Hellenic Robotics Forum
(HRF) 2026

Unlocking the Matrix Form of the Quaternion Fourier Transform and Quaternion Convolution: Properties, connections, and application to Lipschitz constant bounding

Giorgos Sfikas gsfikas@uniwa.gr
Department of Surveying and Geoinformatics, School of Engineering
University of West Attica
Athens, Greece

George Retsinas gretsinas@central.ntua.gr
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece

Originally published in Transactions of Machine Learning Research (TMLR), November 2025

Presentation in Hellenic Robotics Forum (HRF), July 2026

Quaternions..

...are a representational unit that is used extensively in fields that include:

Quaternions..

...are a representational unit that is used extensively in fields that include:

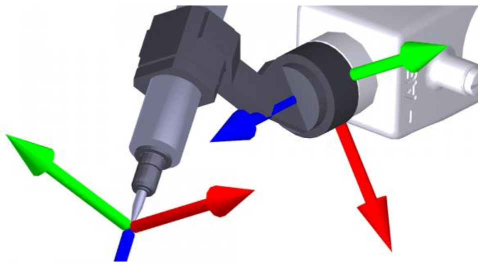
- 🎨 Graphics (e.g. Gaussian Splatting - Kerbl et al., “3D Gaussian Splatting for Real-Time Radiance Field Rendering”, ACM SIGGRAPH 2023)

Quaternions..

...are a representational unit that is used extensively in fields that include:

- 🎨 Graphics (e.g. Gaussian Splatting - Kerbl et al., "3D Gaussian Splatting for Real-Time Radiance Field Rendering", ACM SIGGRAPH 2023)

- 🤖 3D Rotation estimation & Robotics (e.g. Liu et al., "Delving into Discrete Normalizing Flows on SO3 Manifold for Probabilistic Rotation Modeling", IEEE/CVF CVPR 2023)

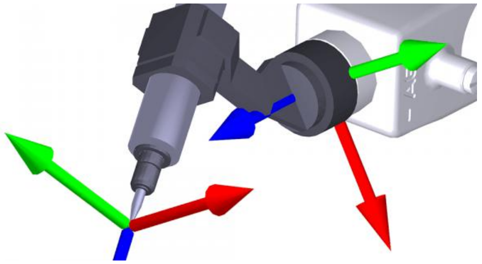


Quaternions..

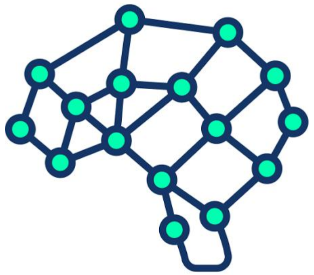
...are a representational unit that is used extensively in fields that include:

- 🎨 Graphics (e.g. Gaussian Splatting - Kerbl et al., “3D Gaussian Splatting for Real-Time Radiance Field Rendering”, ACM SIGGRAPH 2023)

- 🤖 3D Rotation estimation & Robotics (e.g. Liu et al., “Delving into Discrete Normalizing Flows on SO3 Manifold for Probabilistic Rotation Modeling”, IEEE/CVF CVPR 2023)



- 🧠 Hypercomplex or SO3-equivariant Neural Networks (e.g. “Fast Quaternion Product Units for Learning Disentangled Representations in SO(3), IEEE TPAMI 2023)



Quaternions: elements

Basic form

Quaternions $q \in \mathbb{H}$ share the basic form:

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$$

where $a, b, c, d \in \mathbb{R}$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are independent imaginary units.

Generalizations of complex numbers

- Two ways to see this:
  - 2 extra imaginary units ($\boldsymbol{j}, \boldsymbol{k}$). Real and complex numbers can be regarded as quaternions with $b, c, d = 0$ or $c, d = 0$ respectively.
  - OR: $a + b\boldsymbol{j}$ with $a, b \in \mathbb{C}$ (Cayley-Dickson form)

