# Benefits of (Deep) Convolutional Models: a Kernel Perspective

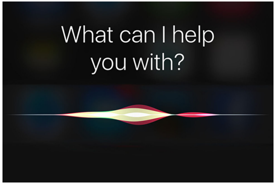
Alberto Bietti

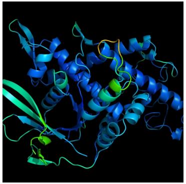
NYU

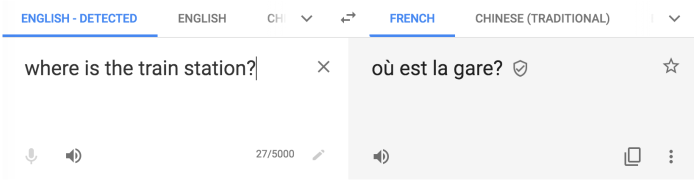
ML seminar, University of Minnesota. February 2, 2022.

# 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, ...)

$$f(x) = W_n \sigma(W_{n-1} \cdots \sigma(W_1 x) \cdots)$$

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

# Success of Deep Learning

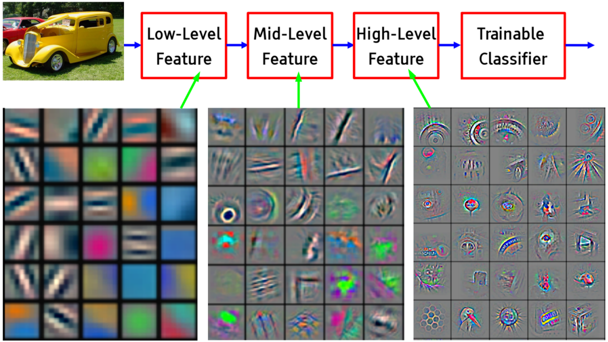
**State-of-the-art models** in various domains (images, speech, language, biology, ...)

$$f(x) = W_n \sigma(W_{n-1} \cdots \sigma(W_1 x) \cdots)$$

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

**Q: Why do they work?**

# Exploiting Data Structure through Architectures

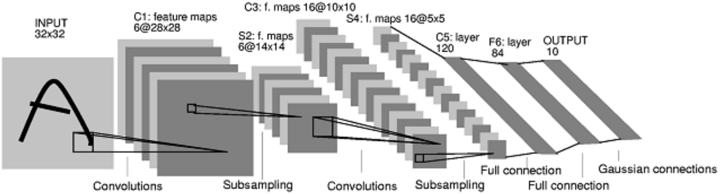



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

## Convolutional networks (CNNs)

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

# Exploiting Data Structure through Architectures

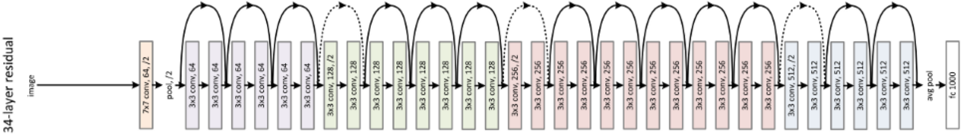



(LeCun et al., 1998)

**Convolutional networks (CNNs)**

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

# Exploiting Data Structure through Architectures




(He et al., 2016)

## Convolutional networks (CNNs)

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

# Understanding Deep Learning

## The challenge of deep learning theory

- **Over-parameterized** (millions 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 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

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

# Understanding Deep Learning

## The challenge of deep learning theory

- **Over-parameterized** (millions 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!

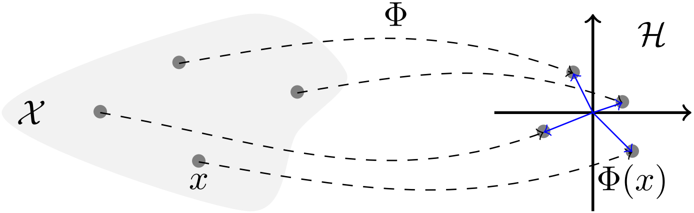
## A functional viewpoint

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

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

**Q: What is an appropriate functional space / norm $\Omega$?**

# 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

# Why Kernels?

## Clean and well-developed theory

- Tractable (convex) **optimization** algorithms
- **Statistical** and **approximation** properties are well understood for many kernels
  - *e.g.*, Sobolev spaces, interaction splines (Wahba, 1990; Caponnetto and De Vito, 2007)
- Related to over-parameterized networks in certain regimes (Neal, 1996; Jacot et al., 2018)

# Why Kernels?

## Clean and well-developed theory

- Tractable (convex) **optimization** algorithms
- **Statistical** and **approximation** properties are well understood for many kernels
  - *e.g.*, Sobolev spaces, interaction splines (Wahba, 1990; Caponnetto and De Vito, 2007)
- Related to over-parameterized networks in certain regimes (Neal, 1996; Jacot et al., 2018)
- Guarantees for *optimization*, *statistics*, and *approximation* **together** are rare in deep learning! *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 (convex) **optimization** algorithms
- **Statistical** and **approximation** properties are well understood for many kernels
  - *e.g.*, Sobolev spaces, interaction splines (Wahba, 1990; Caponnetto and De Vito, 2007)
- Related to over-parameterized networks in certain regimes (Neal, 1996; Jacot et al., 2018)
- Guarantees for *optimization*, *statistics*, and *approximation* **together** are rare in deep learning! *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

