

# Deep Learning in Spatial Transcriptomics

## Introductions

Caleb Hallinan

01/06/2024

# Quick Intro to Me



- 3rd year BME PhD Student in Dr. Jean Fan's lab
- Went to UVA for undergrad, lived in Boston for 2 years
- Research work is in deep learning for spatial transcriptomics :)
- Owner of a crazy dog named Eeko
- Celia, my wife, is a BCMB PhD Student in Dr. Mikala Egeblad's lab
- Love to hang with friends and play board games and sports!



# Quick Intro of you!

- Let's go around and share your:
  1. Name
  2. Year
  3. Why you chose to take this class
  4. A fun fact!

Introduce  
yourself



# Goals of the Class (on paper)

- Introduce the field of spatial transcriptomics and how the varying technologies work
- Teach the basics of deep learning, including different neural networks and how they're used in spatial transcriptomics
- Develop proficiency in PyTorch, a python-based deep learning library, to enable students to effectively utilize neural networks
- Guide students through completing a deep learning project using real-world spatial transcriptomics data, and improve their presentation skills by sharing their work

# Goals of the Class (my hope)

- Teach you about spatial transcriptomics and deep learning and how they interact
- Working knowledge of pytorch (DL framework in python) and spatial transcriptomics data in python
- But what do you want to get out of this class the most?
  - Survey at end class

# Syllabus

- No hws, no outside readings, no extra work
- All you have to do is show up and participate ☺
- 10 classes total, outline tbd
- Always happy to send you more material or chat, just email me
  - [challin1@jh.edu](mailto:challin1@jh.edu)

# GitHub for class

The screenshot shows a GitHub repository page for the user 'calebhallinan' named 'pytorch4st'. The repository is private. The main interface includes a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository summary shows 1 branch and 0 tags. A list of recent commits by 'calebhallinan' is displayed, including edits to 'policies', updates to '.gitignore', an initial commit to 'LICENSE', edits to 'README.md', and the addition of 'misc.py' to the repo. The 'About' section notes that there is no description, website, or topics provided. It also lists the Readme, Apache-2.0 license, activity (0 stars, 1 watching, 0 forks), releases (none published), packages (none published), and languages (Python 100.0%). A 'Suggested workflows' section based on the tech stack is shown at the bottom.

calebhallinan / pytorch4st

Type  to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Unwatch 1 Fork 0 Star 0

pytorch4st Private

main 1 Branch 0 Tags Go to file Add file Code

calebhallinan edit readme 8888b80 · 19 hours ago 7 Commits

policies edits 19 hours ago

.gitignore updates 8 months ago

LICENSE Initial commit last year

README.md edit readme 19 hours ago

misc.py add misc.py to the repo 9 months ago

README Apache-2.0 license

## HEART: Deep Learning for Spatial Transcriptomics

Class lectures, code, data, etc.

About

No description, website, or topics provided.

Readme Apache-2.0 license Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

Python 100.0%

Suggested workflows

Based on your tech stack

Publish Python Configure

# Please ask questions at any time!

- I may not always know the answer, but will do my best and get back to you if I don't ☺

# Outline for today

- Intro to the git/github and how to work with it
- Brief overview of Deep Learning
- Brief overview of Spatial Transcriptomics

# Github Intro

# GitHub Overview

- All class material will be posted on GitHub
- Will go through a quick GitHub tutorial here!
- Next slides are heavily based on Dr. Stephanie Hicks' git slides!
  - [https://docs.google.com/presentation/d/1MyhQ9JMXwpkv6xyOLs-FTE5ZPzweikj-NAaHIZpbZBQ/edit#slide=id.g31018913631\\_0\\_73](https://docs.google.com/presentation/d/1MyhQ9JMXwpkv6xyOLs-FTE5ZPzweikj-NAaHIZpbZBQ/edit#slide=id.g31018913631_0_73)
- Who has a GitHub account and has worked with github?

# "FINAL".doc



FINAL.doc!



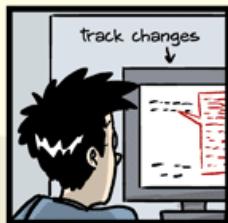
FINAL\_rev.2.doc



↑  
FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



track changes  
↓  
JORGE CHAM © 2012  
FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.#@\$%WHYDID  
ICOMETOGRAD SCHOOL????.doc



# git



- **git** is a version control system used primarily to manage and track changes in files and source code during software development
  - Easy to work with collaborators
  - Secure backup and recovery
- Technically, git is a command-line tool or has a CLI (command line interface)
- Similar to the “track changes” features in Microsoft Word, but more rigorous, powerful, and scaled up to multiple files
- Version control systems start with a base version of the document and then record changes you make each step of the way

# GitHub

- GitHub is a hosting service on internet for git-aware folders and projects
  - Similar to DropBox or Google, but more structured, powerful, and programmatic
  - Great for solo or collaborative work!
  - GitHub is distinct from Git. However, GitHub is in some sense the interface and Git the underlying engine (a bit like RStudio and R).
- Note: Other interfaces to Git exist, e.g., Gitlab, Bitbucket, etc

This screenshot shows a GitHub profile for a user named 'calebhallinan'. The profile includes a large circular profile picture of a smiling man holding a dog. Below the picture, the user's name 'Caleb Hallinan' and GitHub handle 'calebhallinan' are displayed, along with the text 'BME Ph.D. student at JHU'. A 'Edit profile' button is present. The 'Pinned' section lists four repositories: 'appliedgenomics2024', 'calebhallinan.github.io', 'jefworks-lab.github.io', and 'applied\_genomics\_final\_project'. The 'Contribution activity' section features a heatmap showing contributions over time, with a legend indicating 'Less' and 'More'. The 'Achievements' section shows two achievements, each with a blue trophy icon.

This screenshot shows a GitHub repository page for 'pytorch4st'. The repository has 1 branch and 0 tags. The 'About' section contains the text: 'Hopefully will be a template for a pytorch class I teach J-term 2025 at JHU'. It also lists the 'Readme', 'Apache-2.0 license', and 'Activity' sections. The 'Releases' section notes 'No releases published' and 'Create a new release'. The 'Packages' section states 'No packages published' and 'Publish your first package'.