Quaternions: elements

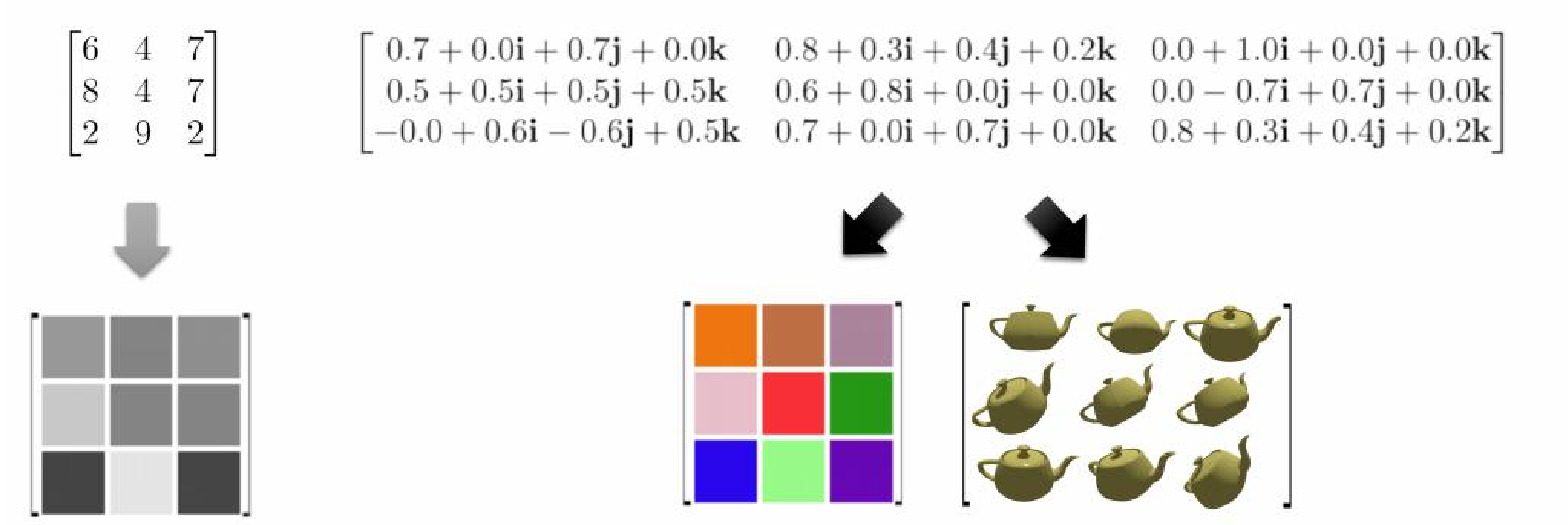


Figure 1: Matrices of real-valued elements (top left) form a standard representational unit for unidimensional data, such as grayscale pixels (bottom left), neural network weights or other unidimensional signals. Matrices of quaternion-valued elements (top right) represent an arrangement of multidimensional data such as color images (Ell & Sangwine, 2007) or 3D spatial rotation (Kerbl et al., 2023) (bottom right).

Motivation and scope of current work

[Sfikas & Retsinas, TMLR 2025]

$$g = Cf \implies Ag = AA^{-1}\Lambda_C Af \implies G = \Lambda_C F$$

Motivation and scope of current work

[Sfikas & Retsinas, TMLR 2025]

$$g = Cf \implies Ag = AA^{-1}\Lambda_C A f \implies G = \Lambda_C F$$

Motivation and scope of current work

[Sfikas & Retsinas, TMLR 2025]

$$g = Cf \implies Ag = AA^{-1}\Lambda_C A f \implies G = \Lambda_C F$$

Motivation and scope of current work

[Sfikas & Retsinas, TMLR 2025]

$$g = Cf \implies Ag = AA^{-1}\Lambda_C A f \implies G = \Lambda_C F$$

Motivation and scope of current work
[Sfikas & Retsinas, TMLR 2025]

- **Research Question**: *How does the well-known relation between Fourier and Convolution, and corresponding matrix forms, generalize to H ?*

$$g = Cf \implies Ag = AA^{-1}\Lambda_C A f \implies G = \Lambda_C F$$

Motivation and scope of current work
[Sfikas & Retsinas, TMLR 2025]

- **Research Question**: *How does the well-known relation between Fourier and Convolution, and corresponding matrix forms, generalize to H ?*

$$g = Cf \implies Ag = AA^{-1}\Lambda_C Af \implies G = \Lambda_C F$$

We provide a matrix framework that allows us to:

Motivation and scope of current work
[Sfikas & Retsinas, TMLR 2025]

- **Research Question**: *How does the well-known relation between Fourier and Convolution, and corresponding matrix forms, generalize to H ?*

$$g = Cf \implies Ag = AA^{-1}\Lambda_C Af \implies G = \Lambda_C F$$

We provide a matrix framework that allows us to:
🔄 Simplify quaternionic operations into standard linear algebra.

Motivation and scope of current work

[Sfikas & Retsinas, TMLR 2025]

- **Research Question**: *How does the well-known relation between Fourier and Convolution, and corresponding matrix forms, generalize to H ?*

$$g = Cf \implies Ag = AA^{-1}\Lambda_C Af \implies G = \Lambda_C F$$

We provide a matrix framework that allows us to:
🔄 Simplify quaternionic operations into standard linear algebra.
🧠 Connect quaternion convolution and Fourier matrix forms