## This talk: kernels for convolutional models (B. and Mairal, 2019a,b; B. et al., 2021; B., 2022)

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

# Kernels for Deep Models: Infinite-Width Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(w_i^\top x), \qquad m \to \infty$$

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

$$K_{RF}(x, x') = \mathbb{E}_w[\rho(w^\top x)\rho(w^\top x')] = \kappa_\rho(x^\top x') \text{ when } x, x' \in \mathbb{S}^{d-1}$$

# Kernels for Deep Models: Infinite-Width Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(w_i^\top x), \qquad m \to \infty$$

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

$$K_{RF}(x, x') = \mathbb{E}_w[\rho(w^\top x)\rho(w^\top x')] = \kappa_\rho(x^\top x') \ \text{ when } x, x' \in \mathbb{S}^{d-1}$$

- **Neural Tangent Kernel** (NTK, Jacot et al., 2018): both $w_i$ and $v_i$ trained in linearized model near initialization $\theta_0 = (w_0, v_0)$ ("lazy training", Chizat et al., 2019)

$$K_{NTK}(x, x') = \mathbb{E}_{\theta_0}[\langle \nabla_\theta f(x), \nabla_\theta f(x') \rangle]$$

# Kernels for Deep Models: Infinite-Width Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(w_i^\top x), \qquad m \to \infty$$

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

$$K_{RF}(x, x') = \mathbb{E}_w[\rho(w^\top x)\rho(w^\top x')] = \kappa_\rho(x^\top x') \text{ when } x, x' \in \mathbb{S}^{d-1}$$

- **Neural Tangent Kernel** (NTK, Jacot et al., 2018): both $w_i$ and $v_i$ trained in linearized model near initialization $\theta_0 = (w_0, v_0)$ ("lazy training", Chizat et al., 2019)

$$K_{NTK}(x, x') = \mathbb{E}_{\theta_0}[\langle \nabla_\theta f(x), \nabla_\theta f(x') \rangle]$$

- RF and NTK extend to deep convolutional architectures (Arora et al., 2019; B. and Mairal, 2019b; Garriga-Alonso et al., 2019; Novak et al., 2019; Yang, 2019)

# Kernels for Deep Models: Hierarchical and Convolutional Kernels

**Hierarchical kernels** (Cho and Saul, 2009)

- Kernels can be constructed **hierarchically**

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

- *e.g.*, dot-product kernels on the sphere

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle) = \kappa_2(\kappa_1(x^\top x'))$$

- For $\kappa_\rho$, corresponds to infinite-width limit of deep *fully-connected* net

# Kernels for Deep Models: Hierarchical and Convolutional Kernels

**Hierarchical kernels** (Cho and Saul, 2009)

- Kernels can be constructed **hierarchically**

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \text{ with } \Phi(x) = \varphi_2(\varphi_1(x))$$

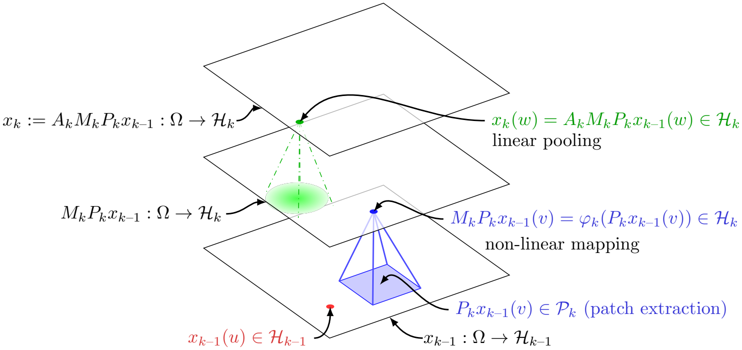
- *e.g.*, dot-product kernels on the sphere

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle) = \kappa_2(\kappa_1(x^\top x'))$$

- For $\kappa_\rho$, corresponds to infinite-width limit of deep *fully-connected* net
- B. and Bach (2021); Chen and Xu (2021): deep = shallow, same RKHS!
- $\implies$ More structure is needed

# Kernels for Deep Models: Hierarchical and Convolutional Kernels

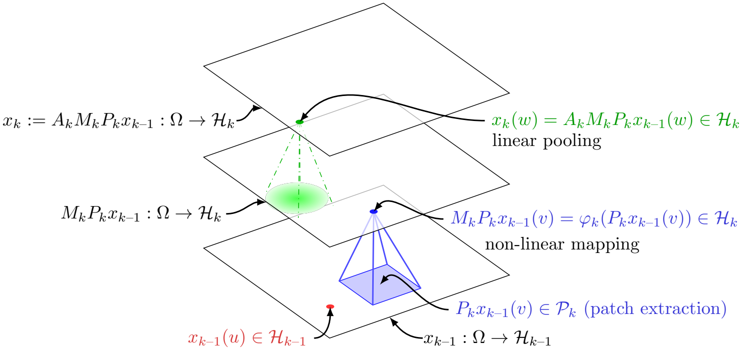
**Convolutional kernels** for images (Mairal et al., 2014; Mairal, 2016)




- Good performance on standard vision tasks (Mairal, 2016; Shankar et al., 2020; B., 2022)

# Kernels for Deep Models: Hierarchical and Convolutional Kernels

**Convolutional kernels** for images (Mairal et al., 2014; Mairal, 2016)




- Good performance on standard vision tasks (Mairal, 2016; Shankar et al., 2020; B., 2022)