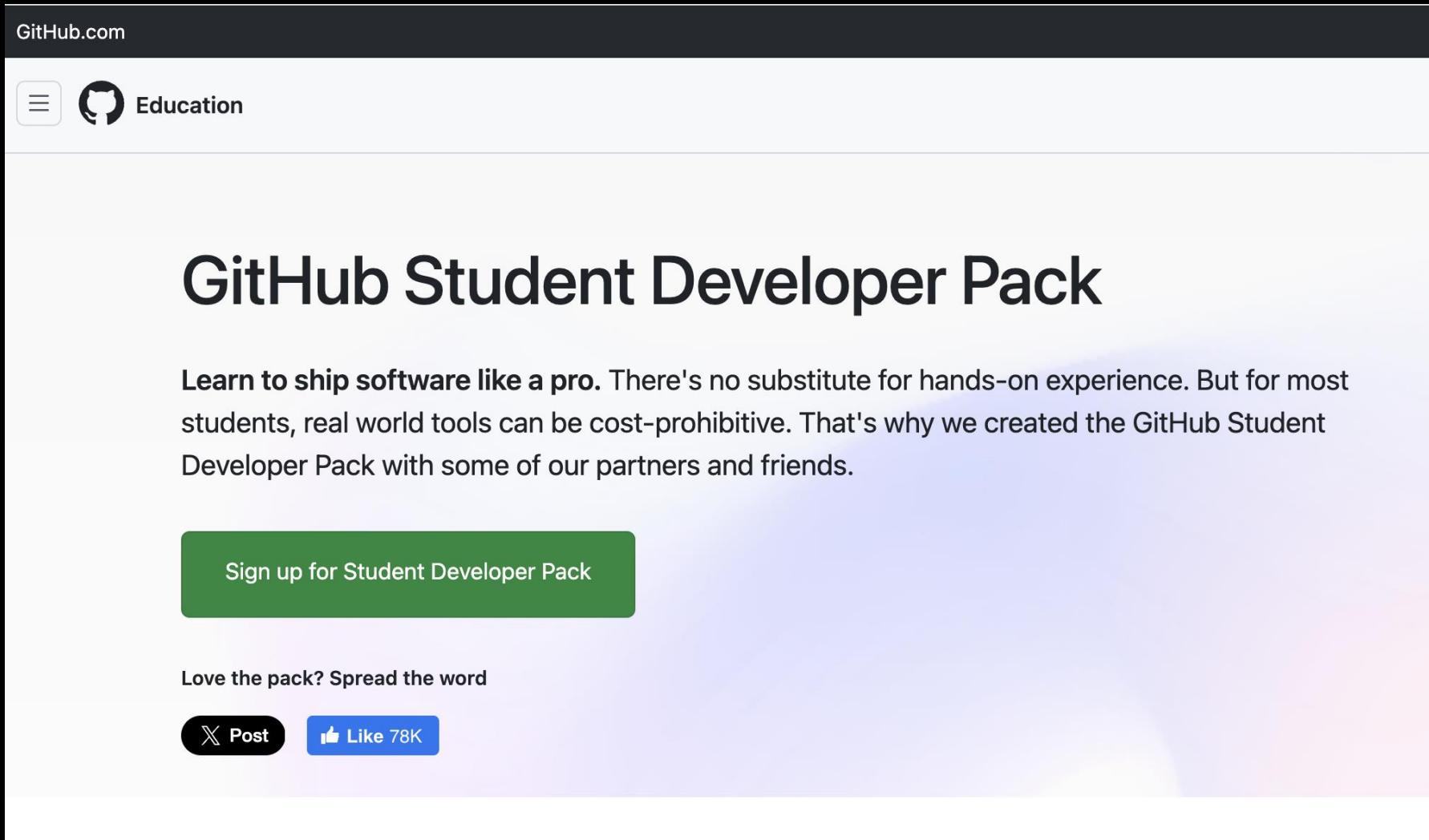
# When to use GitHub

- **GitHub is ideal for:**
  - A project with a fair number of files, most of those files are text files (such as code, LaTeX, etc.) and different people work on different parts of the project.
- **GitHub is less useful for:**
  - Lots of non-text files (e.g. Word or Powerpoint) and when other team members want to edit the same document at the same time
    - Recommend to use Overleaf, Google Docs, Word+Dropbox, Word+Onedrive, etc.

# Let's learn how to use git/GitHub

- Git and GitHub is fundamentally based on commands you type into the command line
  - This is how I almost always interact with git/GitHub
  - I highly recommend learning this way!
- However, many people find this the most confusing way to use git/GitHub. Alternatively, there are GUIs
  - GitHub itself provides a graphical interface with basic functionality
  - RStudio also has GUI git/GitHub integration
  - VSCode has GUI git/Github integration – I use this a lot too

# Free GitHub pro for students!



The image is a screenshot of a web browser displaying the GitHub Student Developer Pack landing page. The page has a dark header with the GitHub logo and 'GitHub.com'. Below the header, there's a navigation bar with a menu icon, the GitHub logo, and the word 'Education'. The main content area features a large, bold title 'GitHub Student Developer Pack' in black font. Below the title is a paragraph of text: 'Learn to ship software like a pro. There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends.' At the bottom of the main section is a green button with white text that says 'Sign up for Student Developer Pack'. Below this button is a light blue footer section containing the text 'Love the pack? Spread the word' and two social sharing buttons: one for 'Post' on a platform that looks like X (formerly Twitter) and another for 'Like 78K' on Facebook.

GitHub.com

≡ GitHub Education

# GitHub Student Developer Pack

Learn to ship software like a pro. There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends.

Sign up for Student Developer Pack

Love the pack? Spread the word

X Post

Like 78K

# Setting up git and GitHub

# git config

- When you use Git on a new computer for the first time, we need to configure a few things:
  - name and email address
  - what your preferred text editor is
  - and that we want to use these settings globally (i.e. for every project)
- On a command line, Git commands are written as git verb options, where verb is what we actually want to do and options is additional optional information.
- Here is how to set up Git on a new laptop with git config:
  - `$ git config --global user.name "My Name"`
  - `$ git config --global user.email "myemail@email.com"`
- This username and email will be associated with your subsequent Git activity



calebhallinan — calebhallinan@Clinks-MacBook-Pro — -zsh — 121x17

~

```
[~ > git config --global user.name  
calebhallinan  
[~ > git config --global user.email  
cph4mb@virginia.edu  
~ > █
```

```
Py pytorch4st ]  
Py pytorch4st ]  
Py pytorch4st
```



# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` .git` folder where git will store this history of the repository



pytorch4st — calebhallinan@Clinks-MacBook-Pro — -zsh — 121x22

+

..es/pytorch4st

```
[~ > git config --global user.name          Py pytorch4st ]
calebhallinan
[~ > git config --global user.email         Py pytorch4st ]
cph4mb@virginia.edu
[~ > cd Desktop/jhu/teach_classes/pytorch4st      Py pytorch4st ]
[~/De/j/teach_classes/pytorch4st main !2 ?3 > ls      Py pytorch4st ]
LICENSE      README.md    assignments code      forms      lectures      misc.py      policies
[~/De/j/teach_classes/pytorch4st main !2 ?3 > ls -lsa      Py pytorch4st ]
total 56
0 drwxr-xr-x  12 calebhallinan  staff   384 Nov 15 19:54 .
0 drwxr-xr-x   3 calebhallinan  staff    96 Jul 29 12:46 ..
16 drwxr-xr-x  1 calebhallinan  staff  6148 Sep  5 14:20 DS_Store
0 drwxr-xr-x  15 calebhallinan  staff   480 Nov 16 14:15 .git
8 -rw-r--r--@  1 calebhallinan  staff     89 Jul 29 12:46 README.md
0 drwxr-xr-x   2 calebhallinan  staff    64 Jul 29 12:49 assignments
0 drwxr-xr-x   3 calebhallinan  staff    96 Aug 29 19:23 code
0 drwxr-xr-x   7 calebhallinan  staff   224 Nov  4 12:58 forms
0 drwxr-xr-x   8 calebhallinan  staff   256 Nov 16 14:14 lectures
8 -rw-r--r--   1 calebhallinan  staff  3012 Nov 15 20:02 misc.py
0 drwxr-xr-x   3 calebhallinan  staff    96 Oct 31 20:19 policies
~/De/j/teach_classes/pytorch4st main !2 ?3 >
```

Py pytorch4st

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` .git` folder where git will store this history of the repository

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` .git` folder where git will store this history of the repository
- **git status** = displays the state of the working directory and the staging area



pytorch4st — calebhallinan@Clinks-MacBook-Pro — -zsh — 121x22

+

..es/pytorch4st

```
0 drwxr-xr-x  7 calebhallinan  staff   224 Nov  4 12:58 forms
0 drwxr-xr-x  8 calebhallinan  staff   256 Nov 16 14:14 lectures
8 -rw-r--r--  1 calebhallinan  staff  3012 Nov 15 20:02 misc.py
0 drwxr-xr-x  3 calebhallinan  staff    96 Oct 31 20:19 policies
```

[~/De/j/teach\_classes/pytorch4st main !2 ?3 > git status

Py pytorch4st ]

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

```
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
  modified:   .DS_Store
  modified:   policies/00_theplan.txt
```

Untracked files:

```
(use "git add <file>..." to include in what will be committed)
  forms/
  lectures/
  misc.py
```

no changes added to commit (use "git add" and/or "git commit -a")

~/De/j/teach\_classes/pytorch4st main !2 ?3 >

Py pytorch4st ]

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` `.git` folder where git will store this history of the repository
- **git status** = displays the state of the working directory and the staging area

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` `.git` folder where git will store this history of the repository
- **git status** = displays the state of the working directory and the staging area