Motivation and scope of current work

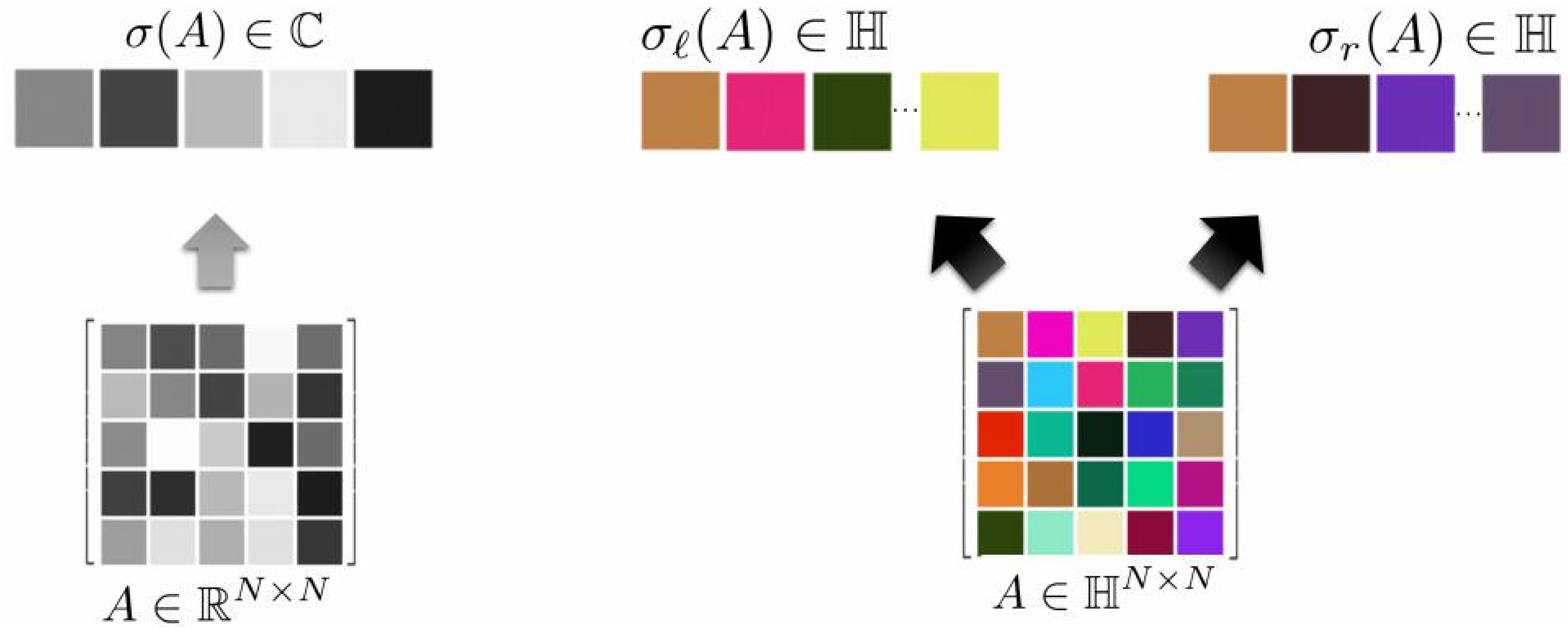
[Sfikas & Retsinas, TMLR 2025]

- **Research Question**: *How does the well-known relation between Fourier and Convolution, and corresponding matrix forms, generalize to H ?*

$$g = Cf \implies Ag = AA^{-1}\Lambda_C Af \implies G = \Lambda_C F$$

We provide a matrix framework that allows us to:
🔄 Simplify quaternionic operations into standard linear algebra.
🧠 Connect quaternion convolution and Fourier matrix forms
📈 Show how properties of the eigenstructure of convolution and Fourier matrices generalize into the significantly more complex structure of quaternion matrices.