**Q: What are the provable benefits of convolutional kernels?**

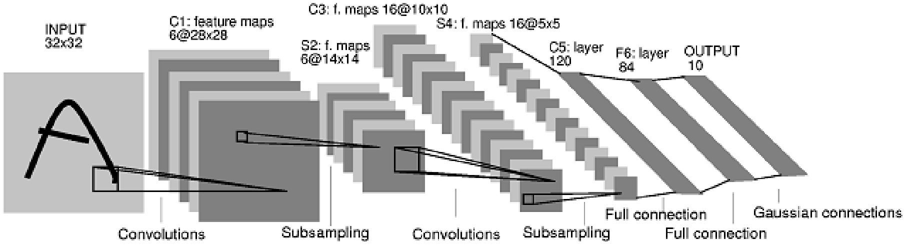
# Outline

1. Invariance and Stability to Deformations (B. and Mairal, 2019a,b)
2. Generalization Benefits under Invariance and Stability (B. et al., 2021)
3. Benefits of Locality and Depth (B., 2022)
4. Concluding Remarks

# Outline

1. Invariance and Stability to Deformations (B. and Mairal, 2019a,b)
2. Generalization Benefits under Invariance and Stability (B. et al., 2021)
3. Benefits of Locality and Depth (B., 2022)
4. Concluding Remarks

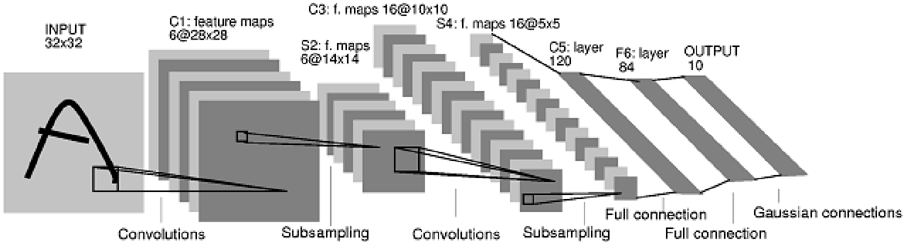
# Folklore Properties of Convolutional Models




**Convolutional architectures**:

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

# Folklore Properties of Convolutional Models




**Convolutional architectures**:

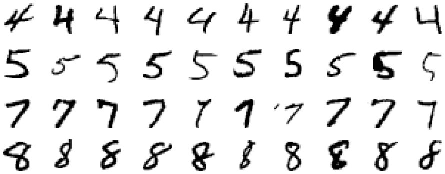
- Capture **multi-scale** structure in natural signals
- Provide some **translation invariance**

**Q: Beyond translation invariance?**

# 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 fixed 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:

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq (C_1 \|\nabla \tau\|_\infty + C_2 \|\tau\|_\infty) \|x\|$$

- $\|\nabla \tau\|_\infty = \sup_u \|\nabla \tau(u)\|$ controls deformation
- $\|\tau\|_\infty = \sup_u |\tau(u)|$ controls translation
- $C_2 \to 0$: translation invariance

# 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:

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq (C_1 \|\nabla \tau\|_\infty + C_2 \|\tau\|_\infty)\|x\|$$

- $\|\nabla \tau\|_\infty = \sup_u \|\nabla \tau(u)\|$ controls deformation
- $\|\tau\|_\infty = \sup_u |\tau(u)|$ controls translation
- $C_2 \to 0$: translation invariance

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

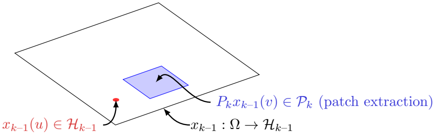
# Deformation Stability with Kernels (B. and Mairal, 2019a)

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

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

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

# Convolutional Kernel Construction (B. and Mairal, 2019a)

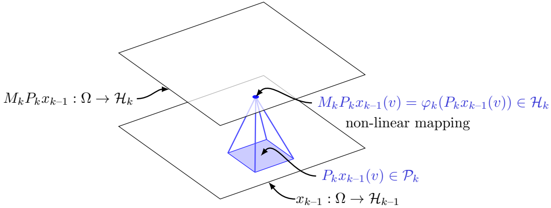



Continuous initial signal $x(u)$

At each layer $k$:

- $P_k$: Extract **patches** of size $|S_k|$

# Convolutional Kernel Construction (B. and Mairal, 2019a)

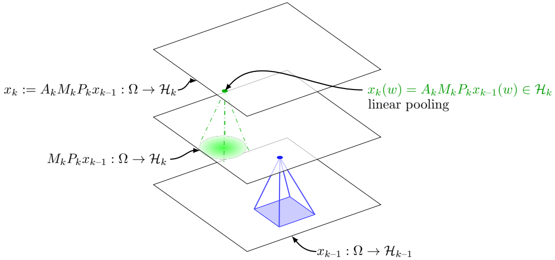



Continuous initial signal $x(u)$

At each layer $k$:

- $P_k$: Extract **patches** of size $|S_k|$
- $M_k$: Apply **kernel map** $\varphi_k$ to patches

