Don't take it lightly: Phasing optical random projections with unknown operators

Sidharth Gupta*, Rémi Gribonval†, Laurent Daudet‡ Ivan Dokmanić*

*University of Illinois at Urbana-Champaign †Univ Rennes, Inria, CNRS, IRISA ‡LightOn, Paris

ILLINOIS Coordinated Science Lab COLLEGE OF ENGINEERING

Inria LightOn We bring Light to AI

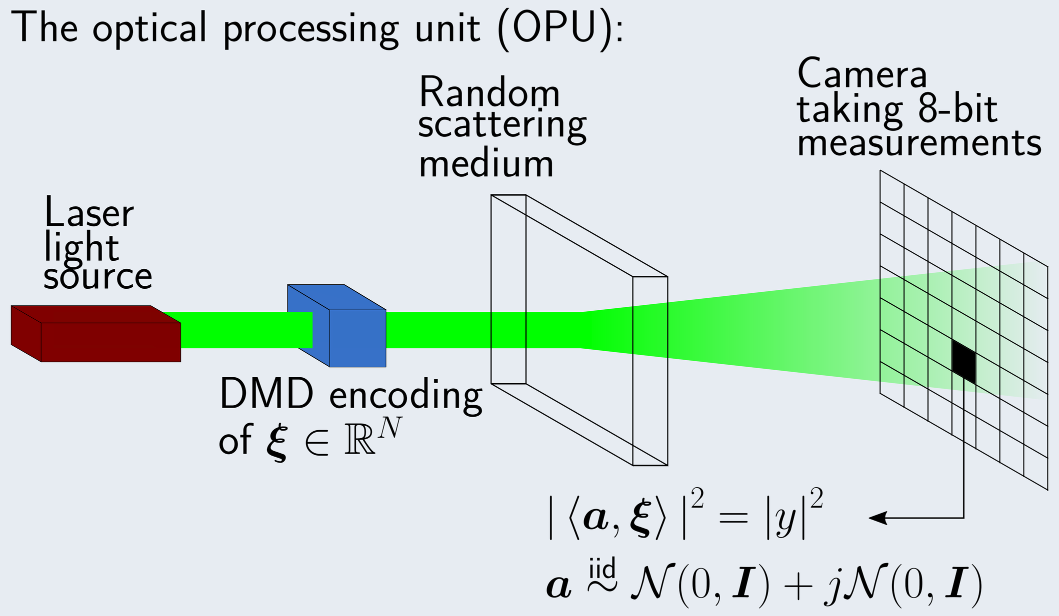
Why fast random projections?

- Key to many algorithms in machine learning, signal processing and numerical linear algebra.
- Applications: classification with random features, kernel approximation, matrix optimization via sketching, randomized linear algebra and many more!
- Bottleneck: memory and computation intensive when applied to images, videos, and modern big data streams.

→ Can we increase the size and speed of such projections?

Random projections at the speed of light

- Optics-based matrix multiplications can be much faster and energy-efficient than the CPU and GPU.
- The optical processing unit (OPU):



- Input data, $\boldsymbol{\xi} \in \mathbb{R}^N$, is "imprinted" onto a coherent light beam using a digital micro-mirror device (DMD) and shined through a multiple scattering medium.
- The scattered lightfield in the sensor plane is written as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{\xi}$$

where $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ has iid standard complex Gaussian entries.
- We can only measure the quantized intensity of scattered light, $|\boldsymbol{y}|^2$, and the phase information is lost.

Measurement phase retrieval (MPR)

We have

$$\boldsymbol{b} = |\boldsymbol{y}|^2 + \boldsymbol{\eta} = |\boldsymbol{A}\boldsymbol{\xi}|^2 + \boldsymbol{\eta},$$

where
$\boldsymbol{b} \in \mathbb{R}^M$ is the measurement
$\boldsymbol{\xi} \in \mathbb{R}^N$ is the input
$\boldsymbol{A} \in \mathbb{C}^{M \times N}$ is an iid standard complex Gaussian matrix
$\boldsymbol{y} \in \mathbb{C}^M$
$\boldsymbol{\eta} \in \mathbb{R}^M$ is noise.