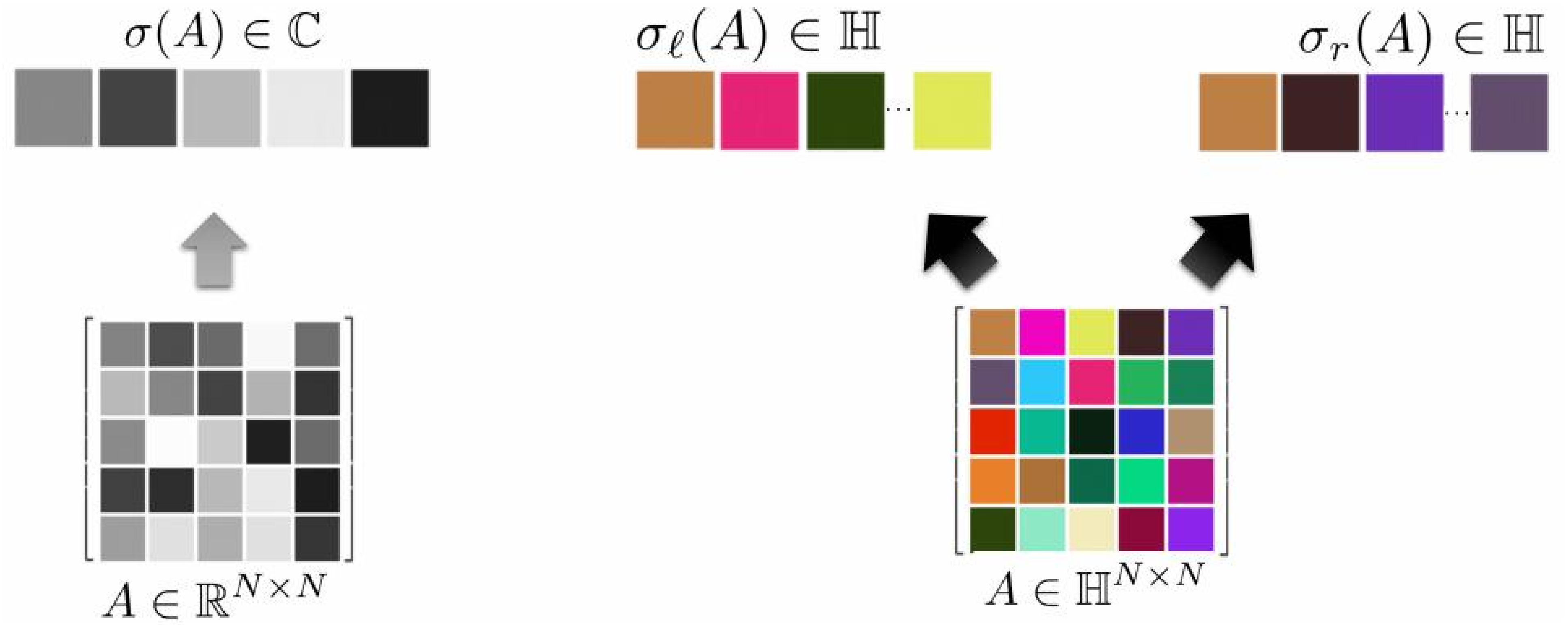
Challenges



The eigenstructure of Quaternion matrices is much more complex than R or C!

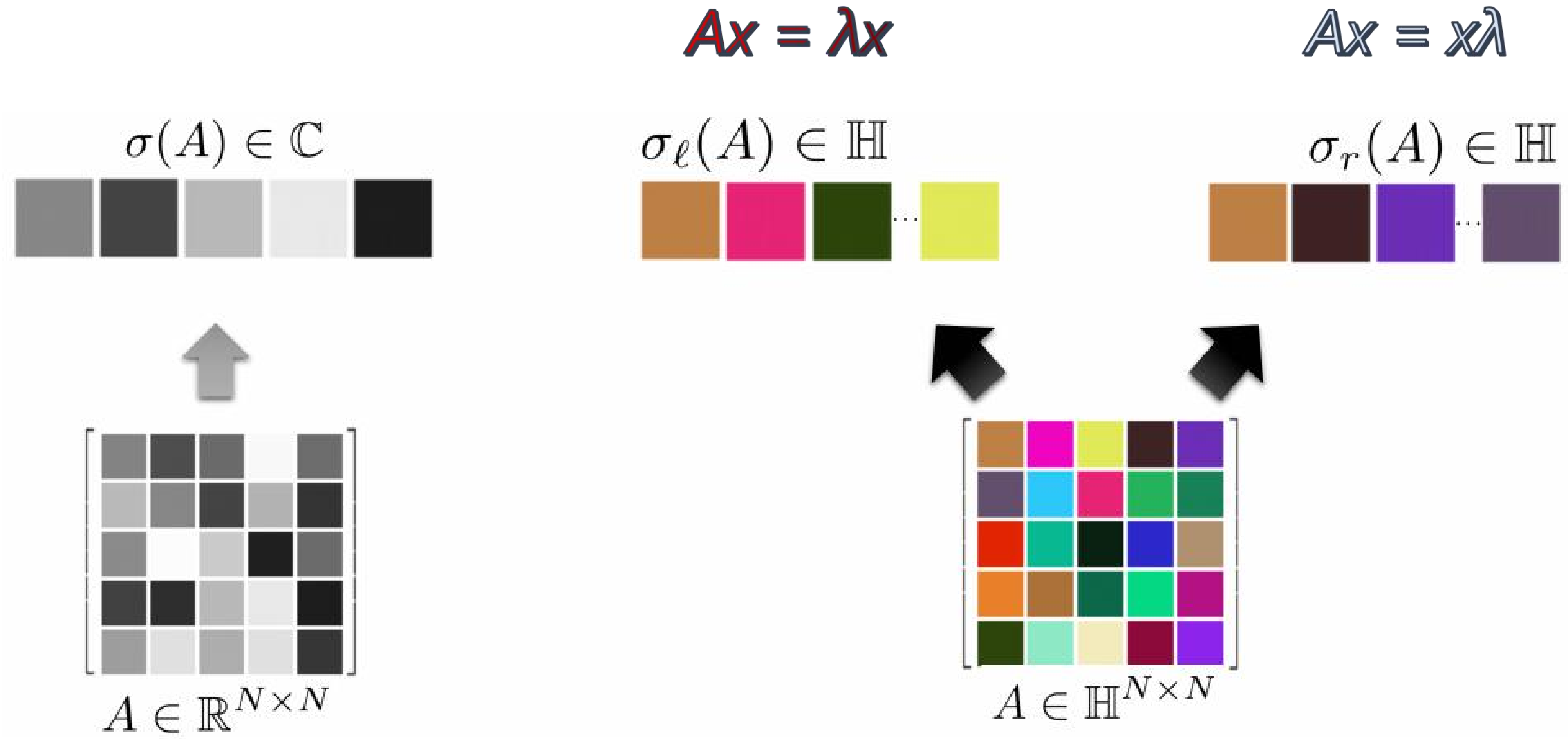
Challenges

$$Ax = \lambda x$$



The eigenstructure of Quaternion matrices is much more complex than R or C!

