))

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \lesssim \left(\frac{1}{|G|n}\right)^{\beta} \qquad \textit{vs.} \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^{\beta}$$

$\implies$ **gains by a factor $|G|$ in sample complexity.**

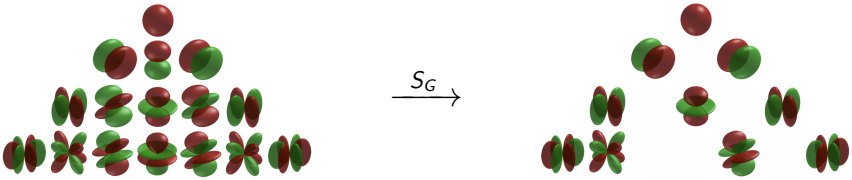
- $|G|$ can be exponential in d for some groups! (*e.g.*, the full permutation group)
- Rate β depends on d and smoothness of f^* (minimax optimal)

Key Technical Ingredient: Counting Invariant Harmonics



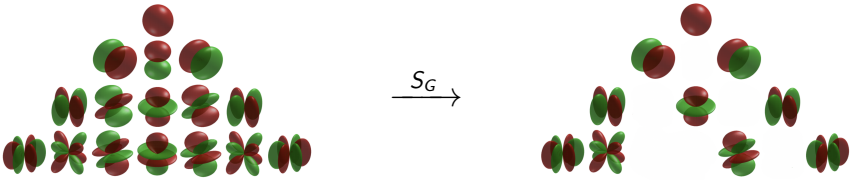
- Expansions in the basis of **spherical harmonics** $Y_{k,j}$ on the sphere $\mathbb{S}^{d-1}$
- N_k: number of harmonics of degree k

Key Technical Ingredient: Counting Invariant Harmonics



- Expansions in the basis of **spherical harmonics** $Y_{k,j}$ on the sphere $\mathbb{S}^{d-1}$
- N_k: number of harmonics of degree k
- Pooling projects down to $\overline{N}_k$ **invariant harmonics**

Key Technical Ingredient: Counting Invariant Harmonics



- Expansions in the basis of **spherical harmonics** $Y_{k,j}$ on the sphere $\mathbb{S}^{d-1}$
- N_k: number of harmonics of degree k
- Pooling projects down to $\overline{N}_k$ **invariant harmonics**
- Key result: decrease in **effective dimensionality** by a factor $|G|$

Theorem (Invariant harmonics (B., Venturi, and Bruna, 2021b))

As $k \to \infty$, we have

$$\frac{\overline{N}_k}{N_k} \to \frac{1}{|G|}$$

Extension to Stability and Discussion

Geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural stability assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Extension to Stability and Discussion

Geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural stability assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Curse of dimensionality

- If the target f^* is non-smooth, *e.g.*, only Lipschitz, the rate is cursed! (and unimprovable)

$$R(\hat{f}_n) - R(f^*) \lesssim n^{-\beta} \quad \text{with } \beta \approx \frac{1}{d}$$

Extension to Stability and Discussion

Geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural stability assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Curse of dimensionality

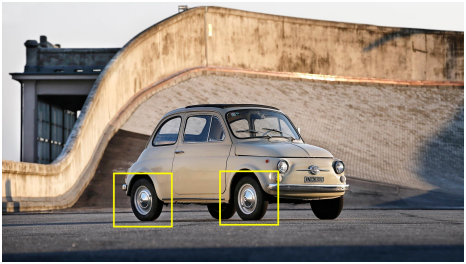
- If the target f^* is non-smooth, *e.g.*, only Lipschitz, the rate is cursed! (and unimprovable)

$$R(\hat{f}_n) - R(f^*) \lesssim n^{-\beta} \quad \text{with } \beta \approx \frac{1}{d}$$

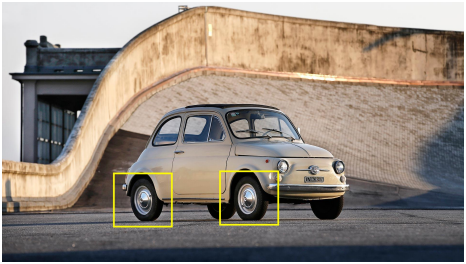
Q: How can we break this curse?

Outline

Locality

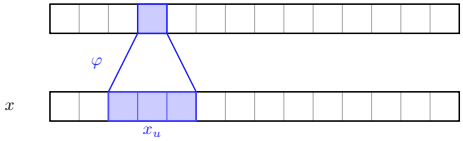


Locality



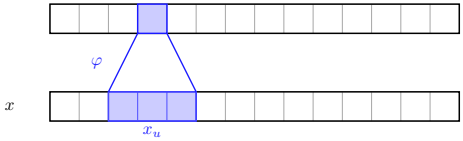
Q: Can locality improve sample complexity?