GOAL: Recover the phase of each complex-valued element of $\boldsymbol{y}$, y_i for $1 \le i \le M$, from its magnitude measurements when $\boldsymbol{\xi}$ is *known* and $\boldsymbol{A}$ is *unknown*.

Recovery up to a reference phase and conjugation

- With $\boldsymbol{A}$ unknown, recovering the absolute phase of $\boldsymbol{A}\boldsymbol{\xi}$ is impossible.
- **Observation**: with $\boldsymbol{A}$ standard complex Gaussian, R($\boldsymbol{A}$) has the same distribution as $\boldsymbol{A}$ where R is a transform that adds a constant phase to each row of its argument (multiplies it by $\mathrm{diag}(e^{j\phi_1}, \ldots, e^{j\phi_m})$) and conjugates a subset of its rows.
- **Approach**: use the same *effective* R for all inputs so that relative phases between recovered measurements are the same.

OBJECTIVE: Given inputs $\xi_1, \ldots, \xi_S$, compute the estimates of random projections $\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_S$ up to a global row-wise phase and conjugation; $\hat{\boldsymbol{y}}_s \approx \mathrm{R}(\boldsymbol{y}_s)$ for all $1 \le s \le S$ for some possibly unknown R.

Measurements are distances

- Input signal $\boldsymbol{\xi} \in \mathbb{R}^N$.
- K reference signals $\boldsymbol{r}_k \in \mathbb{R}^N$ for $1 \le k \le K$ with $\boldsymbol{r}_K = \boldsymbol{0}$
- Build $\boldsymbol{X} \in \mathbb{R}^{N \times Q}$, so that $\boldsymbol{X} = [\boldsymbol{\xi}, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_K]$ and let $Q = K + 1$.
- The qth column of $\boldsymbol{X}$ is denoted $\boldsymbol{x}_q$.
- Consider randomly projecting $(\boldsymbol{x}_q - \boldsymbol{x}_r)$ for all (q, r).
- This gives measurement $|\langle \boldsymbol{a}^m, \boldsymbol{x}_q - \boldsymbol{x}_r \rangle|^2 = |y_{q,m} - y_{r,m}|^2$ for row m of $\boldsymbol{A}$.

KEY FACT: By measuring the squared Euclidean distance between points on the complex plane we can localize points on the complex plane.

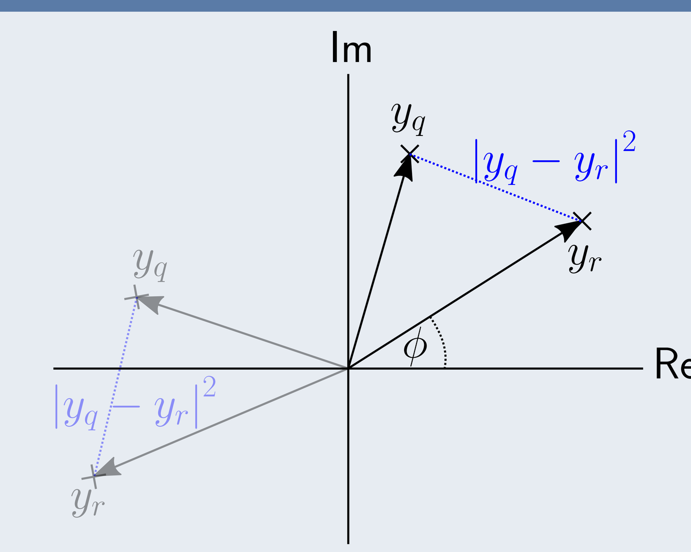
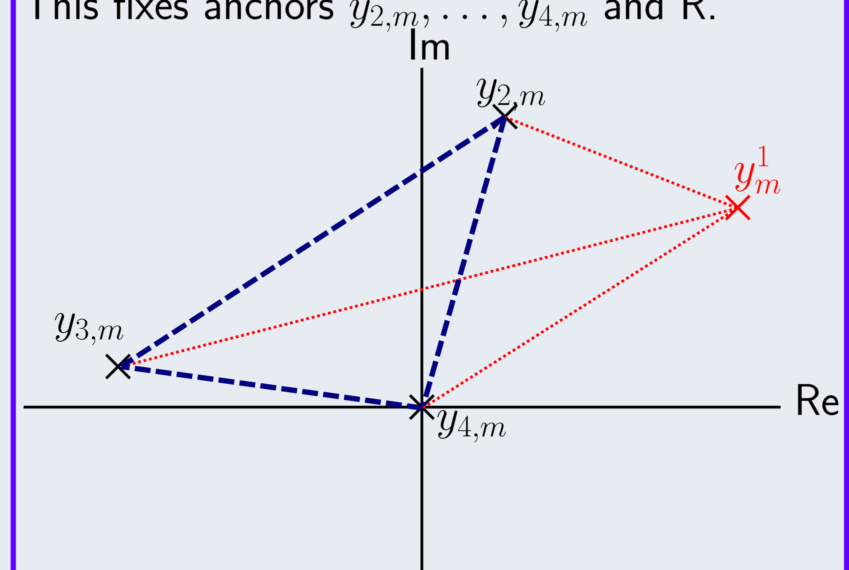