## Tracking changes in a local repo

- **git add** = tell git which files you want to track changes for



pytorch4st — calebhallinan@Clinks-MacBook-Pro — -zsh — 121x22

+

..es/pytorch4st

no changes added to commit (use "git add" and/or "git commit -a")

[~/De/j/teach\_classes/pytorch4st main !2 ?3 > git add misc.py

Py pytorch4st ]

[~/De/j/te/pytorch4st main +1 !2 ?2 > git status

Py pytorch4st ]

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)  
  new file:  misc.py

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
  modified:  .DS\_Store  
  modified:  policies/00\_theplan.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)  
  forms/  
  lectures/

~/De/j/te/pytorch4st main +1 !2 ?2 >

Py pytorch4st

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` `.git` folder where git will store this history of the repository
- **git status** = displays the state of the working directory and the staging area

## Tracking changes in a local repo

- **git add** = tell git which files you want to track changes for

# Creating a local repository

- **git init** = initializes a folder for being git-aware / version control
  - Creates a ` `.git` folder where git will store this history of the repository
- **git status** = displays the state of the working directory and the staging area

## Tracking changes in a local repo

- **git add** = tell git which files you want to track changes for
- **git commit** = tell git to take everything we are tracking (using git add) and stores a copy permanently inside the special .git directory. This permanent copy is called a commit (or revision)



pytorch4st — calebhallinan@Clinks-MacBook-Pro — -zsh — 121x22



..es/pytorch4st

```
[~/De/j/te/pytorch4st main +1 !2 ?2 > git commit -m "add misc.py to the repo"
```

Py pytorch4st ]

[main 2078032] add misc.py to the repo

1 file changed, 72 insertions(+)

create mode 100644 misc.py

```
[~/De/j/te/pytorch4st main +1 !2 ?2 > git status
```

Py pytorch4st ]

On branch main

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: .DS\_Store

modified: policies/00\_theplan.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

forms/

lectures/

no changes added to commit (use "git add" and/or "git commit -a")

```
~/De/j/te/pytorch4st main +1 !2 ?2 >
```

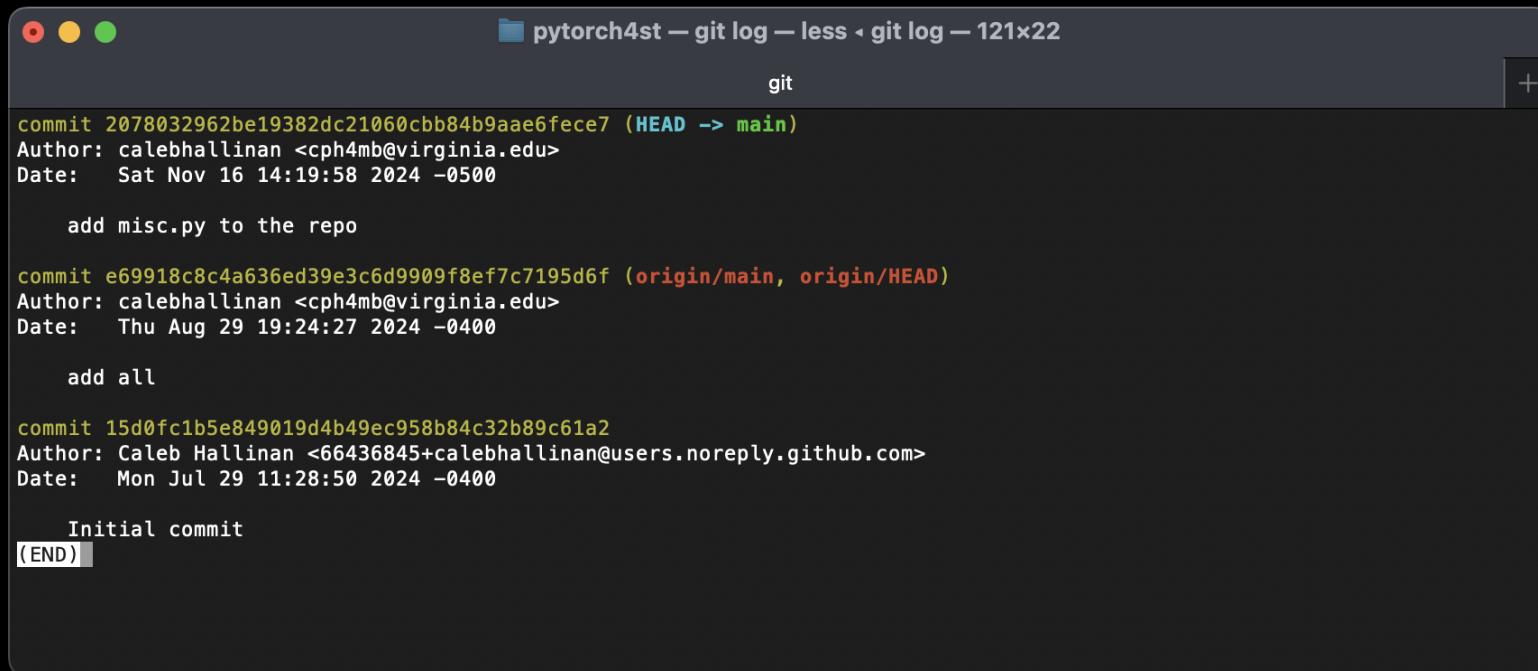
Py pytorch4st ]

# git commit messages

- We use the -m flag (for “message”) to record a short, descriptive, and specific comment that will help us remember later on what we did and why
- If we just run `git commit` without the -m option, git will launch nano (or whatever other editor is configured as `core.editor`) so that we can write a longer message
- Good commit messages start with a brief (<50 characters) statement about the changes made in the commit

# Show git commit history

- **git log** = shows the history of all the commits for the local repository



A screenshot of a terminal window titled "pytorch4st — git log — less - git log — 121x22". The window displays the commit history for the "main" branch. The commits are listed in reverse chronological order, starting from the most recent at the top.

```
commit 2078032962be19382dc21060cbb84b9aae6fece7 (HEAD -> main)
Author: calebhallinan <cph4mb@virginia.edu>
Date:   Sat Nov 16 14:19:58 2024 -0500

    add misc.py to the repo

commit e69918c8c4a636ed39e3c6d9909f8ef7c7195d6f (origin/main, origin/HEAD)
Author: calebhallinan <cph4mb@virginia.edu>
Date:   Thu Aug 29 19:24:27 2024 -0400

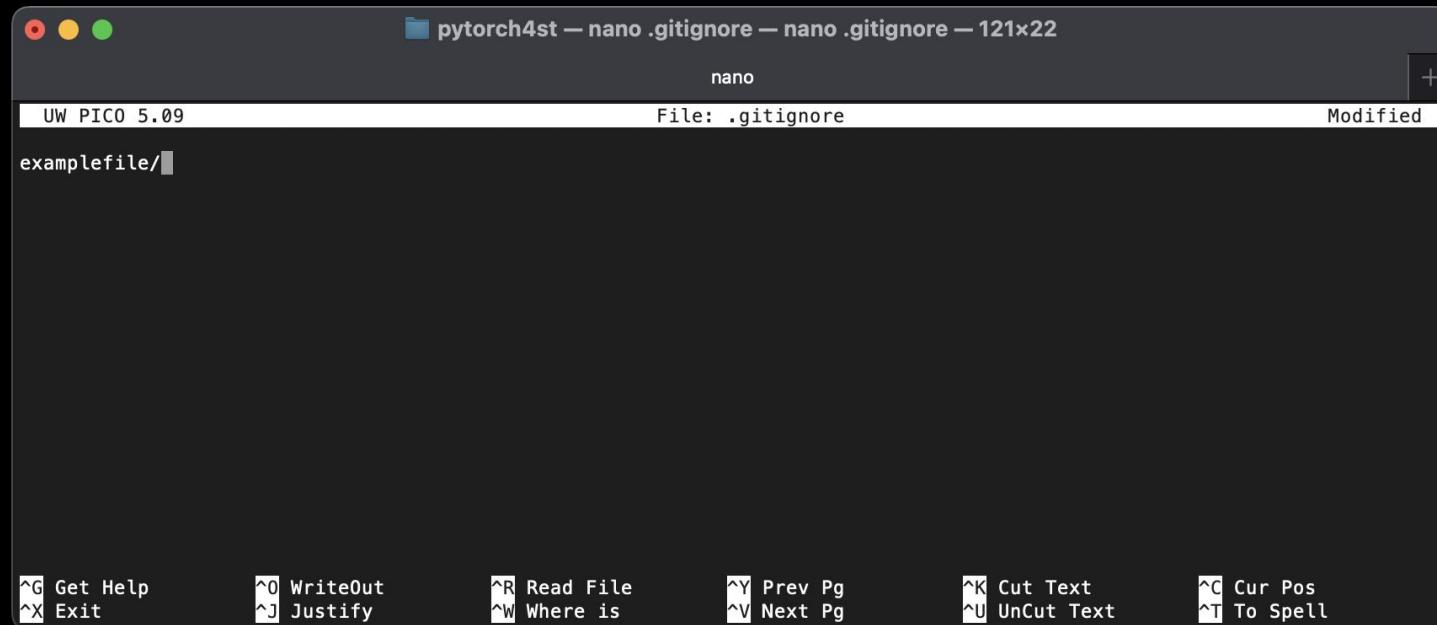
    add all

commit 15d0fc1b5e849019d4b49ec958b84c32b89c61a2
Author: Caleb Hallinan <66436845+calebhallinan@users.noreply.github.com>
Date:   Mon Jul 29 11:28:50 2024 -0400

    Initial commit
(END)
```

