Equivariant Transformers and the Computational Benefits of Equivariance in Neural Networks

Georg Bökman

KAW/WASP Postdoc @ University of Amsterdam
georg-bn.github.io

Munich Geometric Viewpoints Workshop (via the WWW), 2026-09-23

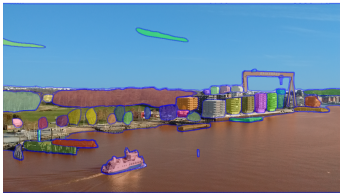
About me

- Master's in applied maths at Chalmers / TUM
- PhD in computer vision at Chalmers.
- *Postdoc in the Machine Learning Lab at UvA.*
- Broad interest: Geometric deep learning
  - Deep learning for geometry (3D reconstruction from images)
  - Geometry for deep learning (equivariant neural networks)

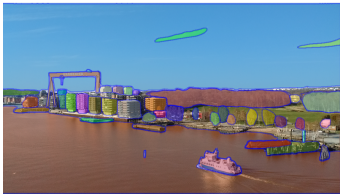
What is equivariance?

$$f(\tau x) = \tau f(x)$$









What is equivariance?

$$f(\tau x) = \tau f(x)$$

In some applications equivariance might be crucial in and of itself.

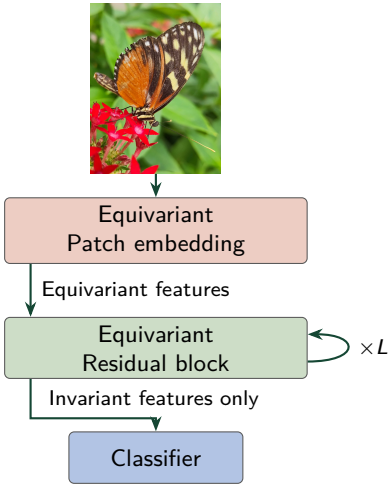
This talk will give an alternative motivation for equivariance, through computational reductions.

Main plan

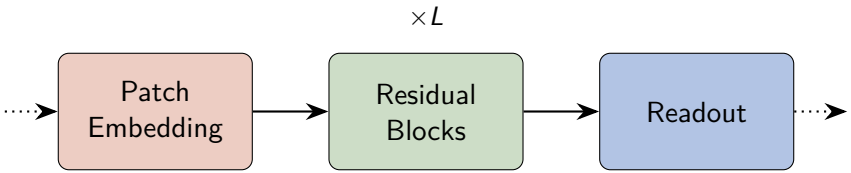
Take standard neural network architecture

Make minimal changes to make it equivariant

Parameterize all linear layers in Fourier domain to make it fast



Vision Transformer (ViT)

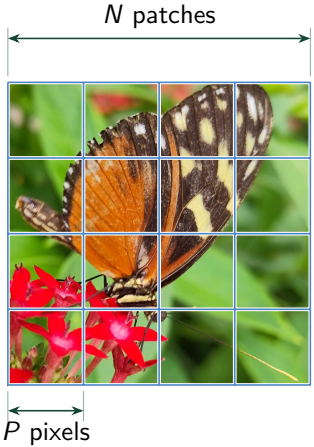


Patch embedding splits the image into patches (“visual tokens”).

Residual blocks alternate self-attention and token-wise MLPs.

Readout uses a class token or pooled tokens, followed by a prediction head.

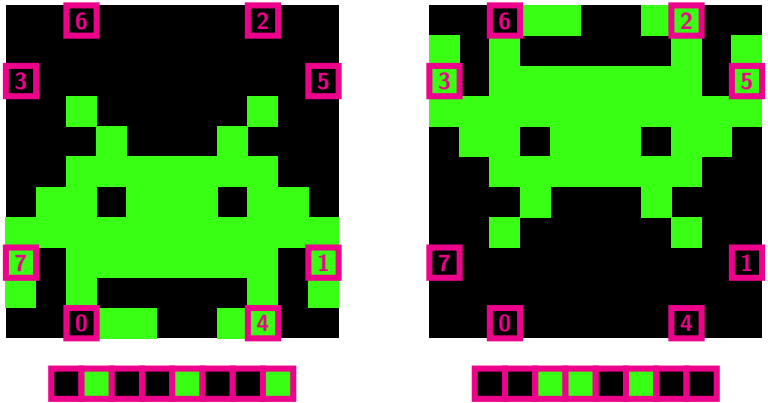
Patchification



An image is processed into N^2 patches/tokens.

