Applied Math
UNIVERSITY OF COLORADO **BOULDER**

Research in Applied Math

Professor Stephen Becker
Dept. of Applied Math
University of Colorado

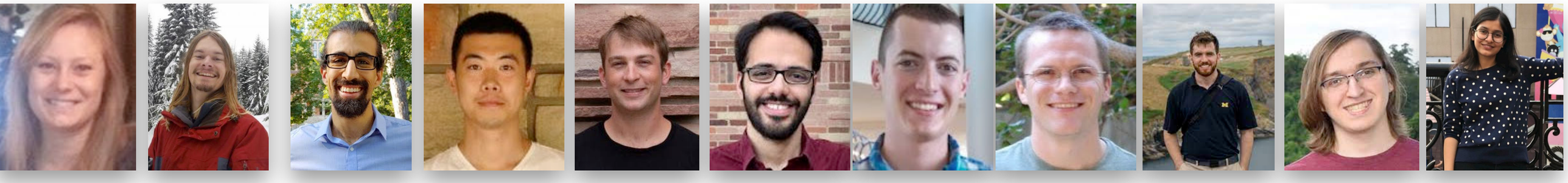
coloradomathcircle

Sept. 15 2024

Part 1: Projects in Applied Math from my CU research group

My research group

Current and former PhD students in my group



Undergrads, MS students, student collaborators, students at Colorado School of Mines, etc.

Source separation

A lot of our work is on **optimization**, and how you can apply it to **image processing**

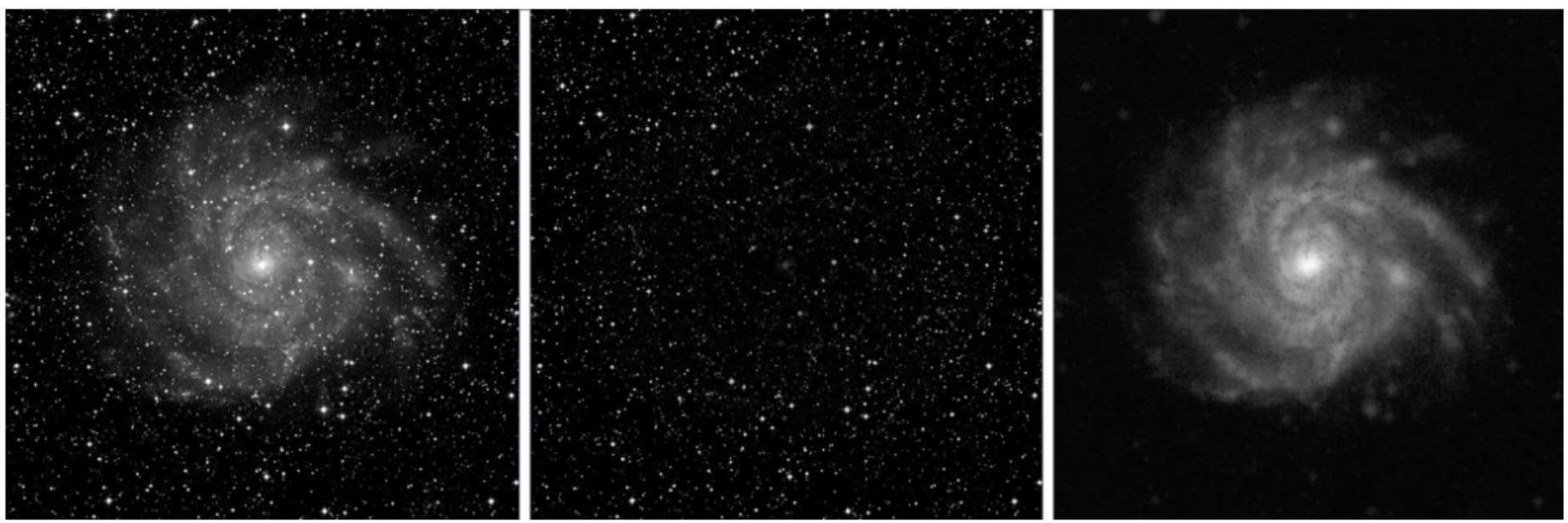


Image credit: NASA

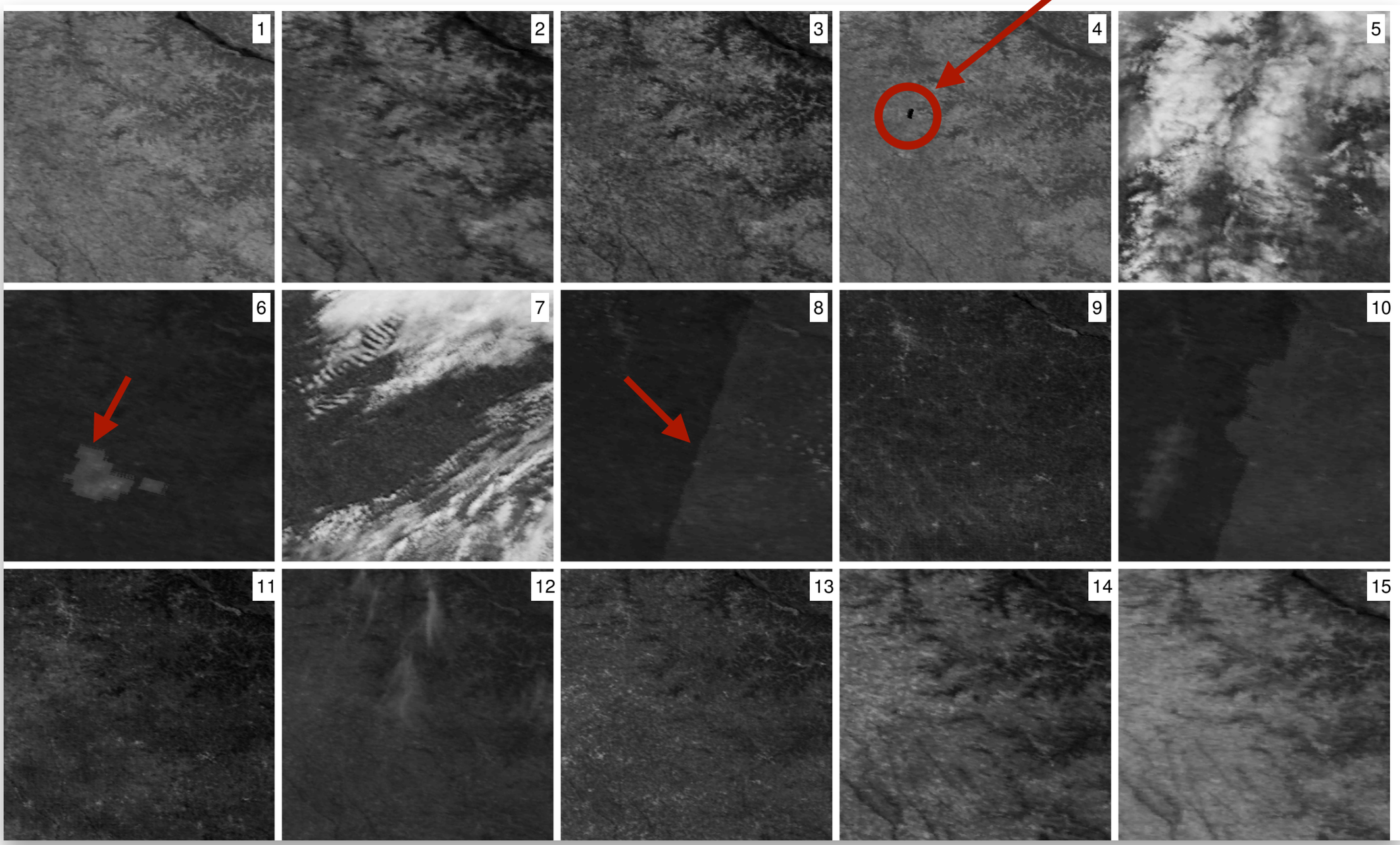
Observation z_0 | Sparse component x_0 | DCT-sparse component y_0

(image from my friend Mike McCoy)

Source separation

Tools:
optimization

Artifacts



MODIS satellite data

Source separation

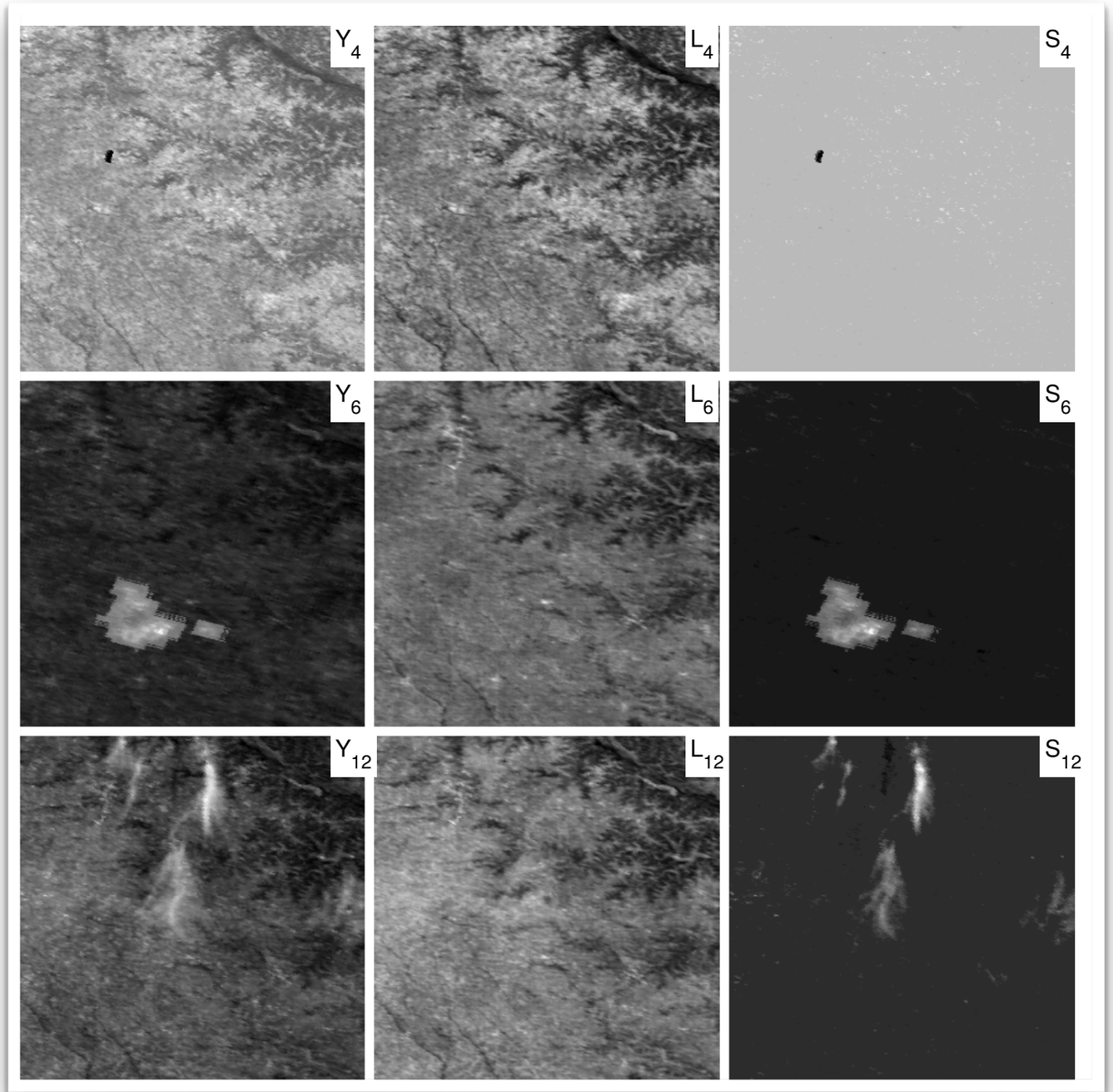
Tools:
optimization

Original | Cleaned | Artifacts

Day #4

Day #6

Day #12



After applying our technique… artifacts are removed!