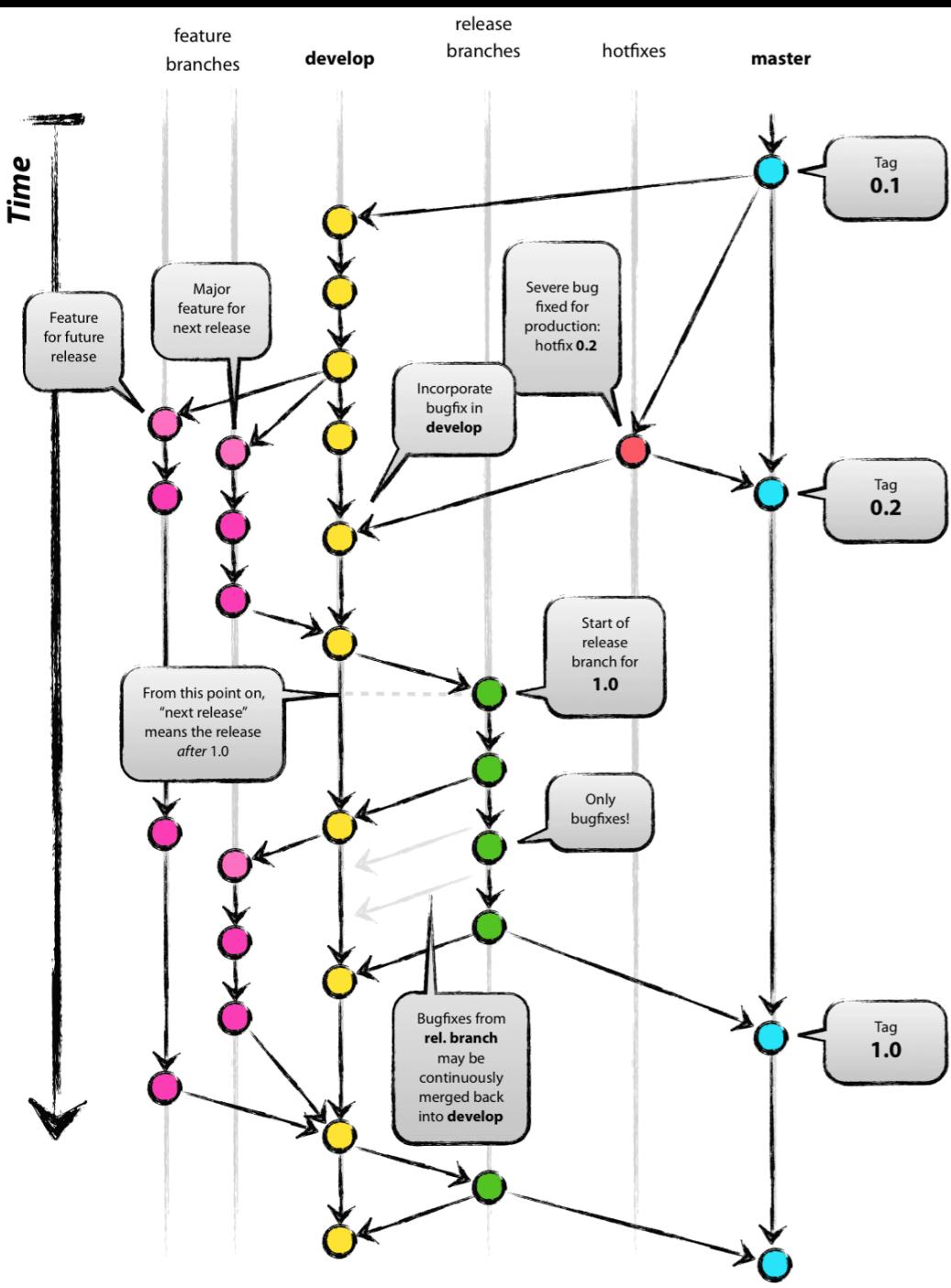
# Ignoring files

- Create a **.gitignore** file to specify specific types of files you do not want tracked



# A few more to know...

- **git branch** = A command to create, list, or delete branches, which are independent lines of development in a Git repository. Branches let you work on different features or fixes without affecting the main codebase.
- **git checkout** = A command to switch between branches or restore files in your working directory to a specific commit. It updates the working directory to match the specified branch or commit.



```
$ git checkout -b myfeature develop
Switched to a new branch "myfeature"

$ git checkout develop
Switched to branch 'develop'
$ git merge --no-ff myfeature
Updating ea1b82a..05e9557
(Summary of changes)
$ git branch -d myfeature
Deleted branch myfeature (was 05e9557).
$ git push origin develop
```

# Installing conda and necessary packages

- Everyone will need to have a package manager
  - I personally use conda, but jupyter env is fine as well if you prefer that
- Here is a tutorial on how to install conda on various machines
  - <https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html>
- Please have this set up for our later classes, if you need help reach out to me and we will sort it out!

# Needed packages

- Numpy
- Seaborn
- Matplotlib
- Scanpy
- Pytorch
  - <https://pytorch.org/get-started/locally/>
- Maybe more tbd

# VSCode

- I HIGHLY, HIGHLY recommend VSCode for coding
  - <https://code.visualstudio.com/>
- Awesome interface, handles multiple languages and formats, easily downloadable extensions for use



Any questions about git/GitHub?

# Deep Learning Intro

A lot of these slides were  
borrowed from the MIT course

<https://introtodeeplearning.com/>

# Questions

- Who here has used DL before?
  - How?
- Would you say you know how it works well?
- Can you give me a few applications of DL you know about?