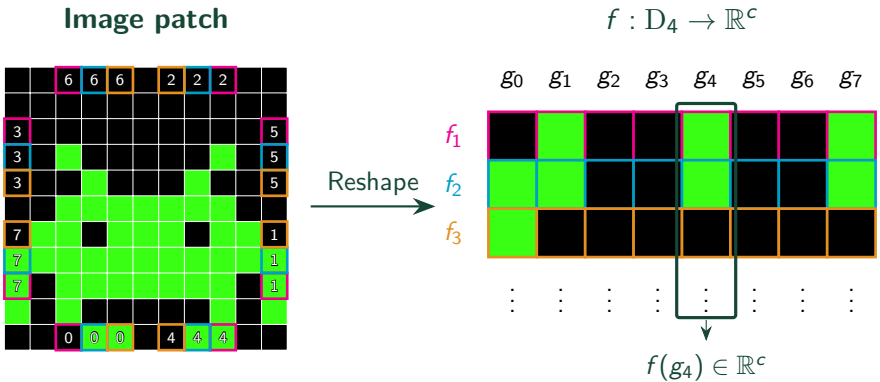
A token f is a c-channel signal over a finite group G, i.e. $f : G \to \mathbb{R}^c$, where $c = P^2/|G|$.

Image patches as signals over groups



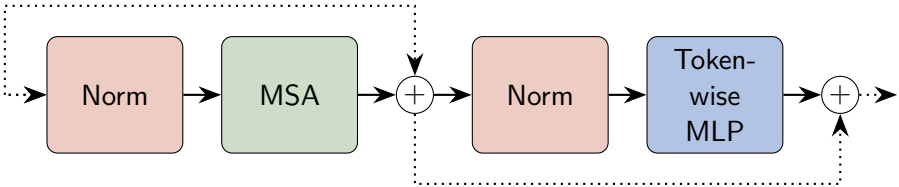
Magenta vector is a signal $f : \mathrm{D}_4 \to \mathbb{R}$

Reshaping a patch into a D_4 signal



Each eight-pixel orbit becomes a channel

Residual blocks



MSA contains linear projections to queries/keys/values and an "out projection".
MLP contains of two linear layers separated by GELU.

(Feature dimension $\gg$ number of tokens) $\Rightarrow$ Linear layers dominate compute.

Make all these tokenwise linear layers fast through equivariance!

Review of circular convolutions

Circular convolution

Convolution as matrix multiplication by circulant matrix

$$(k * f)(x) = \sum_{y=0}^{n-1} k(x-y) f(y)$$

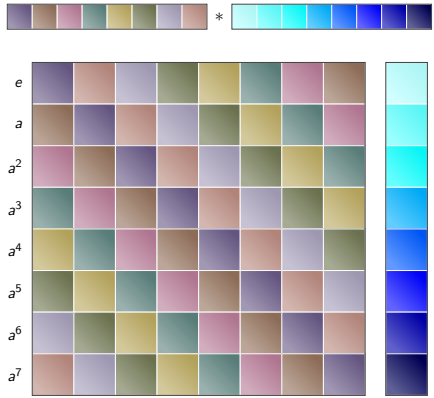
$$(\mathbf{M}_k)_{x,y} := k(x-y)$$

$$k * f = \mathbf{M}_k f$$

Translation equivariant linear maps = Convolutions

$$k * (f * \delta_t) = (k * f) * \delta_t$$

Example for $k, f : \mathbb{Z}_8 \to \mathbb{C}$



Discrete Fourier Transform

Convolution over $\mathbb{Z}_n$ is commutative: $f * k = k * f$. Since all $\mathbf{M}_k$ commute, they are simultaneously diagonalizable.

Diagonalizing basis is called the *Discrete Fourier Transform*.

$$\mathbf{M}_k f = \mathrm{IDFT} \cdot \mathbf{D}_k \cdot \mathrm{DFT} f$$

Discrete Fourier Transform

Convolution over $\mathbb{Z}_n$ is commutative: $f * k = k * f$. Since all $\mathbf{M}_k$ commute, they are simultaneously diagonalizable.

Diagonalizing basis is called the *Discrete Fourier Transform*.

$$\mathbf{M}_k f = \text{IDFT} \cdot \mathbf{D}_k \cdot \text{DFT} f$$

$\mathbf{M}_k$: n^2 multiplications

DFT: n^2, diagonal multiplication: n, IDFT: n^2

Discrete Fourier Transform

Convolution over $\mathbb{Z}_n$ is commutative: $f * k = k * f$. Since all $\mathbf{M}_k$ commute, they are simultaneously diagonalizable.

Diagonalizing basis is called the *Discrete Fourier Transform*.

$$\mathbf{M}_k f = \mathrm{IDFT} \cdot \mathbf{D}_k \cdot \mathrm{DFT} f$$

$\mathbf{M}_k$: n^2 multiplications

DFT: n^2, diagonal multiplication: n, IDFT: n^2

The DFT has exploitable structure — *Fast Fourier Transform*.
FFT/IFFT: $n \log n$,

Multi-channel to multi-channel circular convolution

$f : \mathbb{Z}_n \to \mathbb{C}^c, \quad K : \mathbb{Z}_n \to \mathbb{C}^{c \times c}, \quad (K * f) : \mathbb{Z}_n \to \mathbb{C}^c$

$\mathbf{M}_K \in \mathbb{C}^{nc \times nc}$

$$\mathbf{M}_K f = \mathrm{IDFT} \cdot \mathbf{D}_K \cdot \mathrm{DFT} f$$

c^2 parallel convolutions: $c^2 n^2$ multiplications.