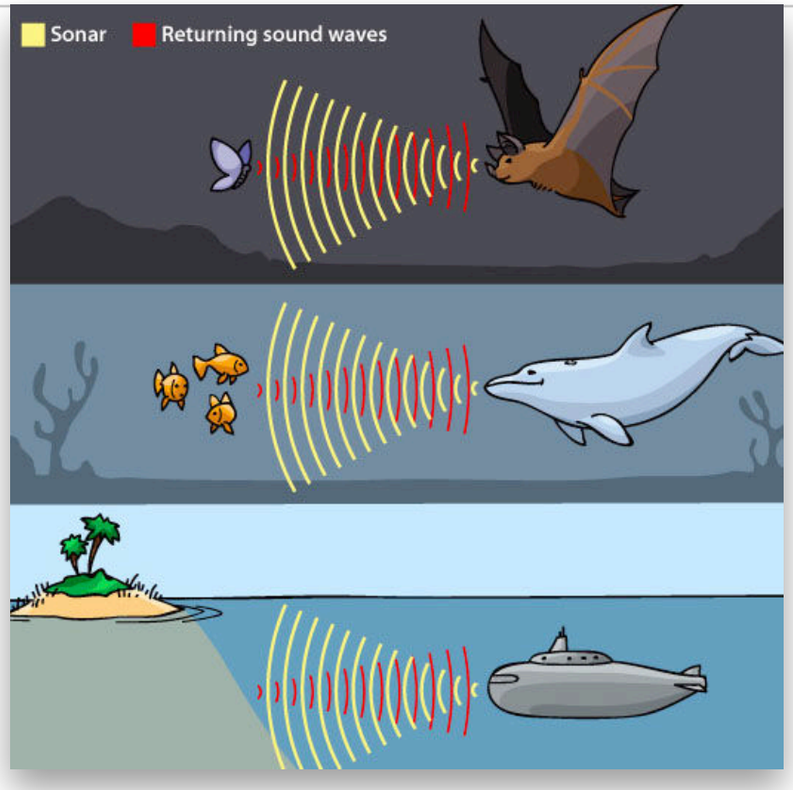
Source separation

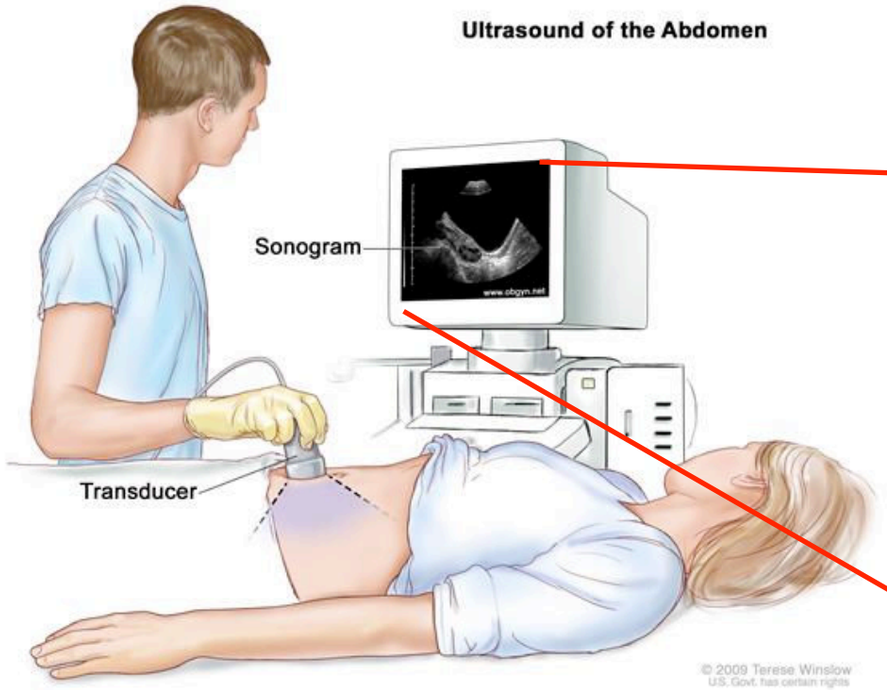
Tools:
optimization

Original Background Foreground

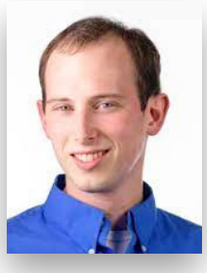
Ultrasound

Ultrasound imaging





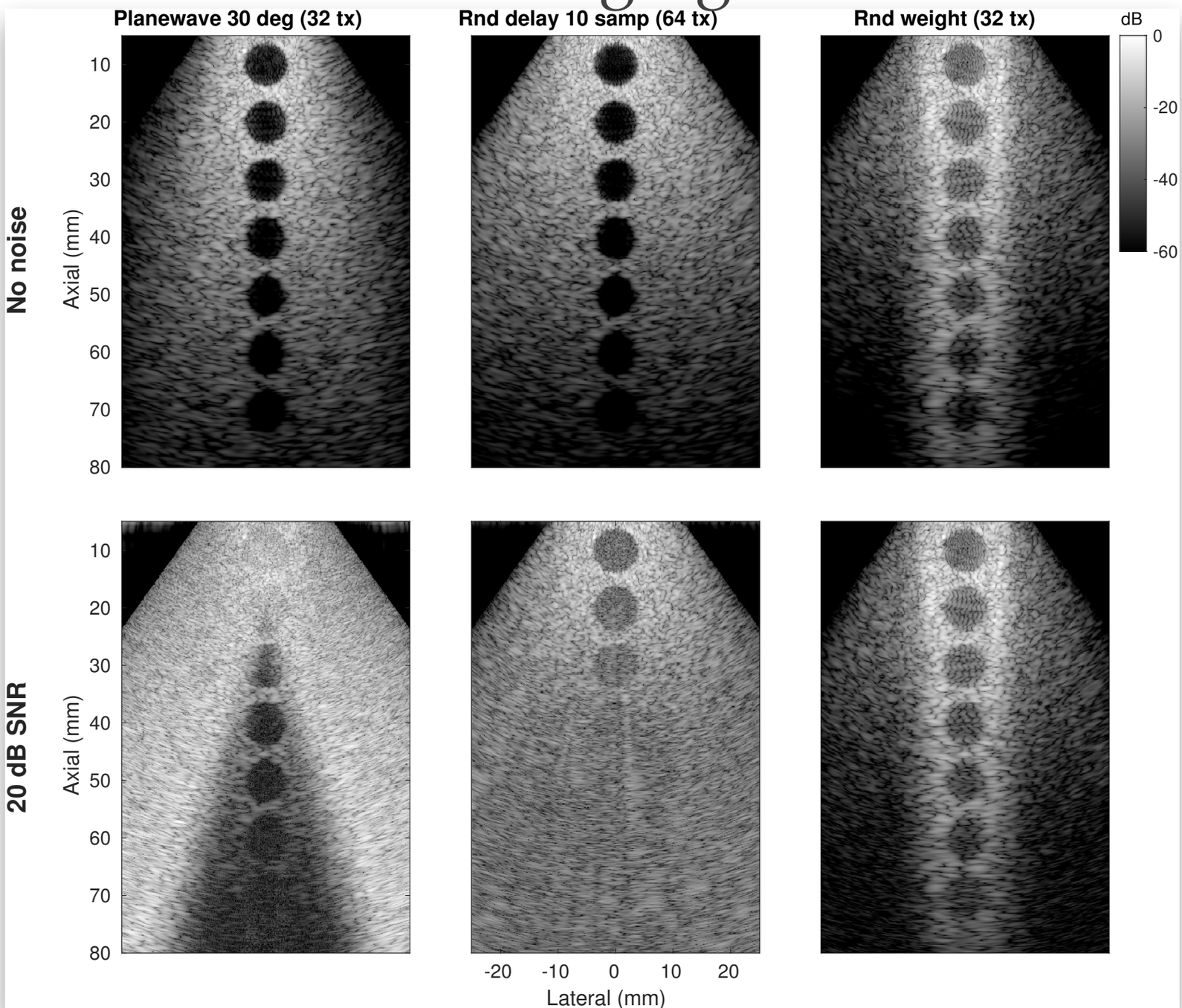
with Prof. Nick Bottenus
Mechanical Engineering



Ultrasound

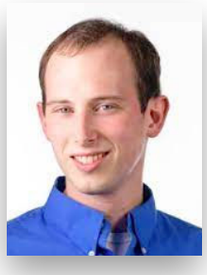
Tools:
optimization

Ultrasound imaging



Project: send out better signals

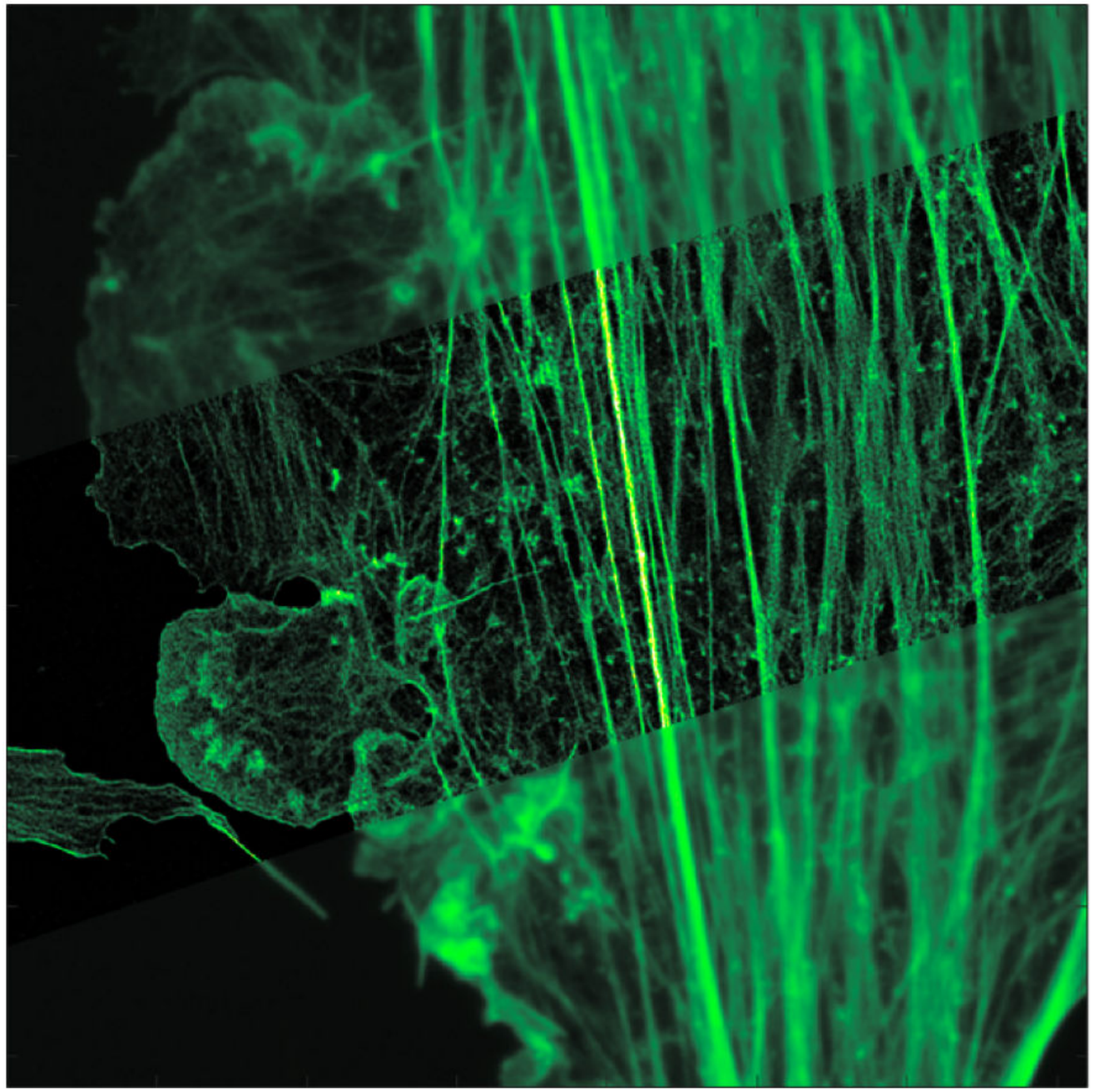
with Prof. Nick Bottenus
Mechanical Engineering