# Evolution of AI

March 2023



April 2025



# What is Deep Learning?

## ARTIFICIAL INTELLIGENCE

Any technique that enables computers to mimic human behavior



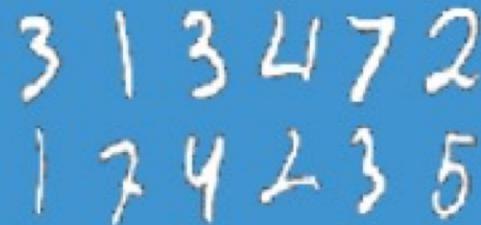
## MACHINE LEARNING

Ability to learn without explicitly being programmed



## DEEP LEARNING

Extract patterns from data using neural networks



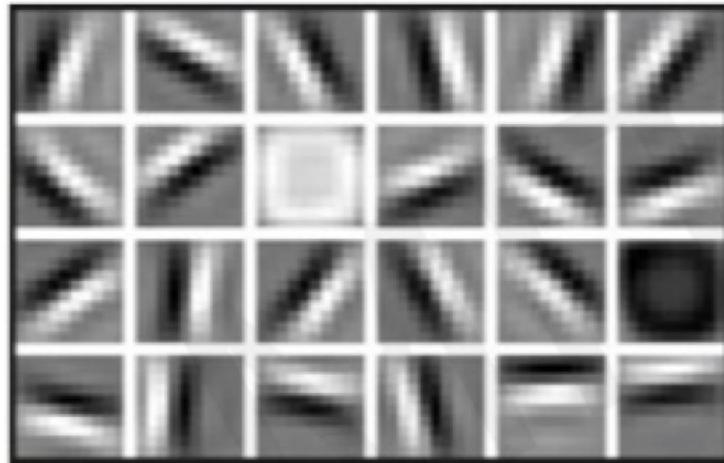
Teaching computers how to **learn a task** directly from **raw data**

# Why Deep Learning?

Hand engineered features are time consuming, brittle, and not scalable in practice

Can we learn the **underlying features** directly from data?