One-Layer Convolutional Kernels on Patches



- **Patches**: $x_u \in \mathbb{R}^p$ at positions $u \in \Omega$
- **Features**: $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$, **patch kernel**: $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$

One-Layer Convolutional Kernels on Patches



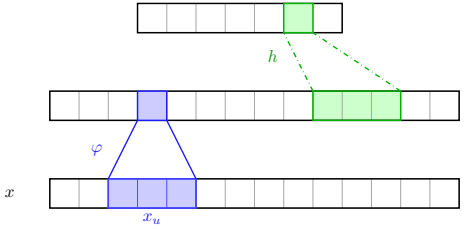
- **Patches**: $x_u \in \mathbb{R}^p$ at positions $u \in \Omega$
- **Features**: $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$, **patch kernel**: $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$
- **Convolutional network**:

$$f(x) = \sum_{u \in \Omega} \langle v_u, \varphi(x_u) \rangle =: \langle v, \Phi(x) \rangle$$

- **Convolutional kernel**:

$$K(x, x') = \sum_{u \in \Omega} k(x_u, x'_u)$$

One-Layer Convolutional Kernels on Patches



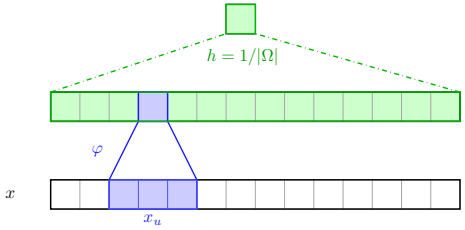
- **Patches**: $x_u \in \mathbb{R}^p$ at positions $u \in \Omega$
- **Features**: $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$, **patch kernel**: $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$
- **Convolutional network**: with pooling filter h

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, \sum_v h[u - v] \varphi(x_v) \rangle$$

- **Convolutional kernel**:

$$K_h(x, x') = \sum_{u \in \Omega} \sum_{v, v'} h[u - v] h[u - v'] k(x_v, x'_{v'})$$

One-Layer Convolutional Kernels on Patches



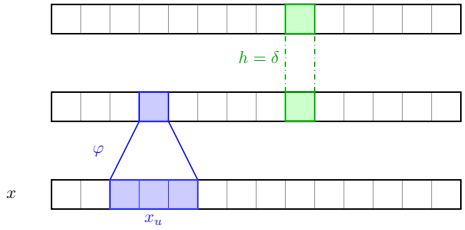
- **Patches**: $x_u \in \mathbb{R}^p$ at positions $u \in \Omega$
- **Features**: $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$, **patch kernel**: $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$
- **Convolutional network**: with global pooling ($h = 1/|\Omega|$)

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, |\Omega|^{-1} \sum_v \varphi(x_v) \rangle$$

- **Convolutional kernel**:

$$K_h(x, x') = |\Omega|^{-1} \sum_{v, v'} k(x_v, x'_{v'})$$

One-Layer Convolutional Kernels on Patches



- **Patches**: $x_u \in \mathbb{R}^p$ at positions $u \in \Omega$
- **Features**: $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$, **patch kernel**: $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$
- **Convolutional network**: with no pooling (Dirac $h = \delta$)

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, \varphi(x_u) \rangle$$

- **Convolutional kernel**:

$$K_h(x, x') = \sum_{u \in \Omega} k(x_u, x'_u)$$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

  (global pool) $K_g(x, x') = \sum_{v,v'} k(x_v, x'_{v'})$ vs (no pool) $K_\delta(x, x') = \sum_u k(x_u, x'_u)$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

  (global pool) $K_g(x, x') = \sum_{v,v'} k(x_v, x'_{v'})$ vs (no pool) $K_\delta(x, x') = \sum_u k(x_u, x'_u)$

Theorem (Generalization with one-layer (B., 2022))

Assume non-overlapping patches on $\mathbb{S}^{p-1}$. Learning with K_h yields

$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^\beta \quad \textit{vs} \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \lesssim \left(\frac{|\Omega|}{n}\right)^\beta$$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

  (global pool) $K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'})$ vs (no pool) $K_\delta(x, x') = \sum_u k(x_u, x'_u)$

Theorem (Generalization with one-layer (B., 2022))

Assume non-overlapping patches on $\mathbb{S}^{p-1}$. Learning with K_h yields

$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^{\beta} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \lesssim \left(\frac{|\Omega|}{n}\right)^{\beta}$$

- Rate β only depends on $p \ll d = p|\Omega|$ (**breaks the curse of dimensionality**!)