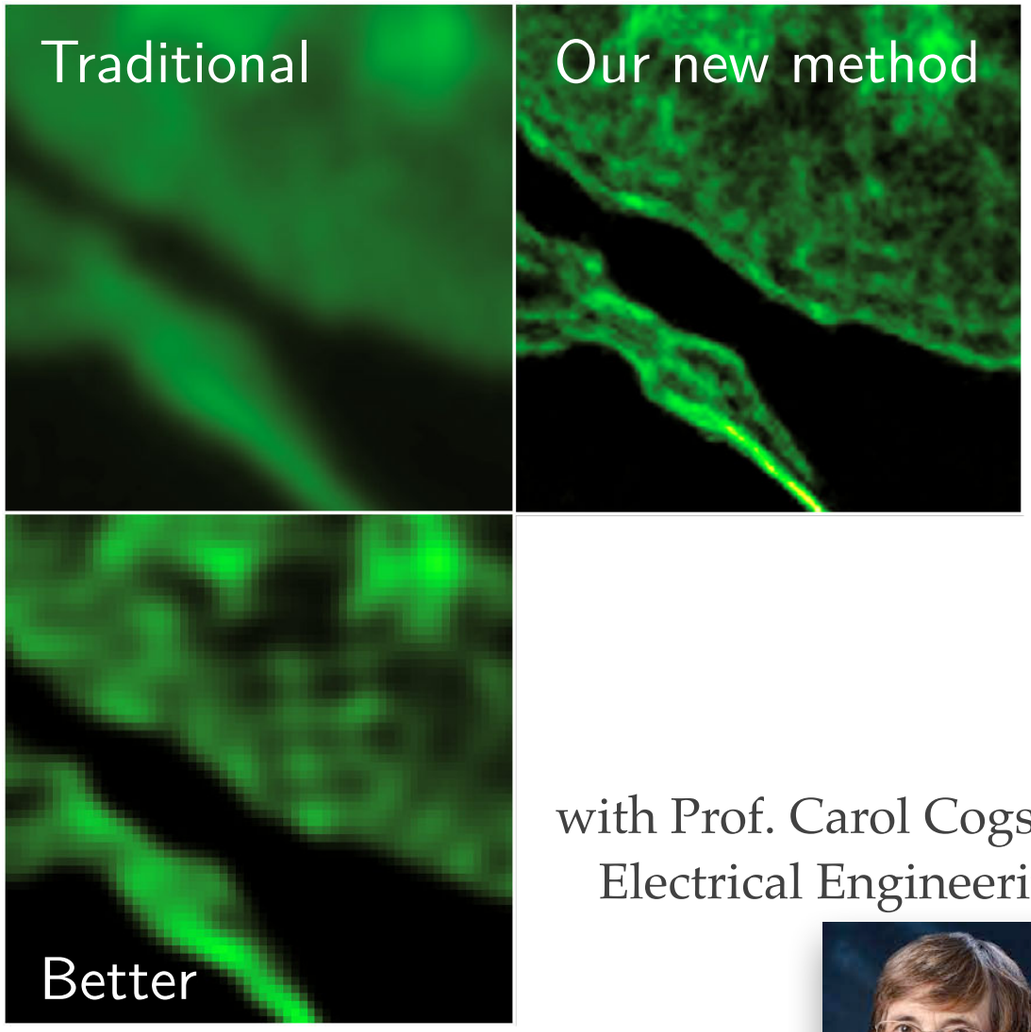
Microscopy

Tools:
optimization

Superresolution fluorescent microscopy



Bovine Pulmonary Artery Endothelial Cells (BPAE) labeled for actin

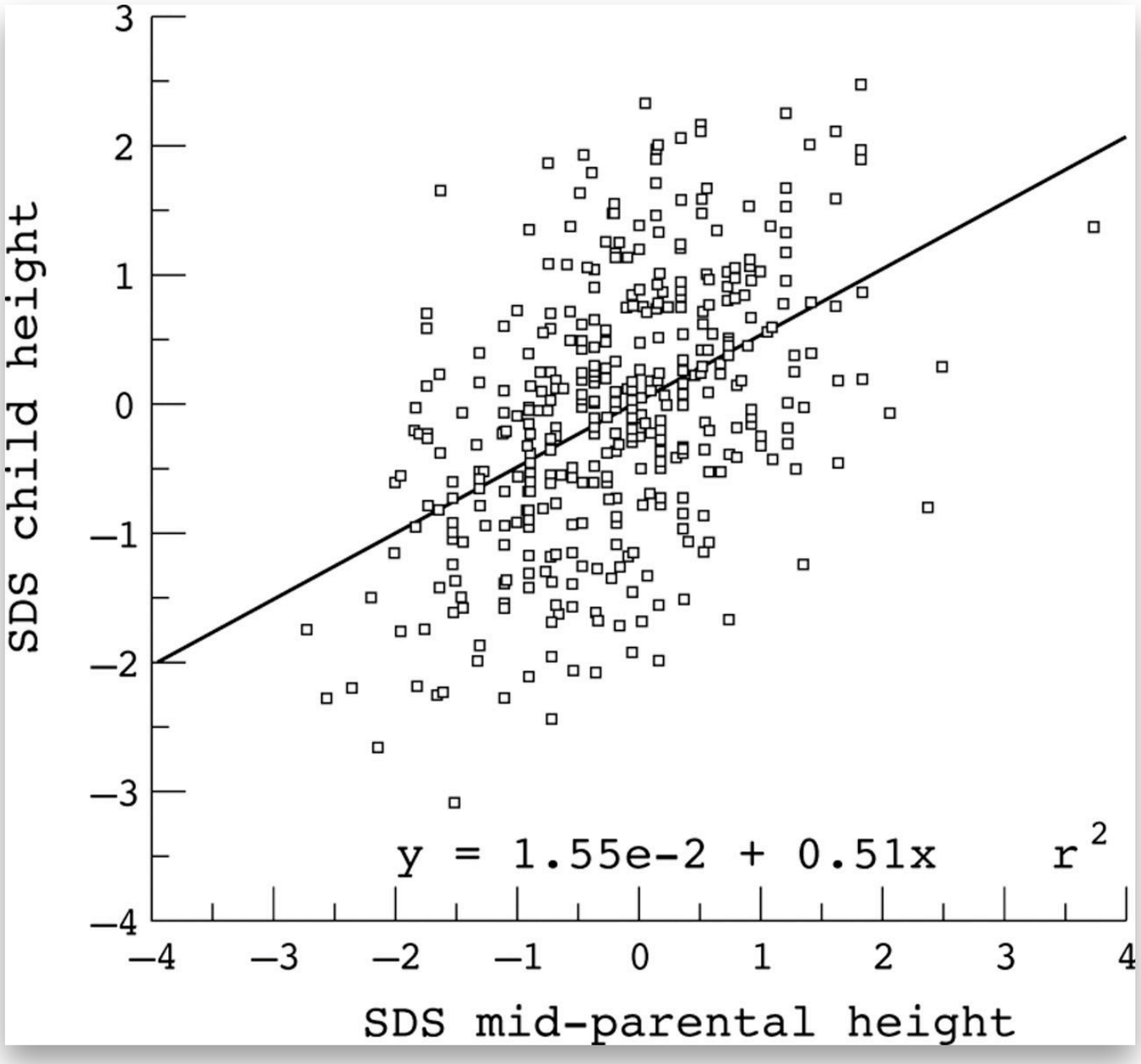


New method is mostly better post-processing

with Prof. Carol Cogswell
Electrical Engineering

Genetics

Does the parents height completely determine their child's height?



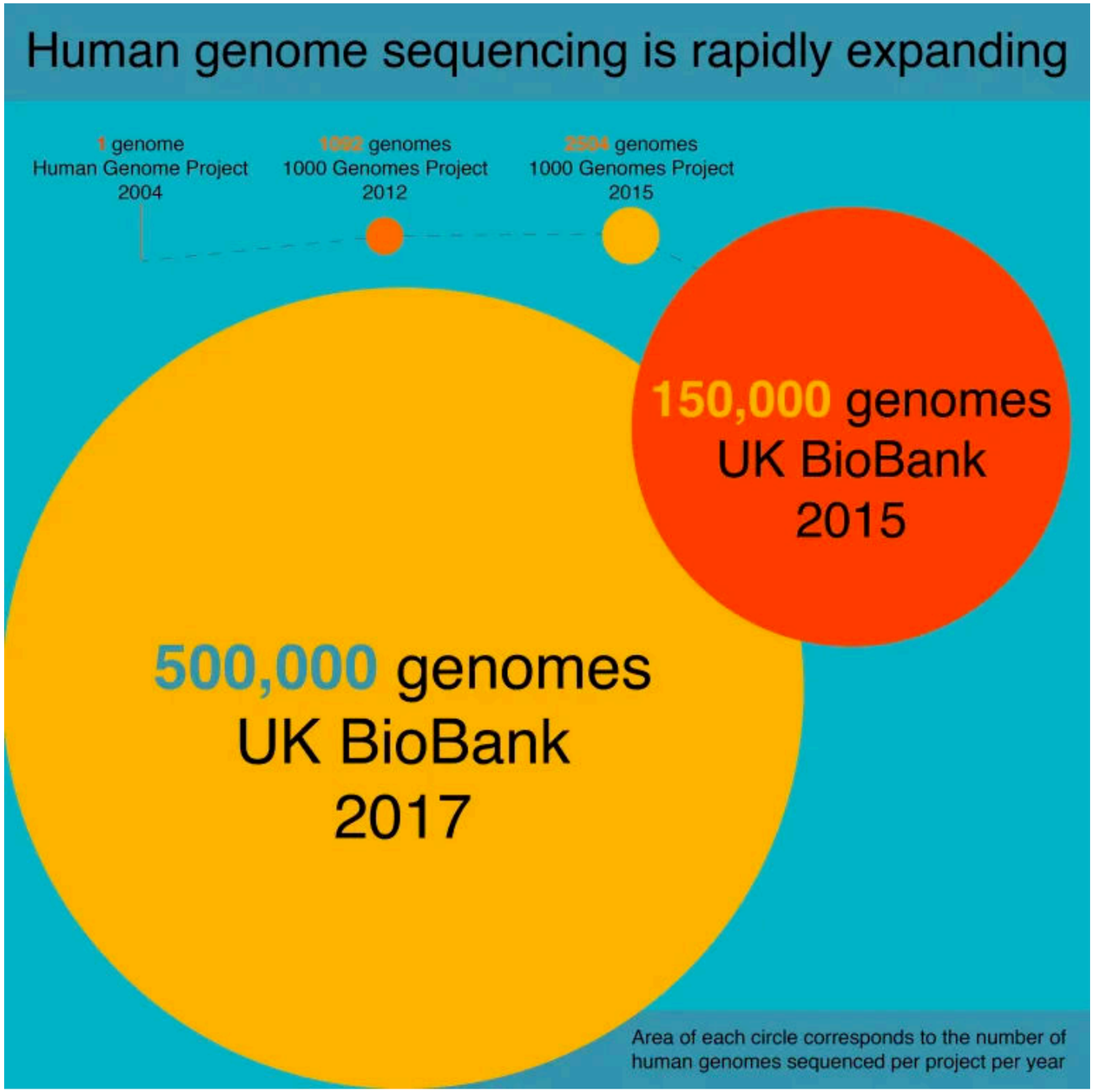
If not… *how much* effect do the parents have?

i.e., how **heritable** is height?

how **heritable** is schizophrenia?