Low Level Features



Lines & Edges

Mid Level Features



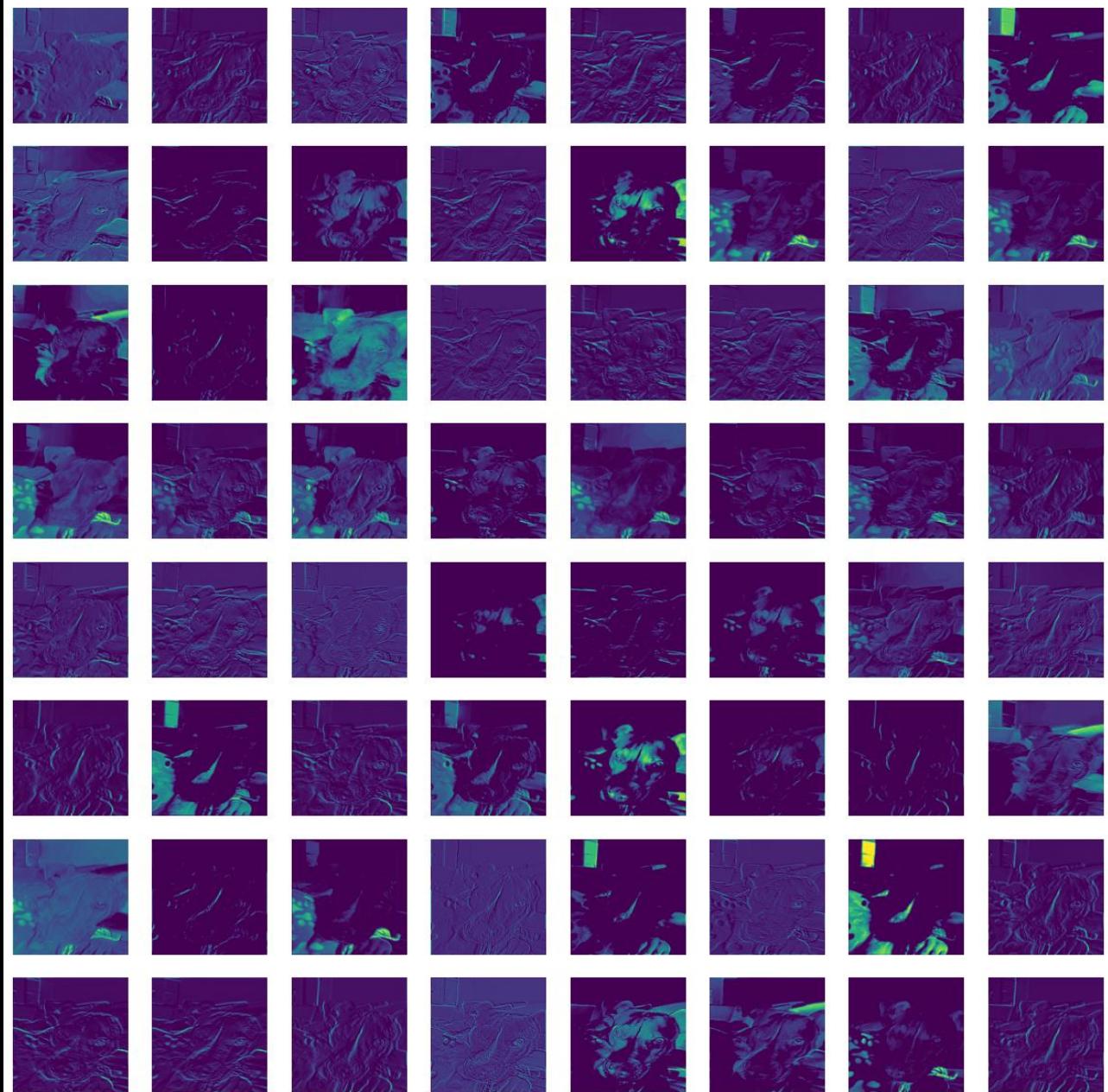
Eyes & Nose & Ears

High Level Features

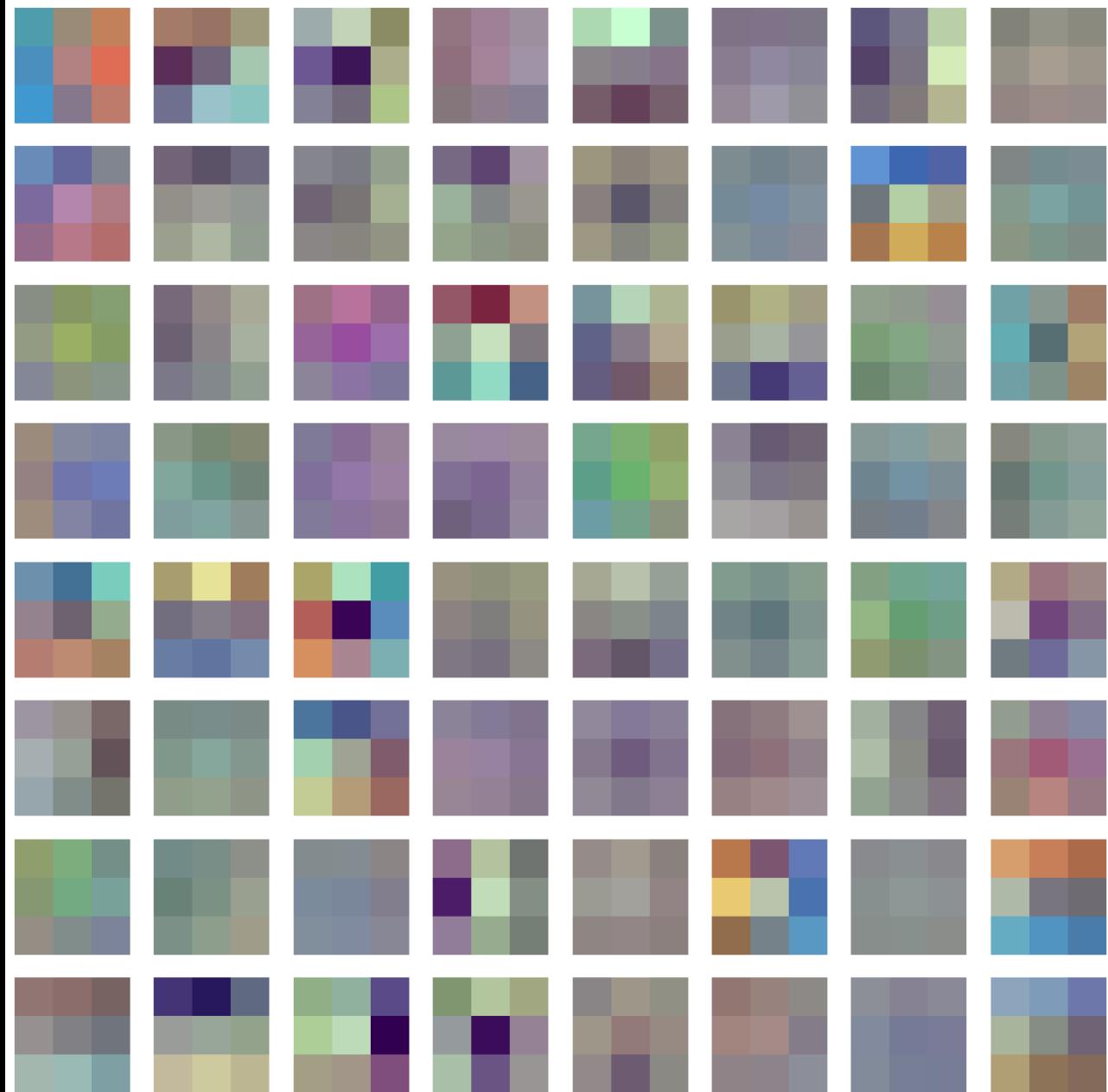


Facial Structure

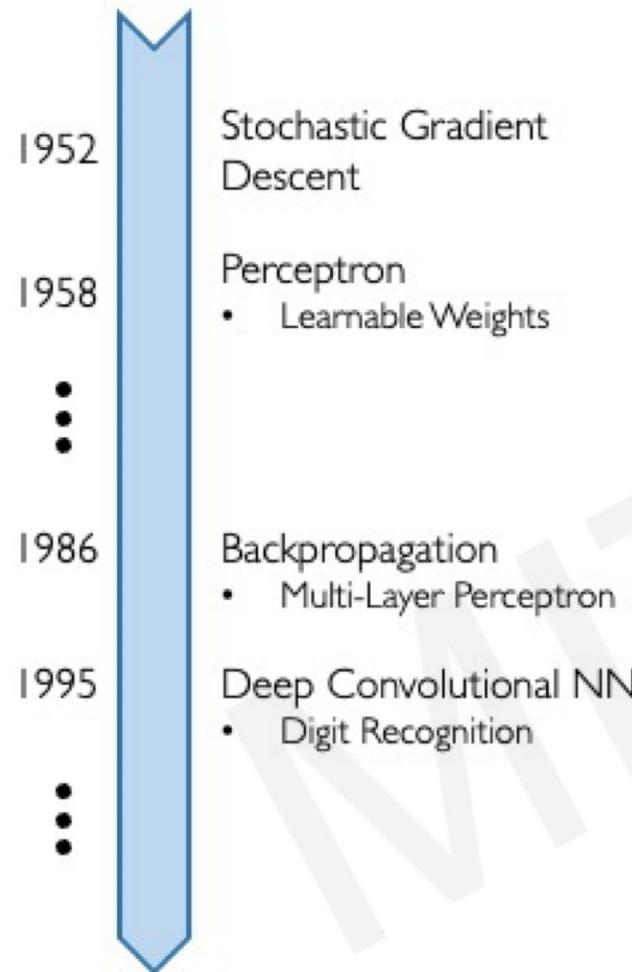
Feature Maps from the First Convolutional Layer



Weights of the First Convolutional Layer



# Why Now?



Neural Networks date back decades, so why the dominance?

## I. Big Data

- Larger Datasets
- Easier Collection & Storage

IM<sup>2</sup>GENET



## 2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable



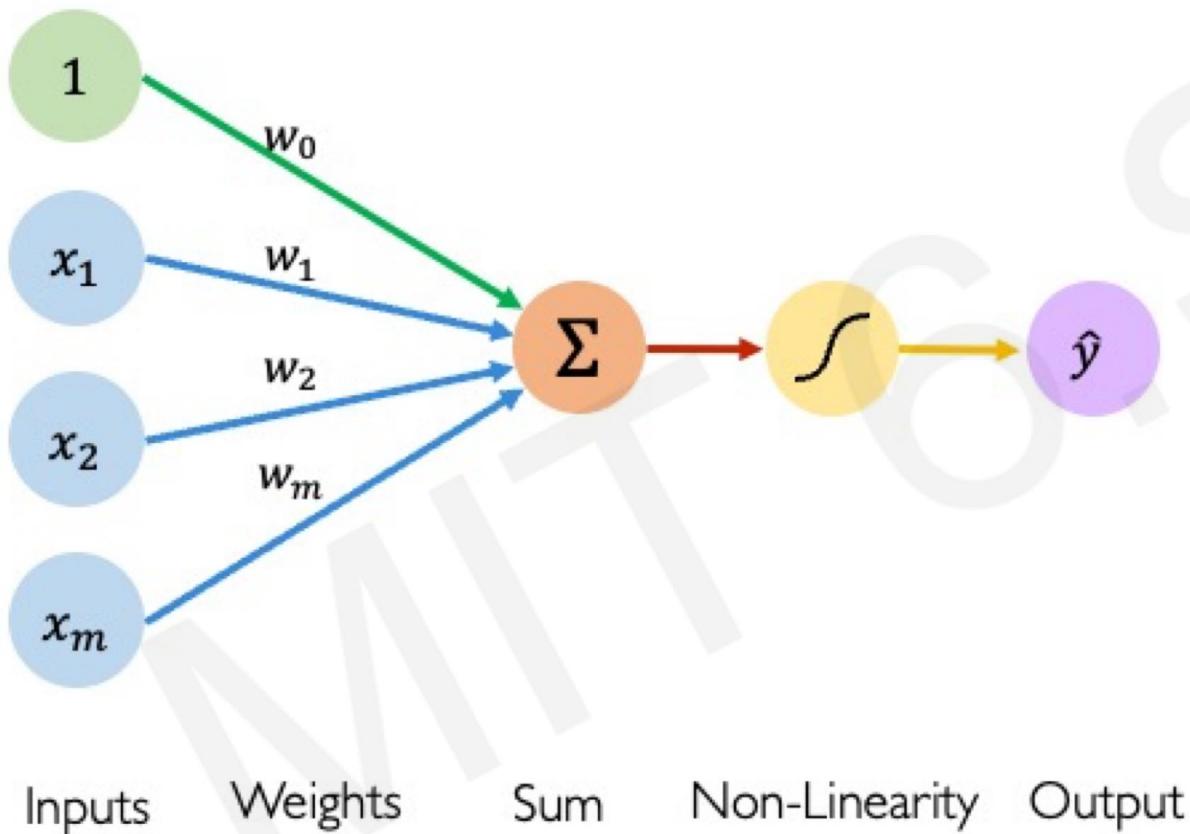
## 3. Software

- Improved Techniques
- New Models
- Toolboxes



What is the basic unit of DL?