Challenges



The eigenstructure of Quaternion matrices is much more complex than R or C!

Our «protagonists»

$$C^\top = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \cdots & c_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{pmatrix}$$

Quaternion Circulant matrix

$$Q_N^\mu = \frac{1}{\sqrt{N}} \begin{pmatrix} w_{N\mu}^{0\cdot 0} & w_{N\mu}^{1\cdot 0} & \cdots & w_{N\mu}^{(N-1)\cdot 0} \\ w_{N\mu}^{0\cdot 1} & w_{N\mu}^{1\cdot 1} & \cdots & w_{N\mu}^{(N-1)\cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N\mu}^{0\cdot (N-1)} & w_{N\mu}^{1\cdot (N-1)} & \cdots & w_{N\mu}^{(N-1)\cdot (N-1)} \end{pmatrix}$$

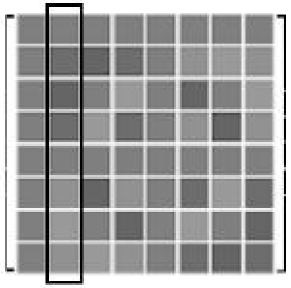
$$w_{N\mu} = e^{-\mu 2\pi N^{-1}}$$

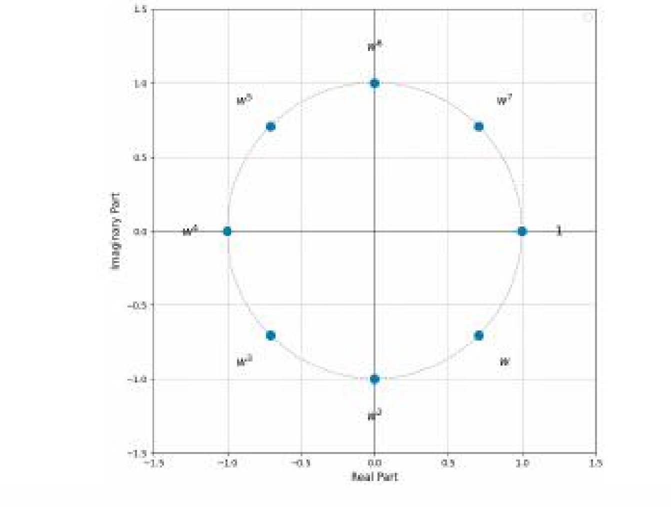
Quaternion Fourier matrix

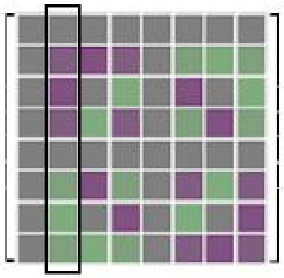
Basic theoretical results

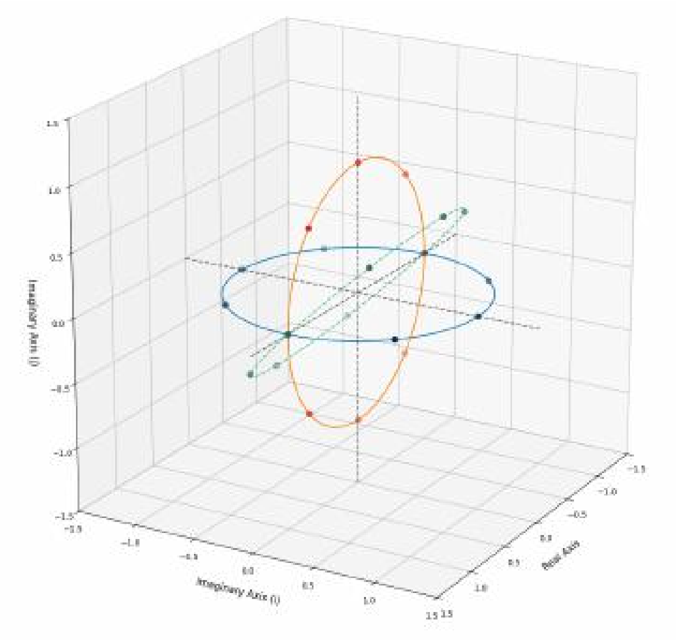
Proposition 3.4. *Let Q_N^{μ}, Q_N^{ν} Quaternionic Fourier matrices with non-collinear axes μ, ν. We can always find unit $p \in \mathbb{H}$ such that*