Genetics

Tools:
optimization



Heritability is a tricky statistics problems (many *confounding* factors)

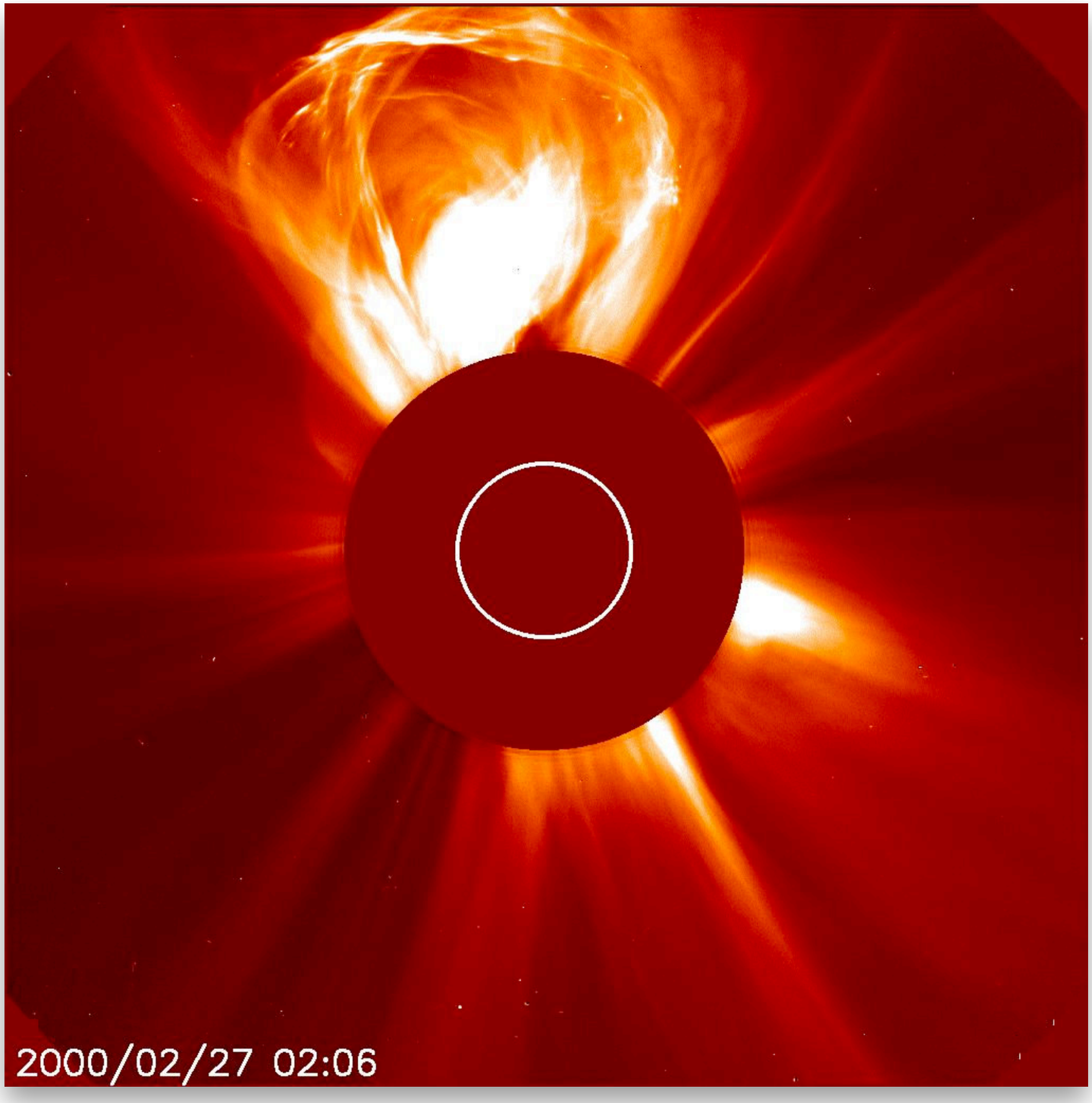
Best approaches exploit new giant datasets

Difficulties: optimization in a million dimensions

My group focuses on the tools to enable these computations

Solar superstorms

Coronal Mass Ejection



"All models are wrong, but some are useful" (George Box)

Tools:
optimization,
differential equations

Solar superstorms

How to model a **plasma**?

	Magnetohydrodynamics	Two Fluids	Gyrokinetics	Kinetics	Everything
Description	The plasma is one continuous fluid - ions have all the mass, but electron carry all the current.	Break the ions & electrons into two continuous, mingling fluids.	Only track superparticles' straight motion - and ignore the corkscrewing.	Assign particles a speed and location based on a distribution. Track super particles through space.	Track every particle, at all times.
Strengthens	~~Easily solved.~~	Simple bulk effects like drift waves & reconnection can be understood.	Captures most of kinetic model, but much easier to solve - can model an entire Tokamak.	Many things captured, can get powerful results like the linear velocity-space instabilities.	Most accurate model possible.
Weakness	Most things not captured: most plasma waves, leakage, kinetic instabilities, structures etc.	Many things not captured: plasma instabilities, large effects & non-equilibrium effects. Assumes bell curves.	Non-physical behavior over long times: resonances & adiabatic invariants can be lost.	Tough to solve: hard to apply to full-size reactors. Loses some effects: like plasma microdensity and collective thomson scattering.	Typically impossible to solve.
Mathematics	Navier-stokes, Lorentz force, Maxwells' equations.	Navier-stokes, Lorentz force, Maxwells' equations.	Vlasov-Maxwell Expansion Equation	Vlasov-Maxwell Equation	Klimontovich Model

Plasma as a fluid (~~Chalkboard~~) — Plasma as a gas (Computer Required)

Simplicity — Detail

Recent project: develop new **reduced order models** for plasmas (for space weather prediction)

Involves: Michigan State, University of Colorado Boulder, University of Washington, University of Delaware, Sandia National Laboratories, Oak Ridge National Laboratory, Los Alamos National Laboratory, Lawrence Livermore National Laboratory

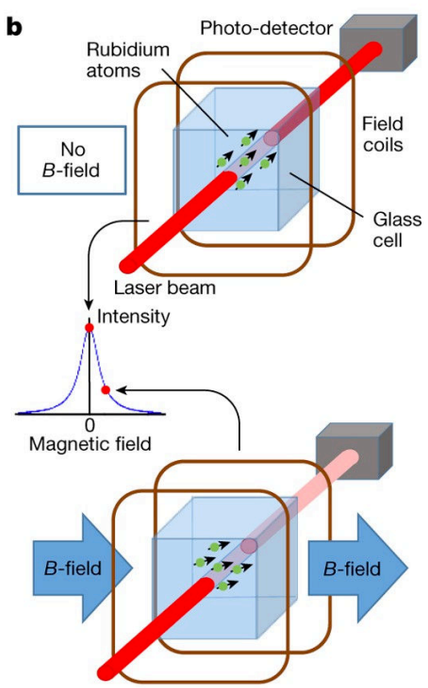
Brain imaging

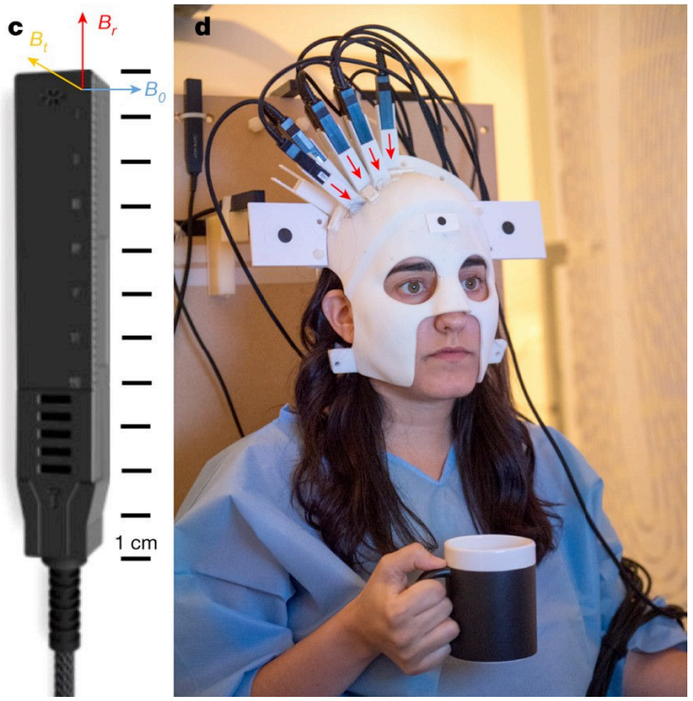
Magnetoencephalography (MEG)

Tools: optimization

Detect magnetic fields created by neurons firing in the brain

Our job: from the sensor readings, determine *where* those neurons are





with Prof. Svenja Knappe
Mechanical Engineering

Applications:

- Human-Computer Interaction
- Research on how brain works
- Surgery
- Neurofeedback … and mind reading (in 100 years maybe)

MRI

Magnetic Resonance Imaging

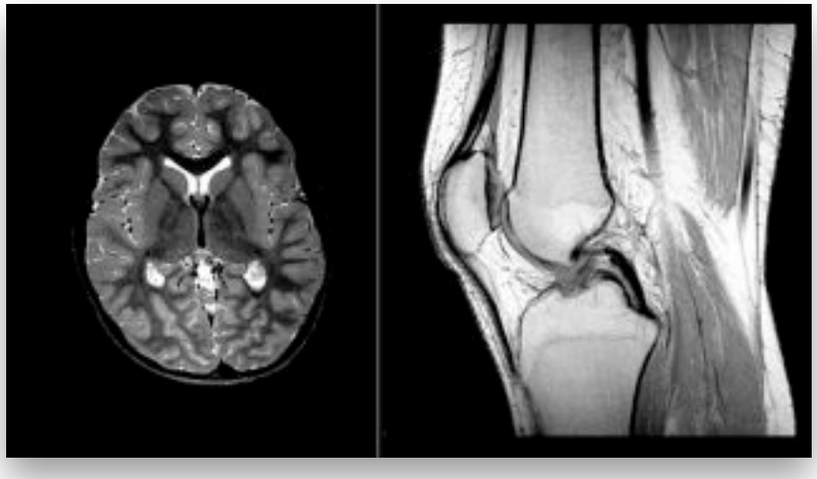
Tools:
optimization, ML

MRI gives excellent resolution… but it's slow

If you want to image the heart, and it takes a minute, that's a problem!