Multi-channel to multi-channel circular convolution

$f : \mathbb{Z}_n \to \mathbb{C}^c, \quad K : \mathbb{Z}_n \to \mathbb{C}^{c \times c}, \quad (K * f) : \mathbb{Z}_n \to \mathbb{C}^c$

$\mathbf{M}_K \in \mathbb{C}^{nc \times nc}$

$$\mathbf{M}_K f = \text{IDFT} \cdot \mathbf{D}_K \cdot \text{DFT} f$$

c^2 parallel convolutions: $c^2 n^2$ multiplications.

c DFTs (without FFT): cn^2, c^2 diagonal multiplications: $c^2 n$, c IDFTs: cn^2
Total: $(c + 2n)cn$

Multi-channel to multi-channel circular convolution

$f : \mathbb{Z}_n \to \mathbb{C}^c, \quad K : \mathbb{Z}_n \to \mathbb{C}^{c \times c}, \quad (K * f) : \mathbb{Z}_n \to \mathbb{C}^c$

$\mathbf{M}_K \in \mathbb{C}^{nc \times nc}$

$$\mathbf{M}_K f = \text{IDFT} \cdot \mathbf{D}_K \cdot \text{DFT} f$$

c^2 parallel convolutions: $c^2 n^2$ multiplications.

c DFTs (without FFT): cn^2, c^2 diagonal multiplications: $c^2 n$, c IDFTs: cn^2
Total: $(c + 2n)cn$

Saving is $\frac{c^2 n^2}{(c+2n)cn} = \frac{cn}{c+2n} \xrightarrow[c \to \infty]{} n$ times

Multi-channel to multi-channel circular convolution

$f : \mathbb{Z}_n \to \mathbb{C}^c, \quad K : \mathbb{Z}_n \to \mathbb{C}^{c \times c}, \quad (K * f) : \mathbb{Z}_n \to \mathbb{C}^c$

$\mathbf{M}_K \in \mathbb{C}^{nc \times nc}$

$$\mathbf{M}_K f = \text{IDFT} \cdot \mathbf{D}_K \cdot \text{DFT} f$$

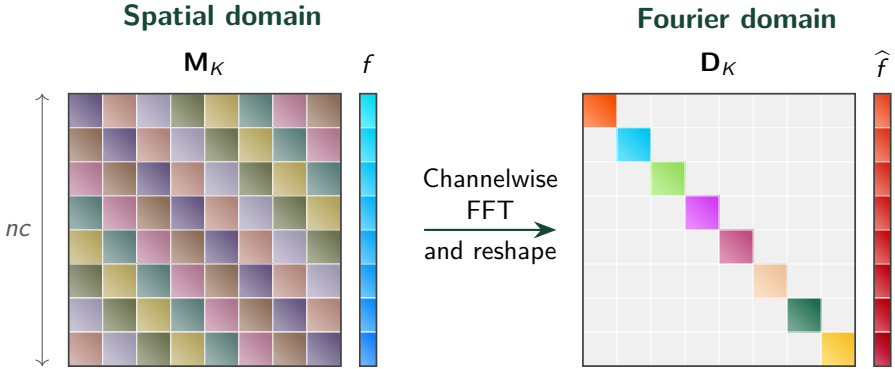
c^2 parallel convolutions: $c^2 n^2$ multiplications.

c DFTs (without FFT): cn^2, c^2 diagonal multiplications: $c^2 n$, c IDFTs: cn^2
Total: $(c + 2n)cn$

Saving is $\frac{c^2 n^2}{(c+2n)cn} = \frac{cn}{c+2n} \xrightarrow[c \to \infty]{} n$ times

(with FFT: $\frac{cn}{c+2\log n} \to n$)