# Convolutional Kernel Construction (B. and Mairal, 2019a)

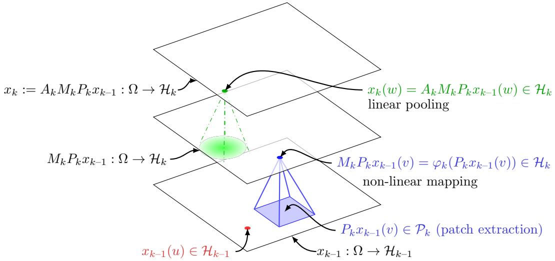



Continuous initial signal $x(u)$

At each layer $k$:

- $P_k$: Extract **patches** of size $|S_k|$
- $M_k$: Apply **kernel map** $\varphi_k$ to patches
- $A_k$: Gaussian **pooling** at scale $\sigma_k$

# Convolutional Kernel Construction (B. and Mairal, 2019a)

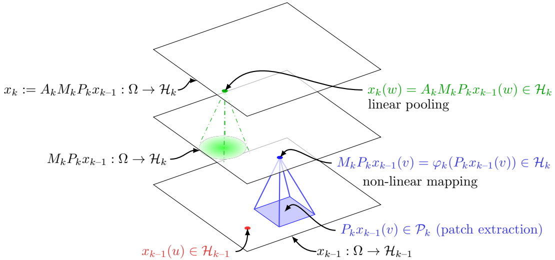



Continuous initial signal $x(u)$

At each layer $k$:

- $P_k$: Extract **patches** of size $|S_k|$
- $M_k$: Apply **kernel map** $\varphi_k$ to patches
- $A_k$: Gaussian **pooling** at scale $\sigma_k$

# Convolutional Kernel Construction (B. and Mairal, 2019a)




Continuous initial signal $x(u)$

At each layer $k$:

- $P_k$: Extract **patches** of size $|S_k|$
- $M_k$: Apply **kernel map** $\varphi_k$ to patches
- $A_k$: Gaussian **pooling** at scale $\sigma_k$

## Multi-layer construction

$$\Phi(x) = A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 x$$

- $|S_k|$, $\sigma_k$ typically exponential in $k$, fixed "**patch size**" $\beta := |S_k|/\sigma_{k-1}$
- In practice, discretize with **subsampling** $\leq$ **patch size** to preserve information

# Stability of Convolutional Kernels

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

*Let $\Phi$ be a n-layer conv kernel with initial* ***anti-aliasing*** *at $\sigma_0$. If $\|\nabla\tau\|_\infty \leq 1/2$,*

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq \left( C_1 \beta^3 (n+1) \|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n} \|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large $\sigma_n$
- Patch size: $\beta \approx \sigma_{k+1}/\sigma_k$
- Signal preservation/universal approximation: subsampling factor $\approx$ patch size
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $\beta \approx \sigma_n/\sigma_0 \implies O((\sigma_n/\sigma_0)^3)$
  - Deep: $\beta = O(1)$, $n \approx \log(\sigma_n/\sigma_0)/\log\beta \implies O(\log(\sigma_n/\sigma_0))$

# Stability of Convolutional Kernels

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

*Let $\Phi$ be a n-layer conv kernel with initial* ***anti-aliasing*** *at $\sigma_0$. If $\|\nabla\tau\|_\infty \leq 1/2$,*

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq \left( C_1 \beta^3 (n+1) \|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n}\|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large $\sigma_n$
- Patch size: $\beta \approx \sigma_{k+1}/\sigma_k$
- Signal preservation/universal approximation: subsampling factor $\approx$ patch size
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $\beta \approx \sigma_n/\sigma_0 \implies O((\sigma_n/\sigma_0)^3)$
  - Deep: $\beta = O(1)$, $n \approx \log(\sigma_n/\sigma_0)/\log\beta \implies O(\log(\sigma_n/\sigma_0))$
- Achieved by controlling operator norm of a **commutator** $[L_\tau, P_k A_{k-1}]$
  - Extend result by Mallat (2012) for controlling $\|[L_\tau, A]\|$, need $|S_k| \leq \beta\sigma_{k-1}$

# Stability of Convolutional Kernels

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

*Let $\Phi$ be a n-layer conv kernel with initial* ***anti-aliasing*** *at $\sigma_0$. If $\|\nabla\tau\|_\infty \leq 1/2$,*

$$\|\Phi(\tau \cdot x) - \Phi(x)\| \leq \left( C_1 \beta^3 (n+1) \|\nabla \tau\|_\infty + \frac{C_2}{\sigma_n} \|\tau\|_\infty \right) \|x\|$$

- Translation invariance: large $\sigma_n$
- Patch size: $\beta \approx \sigma_{k+1}/\sigma_k$
- Signal preservation/universal approximation: subsampling factor $\approx$ patch size
- **Exponential benefits of depth for stability**:
  - Shallow: $n = 1$, $\beta \approx \sigma_n/\sigma_0 \implies O((\sigma_n/\sigma_0)^3)$
  - Deep: $\beta = O(1)$, $n \approx \log(\sigma_n/\sigma_0)/\log \beta \implies O(\log(\sigma_n/\sigma_0))$
- Achieved by controlling operator norm of a **commutator** $[L_\tau, P_k A_{k-1}]$
  - Extend result by Mallat (2012) for controlling $\|[L_\tau, A]\|$, need $|S_k| \leq \beta\sigma_{k-1}$
- Extensions to other transformation groups, *e.g.*, roto-translations (B. and Mairal, 2019a)
- Similar stability results hold for convolutional NTK (B. and Mairal, 2019b)