Goal: imaging fetal twins *in utero* for interventions

Use optimization, modeling, statistics and machine learning to make do with fewer measurements …so faster



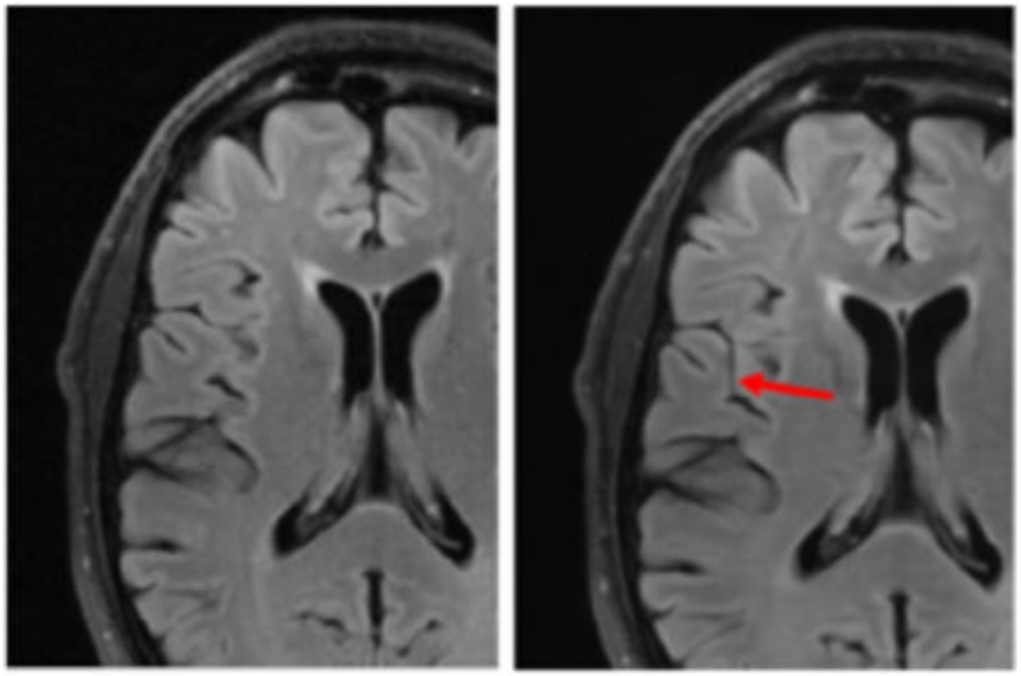
with Prof. Nick Dwork
CU Anschutz

MRI

Fixing problems with Deep Learning methods

Hallucinating anatomies

Ground truth Deep Learning



False sulcus from ResoNNance model. cf. Buckley et al. "Results of the 2020 **fastMRI** challenge for machine learning MR image reconstruction," IEEE Trans. Medical Imaging '21

Eliminating pathologies

Ground truth Sparse-SENSE Deep Learning

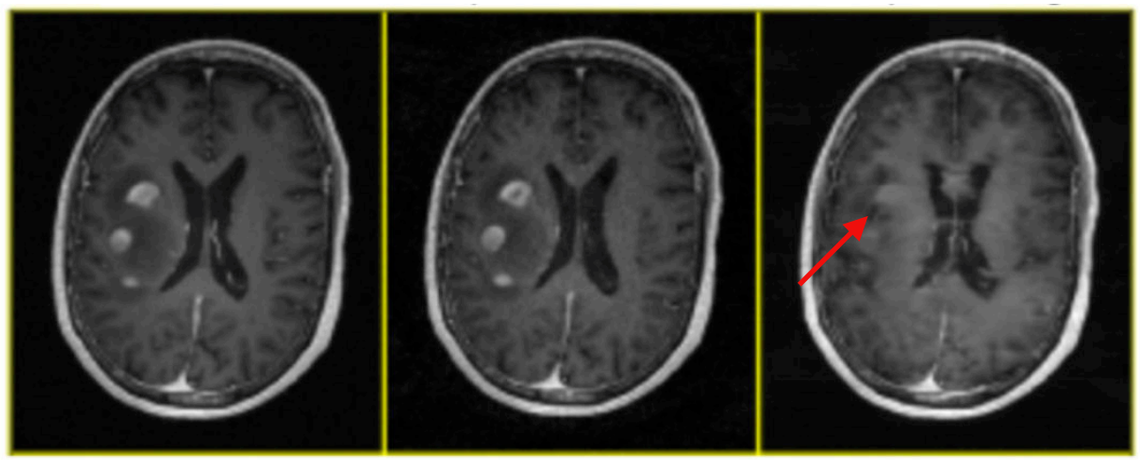
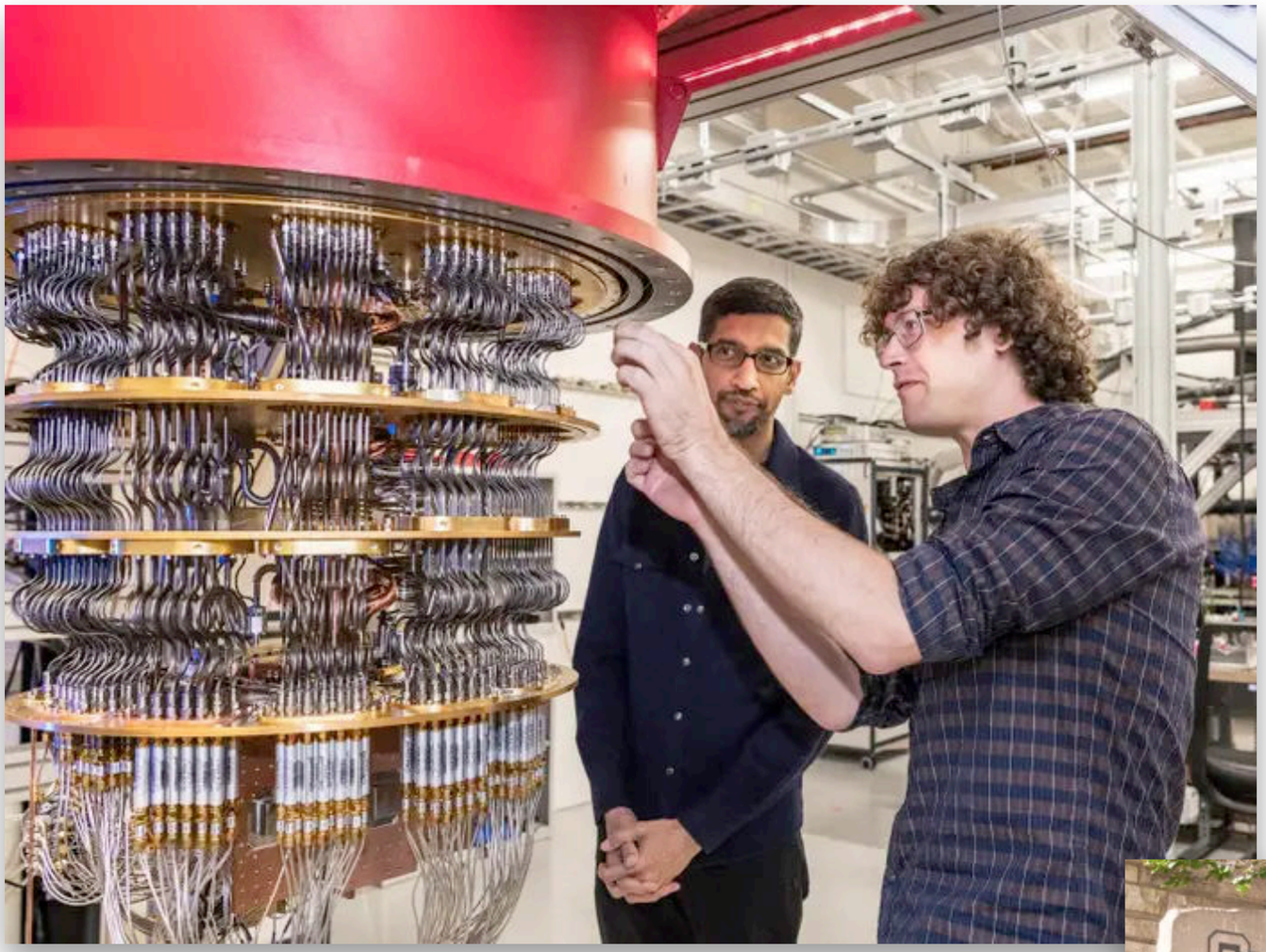


Figure from Nicholas Dwork

Quantum computing

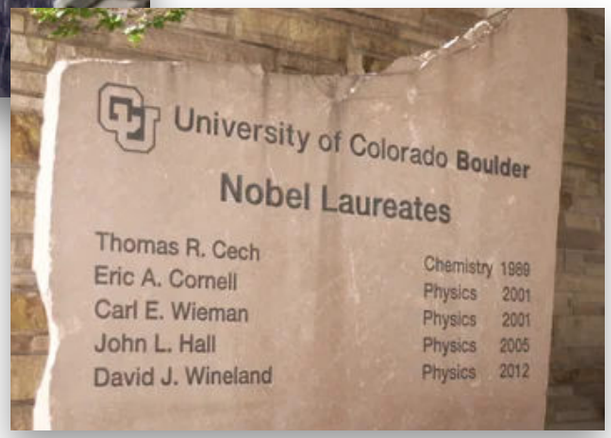
Tools:
- optimization
- statistics



Quantum computers are thought to be fundamentally more powerful than classical computers

"Seeing" what you just made isn't easy… quantum mechanics gives indirect and noisy measurements.

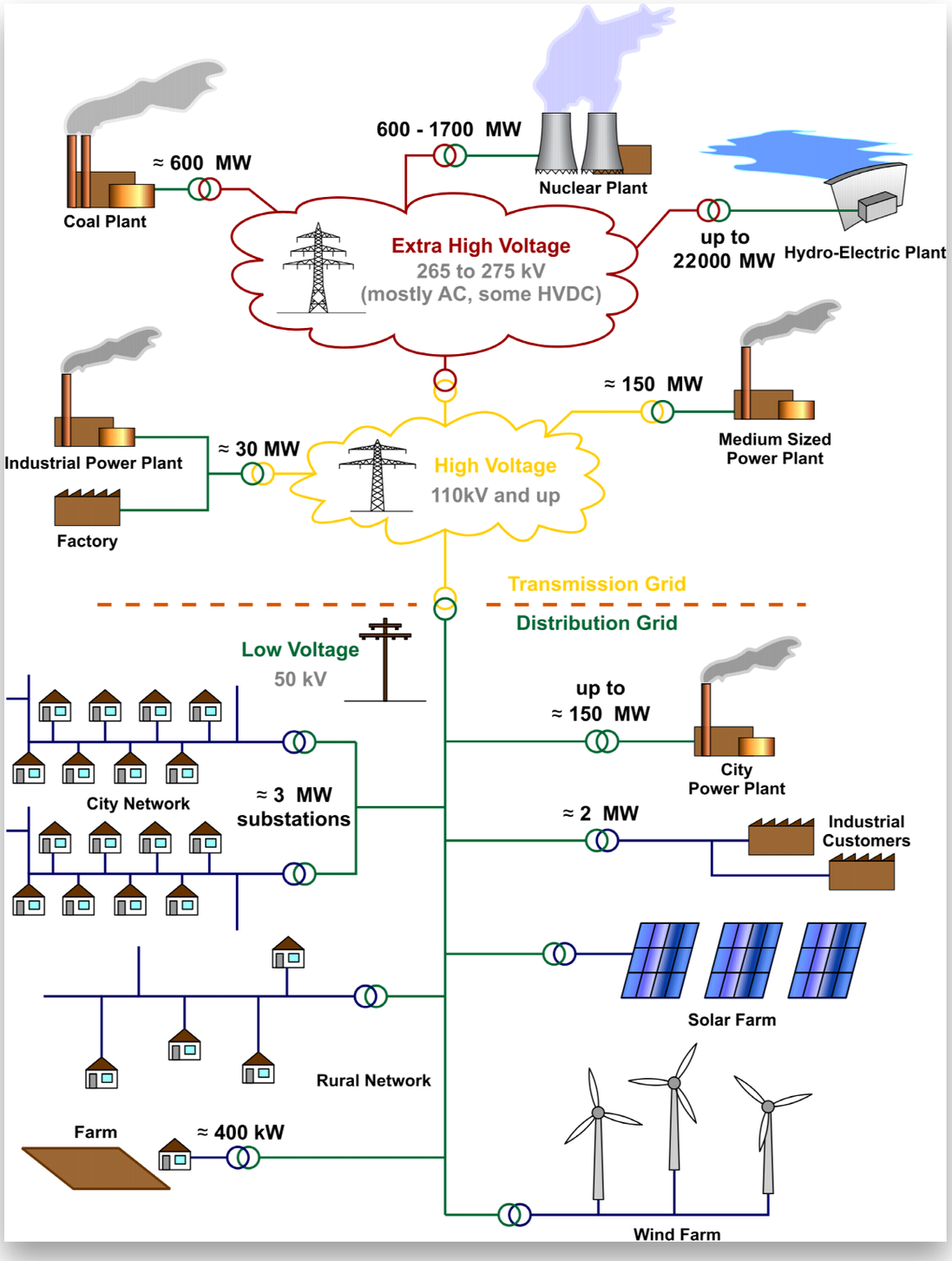
Our group use statistics and optimization to give a better diagnosis tool (a better "debugger")



with my PhD student
Akshay Seshadri
(who also works at NIST)