Multi-channel convolution in the Fourier domain



Shared $c \times c$ kernel blocks

One $c \times c$ block per frequency

Finite group convolutions

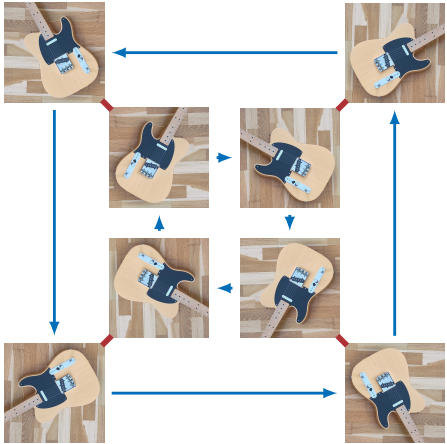
Example: Dihedral group

Cyclic group $\mathbb{Z}_n$

$\rightsquigarrow$

Arbitrary finite group G with n elements

Example: Dihedral group D_4 of reflections and 90° rotations



Arbitrary finite groups

Convolution as matrix multiplication by G-circulant matrix

$$(k * f)(x) = \sum_{y \in G} k(y^{-1}x) f(y)$$

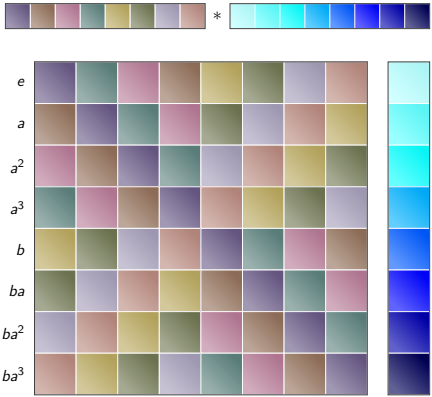
$$(\mathbf{M}_k)_{x,y} = k(y^{-1}x)$$

$$k * f = \mathbf{M}_k f$$

G-equivariant linear maps = Convolutions

$$k * (f * \delta_t) = (k * f) * \delta_t$$

Example for $k, f : \mathrm{D}_4 \to \mathbb{R}$



Arbitrary finite groups

If G is not commutative, then convolutions do not commute. Convolutions do not simultaneously diagonalize in this case!

Arbitrary finite groups

If G is not commutative, then convolutions do not commute. Convolutions do not simultaneously diagonalize in this case!

However, Schur's lemma gives simultaneous *block-diagonalization* of all convolutions, with block-structure depending on G.

$$k * f = \mathbf{M}_k f = \mathrm{IDFT} \cdot \mathbf{B}_k \cdot \mathrm{DFT} f$$

Arbitrary finite groups

Multi-channel case:

$$K * f = \mathrm{IDFT} \cdot \mathbf{B}_K \cdot \mathrm{DFT} f$$

c^2 block-diagonals: $(c^2|G|^2)/b_G$ multiplications, $b_G > 1$

Arbitrary finite groups

Multi-channel case:

$$K * f = \mathrm{IDFT} \cdot \mathbf{B}_K \cdot \mathrm{DFT} f$$

c^2 block-diagonals: $(c^2|G|^2)/b_G$ multiplications, $b_G > 1$

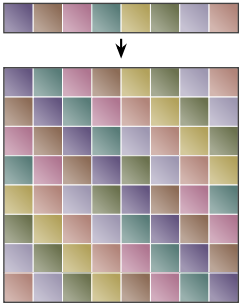
Savings as $c \to \infty$:

$$\frac{c^2|G|^2}{(c/b_G+2)c|G|^2} \to b_G$$

Example: Dihedral group

Regular

$H = \{e\}$

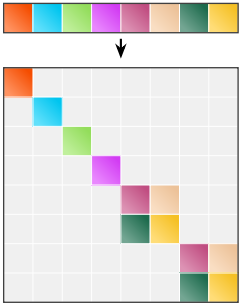


$b_G = 16/3 \approx 5.33$

81% sparsity

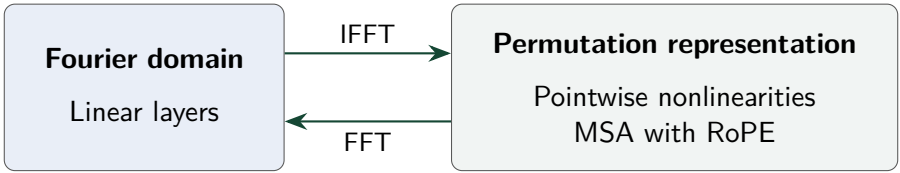
Fourier

$H = G$



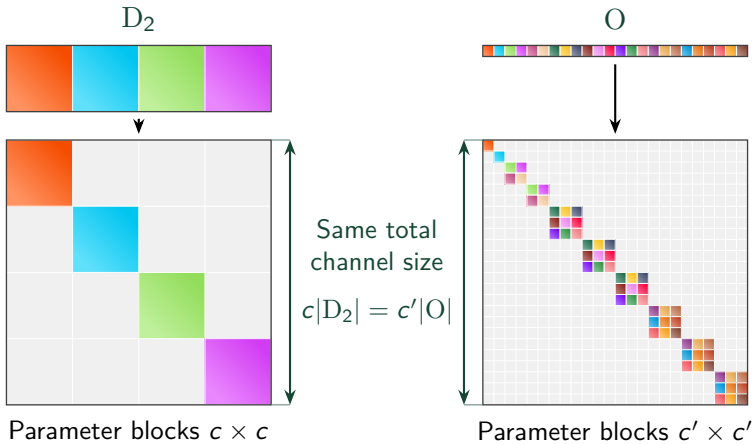
The structure of the blocks depends on the group.