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

  (global pool) $K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'})$ vs (no pool) $K_\delta(x, x') = \sum_u k(x_u, x'_u)$

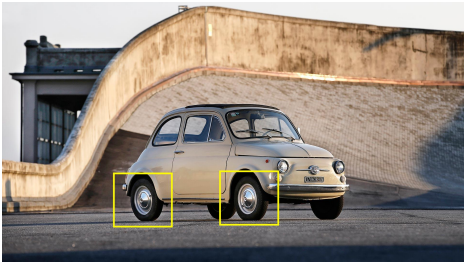
Theorem (Generalization with one-layer (B., 2022))

Assume non-overlapping patches on $\mathbb{S}^{p-1}$. Learning with K_h yields

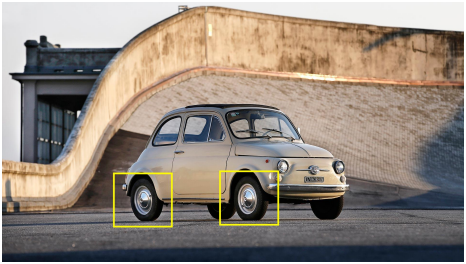
$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \lesssim \left(\frac{1}{n}\right)^{\beta} \quad \textit{vs} \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \lesssim \left(\frac{|\Omega|}{n}\right)^{\beta}$$

- Rate β only depends on $p \ll d = p|\Omega|$ (**breaks the curse of dimensionality**!)
- With local pooling, we can also learn $f^*(x) = \sum_{u \in \Omega} g^*_u(x_u)$ with different g^*_u

Long-Range Interactions

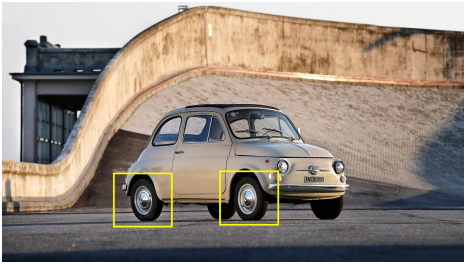


Long-Range Interactions



Q: How to capture interactions between multiple patches?

Long-Range Interactions

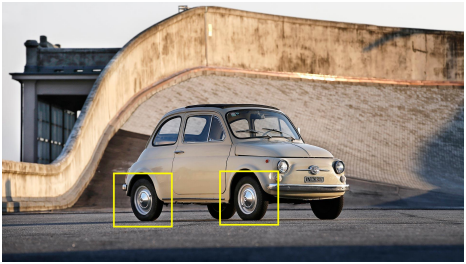


Q: How to capture interactions between multiple patches?

→ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

$$K(x, x') = \langle \varphi_2(\varphi_1(x)), \varphi_2(\varphi_1(x')) \rangle$$

Long-Range Interactions

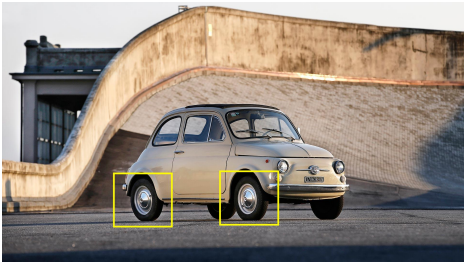


Q: How to capture interactions between multiple patches?

→ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle)$$

Long-Range Interactions



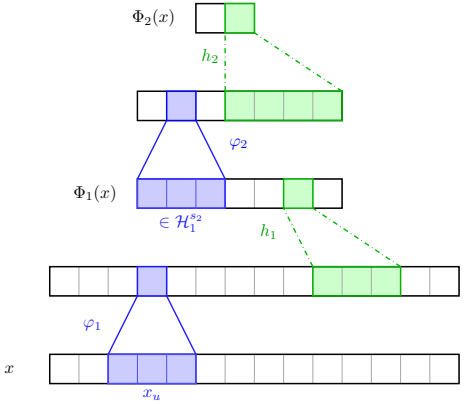
Q: How to capture interactions between multiple patches?

→ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

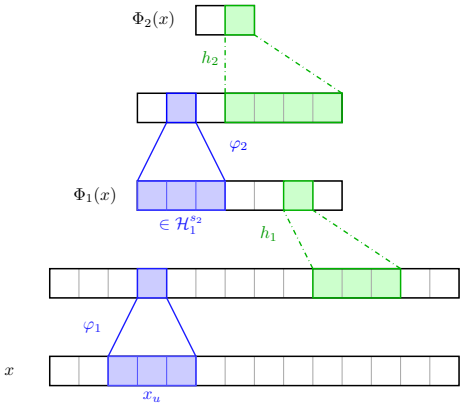
$$K(x, x') = \kappa_2(\kappa_1(\langle x, x' \rangle))$$

RKHS of Two-Layer Convolutional Kernels (B., 2022)

- φ_2/κ_2 captures **interactions** between patches



RKHS of Two-Layer Convolutional Kernels (B., 2022)

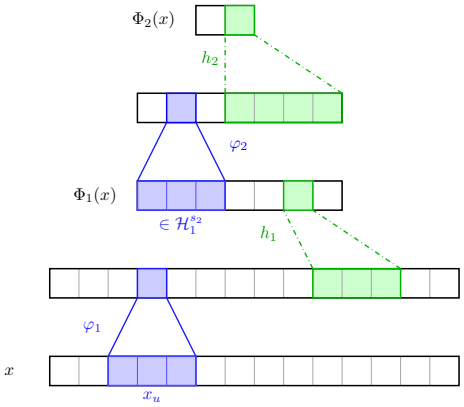


- φ_2/κ_2 captures **interactions** between patches
- Take $\kappa_2(u) = u^2$. RKHS contains

$$f(x) = \sum_{|u-v| \le r} g_{u,v}(x_u, x_v)$$

- Receptive field r depends on h_1 and s_2
- $g_{u,v} \in \mathcal{H}_1 \otimes \mathcal{H}_1$

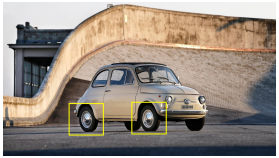
RKHS of Two-Layer Convolutional Kernels (B., 2022)



- φ_2/κ_2 captures **interactions** between patches
- Take $\kappa_2(u) = u^2$. RKHS contains

$$f(x) = \sum_{|u-v| \le r} g_{u,v}(x_u, x_v)$$

- Receptive field r depends on h_1 and s_2
- $g_{u,v} \in \mathcal{H}_1 \otimes \mathcal{H}_1$



- Pooling h_1: invariance to **relative** position
- Pooling h_2: invariance to **global** position

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix and run Kernel Ridge Regression.[1]

[1] `https://github.com/albietz/ckn_kernel`

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix and run Kernel Ridge Regression.[1]

2-layers, patch sizes (3, 5), Gaussian pooling factors (2,5).

κ_1	κ_2	Test acc.
Gauss	Gauss	88.3%
Gauss	Poly4	88.3%
Gauss	Poly3	88.2%
Gauss	Poly2	87.4%
Gauss	Linear	80.9%

[1] `https://github.com/albietz/ckn_kernel`

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix and run Kernel Ridge Regression.[1]

2-layers, patch sizes (3, 5), Gaussian pooling factors (2,5).

κ_1	κ_2	Test acc.
Gauss	Gauss	88.3%
Gauss	Poly4	88.3%
Gauss	Poly3	88.2%
Gauss	Poly2	87.4%
Gauss	Linear	80.9%

- **Polynomial kernels at second layer suffice!**

[1] `https://github.com/albietz/ckn_kernel`

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix and run Kernel Ridge Regression.[1]

2-layers, patch sizes (3, 5), Gaussian pooling factors (2,5).

κ_1	κ_2	Test acc.
Gauss	Gauss	88.3%
Gauss	Poly4	88.3%
Gauss	Poly3	88.2%
Gauss	Poly2	87.4%
Gauss	Linear	80.9%

- **Polynomial kernels at second layer suffice!**
- **State-of-the-art for kernels on Cifar10**
  - Shankar et al. (2020): 88.2% with 10 layers (90% with data augmentation)

[1] `https://github.com/albietz/ckn_kernel`

Generalization Benefits with Two Layers (B., 2022)

- Consider **invariant** $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$)

Generalization Benefits with Two Layers (B., 2022)

- Consider **invariant** $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$)

Generalization bounds (informal) when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$

h_1	h_2	s_2	$R(\hat{f}_n) - R(f^*)$
δ	δ	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2.5} / \sqrt{n}$
δ	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2} / \sqrt{n}$
global	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert / \sqrt{n}$
global	global or δ	1	$\lVert g^* \rVert / \sqrt{n}$

Generalization Benefits with Two Layers (B., 2022)

- Consider **invariant** $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$)

Generalization bounds (informal) when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$

h_1	h_2	s_2	$R(\hat{f}_n) - R(f^*)$
δ	δ	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2.5}/\sqrt{n}$
δ	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2}/\sqrt{n}$
global	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert/\sqrt{n}$
global	global or δ	1	$\lVert g^* \rVert/\sqrt{n}$

Polynomial gains in $|\Omega|$ when using the right architecture![2]

[2] Best $\approx$ deep sets (Zaheer et al., 2017)