Tools:
optimization

Power Grid control



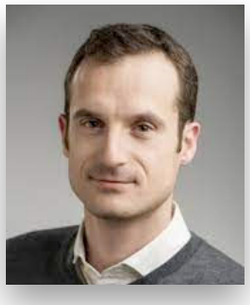
The power grid is old and prone to failure (blackouts), and ill-equipped to handle solar

One part of the solution is to **control** the grid on a faster time scale

(*control* can mean adjusting the output of a conventional power plant, typically done every 15 minutes)

Since solar output can change in less than a second (clouds!), we should control the grid every second

Our group helps with algorithms that can handle this fast pace



with Prof. Emiliano Dall'Anese
Electrical Engineering

Tools:
optimization

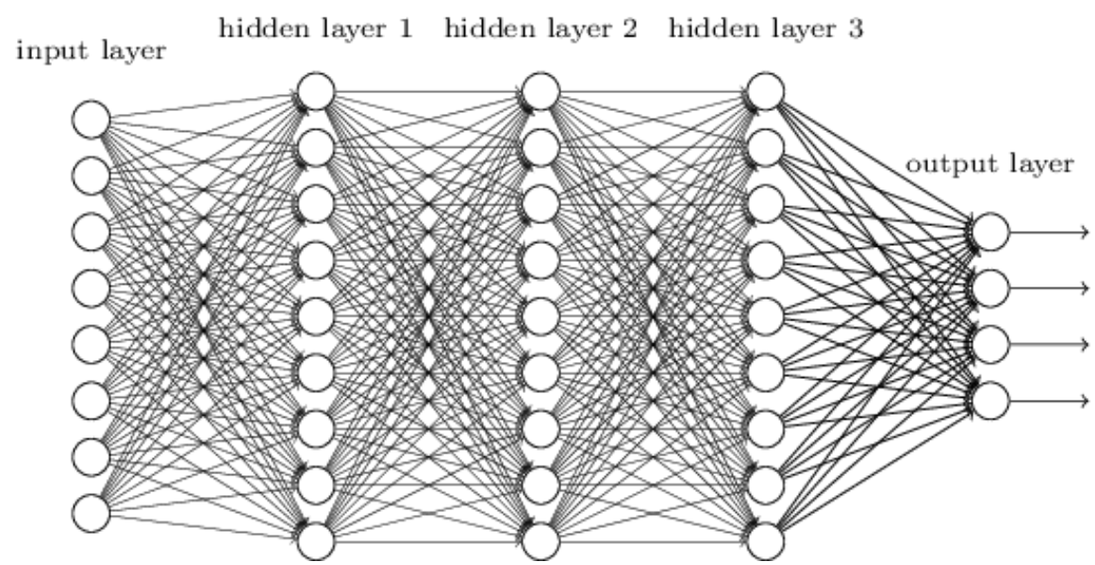
Training neural nets

Artificial neural nets are loosely inspired by our brain

In theory, if big enough, they can approximate any kind of relation…

… but training them is tricky. It's time consuming, and we don't understand why it works

The training is all about optimization (using **random** algorithms)

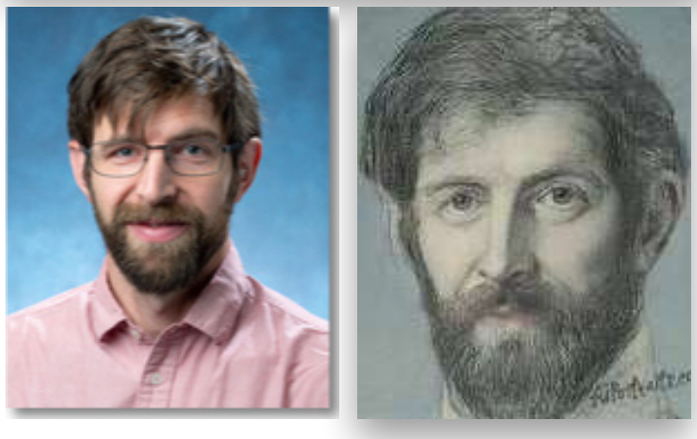


Some of our contributions are not new *tools* or *algorithms*, but **new theorems** *(guarantees)*, e.g.,

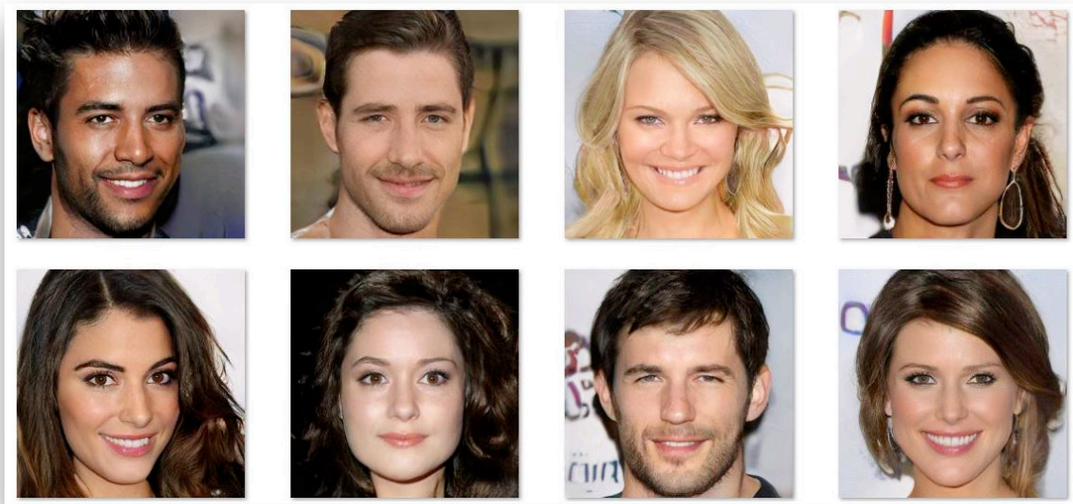
<u>Theorem</u>: choose any $0 < \delta < 1$. Then with probability greater than $1 - \delta$, after T iterations, the accuracy will be on the order of $\log(1/\delta)/T$ if using [this particular algorithm].

Neural net applications

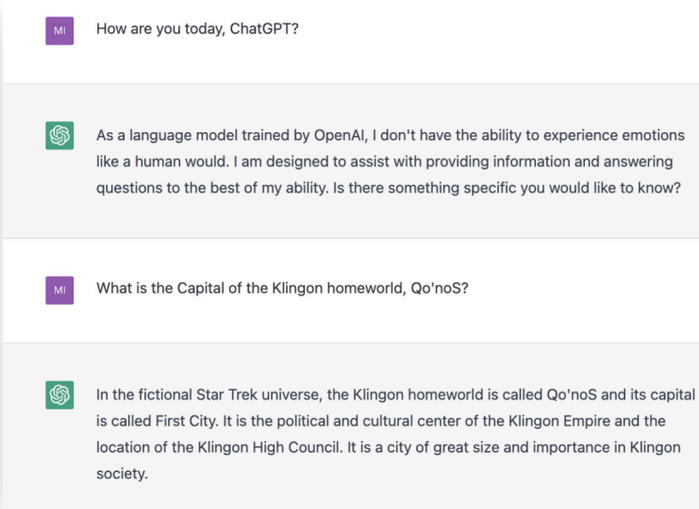
Style transfer



Creating fake celebrity faces



ChatGPT



Deep fakes

As well as many useful and serious applications:

- digital assistants (Alexa, Siri…), NLP
- scientific computing
- self-driving cars
- language translation
- healthcare
- fraud detection, cybersecurity, spam filtering

Randomized Linear Algebra

Tools:
linear algebra,
prob / stat

Classical (deterministic) framework

Orderly input
(e.g. a "basis")

Works great… if you give it enough input

New (stochastic) framework

Good enough, even with a few inputs

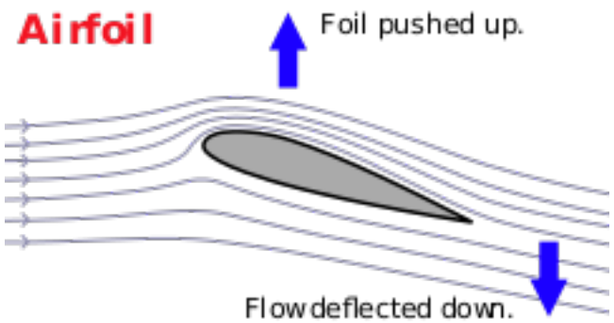
Why?

- Faster computation (or easier to parallelize)
- For physical systems (quantum, or geology, etc.)

Scientific Simulation

Collaboration with CU's aerospace engineering department

Tools:
optimization,
differential equations



Air is a fluid. Study **fluid dynamics** for understanding airplane wings etc. (and designing better ones)

Must solve these equations numerically ("Navier Stokes" differential equations). $1,000,000 prize for proving properties about their solutions

Challenges:

- Most accurate simulations are too slow… use reduced models to speed things up
- Even so, we run on giant computers (recently gained access to Aurora)
- Storing the data is a problem… need specialized **compression algorithms**

10^{18} 10^{9} 10^{12}

$500M, 2 exaFlops, 9000 multicore nodes (an i7 CPU is 289 gigaFlops, Playstation 5 is 10 teraFlops)

Scientific Simulation

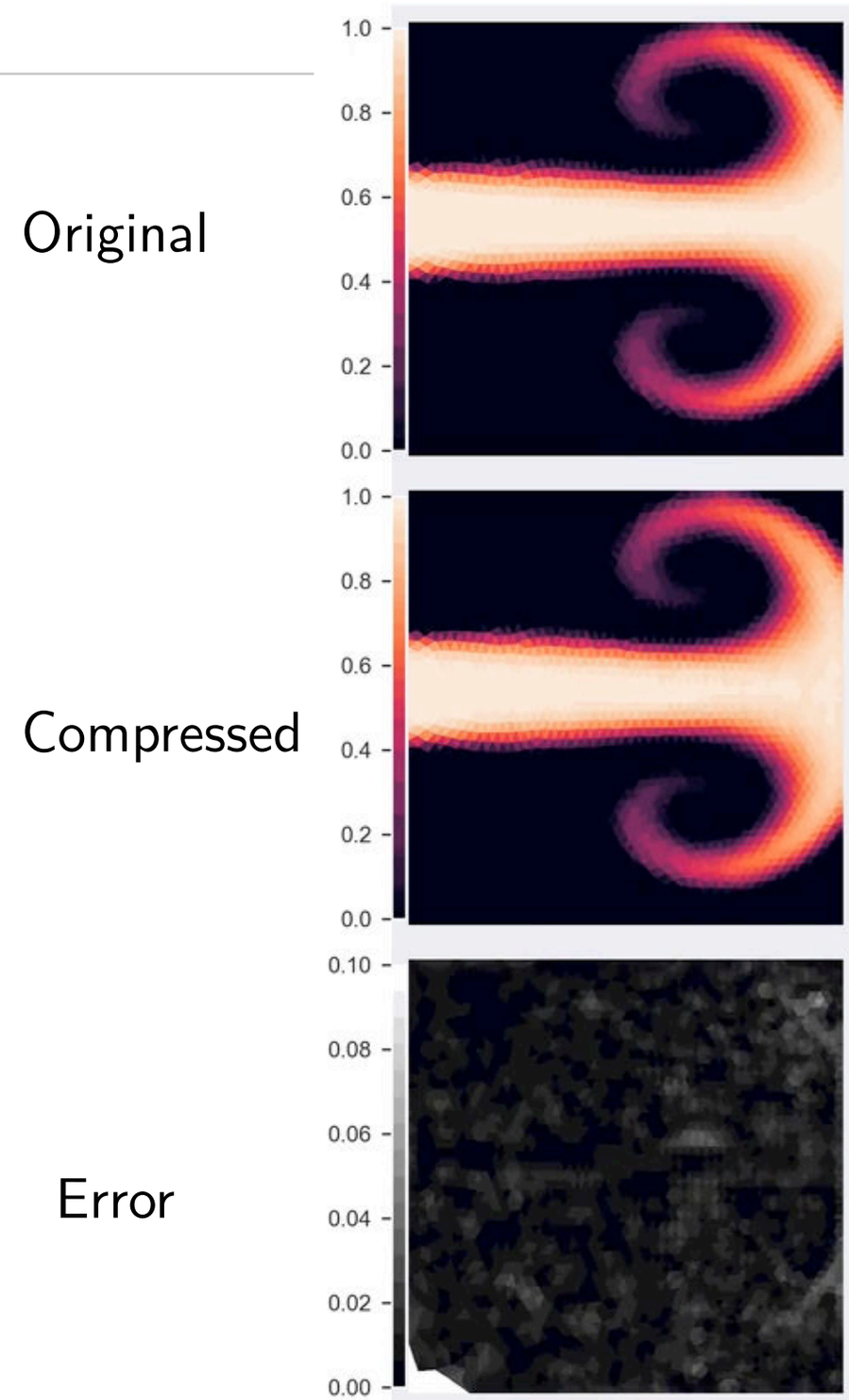
We have two main approaches to compression:

1. Randomized numerical linear algebra
2. Machine learning

both have advantages and disadvantages.

Our goal is to do the compression "on-the-fly" during a simulation

880x compression



Part 2

Applied Mathematics

COLLEGE OF ARTS AND SCIENCES

Academics | Research | News & Events | Organizations

Research

Research by members of the Department of Applied Mathemat
mathematical modeling, analysis, and computation—in order to
is identified with at least one of six primary research areas liste

Research Areas

- Computational Mathematics
- Mathematical Biology
- Mathematical Geosciences
- Applied Nonlinear PDEs and Dynamics
- Statistics and Data Science
- Stochastic Processes and Applications

my core interests

Most researchers
span several subfields

Q: is there **math** behind **computing?** Isn't this just software development?

or "numerical analyst"

Part 2: What does a "computational mathematician" do

A famous example, from Volker Strassen (1969)
[also a probabilist and "theoretical computer scientist"]

Matrix Multiplication

0	8	9	4	7
-6	5	2	18	-4
3	6	9	-2	4
9	-6	78	23	8
8	2	-3	9	11

×

2	9	5	9	0
8	-2	-6	8	99
23	5	5	-9	12
48	6	8	9	3
2	1	-5	7	6

=

…	52	…		

3 x 9 + 6 x -2 + 9 x 5 + -2 x 6 + 4 x 1 = 52

Matrix Multiplication

0	8	9	4	7
-6	5	2	18	-4
3	6	9	-2	4
9	-6	78	23	8
8	2	-3	9	11

$\times$

2	9	5	9	0
8	-2	-6	8	99
23	5	5	-9	12
48	6	8		
2	1	-5		

$=$

	52			

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,206.00	1,714.81	22,786
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

TOP 500 The List.

Why is this important?

- It represents **linear transformations** and is the backbone of science
- It's the building block of most **scientific codes** (e.g., climate modeling)
- … as well as behind all **machine learning** training (ChatGPT…)

Matrix Multiplication

n $\longleftrightarrow$ n $\updownarrow$

0	8	9	4	7
-6	5	2	18	-4
3	6	9	-2	4
9	-6	78	23	8
8	2	-3	9	11

$\times$

n $\longleftrightarrow$

2	9	5	9	0
8	-2	-6	8	99
23	5	5	-9	12
48	6	8	9	3
2	1	-5	7	6

$=$

…	52	…		

About **how many operations are required**?

e.g., $\mathcal{O}(2^n)$? $\mathcal{O}(e^n)$? $\mathcal{O}(\log(n))$?

$\mathcal{O}(n)$? $\mathcal{O}(n^2)$? $\mathcal{O}(n^3)$? $\mathcal{O}(n^4)$?

$\mathcal{O}(n!)$?

Aside: "big O" notation

$f(n) = \mathcal{O}\big(g(n)\big)$ means (informally) that f "grows" no faster than a constant times g

e.g., $4n^2 + 3n - 2 = \mathcal{O}(n^2)$ $\qquad$ $2^n \neq \mathcal{O}(n^{100})$

Matrix Multiplication

$$
\begin{array}{c}
\longleftarrow n \longrightarrow \\
\uparrow \\
n \\
\downarrow
\end{array}
\begin{bmatrix}
0 & 8 & 9 & 4 & 7 \\
-6 & 5 & 2 & 18 & -4 \\
3 & 6 & 9 & -2 & 4 \\
9 & -6 & 78 & 23 & 8 \\
8 & 2 & -3 & 9 & 11
\end{bmatrix}
\times
\begin{array}{c}
\longleftarrow n \longrightarrow \\
\begin{bmatrix}
2 & 9 & 5 & 9 & 0 \\
8 & -2 & -6 & 8 & 99 \\
23 & 5 & 5 & -9 & 12 \\
48 & 6 & 8 & 9 & 3 \\
2 & 1 & -5 & 7 & 6
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
 & & & & \\
 & & & & \\
\dots & 52 & \dots & & \\
 & & & & \\
 & & & &
\end{bmatrix}
$$

About **how many operations are required**?

e.g., $\mathcal{O}(2^n)$? $\mathcal{O}(e^n)$? $\mathcal{O}(\log(n))$?

$\mathcal{O}(n)$? $\mathcal{O}(n^2)$? $\mathcal{O}(n^3)$? $\mathcal{O}(n^4)$?

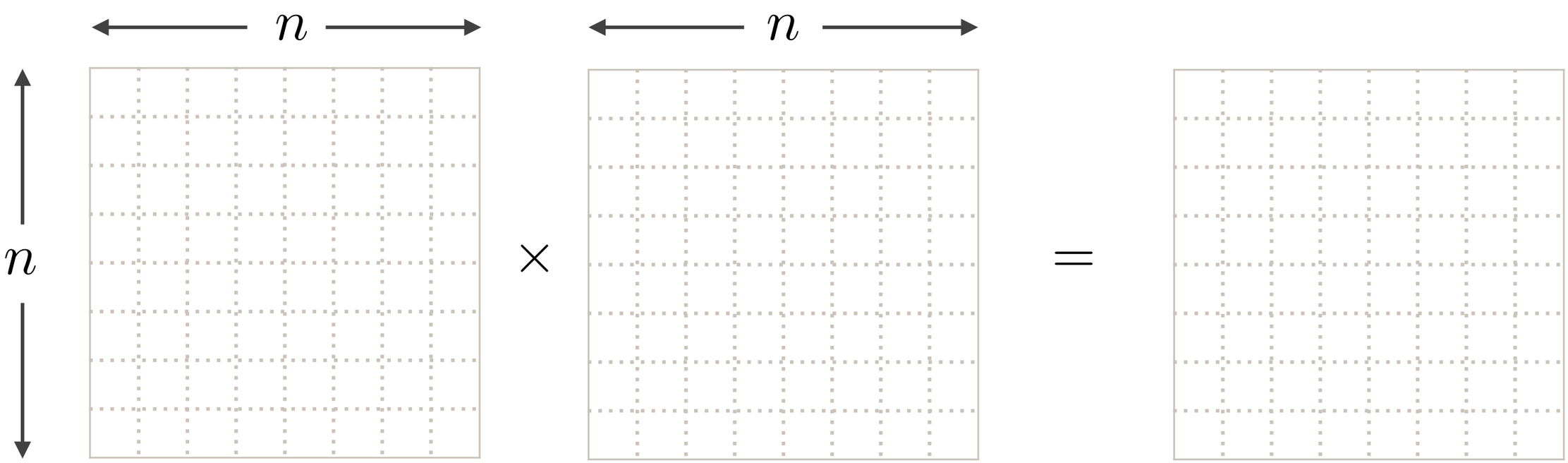
$\mathcal{O}(n!)$?

And how many operations to **add** two matrices?

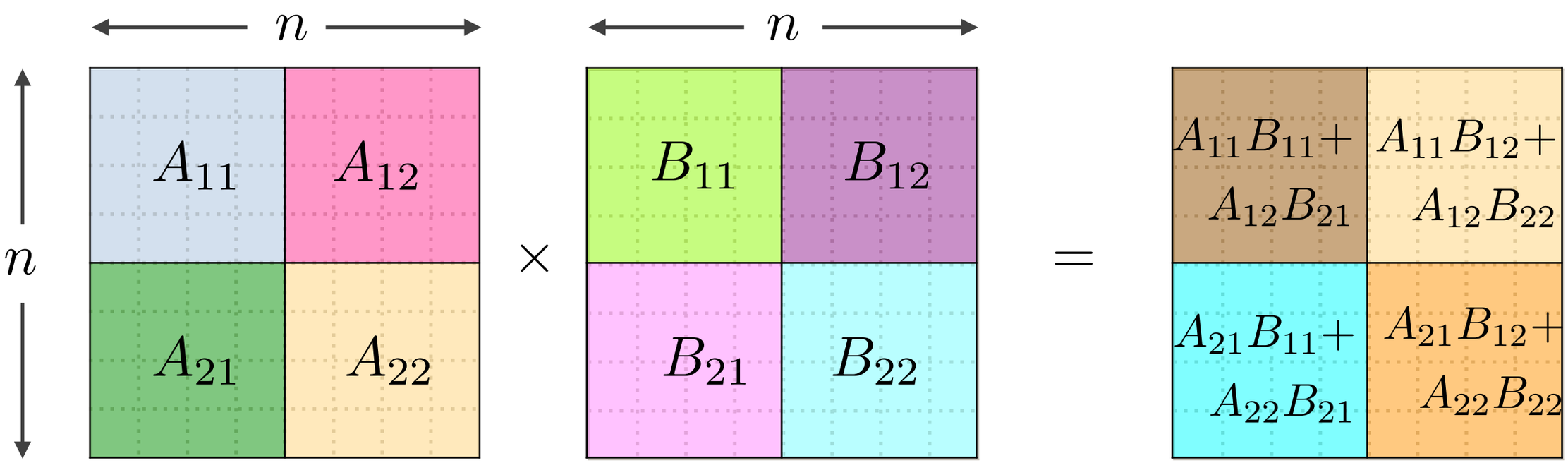
$\mathcal{O}(n^2)$

(Matrix) addition is **cheaper** than (matrix) multiplication!