# Outline

# Non-parametric Regression on the Sphere

## Problem setup

- **Goal**: bound on the excess risk $R(\hat{f}) - R(f^*) = \mathbb{E}_{x\sim\tau}[(\hat{f}(x) - f^*(x))^2]$
- $f^*(x) := \mathbb{E}[y|x]$ and $x \sim \tau$: uniform distribution on the sphere $\mathbb{S}^{d-1}$
- *Kernel ridge regression* estimator for some kernel $K$ on data $\{(x_i, y_i)\}_{i=1}^n$:

$$\hat{f}_{K,n} := \arg\min_{f\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda\|f\|^2_{\mathcal{H}_K}$$

# Non-parametric Regression on the Sphere

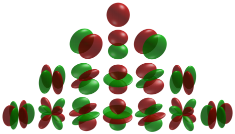
## Problem setup

- **Goal**: bound on the excess risk $R(\hat{f}) - R(f^*) = \mathbb{E}_{x\sim\tau}[(\hat{f}(x) - f^*(x))^2]$
- $f^*(x) := \mathbb{E}[y|x]$ and $x \sim \tau$: uniform distribution on the sphere $\mathbb{S}^{d-1}$
- *Kernel ridge regression* estimator for some kernel $K$ on data $\{(x_i, y_i)\}_{i=1}^n$:

$$\hat{f}_{K,n} := \arg\min_{f\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda\|f\|^2_{\mathcal{H}_K}$$

## Harmonic analysis on the sphere

- $L^2(\tau)$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree $k$, form a basis of $V_{d,k}$
- Diagonalizes dot-product kernels $K(x,x') = \kappa(\langle x, x'\rangle)$
- Assume $f^*$ is smooth $\leftrightarrow$ decay of coefficients of $f^*$



# Non-parametric Regression on the Sphere

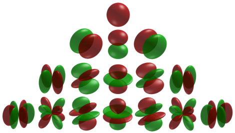
## Problem setup

- **Goal**: bound on the excess risk $R(\hat{f}) - R(f^*) = \mathbb{E}_{x\sim\tau}[(\hat{f}(x) - f^*(x))^2]$
- $f^*(x) := \mathbb{E}[y|x]$ and $x \sim \tau$: uniform distribution on the sphere $\mathbb{S}^{d-1}$
- *Kernel ridge regression* estimator for some kernel $K$ on data $\{(x_i, y_i)\}_{i=1}^n$:

$$\hat{f}_{K,n} := \arg\min_{f\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda\|f\|_{\mathcal{H}_K}^2$$

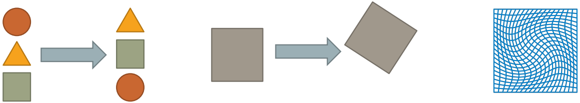
## Harmonic analysis on the sphere

- $L^2(\tau)$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree $k$, form a basis of $V_{d,k}$
- Diagonalizes dot-product kernels $K(x,x') = \kappa(\langle x, x'\rangle)$
- Assume $f^*$ is smooth $\leftrightarrow$ decay of coefficients of $f^*$



**Q: How can we encode invariance and stability?**

# Geometric Priors

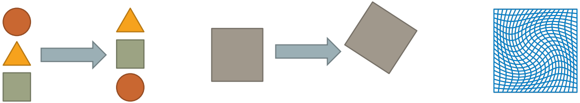


**Functions $f : \mathcal{X} \to \mathbb{R}$ that are "smooth" along known transformations of input $x$**

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

$$(\sigma \cdot x)[u] = x[\sigma^{-1}(u)]$$

# Geometric Priors



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

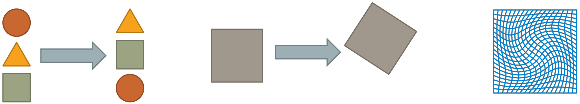
- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)[u] = x[\sigma^{-1}(u)]$$

**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 Priors



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

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

$$(\sigma \cdot x)[u] = x[\sigma^{-1}(u)]$$

**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

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

# Geometric Priors: Pooling Operator

$$S_G f(x) := \frac{1}{|G|} \sum_{\sigma \in G} f(\sigma \cdot x)$$

# Geometric Priors: Pooling Operator

$$S_G f(x) := \frac{1}{|G|} \sum_{\sigma \in G} f(\sigma \cdot x)$$

**Invariant spherical harmonics, when $G$ is a group** (Meyer, 1954; Mei et al., 2021)

- $S_G$ acts as a projection operator
- $\overline{N}(d, k)$ **invariant harmonics** of degree $k$, form a basis of $\overline{V}_{d,k} = S_G V_{d,k}$
- $f^*$ is $G$-invariant $\leftrightarrow f^* = S_G f^*$

# Geometric Priors: Pooling Operator

$$S_G f(x) := \frac{1}{|G|} \sum_{\sigma \in G} f(\sigma \cdot x)$$

**Invariant spherical harmonics, when $G$ is a group** (Meyer, 1954; Mei et al., 2021)

- $S_G$ acts as a projection operator
- $\overline{N}(d,k)$ **invariant harmonics** of degree $k$, form a basis of $\overline{V}_{d,k} = S_G V_{d,k}$
- $f^*$ is $G$-invariant $\leftrightarrow f^* = S_G f^*$