$$Q_N^{\mu} = pQ_N^{\nu}\bar{p} \tag{20}$$









Basic theoretical results

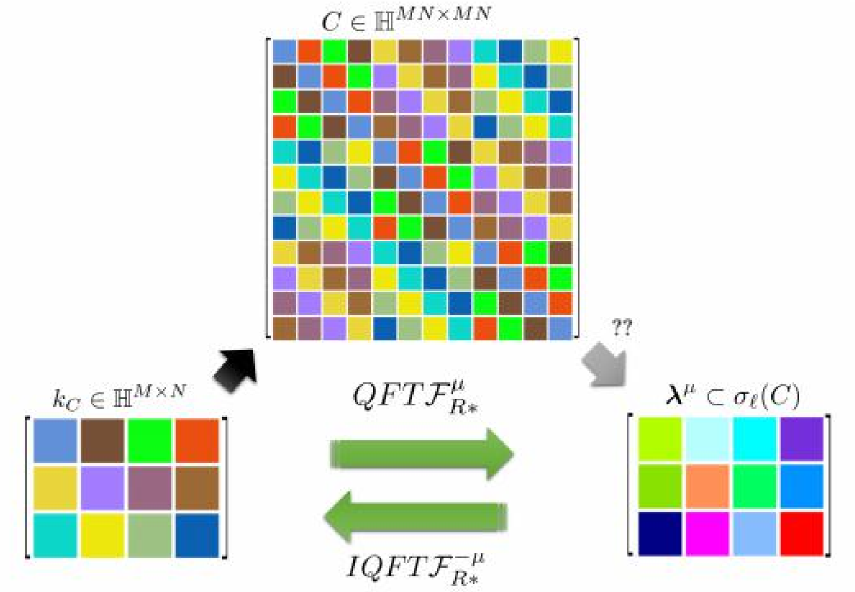
3.3 Connection of Circulant and Fourier matrices

The following results can then be proved, underpinning the relation between Quaternionic Circulant and Quaternionic Fourier matrices, in particular with respect to the left spectrum of the former:

Proposition 3.5 (Circulant & Fourier Matrices). *For any $C \in \mathbb{H}^{N\times N}$ that is circulant, and any pure unit $\mu \in \mathbb{H}$,*

(1) *Any column $k = 0, 1, \ldots, N-1$ of the inverse QFT matrix $Q_N^{-\mu}$ is an eigenvector of C. Column k corresponds to the k^{th} component of the vector of* left *eigenvalues $\boldsymbol{\lambda}^\mu = [\lambda_1^\mu \lambda_2^\mu \cdots \lambda_N^\mu]^T \in \mathbb{H}^N$. Vector $\boldsymbol{\lambda}^\mu$ is equal to the* right *QFT $\mathcal{F}_{R*}^\mu$ of the kernel of C.*

(2) *Any column $k = 0, 1, \ldots, N-1$ of the inverse QFT matrix $Q_N^{-\mu}$ is an eigenvector of C^H. Column k corresponds to the k^{th} component of the vector of* left *eigenvalues $\boldsymbol{\kappa}^\mu = [\kappa_1^\mu \kappa_2^\mu \cdots \kappa_N^\mu]^T \in \mathbb{H}^N$. The conjugate of the vector $\boldsymbol{\kappa}^\mu$ is equal to the* left *QFT $\mathcal{F}_{L*}^\mu$ of the kernel of C.*



Basic theoretical results

Proposition 3.7 (Eigenstructure of sums, products and inverse of circulant matrices)**.**

Let $L, K \in \mathbb{H}^{N\times N}$ circulant matrices, then the following propositions hold. (Analogous results hold for L, K doubly-block circulant).

(a) The sums $L+K$ and products LK, KL are also circulant. Any scalar product pL or Lp where $p \in \mathbb{H}$ also results to a circulant matrix.

(b) Given some pure unit axis $\mu \in \mathbb{H}$, let w be the i^{th} column of the inverse QFT matrix $Q_N^{-\mu}$, and $\lambda_i^\mu, \kappa_i^\mu$ are left eigenvalues of L, K with respect to the shared eigenvector w. Then, w is an eigenvector of...

i. $L+K$ with left eigenvalue equal to $\lambda_i^\mu + \kappa_i^\mu$.

ii. pL with left eigenvalue equal to $p\lambda_i^\mu$.