Matrix Multiplication… via recursion



Matrix Multiplication… via recursion



That's interesting, but it doesn't reduce the number of operations.
Say, we can multiply n x n matrices in exactly n^3operations

$$A_{11}B_{11} + A_{12}B_{21}$$

$\frac{n}{2} \times \frac{n}{2}$ sized matrices

$$8 \times \frac{n^3}{8} = n^3$$

$\left(\frac{n}{2}\right)^3 = \frac{n^3}{8}$ operations to multiply

[activity]

Multiply two complex numbers

$$\underbrace{(a + ib)}_{z_1} \times \underbrace{(c + id)}_{z_2} = (ac - bd) + i(bc + ad)$$

The standard formula has **4** multiplies

Can you do it in less? *Hint: the standard formula has 2 <u>additions</u>. You'll need* ***more***

[activity]

Multiply two complex numbers

$$\underbrace{(a+ib)}_{z_1} \times \underbrace{(c+id)}_{z_2} = (ac-bd) + i(bc+ad)$$

The standard formula has **4** multiplies

Can you do it in less? *Hint: the standard formula has 2 <u>additions</u>. You'll need* ***more***

ONE SOLUTION

$$(ac-bd) = \overbrace{a(c-d)}^{w} + \underbrace{d(a-b)}_{x}$$

$$(bc+ad) = \underbrace{b(c+d)}_{y} + \overbrace{d(a-b)}^{x}$$

Only 3 multiplies, but 5 additions

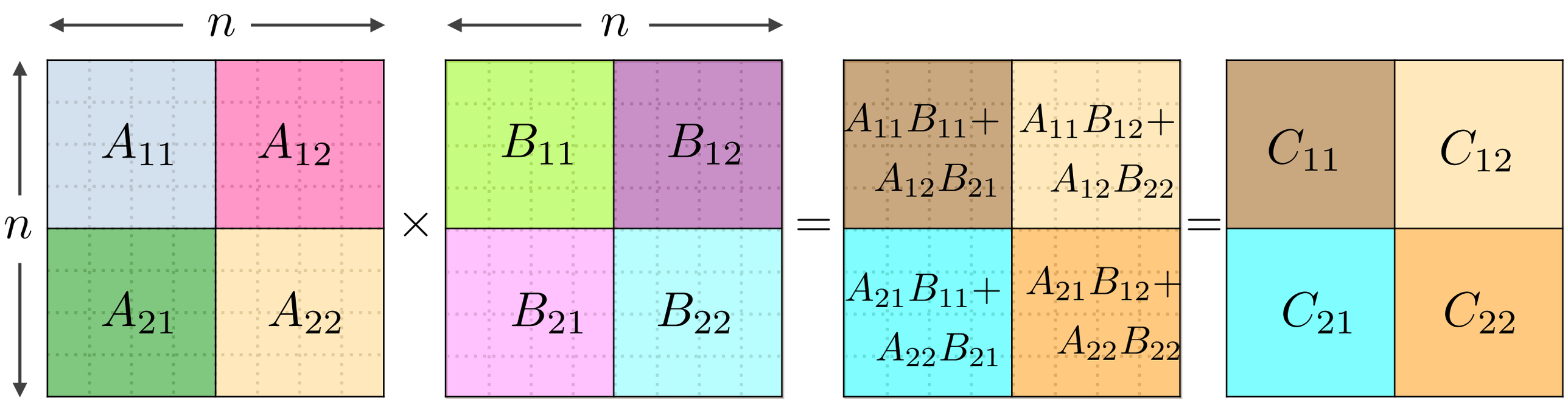
ANOTHER SOLUTION

$$(ac-bd) = \overbrace{(a+b)(c-d)}^{w} + \overbrace{ad}^{x} - \overbrace{bc}^{y}$$

$$(bc+ad) = \underbrace{bc}_{y} + \underbrace{ad}_{x}$$

Due to Peter Unger 1963; similar to fast two-digit multiplication (Anatoly Karatsuba 1960)

Matrix Multiplication… via recursion



Can we write this in fewer than 8 (block) multiplies?

$$M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22})$$
$$M_2 = (A_{21} + A_{22}) \times B_{11}$$
$$M_3 = A_{11} \times (B_{12} - B_{22})$$
$$M_4 = A_{22} \times (B_{21} - B_{11})$$
$$M_5 = (A_{11} + A_{12}) \times B_{22}$$
$$M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12})$$
$$M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22})$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{bmatrix}$$

Yes! 7 multiplies (though 18 instead of 4 additions)

(assuming n is a power of 2 for simplicity)

Complexity analysis

a constant, e.g., 18

recursion relationship

$$w(n) = 7w\left(\frac{n}{2}\right) + cn^2$$

w is *work* (i.e., # of operations)

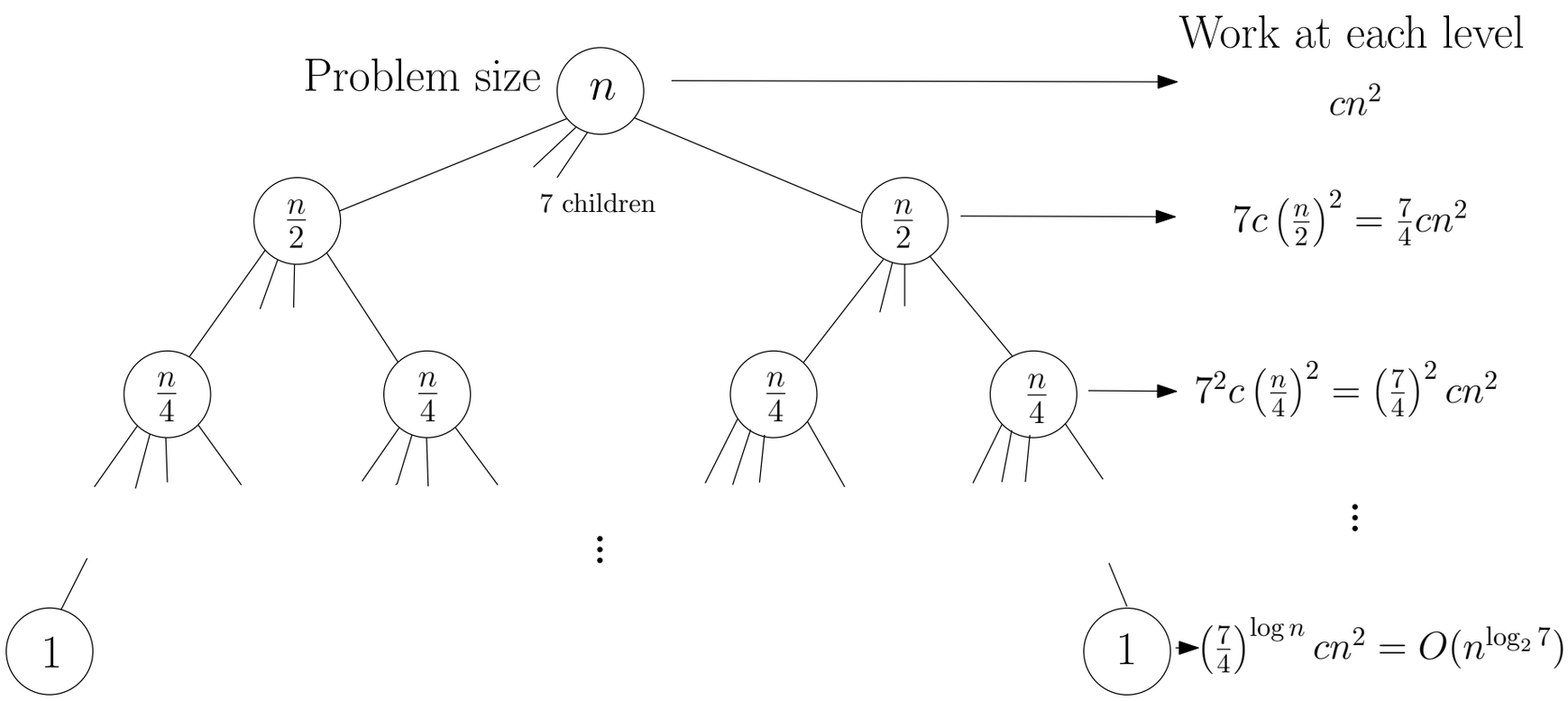


figure from https://www.cs.cmu.edu/afs/cs/academic/class/15750-s17/ScribeNotes/lecture1.pdf, scribe David Witmer, Lecturer: Gary Miller

$$w(n) = cn^2\left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right)$$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$w(n) = cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right)$$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$w(n) = cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right)$$

$$= cn^2 \frac{\frac{7}{4}^{\log_2(n)+1} - 1}{\frac{7}{4} - 1}$$

recall $\displaystyle\sum_{k=0}^{m} r = \frac{r^{m+1} - 1}{r - 1}$

if $r \neq 1$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$
\begin{aligned}
w(n) &= cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right) \\
&= cn^2 \, \frac{\frac{7}{4}^{\log_2(n)+1} - 1}{\frac{7}{4} - 1} \\
&\leq \tilde{c} n^2 \left(\frac{7}{4}\right)^{\log_2(n)} \\
&= \tilde{c} \, 7^{\log_2(n)} \qquad \text{since } \; 4^{\log_2(n)} = 2^{2\log_2(n)} = \left(2^{\log_2(n)}\right)^2 = n^2
\end{aligned}
$$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$
\begin{aligned}
w(n) &= cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right) \\
&= cn^2 \, \frac{\frac{7}{4}^{\log_2(n)+1} - 1}{\frac{7}{4} - 1} \\
&\leq \tilde{c} n^2 \left(\frac{7}{4}\right)^{\log_2(n)} \\
&= \tilde{c} \, 7^{\log_2(n)} \\
&= \tilde{c} \, (2^{\log_2(7)})^{\log_2(n)} \\
&= \tilde{c} \, (2^{\log_2(n)})^{\log_2(7)}
\end{aligned}
$$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$
\begin{aligned}
w(n) &= cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right) \\
&= cn^2 \, \frac{\frac{7}{4}^{\log_2(n)+1} - 1}{\frac{7}{4} - 1} \\
&\leq \tilde{c} n^2 \left(\frac{7}{4}\right)^{\log_2(n)} \\
&= \tilde{c}\, 7^{\log_2(n)} \\
&= \tilde{c}\, (2^{\log_2(7)})^{\log_2(n)} \\
&= \tilde{c}\, (2^{\log_2(n)})^{\log_2(7)} \\
&= \tilde{c}\, n^{\log_2(7)} \\
&\approx \tilde{c}\, n^{2.807}
\end{aligned}
$$

(assuming n is a power of 2 for simplicity)

Complexity analysis

$$
\begin{aligned}
w(n) &= cn^2 \left(1 + \frac{7}{4} + \left(\frac{7}{4}\right)^2 + \left(\frac{7}{4}\right)^3 + \ldots + \left(\frac{7}{4}\right)^{\log_2(n)}\right) \\
&= cn^2 \, \frac{\frac{7}{4}^{\log_2(n)+1} - 1}{\frac{7}{4} - 1} \\
&\leq \tilde{c} n^2 \left(\frac{7}{4}\right)^{\log_2(n)} \\
&= \tilde{c}\, 7^{\log_2(n)} \\
&= \tilde{c}\, (2^{\log_2(7)})^{\log_2(n)} \\
&= \tilde{c}\, (2^{\log_2(n)})^{\log_2(7)} \\
&= \tilde{c}\, n^{\log_2(7)} \\
&\approx \tilde{c}\, n^{2.807}
\end{aligned}
$$

Complexity of Strassen multiplication is $\mathcal{O}(n^{2.807})$

Can also use the "Master Theorem" or guess-and-prove-via-induction

Going further

There are even better algorithms than Strassen!

As of January 2024, the fastest known is $\mathcal{O}(n^{2.371})$

It's conjectured that for any $\epsilon > 0$ there is an algorithm that runs in $\mathcal{O}(n^{2+\epsilon})$

- Most are not practical. Strassen is one of the more practical ones but still not used often in practice for various reasons

- Finding faster algorithms is related to **tensor factorization**
  - Intersection of *computer science* and *numerical analysis*
  - Prof. Josh Grochow in CS dept (and myself) work on it

- Recent hype over using machine learning (specifically, reinforcement learning) to find good algorithms!


nature

Explore content | About the journal | Publish with us

nature > articles > article

Article | Open access | Published: 05 October 2022

Discovering faster matrix multiplication algorithms with reinforcement learning

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis & Pushmeet Kohli

Nature **610**, 47–53 (2022) | Cite this article

593k Accesses | **192** Citations | **3580** Altmetric | Metrics