Nonlinear layers in transformers need permutation representations

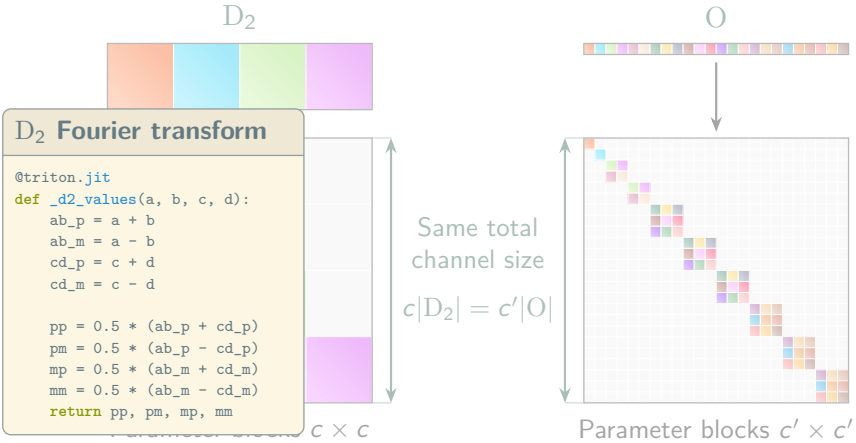


Overhead: FFT/IFFT and tensor reshaping.

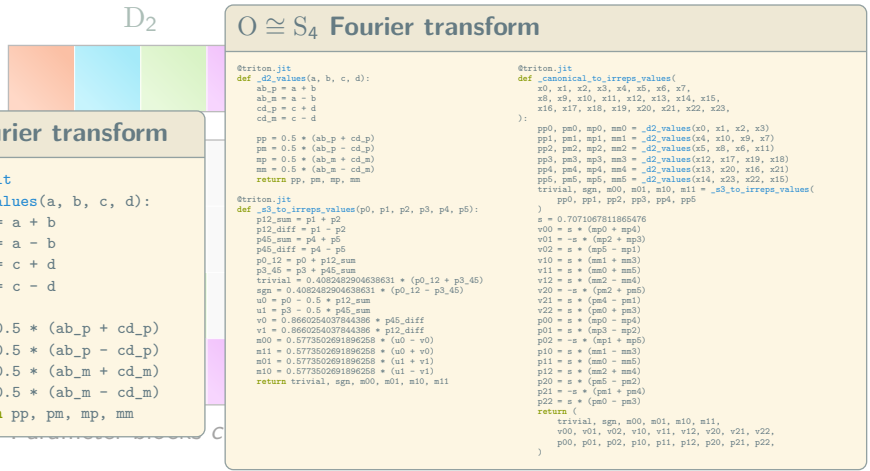
Fourier domain: D_2 and $\mathrm{O} \cong \mathrm{S}_4$



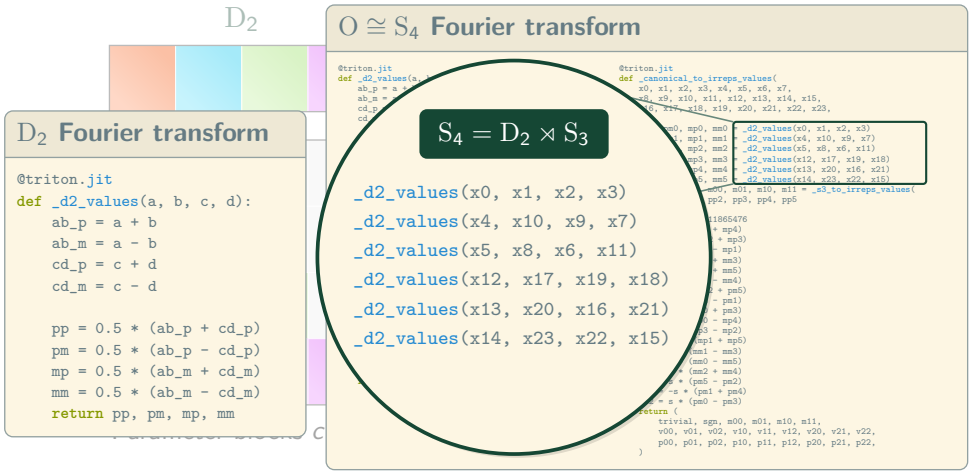
Fourier domain: D_2 and $\mathrm{O} \cong \mathrm{S}_4$