Generalization Benefits with Two Layers (B., 2022)

- Consider **invariant** $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$)

Generalization bounds (informal) when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$

h_1	h_2	s_2	$R(\hat{f}_n) - R(f^*)$
δ	δ	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2.5}/\sqrt{n}$
δ	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert^{2}/\sqrt{n}$
global	global	$\lvert\Omega\rvert$	$\lVert g^* \rVert \lvert\Omega\rvert/\sqrt{n}$
global	global or δ	1	$\lVert g^* \rVert/\sqrt{n}$

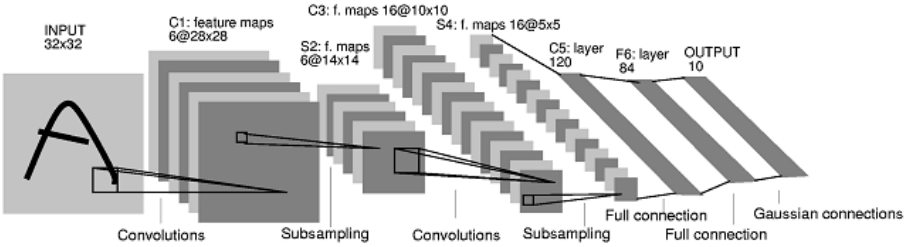
Polynomial gains in $|\Omega|$ when using the right architecture![2]

Q: Can more layers help with multi-scale structure?

[2] Best $\approx$ deep sets (Zaheer et al., 2017)

Outline

Beyond Translation Invariance



Convolutional architectures:

- Capture **multi-scale** structure in natural signals
- Provide near-**translation invariance**

Q: Beyond translation invariance using multi-scale structure?

Stability to Deformations

Deformations

- $\tau : \Omega \to \Omega$: smooth vector field
- $\tau \cdot x(u) = x(u - \tau(u))$: deformation operator
- Much richer group of transformations than translations

- Studied for wavelet-based scattering transform (Mallat, 2012; Bruna and Mallat, 2013)

Stability to Deformations

Deformations

- $\tau : \Omega \to \Omega$: smooth vector field
- $\tau \cdot x(u) = x(u - \tau(u))$: deformation operator
- Much richer group of transformations than translations

Definition of stability

- Representation $\Phi(\cdot)$ is **stable** (Mallat, 2012) if:

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq (\underbrace{C_{\text{stab}} \|\nabla \tau\|_\infty}_{\text{deformation stability}} + \underbrace{C_{\text{inv}} \|\tau\|_\infty}_{C_{\text{inv}} \to 0\text{: translation invariance}}) \|x\|$$

Stability to Deformations

Deformations

- $\tau : \Omega \to \Omega$: smooth vector field
- $\tau \cdot x(u) = x(u - \tau(u))$: deformation operator
- Much richer group of transformations than translations

Definition of stability

- Representation $\Phi(\cdot)$ is **stable** (Mallat, 2012) if:

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq (\underbrace{C_{\text{stab}} \|\nabla \tau\|_\infty}_{\text{deformation stability}} + \underbrace{C_{\text{inv}} \|\tau\|_\infty}_{C_{\text{inv}} \to 0\text{: translation invariance}}) \|x\|$$

Q: Can we achieve this along with approximation using kernels?

Stability with Kernels (B. and Mairal, 2019a)

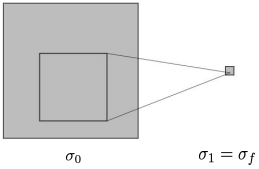
Geometry of the kernel mapping: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \cdot \|\Phi(x) - \Phi(x')\|_{\mathcal{H}}$$

- $\Phi(x)$ encodes CNN **architecture** independently of the model
- $\|f\|_{\mathcal{H}}$ controls **complexity** of the model

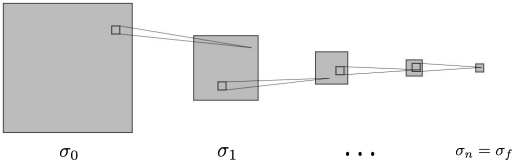
Deep vs Shallow Convolutional Models

Shallow



- $n = 1$
- $s \approx \sigma_f / \sigma_0$

Deep



- $s = O(1)$
- $n \approx \log(\sigma_f / \sigma_0) / \log s$

- n layers, multiple scales $\sigma_0, \ldots, \sigma_n = \sigma_f$
- Patch size $s \approx \sigma_k / \sigma_{k-1}$ = pooling/downsampling factor
  - Ensures **universal approximation** and **invariance**

Stability of Convolutional Kernels

Theorem (Stability of Convolutional Kernel (B. and Mairal, 2019a))