**Invariant kernels with pooling** (Haasdonk and Burkhardt, 2007; Mroueh et al., 2015)

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

- If $\kappa = \kappa_\rho$, corresponds to CNN with pooling $f(x) = \frac{1}{|G|} \sum_{\sigma \in G} \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \rho(\langle w_i, \sigma \cdot x \rangle)$

# Generalization Benefits of Pooling

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

**Theorem (Benefits of pooling (B., Venturi, and Bruna, 2021))**

*Assume $f^*$ is **invariant** to a group $G$ and **smooth** of order $s$.*
*Ridge regression with kernel $K_G$ vs $K$ achieves*

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \leq C_d \left( \frac{\nu_d(n)}{n} \right)^{\frac{2s}{2s+d-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \leq C_d \left( \frac{1}{n} \right)^{\frac{2s}{2s+d-1}},$$

*with $\nu_d(n) = \frac{1}{|G|} + o(1)$.*

# Generalization Benefits of Pooling

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

**Theorem (Benefits of pooling (B., Venturi, and Bruna, 2021))**

*Assume $f^*$ is **invariant** to a group $G$ and **smooth** of order $s$.*
*Ridge regression with kernel $K_G$ vs $K$ achieves*

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \leq C_d \left(\frac{\nu_d(n)}{n}\right)^{\frac{2s}{2s+d-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \leq C_d \left(\frac{1}{n}\right)^{\frac{2s}{2s+d-1}},$$

*with $\nu_d(n) = \frac{1}{|G|} + o(1)$.*

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

- $|G|$ can be exponential in $d$!
- Rate and constant $C_d$ are minimax optimal: curse of dimensionality

# Ingredients: Counting Invariant Harmonics

**Proposition ((B., Venturi, and Bruna, 2021))**

*As $k \to \infty$, we have*

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + O(k^{-d+\chi}),$$

*where $\chi$ is the maximal number of cycles of any permutation $\sigma \in G \setminus \{Id\}$.*

# Ingredients: Counting Invariant Harmonics

**Proposition ((B., Venturi, and Bruna, 2021))**

*As $k \to \infty$, we have*

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + O(k^{-d+\chi}),$$

*where $\chi$ is the maximal number of cycles of any permutation $\sigma \in G \setminus \{Id\}$.*

- Asymptotic rate of improvement can be quantified in terms of $\chi$
- Relies on singularity analysis of density of $\langle \sigma \cdot x, x \rangle$ (Saldanha and Tomei, 1996)

# Ingredients: Counting Invariant Harmonics

**Proposition ((B., Venturi, and Bruna, 2021))**

*As $k \to \infty$, we have*

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + O(k^{-d+\chi}),$$

*where $\chi$ is the maximal number of cycles of any permutation $\sigma \in G \setminus \{Id\}$.*

- Asymptotic rate of improvement can be quantified in terms of $\chi$
- Relies on singularity analysis of density of $\langle \sigma \cdot x, x \rangle$ (Saldanha and Tomei, 1996)
- Related to Mei et al. (2021), but different regimes
  - They study $d \to \infty$ with fixed $k$ ($\gamma_d(k) = \Theta_d(d^{-\alpha})$), gains at most polynomial in $d$
  - We study $k \to \infty$ with fixed $d$, gain $|G|$ can be exponential in $d$.

# Extension to Stability and Discussion

**Extension to geometric stability ($G$ is not a group)**

- Pooling operator $S_G$ is no longer a projection, has eigenvalues $\lambda_{k,j} \in [0, 1]$
- Different assumption: $f^* = S_G^r g$ for some $g$ and $r > 0$
- Leads to similar bounds with effective sample size $n|G|$ instead of $n$
- $|G|$ is exponential in $d$ for a simple model of deformations!

# Extension to Stability and Discussion

## Extension to geometric stability ($G$ is not a group)

- Pooling operator $S_G$ is no longer a projection, has eigenvalues $\lambda_{k,j} \in [0,1]$
- Different assumption: $f^* = S_G^r g$ for some $g$ and $r > 0$
- Leads to similar bounds with effective sample size $n|G|$ instead of $n$
- $|G|$ is exponential in $d$ for a simple model of deformations!

## Curse of dimensionality

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

$$R(\hat{f}) - f(f^*) \lesssim n^{-\frac{2}{2+d-1}}$$

# Extension to Stability and Discussion

## Extension to geometric stability ($G$ is not a group)

- Pooling operator $S_G$ is no longer a projection, has eigenvalues $\lambda_{k,j} \in [0, 1]$
- Different assumption: $f^* = S_G^r g$ for some $g$ and $r > 0$
- Leads to similar bounds with effective sample size $n|G|$ instead of $n$
- $|G|$ is exponential in $d$ for a simple model of deformations!