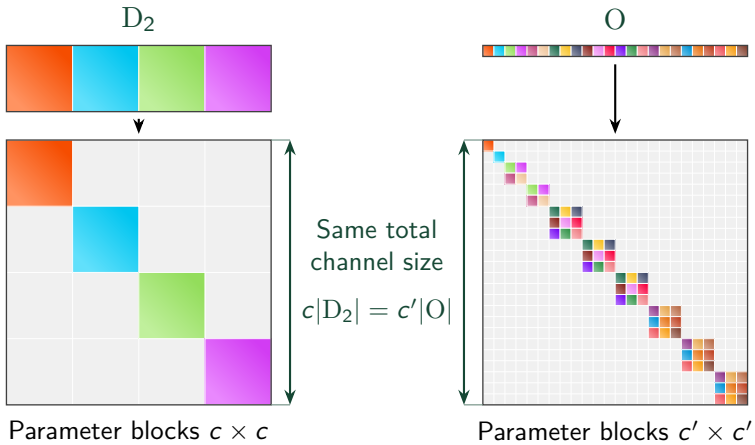
Fourier domain: D_2 and $O \cong S_4$



Fourier domain: D_2 and $\mathrm{O} \cong \mathrm{S}_4$



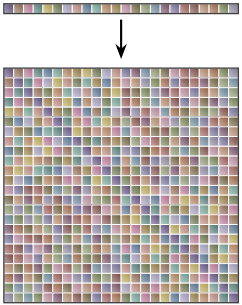
Fourier domain: D_2 and $\mathrm{O} \cong \mathrm{S}_4$



Lemon parameterization of O

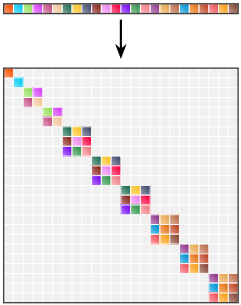
Regular

$H = \{e\}$



Fourier

$H = G$

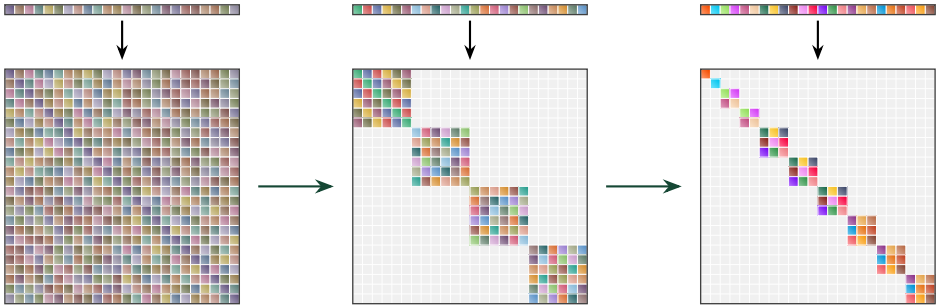


Lemon parameterization of O

Regular
$H = \{e\}$

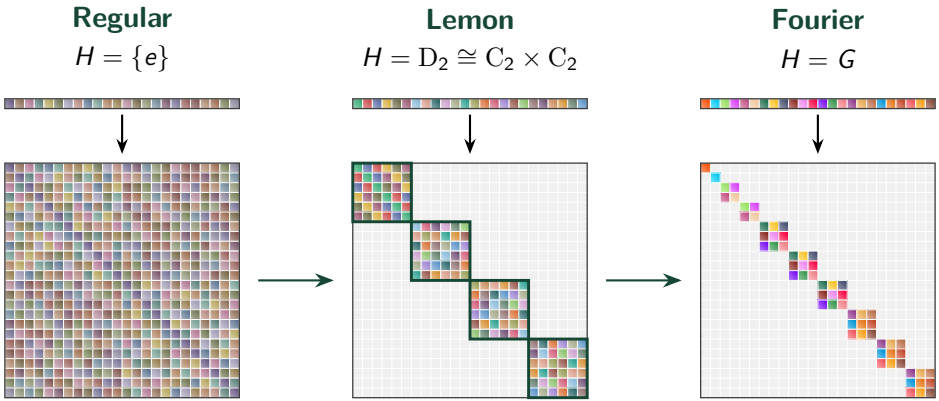
Lemon
$H = \mathrm{D}_2 \cong \mathrm{C}_2 \times \mathrm{C}_2$

Fourier
$H = G$



G-equivariance $\Rightarrow$ H-equivariance

Lemon parameterization of O



G-equivariance $\Rightarrow$ H-equivariance

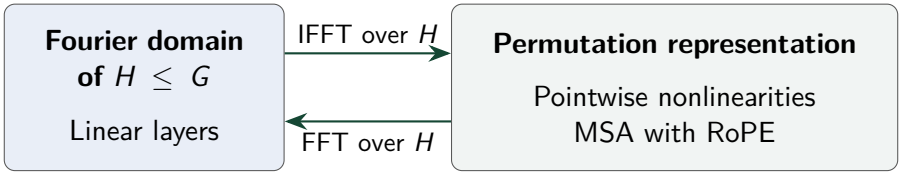
H-equivariance + weight-tying $\Rightarrow$ G-equivariance

Lemon parameterizations

Tradeoffs

- Fewer FLOPs vs. larger, more efficient matmuls
- Simpler block structure, easier implementation
- Smaller transforms, faster FFTs

Nonlinear layers in transformers need permutation representations

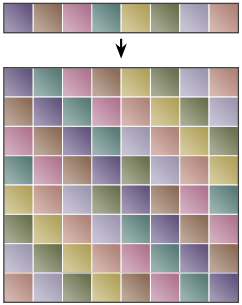


Smaller $H \Rightarrow$ less FFT/IFFT and reshaping overhead, more FLOPs in matmuls.