Let Φ be a n-layer conv kernel with scales $\sigma_0, \dots, \sigma_f$, and patch size s.

$$\|\Phi(\tau \cdot x) - \Phi(x)\|_{\mathcal{H}} \lesssim \left(n \cdot s^3 \|\nabla \tau\|_\infty + \frac{1}{\sigma_f} \|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large σ_f

Stability of Convolutional Kernels

Theorem (Stability of Convolutional Kernel (B. and Mairal, 2019a))

Let Φ be a n-layer conv kernel with scales $\sigma_0, \dots, \sigma_f$, and patch size s.

$$\|\Phi(\tau \cdot x) - \Phi(x)\|_{\mathcal{H}} \lesssim \left(n \cdot s^3 \|\nabla \tau\|_\infty + \frac{1}{\sigma_f} \|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large σ_f
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $s \approx \sigma_f / \sigma_0 \implies O((\sigma_f / \sigma_0)^3)$
  - Deep: $s = O(1)$, $n \approx \log(\sigma_f / \sigma_0) / \log s \implies O(\log(\sigma_f / \sigma_0))$

Stability of Convolutional Kernels

Theorem (Stability of Convolutional Kernel (B. and Mairal, 2019a))

Let Φ be a n-layer conv kernel with scales $\sigma_0, \dots, \sigma_f$, and patch size s.

$$\|\Phi(\tau \cdot x) - \Phi(x)\|_{\mathcal{H}} \lesssim \left(n \cdot s^3 \|\nabla \tau\|_\infty + \frac{1}{\sigma_f} \|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large σ_f
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $s \approx \sigma_f / \sigma_0 \implies O((\sigma_f/\sigma_0)^3)$
  - Deep: $s = O(1)$, $n \approx \log(\sigma_f/\sigma_0)/\log s \implies O(\log(\sigma_f/\sigma_0))$
- Extensions to other transformation groups (B. and Mairal, 2019a)
- Similar stability results hold for convolutional NTK (B. and Mairal, 2019b)

Stability of Convolutional Kernels

Theorem (Stability of Convolutional Kernel (B. and Mairal, 2019a))

Let Φ be a n-layer conv kernel with scales $\sigma_0, \dots, \sigma_f$, and patch size s.

$$\|\Phi(\tau \cdot x) - \Phi(x)\|_{\mathcal{H}} \lesssim \left(n \cdot s^3 \|\nabla \tau\|_\infty + \frac{1}{\sigma_f} \|\tau\|_\infty \right) \|x\|$$

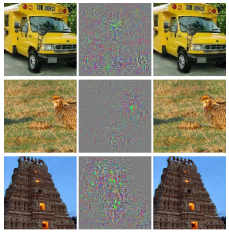
- Translation invariance: large σ_f
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $s \approx \sigma_f / \sigma_0 \implies O((\sigma_f/\sigma_0)^3)$
  - Deep: $s = O(1)$, $n \approx \log(\sigma_f/\sigma_0)/\log s \implies O(\log(\sigma_f/\sigma_0))$
- Extensions to other transformation groups (B. and Mairal, 2019a)
- Similar stability results hold for convolutional NTK (B. and Mairal, 2019b)

Q: What about stability of predictions?

Stability and Regularization in Deep Learning Practice

Two common issues with deep learning models:

- **Lack of robustness** to adversarial perturbations



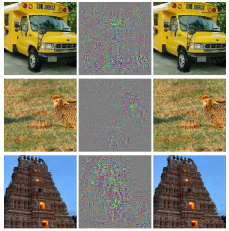
clean + noise → "**ostrich**"

(Szegedy et al., 2014)

Stability and Regularization in Deep Learning Practice

Two common issues with deep learning models:

- **Lack of robustness** to adversarial perturbations
- Poor performance on **small datasets**



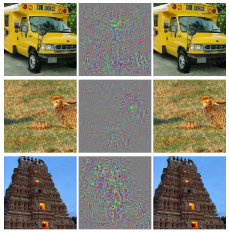
clean + noise → "**ostrich**"

(Szegedy et al., 2014)

Stability and Regularization in Deep Learning Practice

Two common issues with deep learning models:

- **Lack of robustness** to adversarial perturbations
- Poor performance on **small datasets**
- $\rightarrow$ **Implicit regularization is not enough!**



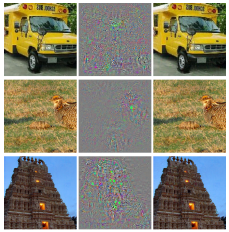
clean + noise $\rightarrow$ "**ostrich**"

(Szegedy et al., 2014)

Stability and Regularization in Deep Learning Practice

Two common issues with deep learning models:

- **Lack of robustness** to adversarial perturbations
- Poor performance on **small datasets**
- $\rightarrow$ **Implicit regularization is not enough!**



clean + noise $\rightarrow$ "**ostrich**"

(Szegedy et al., 2014)

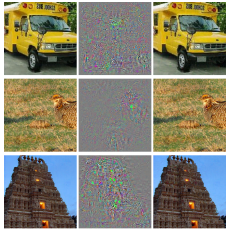
New approach to regularization (B., Mialon, Chen, and Mairal, 2019):

- View generic deep network f_θ as an element of a RKHS $\mathcal{H}$
- Regularize using (approximations of) RKHS norm $\|f_\theta\|_{\mathcal{H}}$

Stability and Regularization in Deep Learning Practice

Two common issues with deep learning models:

- **Lack of robustness** to adversarial perturbations
- Poor performance on **small datasets**
- $\rightarrow$ **Implicit regularization is not enough!**



clean + noise $\rightarrow$ "**ostrich**"

(Szegedy et al., 2014)

New approach to regularization (B., Mialon, Chen, and Mairal, 2019):

- View generic deep network f_θ as an element of a RKHS $\mathcal{H}$
- Regularize using (approximations of) RKHS norm $\|f_\theta\|_{\mathcal{H}}$
- **New practical regularizers**, outperform existing ones on:
  - Small data problems (vision, biology)
  - Adversarial robustness (**state-of-the-art** on Cifar10)

Outline

Concluding Remarks

Benefits of deep convolutional models

- Pooling improves generalization under invariance and stability
- Locality + depth + pooling capture structured interaction models with invariances
- Depth improves deformation stability in convolutional models

Concluding Remarks

Benefits of deep convolutional models

- Pooling improves generalization under invariance and stability
- Locality + depth + pooling capture structured interaction models with invariances
- Depth improves deformation stability in convolutional models

Kernels can help us understand deep learning architectures

Deep Learning Structure Beyond Kernels

Convolutional networks beyond kernels

- Limitations of kernels:
  - No feature selection (Ghorbani et al., 2019; Yehudai and Shamir, 2019)
  - No hierarchy (Allen-Zhu and Li, 2020; B. and Bach, 2021; Chen and Xu, 2021)

Deep Learning Structure Beyond Kernels

Convolutional networks beyond kernels

- Limitations of kernels:
  - No feature selection (Ghorbani et al., 2019; Yehudai and Shamir, 2019)
  - No hierarchy (Allen-Zhu and Li, 2020; B. and Bach, 2021; Chen and Xu, 2021)
- Feature learning in CNNs to learn **multi-scale hierarchical structure**
  - "Discover" filters at the first layer, interactions and invariances at following layers
  - Leverage recent tools (Chizat and Bach, 2018; Mei et al., 2019; Allen-Zhu and Li, 2020; Abbe et al., 2021)

Deep Learning Structure Beyond Kernels

Convolutional networks beyond kernels

- Limitations of kernels:
  - No feature selection (Ghorbani et al., 2019; Yehudai and Shamir, 2019)
  - No hierarchy (Allen-Zhu and Li, 2020; B. and Bach, 2021; Chen and Xu, 2021)
- Feature learning in CNNs to learn **multi-scale hierarchical structure**
  - "Discover" filters at the first layer, interactions and invariances at following layers
  - Leverage recent tools (Chizat and Bach, 2018; Mei et al., 2019; Allen-Zhu and Li, 2020; Abbe et al., 2021)

Generative models, representation learning

- How do DL algorithms discover structure without labels?
- Feature learning in energy-based models (Domingo-Enrich et al., 2021b,a)
- Role of transformations in contrastive self-supervised learning

Understanding Architectures

Graph Neural Networks

- Graph data are rich and varied (*e.g.*, social networks, chemistry, drug discovery, graphics)
- B. et al. (2020); Keriven et al. (2021): we show **stability** and **approximation** benefits for GNNs on large random graph models
- How do GNNs exploit such structure for **statistical** efficiency?

Understanding Architectures

Graph Neural Networks

- Graph data are rich and varied (*e.g.*, social networks, chemistry, drug discovery, graphics)
- B. et al. (2020); Keriven et al. (2021): we show **stability** and **approximation** benefits for GNNs on large random graph models
- How do GNNs exploit such structure for **statistical** efficiency?

Transformers

- What interaction structure do they model differently than CNNs?

Understanding Architectures

Graph Neural Networks

- Graph data are rich and varied (*e.g.*, social networks, chemistry, drug discovery, graphics)
- B. et al. (2020); Keriven et al. (2021): we show **stability** and **approximation** benefits for GNNs on large random graph models
- How do GNNs exploit such structure for **statistical** efficiency?