Figure: We can measure distances $|y_q - y_r|^2$.

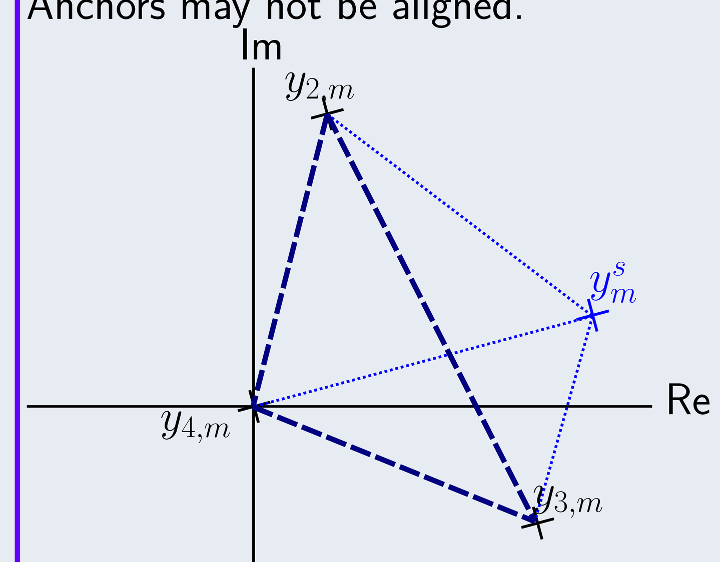
Distance geometry MPR algorithm

OVERVIEW: Obtain distances between random projections and known references to localize random projection and obtain relative measurement phase.

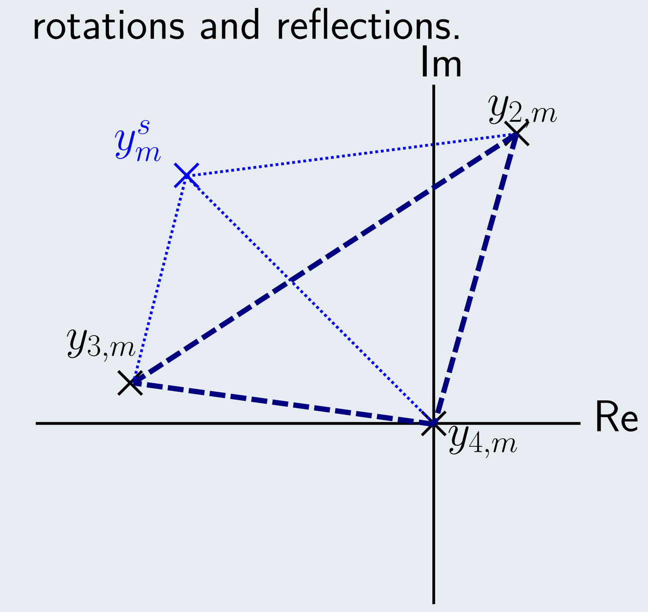
Step 1)
Localize first input ($s = 1$), $y_m^1 = \langle \boldsymbol{a}^m, \boldsymbol{\xi}_1 \rangle$.
This fixes anchors $y_{2,m}, \ldots, y_{4,m}$ and R.