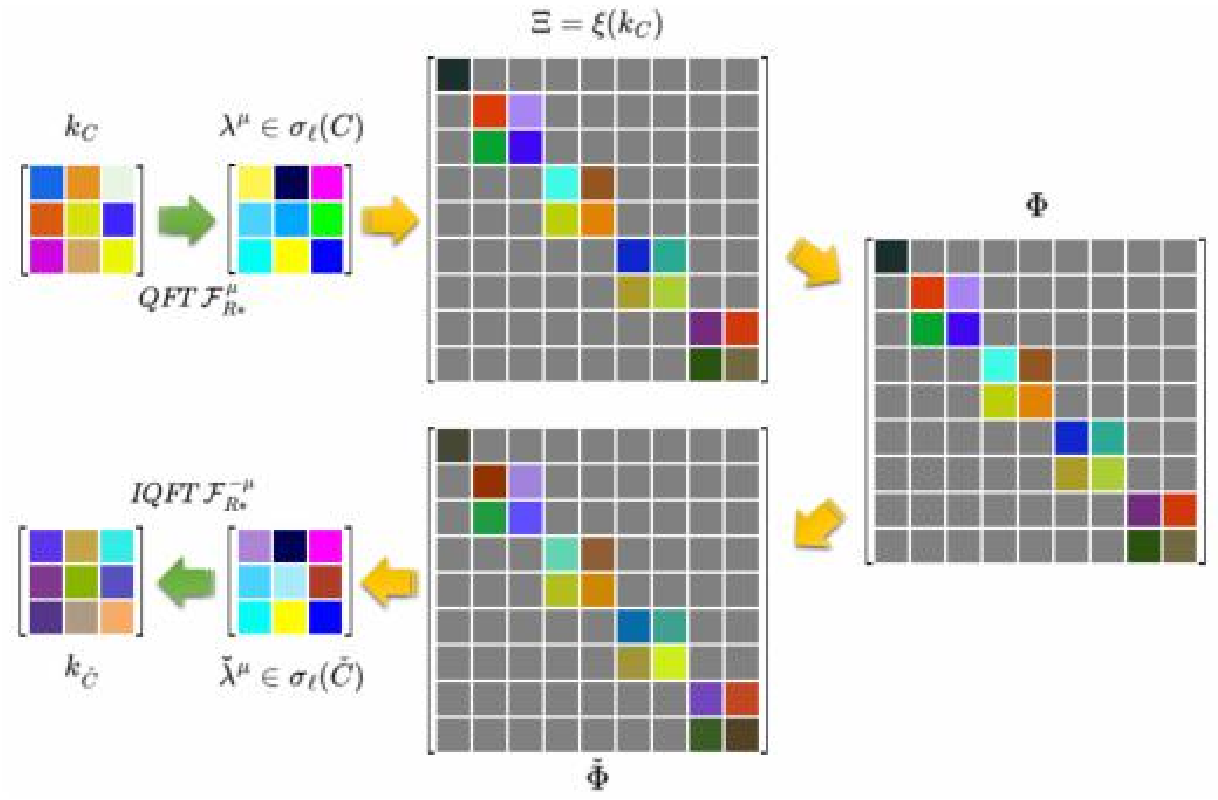
iii. Lp with left eigenvalue equal to $\lambda_i^{p\mu p^{-1}} p$.

iv. LK with left eigenvalue equal to $\lambda_i^{\kappa_i^\mu \mu [\kappa_i^\mu]^{-1}} \kappa_i^\mu$ if $\kappa_i^\mu \neq 0$ and equal to 0 if $\kappa_i^\mu = 0$.

v. Provided L^{-1} exists, $\lambda_i^\mu w (\lambda_i^\mu)^{-1}$ is an eigenvector of L^{-1} with left eigenvalue equal to $(\lambda_i^\mu)^{-1}$.