Transformers

- What interaction structure do they model differently than CNNs?

Interpretability

- Use insights from kernels to understand mechanisms behind trained networks?

Robust and Efficient Machine Learning Systems

Long term goal: make ML systems more robust and efficient

Robust and Efficient Machine Learning Systems

Long term goal: make ML systems more robust and efficient

- Work on understanding deep learning is crucial!

Robust and Efficient Machine Learning Systems

Long term goal: make ML systems more robust and efficient

- Work on understanding deep learning is crucial!
- Understanding and improving robustness using kernels
  - New empirical regularizers for adversarial robustness (B. et al., 2019)
  - Trade-offs with generalization (Dohmatob and B., 2022)
  - Role of data augmentation for robustness to distribution shifts

Robust and Efficient Machine Learning Systems

Long term goal: make ML systems more robust and efficient

- Work on understanding deep learning is crucial!
- Understanding and improving robustness using kernels
  - New empirical regularizers for adversarial robustness (B. et al., 2019)
  - Trade-offs with generalization (Dohmatob and B., 2022)
  - Role of data augmentation for robustness to distribution shifts
- Robustness and statistical/computational efficiency in exploration problems
  - (B. et al., 2021a; Zenati et al., 2020, 2022)

Robust and Efficient Machine Learning Systems

Long term goal: make ML systems more robust and efficient

- Work on understanding deep learning is crucial!
- Understanding and improving robustness using kernels
  - New empirical regularizers for adversarial robustness (B. et al., 2019)
  - Trade-offs with generalization (Dohmatob and B., 2022)
  - Role of data augmentation for robustness to distribution shifts
- Robustness and statistical/computational efficiency in exploration problems
  - (B. et al., 2021a; Zenati et al., 2020, 2022)
- Stochastic and private optimization
  - (B. et al., 2022; B. and Mairal, 2017)

Thank you!

References I

E. Abbe, E. Boix-Adsera, M. Brennan, G. Bresler, and D. Nagaraj. The staircase property: How hierarchical structure can guide deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Z. Allen-Zhu and Y. Li. Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*, 2020.

A. B. Approximation and learning with deep convolutional models: a kernel perspective. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

A. B. and F. Bach. Deep equals shallow for relu networks in kernel regimes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

A. B. and J. Mairal. Stochastic optimization with variance reduction for infinite datasets with finite sum structure. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

A. B. and J. Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research (JMLR)*, 20(25):1–49, 2019a.

A. B. and J. Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

A. B., G. Mialon, D. Chen, and J. Mairal. A kernel perspective for regularizing deep neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

References II

A. B., N. Keriven, and S. Vaiter. Convergence and stability of graph convolutional networks on large random graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A. B., A. Agarwal, and J. Langford. A contextual bandit bake-off. *Journal of Machine Learning Research (JMLR)*, 2021a.

A. B., L. Venturi, and J. Bruna. On the sample complexity of learning with geometric stability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

A. B., C. Wei, M. Dudik, J. Langford, and S. Wu. Personalization improves privacy-accuracy tradeoffs in federated optimization. *in submission*, 2022.

J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1872–1886, 2013.

L. Chen and S. Xu. Deep neural tangent kernel and laplace kernel have the same rkhs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

References III

E. Dohmatob and A. B. On the (non-)robustness of two-layer neural networks in different learning regimes. *in submission*, 2022.

C. Domingo-Enrich, A. B., M. Gabrié, J. Bruna, and E. Vanden-Eijnden. Dual training of energy-based models with overparametrized shallow neural networks. *arXiv preprint arXiv:2107.05134*, 2021a.

C. Domingo-Enrich, A. B., E. VE, and J. Bruna. On energy-based models with overparametrized shallow neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021b.

R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory (COLT)*, 2016.

B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

N. Keriven, A. B., and S. Vaiter. On the universality of graph neural networks on large random graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

References IV

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10): 1331–1398, 2012.

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, 2019.

H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

R. M. Neal. *Bayesian learning for neural networks*. Springer, 1996.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, L. Schmidt, J. Ragan-Kelley, and B. Recht. Neural kernels without tangents. *arXiv preprint arXiv:2003.02237*, 2020.

M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2): 742–769, 2018.

References V

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

G. Yehudai and O. Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

H. Zenati, A. B., M. Martin, E. Diemert, and J. Mairal. Counterfactual learning of stochastic policies with continuous actions: from models to offline evaluation. *arXiv preprint arXiv:2004.11722*, 2020.

H. Zenati, A. B., E. Diemert, J. Mairal, M. Martin, and P. Gaillard. Efficient kernelized UCB for contextual bandits. 2022.