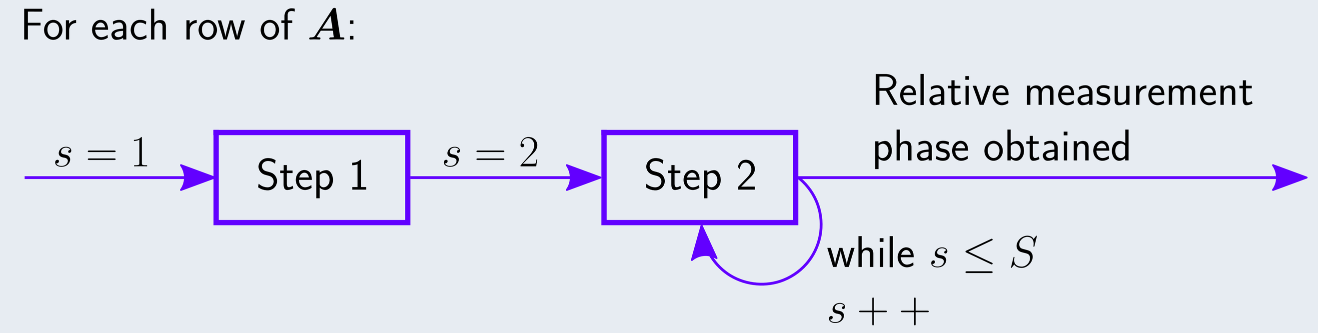
Step 2)
First, localize sth input, $y_m^s = \langle \boldsymbol{a}^m, \boldsymbol{\xi}_s \rangle$.
Anchors may not be aligned.

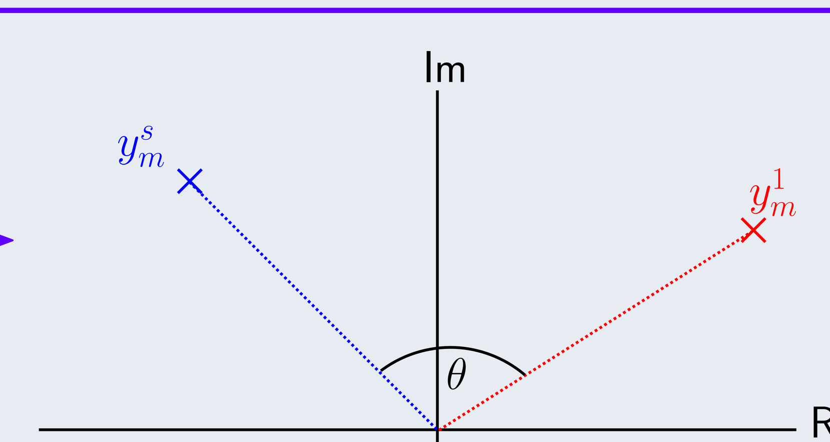


Second, align anchors $y_{2,m}, \ldots, y_{4,m}$ with first input anchors via rotations and reflections.



For each row of $\boldsymbol{A}$:





Multidimensional scaling (MDS) used for localization
Centering to origin because $\boldsymbol{r}_K = \boldsymbol{0}$
Procrustes analysis used to align anchors

Experimental verification

The system is approximately linear when using our method and enables new randomized algorithms.

Linearity.

- Recover $\boldsymbol{y}$ and $\boldsymbol{z}$ from $|\boldsymbol{y}|^2 = |\boldsymbol{A}\boldsymbol{\xi}_1|^2$ and $|\boldsymbol{z}|^2 = |\boldsymbol{A}\boldsymbol{\xi}_2|^2$.
- Do we get $(\boldsymbol{y} + \boldsymbol{z})$ when solving $|\boldsymbol{v}|^2 = |\boldsymbol{A}(\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2)|^2$?
-

$$\text{linearity error} = \frac{1}{M} \sum_{m=1}^{M} \frac{|(y_m + z_m) - v_m|}{|v_m|}.$$