# The Perceptron: Forward Propagation



Linear combination of inputs

$$\hat{y} = g \left( w_0 + \sum_{i=1}^m x_i w_i \right)$$

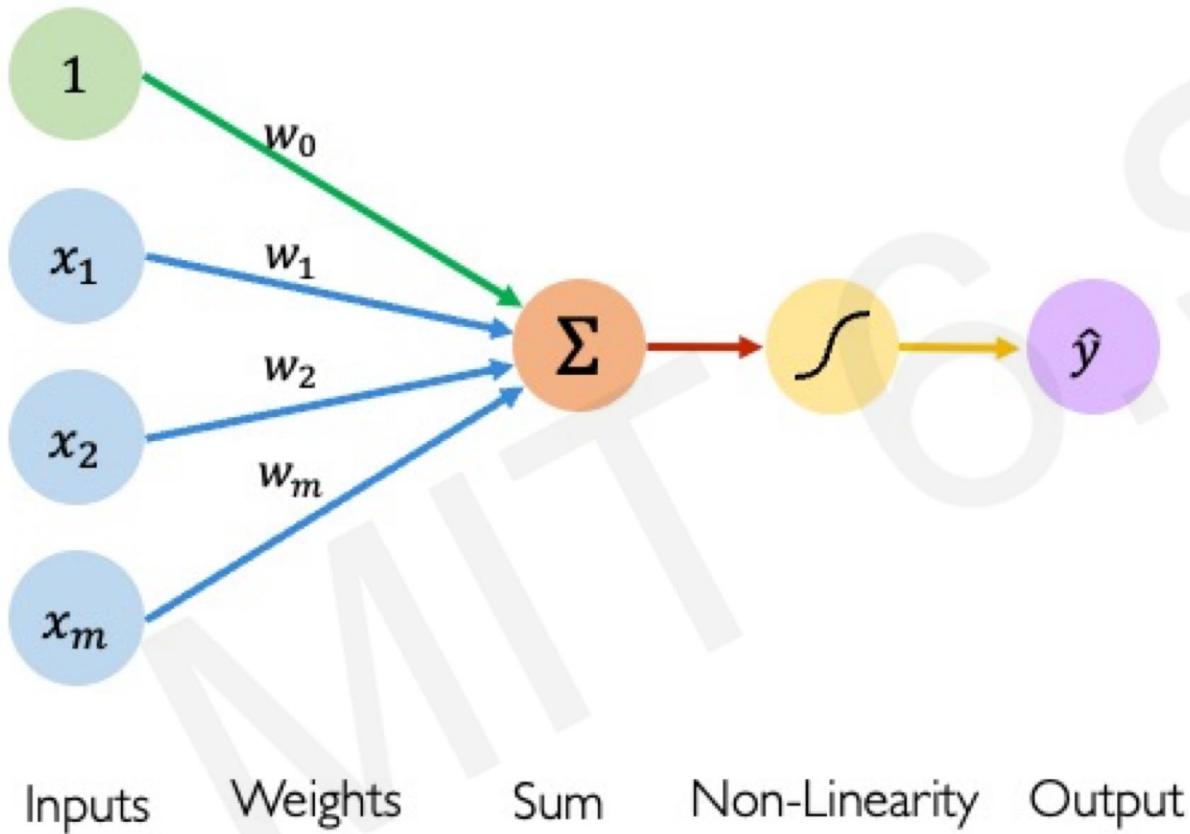
Output

Non-linear activation function

Bias

Bias allows neurons to learn functions that aren't forced through the origin, making the network more flexible and powerful

# The Perceptron: Forward Propagation

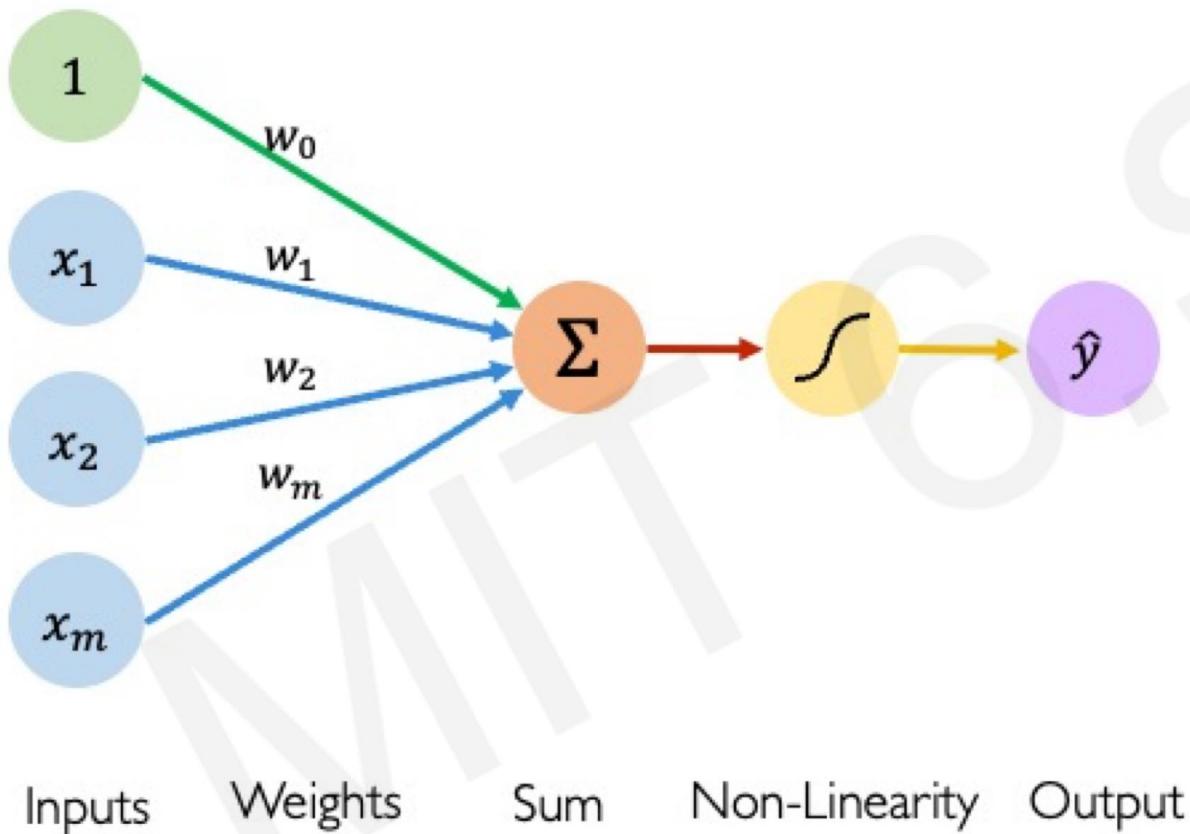


$$\hat{y} = g \left( w_0 + \sum_{i=1}^m x_i w_i \right)$$

$$\hat{y} = g ( w_0 + \mathbf{X}^T \mathbf{W} )$$

where:  $\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$  and  $\mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$

# The Perceptron: Forward Propagation

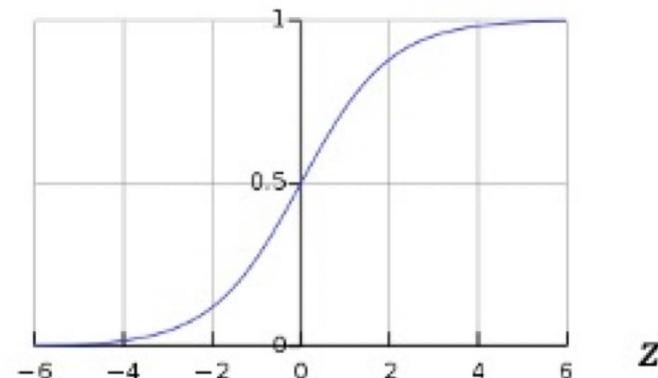


## Activation Functions

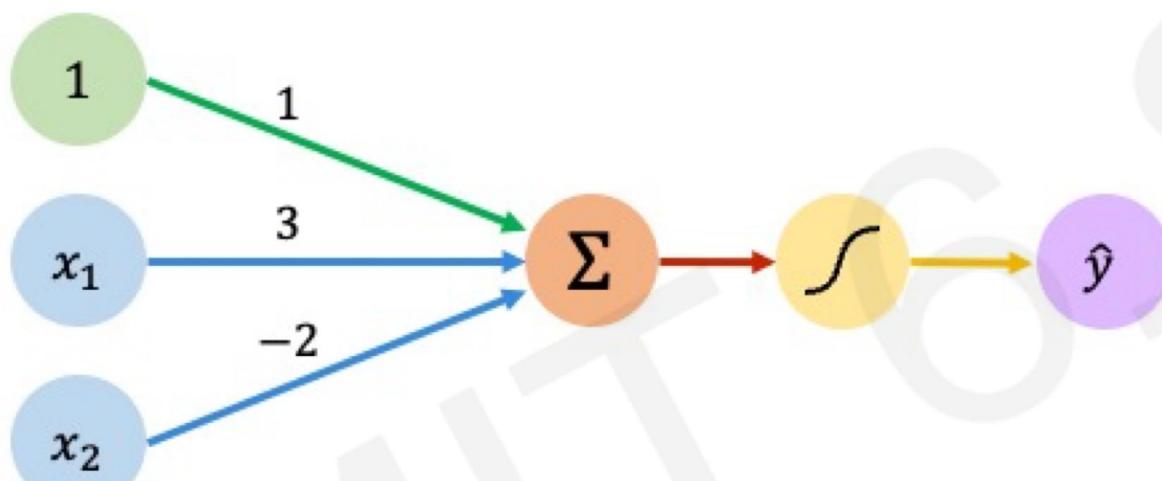
$$\hat{y} = g(w_0 + \mathbf{X}^T \mathbf{W})$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