## Curse of dimensionality

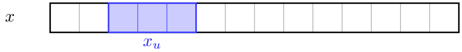
- If the target $f^*$ is non-smooth, *e.g.*, only Lipschitz, the rate is cursed! (and unimprovable)

$$R(\hat{f}) - f(f^*) \lesssim n^{-\frac{2}{2+d-1}}$$

**Q: How can we break this curse?**

# Outline

# One-layer Convolutional Kernels on Patches




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

## One-layer local convolutional kernel

- 1D signal $x[u]$, $u \in \Omega$, **localized** patches $x_u = (x[u], \ldots, x[u+s]) \in \mathbb{R}^p$
- RKHS $\mathcal{H}_K$ contains functions $f(x) = \sum_{u \in \Omega} g_u(x_u)$ with $g_u \in \mathcal{H}_k$

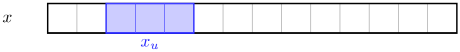
# One-layer Convolutional Kernels on Patches


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

## One-layer local convolutional kernel

- 1D signal $x[u]$, $u \in \Omega$, **localized** patches $x_u = (x[u], \dots, x[u+s]) \in \mathbb{R}^p$
- RKHS $\mathcal{H}_K$ contains functions $f(x) = \sum_{u \in \Omega} g_u(x_u)$ with $g_u \in \mathcal{H}_k$
- Pooling filter $h$ with $\sum_u h[u] = 1$
- **Pooling**: same functions, RKHS norm encourages similarities between the $g_u$

# One-layer Convolutional Kernels on Patches




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

## One-layer local convolutional kernel

- 1D signal $x[u]$, $u \in \Omega$, **localized** patches $x_u = (x[u], \dots, x[u+s]) \in \mathbb{R}^p$
- RKHS $\mathcal{H}_K$ contains functions $f(x) = \sum_{u \in \Omega} g_u(x_u)$ with $g_u \in \mathcal{H}_k$
- Pooling filter $h$ with $\sum_u h[u] = 1$
- **Pooling**: same functions, RKHS norm encourages similarities between the $g_u$
- Global pooling ($h[u] = 1/|\Omega|$): all the $g_u$ must be equal (*translation invariance*)

# Benefits of Locality and Pooling

- Assume non-overlapping patches $x_u$ uniform on the sphere $\mathbb{S}^{p-1}$
- Assume invariant target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- No pooling ($h = \delta$) vs global pooling ($h = \mathbf{1}$)

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

*Assume $g^*$ smooth of order $s$. Kernel ridge regression with $K_h$ yields*

$$\mathbb{E}\, R(\hat{f}_{\mathbf{1},n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

# Benefits of Locality and Pooling

- Assume non-overlapping patches $x_u$ uniform on the sphere $\mathbb{S}^{p-1}$
- Assume invariant target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- No pooling ($h = \delta$) vs global pooling ($h = \mathbf{1}$)

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

*Assume $g^*$ smooth of order $s$. Kernel ridge regression with $K_h$ yields*

$$\mathbb{E}\, R(\hat{f}_{\mathbf{1},n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

- **Breaks the curse** of dimensionality! $p$ instead of $d = p|\Omega|$ in the rate
- With localized pooling, we can also learn $f^*(x) = \sum_{u \in \Omega} g_u(x_u)$ with different $g_u$
- For overlapping patches see (Favero et al., 2021; Misiakiewicz and Mei, 2021)

# Benefits of Locality and Pooling

- Assume non-overlapping patches $x_u$ uniform on the sphere $\mathbb{S}^{p-1}$
- Assume invariant target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- No pooling ($h = \delta$) vs global pooling ($h = \mathbf{1}$)

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

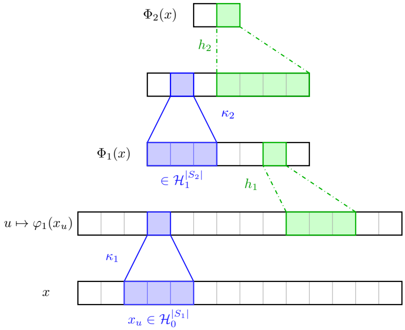
*Assume $g^*$ smooth of order $s$. Kernel ridge regression with $K_h$ yields*

$$\mathbb{E}\, R(\hat{f}_{\mathbf{1},n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

- **Breaks the curse** of dimensionality! $p$ instead of $d = p|\Omega|$ in the rate
- With localized pooling, we can also learn $f^*(x) = \sum_{u \in \Omega} g_u(x_u)$ with different $g_u$
- For overlapping patches see (Favero et al., 2021; Misiakiewicz and Mei, 2021)

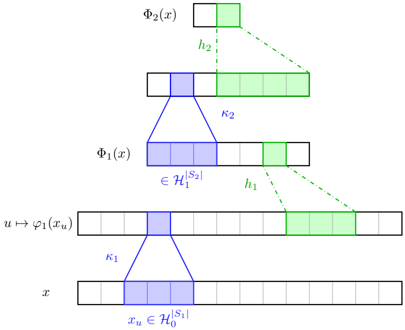
**Q: How can we capture long-range interactions?**

# Two-layer Convolutional Kernels




- Captures **interactions** between different patches

# Two-layer Convolutional Kernels




- Captures **interactions** between different patches
- If $\kappa_2(u) = u^2$, RKHS contains functions

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

- $g_{u,v} \in \mathcal{H}_k \otimes \mathcal{H}_k$
- Receptive field $r$ depends on $h_1$ and $S_2$

# Experiments on Cifar10

2-layers, 3x3 patches, pooling/downsampling sizes (2,5). Patch kernels $\kappa_1$, $\kappa_2$.

| $\kappa_1$ | $\kappa_2$ | Test acc. |
|---|---|---|
| Gauss | Gauss | 87.9% |
| Gauss | Poly3 | 87.7% |
| Gauss | Poly2 | 86.9% |
| Gauss | Poly1 (Linear) | 80.9% |

# Experiments on Cifar10

2-layers, 3x3 patches, pooling/downsampling sizes (2,5). Patch kernels $\kappa_1$, $\kappa_2$.