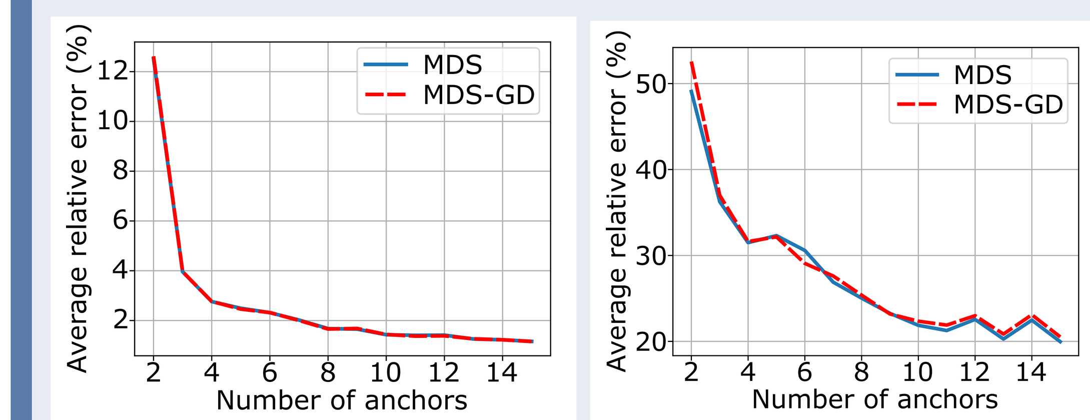


Figure: Experiments in simulation (left) and on a real hardware OPU (right) to evaluate the linearity error. The input signals are of dimension 64^2, M is 100 and the number of reference signals are increased. The classical MDS and MDS with gradient descent (MDS-GD) are used. In all cases the error decreases as the number of anchors increases.

Randomized SVD (RSVD).

- RSVD of matrix $\boldsymbol{B}$ requires multiplying it by a random Gaussian matrix.
- The OPU is used to compute this random matrix multiplication with $\boldsymbol{B} \in \mathbb{R}^{500 \times 28^2}$ being 500 28 × 28 vectorized and stacked samples from the MNIST dataset.

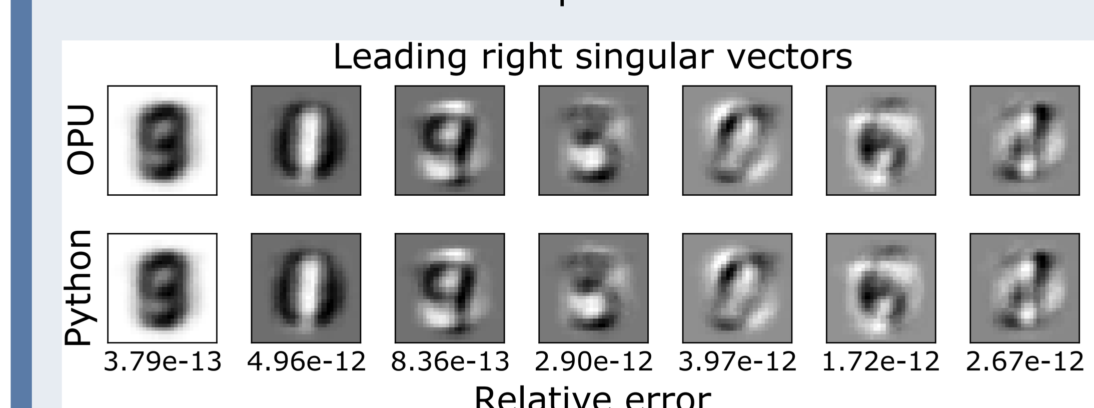


Figure: Reshaped leading right singular vectors of an MNIST matrix of size 500×28^2. The top rows shows the leading right singular vectors after performing RSVD with the OPU and using our algorithm. The bottom row shows the leading right singular vectors from Python. The relative error is below each singular vector.

Sneak peek

- Swapping the roles of $\boldsymbol{A}$ and the input $\boldsymbol{\xi}$ reduces learning $\boldsymbol{A}$ to a linear rather than quadratic problem.
- Concatenate K inputs into Ξ and recover $\boldsymbol{Y}$ and then $\boldsymbol{A}$:

$$\boldsymbol{Y} = \boldsymbol{A}\Xi.$$

- **Dramatic improvements.** State-of-the-art: 3.26 hours. Our approach: 6.15 minutes!

S. Gupta et al. "Fast Optical System Identification by Numerical Interferometry." arXiv 2019