Lemon parameterizations of D_4

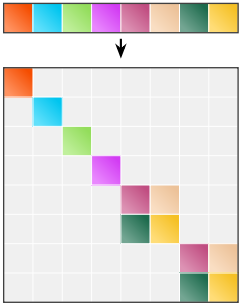
Regular

$H = \{e\}$

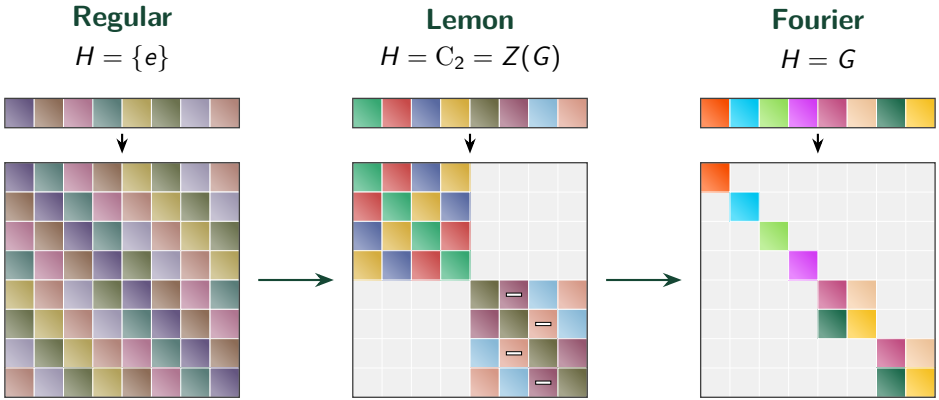


Fourier

$H = G$

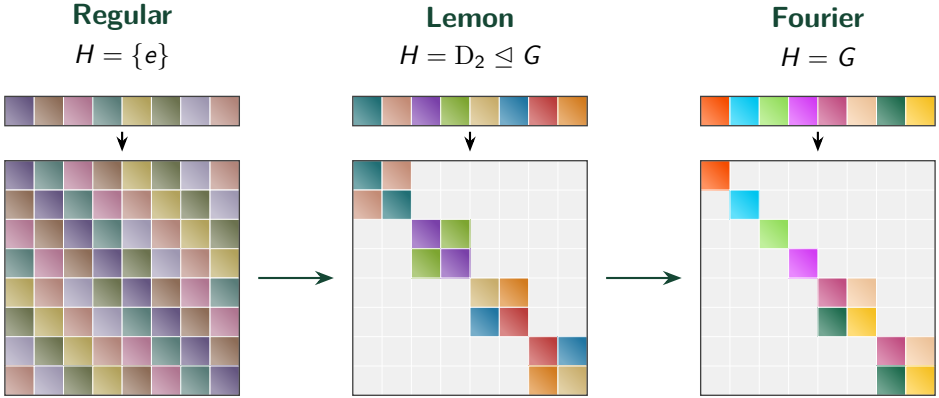


Lemon parameterizations of D_4



Central subgroup: no shared weights across blocks

Lemon parameterizations of D_4

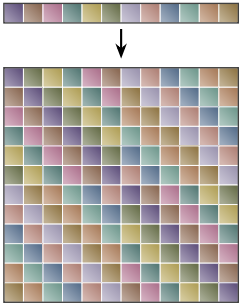


H has a complement: no minus signs

Lemon parameterizations of D_6

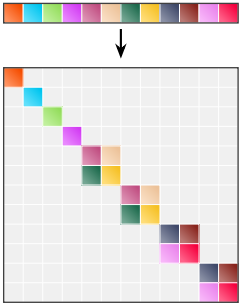
Regular

$H = \{e\}$

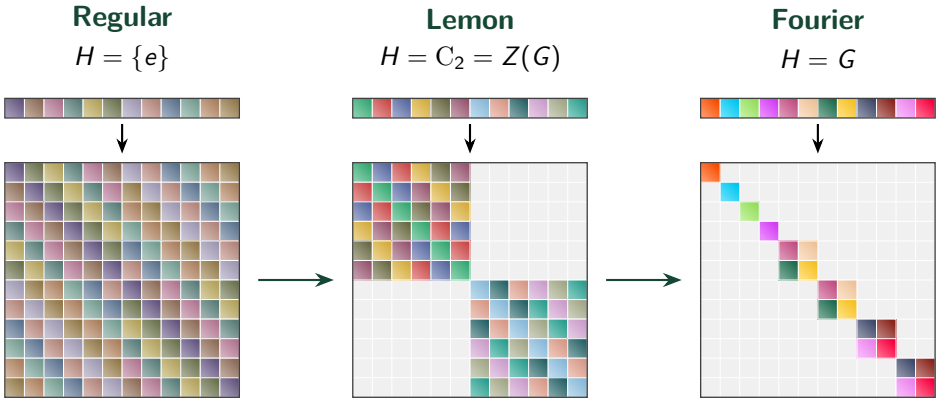


Fourier

$H = G$



Lemon parameterizations of D_6



Central subgroup with a complement

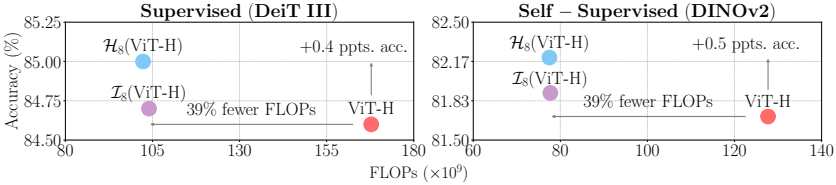
Reality check

At scale ($c \to \infty$), a well-optimized full Fourier implementation will always beat lemon (i.e. Fourier over $H \leq G$).

Ongoing work

Implementations and experiments

D_4 on Images (*Quick ViTs*)



Model	Width	Depth	MLP dim.	Heads	Speedup (bf16)	FLOPs	Peak mem.
ViT-L	1024	24	4096	16	1.32×	1/4.58×	1/2.44×
ViT-H	1280	32	5120	16	1.47×	1/4.58×	1/2.88×
ViT-G	1664	48	8192	16	1.91×	1/4.88×	1/4.36×
ViT-e	1792	56	15360	16	2.37×	1/5.01×	1/5.30×
ViT-22B	6144	36	24576	48	3.54×	1/5.18×	1/5.80×

Nordström, Edstedt, Kahl & Bökman; Quick ViTs: Speeding up Vision Transformers through Equivariance; ECCV 2026

Tetrahedral rotations on molecules (*Platonic Transformers*)

OMol25 4M: validation MAE (↓)

Model	Force	Energy	E/Atom
Tetrahedron	**9.33**	**63.84**	**1.12**
eSEN-sm	12.64	107.78	1.66

Training time per epoch: 20k molecules (seconds)

Hidden dimension	576	864	1152	1440	1728	2016
Spatial (standard)	18	23	29	40	49	63