| $\kappa_1$ | $\kappa_2$ | Test acc. |
|---|---|---|
| Gauss | Gauss | 87.9% |
| Gauss | Poly3 | 87.7% |
| Gauss | Poly2 | 86.9% |
| Gauss | Poly1 (Linear) | 80.9% |

**Polynomial kernels suffice at second layer!**

# Experiments on Cifar10

2-layers, 3x3 patches, pooling/downsampling sizes (2,5). Patch kernels $\kappa_1$, $\kappa_2$.

| $\kappa_1$ | $\kappa_2$ | Test acc. |
|---|---|---|
| Gauss | Gauss | 87.9% |
| Gauss | Poly3 | 87.7% |
| Gauss | Poly2 | 86.9% |
| Gauss | Poly1 (Linear) | 80.9% |

**Polynomial kernels suffice at second layer!**

**Best performance** (B., 2022): **88.3%** (2 layers, larger patches at 2nd layer).

Shankar et al. (2020): 88.2% (10 layers). 90% with data augmentation ($\approx$ AlexNet)

# Generalization Benefits with Two Layers

- Consider $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\delta, \mathbf{1}\}$) and patch sizes ($|S_2|$):

# Generalization Benefits with Two Layers

- Consider $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\delta, \mathbf{1}\}$) and patch sizes ($|S_2|$):

**Generalization bound**: when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$, we have

| $h_1$ | $h_2$ | $\lvert S_2 \rvert$ | $R(\hat{f}_n) - R(f^*)$ (for $\epsilon \to 0$) |
|---|---|---|---|
| $\delta$ | $\delta$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert^{2.5}/\sqrt{n}$ |
| $\delta$ | $\mathbf{1}$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert^{2}/\sqrt{n}$ |
| $\mathbf{1}$ | $\mathbf{1}$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert/\sqrt{n}$ |
| $\mathbf{1}$ | $\delta$ or $\mathbf{1}$ | 1 | $\lVert g^* \rVert/\sqrt{n}$ |

# Generalization Benefits with Two Layers

- Consider $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\delta, \mathbf{1}\}$) and patch sizes ($|S_2|$):

**Generalization bound**: when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$, we have

| $h_1$ | $h_2$ | $\lvert S_2 \rvert$ | $R(\hat{f}_n) - R(f^*)$ (for $\epsilon \to 0$) |
|---|---|---|---|
| $\delta$ | $\delta$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert^{2.5}/\sqrt{n}$ |
| $\delta$ | $\mathbf{1}$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert^{2}/\sqrt{n}$ |
| $\mathbf{1}$ | $\mathbf{1}$ | $\lvert\Omega\rvert$ | $\lVert g^* \rVert \lvert\Omega\rvert/\sqrt{n}$ |
| $\mathbf{1}$ | $\delta$ or $\mathbf{1}$ | 1 | $\lVert g^* \rVert/\sqrt{n}$ |

**Polynomial gains in $|\Omega|$ when using the right architecture!**

# Outline

# Concluding Remarks

## Benefits of deep convolutional models

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

# Concluding Remarks

## Benefits of deep convolutional models

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

**Kernels can help us understand deep learning architectures**

# Concluding Remarks

## Benefits of deep convolutional models

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

**Kernels can help us understand deep learning architectures**

## Future directions

- Convolutional networks beyond kernels (data-adaptive filters, interaction terms)
  - *e.g.*, mean-field regimes (Chizat and Bach, 2018; Mei et al., 2019)
- Extensions to other architectures
  - *e.g.*, GNNs, Transformers
- Role of architecture beyond supervised learning
  - *e.g.*, generative models, self-supervised learning

# Concluding Remarks

## Benefits of deep convolutional models

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

**Kernels can help us understand deep learning architectures**

## Future directions

- Convolutional networks beyond kernels (data-adaptive filters, interaction terms)
  - *e.g.*, mean-field regimes (Chizat and Bach, 2018; Mei et al., 2019)
- Extensions to other architectures
  - *e.g.*, GNNs, Transformers
- Role of architecture beyond supervised learning
  - *e.g.*, generative models, self-supervised learning

**Thanks!**

# References I


S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

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

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

# References II


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.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

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

A. Favero, F. Cagnetta, and M. Wyart. Locality defeats the curse of dimensionality in convolutional teacher-student scenarios. *arXiv preprint arXiv:2106.08619*, 2021.

A. Garriga-Alonso, L. Aitchison, and C. E. Rasmussen. Deep convolutional networks as shallow gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

B. Haasdonk and H. Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine learning*, 68(1):35–61, 2007.

# References III


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.

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.

J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

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.

S. Mei, T. Misiakiewicz, and A. Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory (COLT)*, 2021.

# References IV


B. Meyer. On the symmetries of spherical harmonics. *Canadian Journal of Mathematics*, 6:135–157, 1954.

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

T. Misiakiewicz and S. Mei. Learning with convolution and pooling operations in kernel methods. In *Conference on Learning Theory (COLT)*, 2021.

Y. Mroueh, S. Voinea, and T. A. Poggio. Learning with group invariant features: A kernel perspective. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

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

R. Novak, L. Xiao, Y. Bahri, J. Lee, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

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

N. C. Saldanha and C. Tomei. The accumulated distribution of quadratic forms on the sphere. *Linear algebra and its applications*, 245:335–351, 1996.

# References V


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.

G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

G. Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.