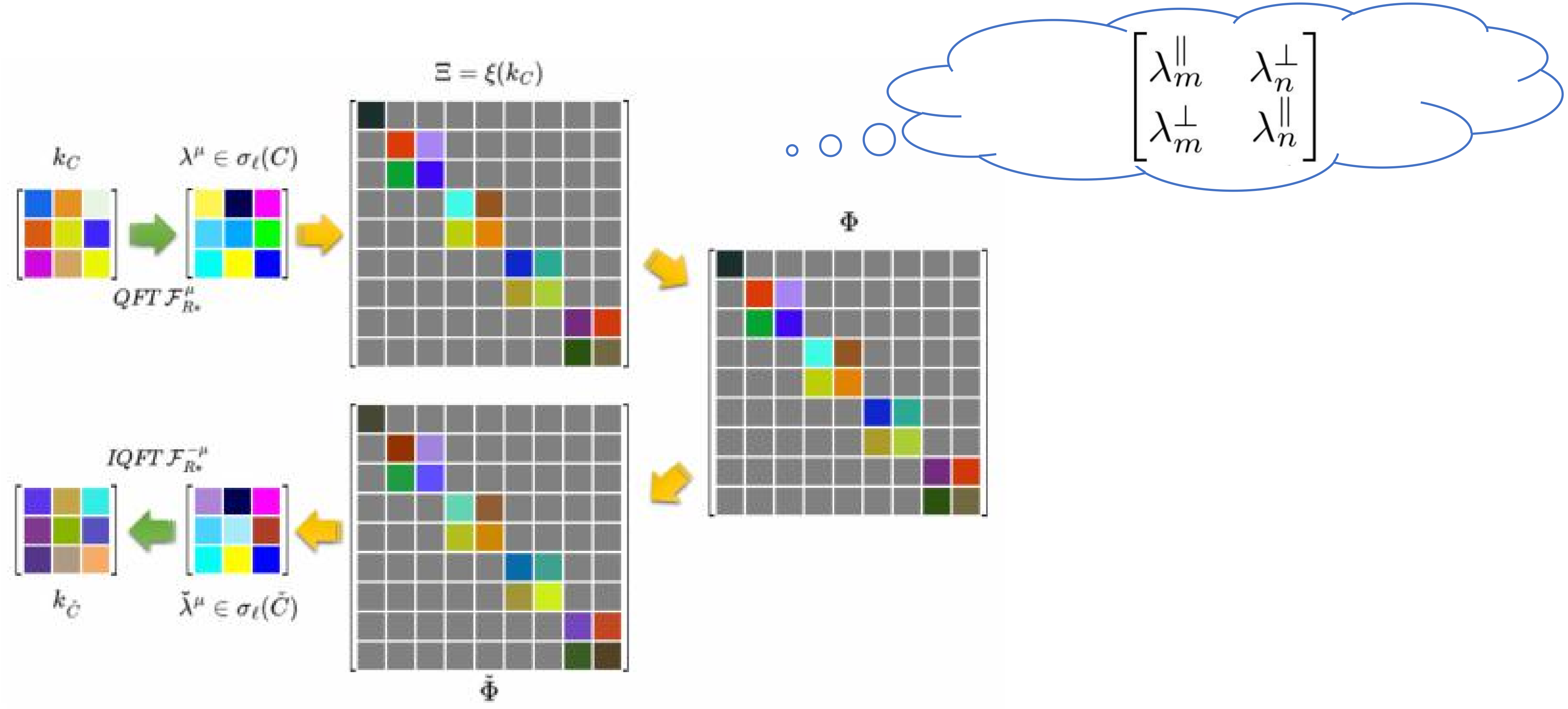
Proof of concept application

Fast computation of singular values, Lipschitz constant bounding



Proof of concept application

Fast computation of singular values, Lipschitz constant bounding



Proof of concept application

Fast computation of singular values, Lipschitz constant bounding

Table 1: Runtime comparison for computation of Quaternionic Convolution Singular Values and Spectral Norm Clipping & Reconstruction. Convolution kernel size refers to $2D$ kernel side length. Average CPU time in milliseconds is reported; an exception are the fields containing an asterisk ($\star$), which denote that required runtime is impractical (measured in hours). Our results enable fast and exact carrying out of both tasks.

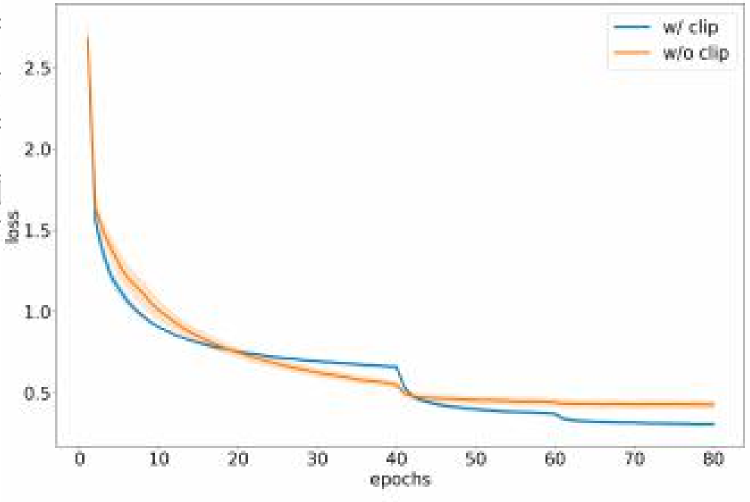
		Required time (in ms) ↓					
Convolution kernel size		4	8	16	32	64	128
Singular Value Computation	ours	4	16	80	210	708	1,795
	brute-force	5	95	1,244	69,319	$\star$	$\star$
Clipping & Reconstruction	ours	5	20	113	280	1,119	2,590
	brute-force	8	121	1,866	150,000	$\star$	$\star$

Proof of concept application

Fast computation of singular values, Lipschitz constant bounding

Table 1: Runtime comparison for computation of Quaternionic Convolution Singular Values and Spectral Norm Clipping & Reconstruction. Convolution kernel size refers to $2D$ kernel side length. Average CPU time in milliseconds is reported; an exception are the fields containing an asterisk ($\star$), which denote that required runtime is impractical (measured in hours). Our results enable fast and exact carrying out of both tasks.

		Required time (in ms) ↓		
Convolution kernel size		4	8	16
Singular Value Computation	ours	4	16	80
	brute-force	5	95	1,244
Cli[illegible] & Recons[illegible]	ours	5	20	113
	brute-force	8	121	1,866



With our results, runs on $O(N^2 \log N)$
(vs previous best time of $O(N^6)$!!)

Thank you for your attention!

Come chat with us by our poster!

-- Poster Session 1 (Wednesday, 11.30 - 12.30, id#3)

paper

Open Review

code

GitHub

Questions ?