Fourier	19	22	27	32	38	45

Islam, Anand, Wessels, de Kruiff, Kuipers, Ying, Sánchez, Vadgama, Bökman & Bekkers;
Platonic Transformers: A Solid Choice For Equivariance; ICML 2026

Why are throughput improvements not as large as FLOP improvements?

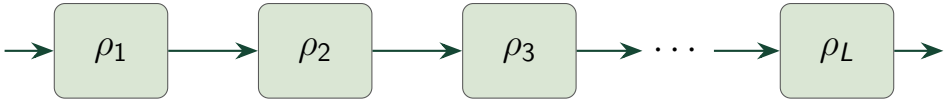
Transforming back and forth from Fourier domain incurs reshape-overheads.

Our implementations are not great (actually, very recent implementations are quite good).

Ongoing work

Main open question in equivariant deep learning

Every layer of the network satisfies equivariance constraints with respect to some representation ρ_i.



We've had $\rho_i =$ multiple copies of the regular representation at every layer i.

How do we choose each ρ_i optimally?

Bökman, Nordström & Kahl; Flopping for FLOPs; ICML 2025.
Shahverdi, Marchetti, Bökman & Kohn; Identifiable Equivariant Nets are Layerwise Equivariant; ICML 2026.
Ông & Bökman; A Unified Framework for Equivariant ViTs; 2606.27864.
Ông; No Equivariant Architecture Covers All Equivariant Attention; 2608.30417.

Main takeaways

- Representation theory of finite groups gives computational boost to state-of-the-art neural networks.
- FLOP reductions are straightforward.
- Throughput improvements require more work to reach full potential.
- Unclear when and why a specific choice of group representation is best in a given layer.

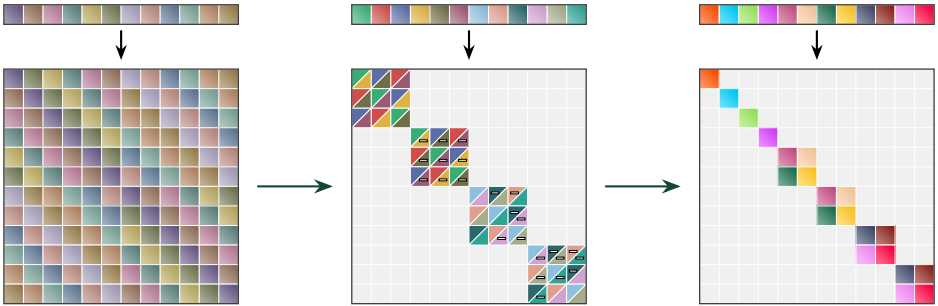
Bonus

Normal core: D_6 with $H = \mathrm{D}_2$

Regular
$H = \{e\}$

Lemon
$H = \mathrm{D}_2$

Fourier
$H = G$



$N = \mathrm{core}_G(H) = \mathrm{C}_2$

Signed sums of two parameter blocks