# The Perceptron: Example

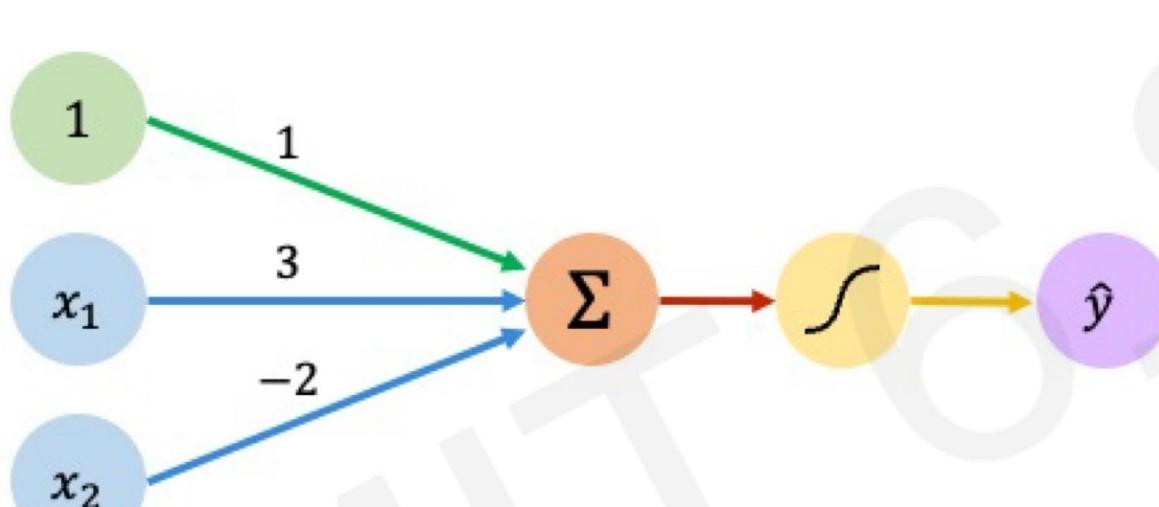


We have:  $w_0 = 1$  and  $\mathbf{w} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$

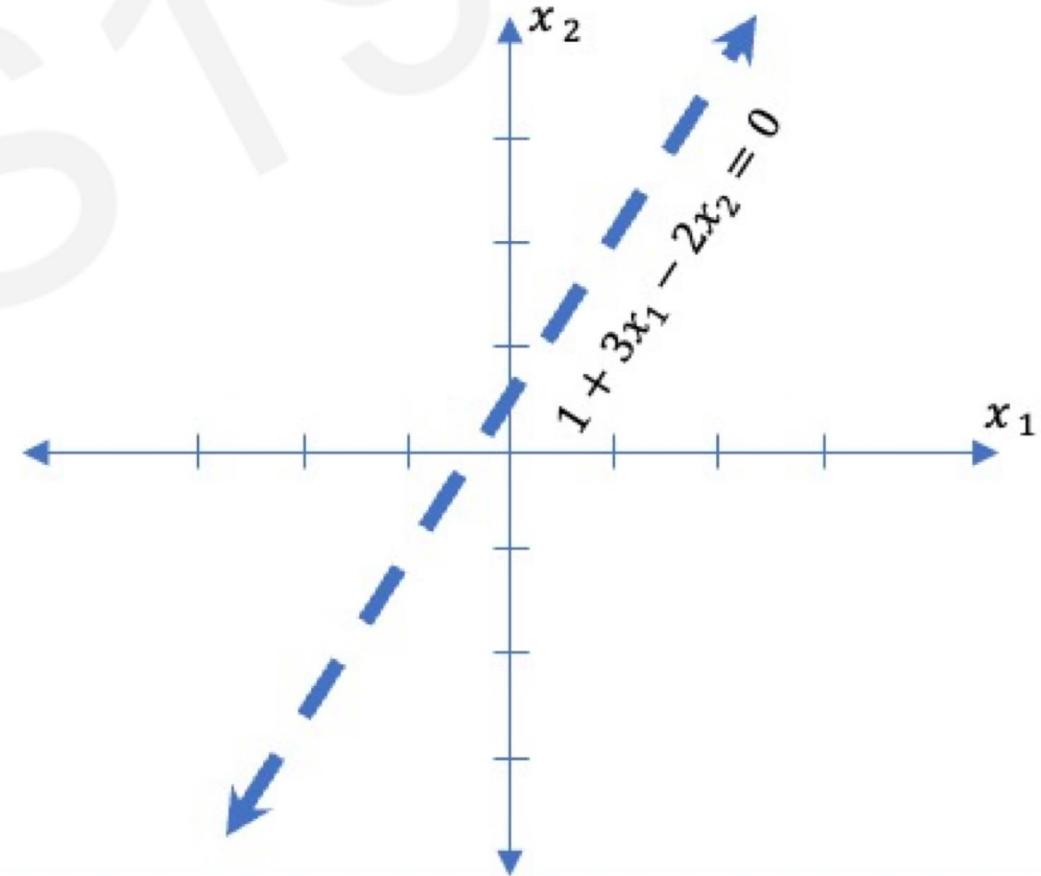
$$\begin{aligned}\hat{y} &= g(w_0 + \mathbf{x}^T \mathbf{w}) \\ &= g\left(1 + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 3 \\ -2 \end{bmatrix}\right) \\ \hat{y} &= g(1 + 3x_1 - 2x_2)\end{aligned}$$

This is just a line in 2D!

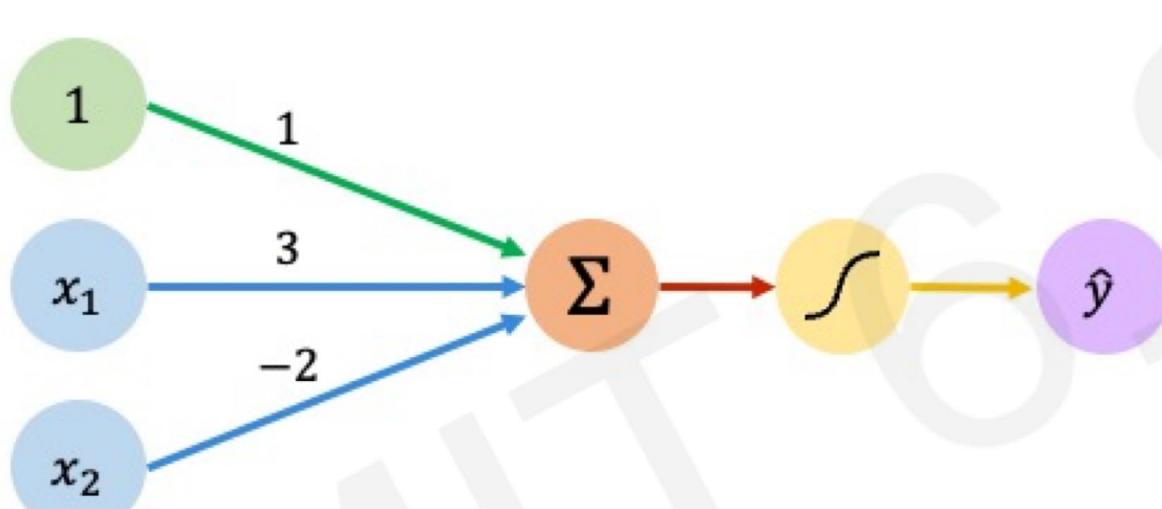
# The Perceptron: Example



$$\hat{y} = g(1 + 3x_1 - 2x_2)$$



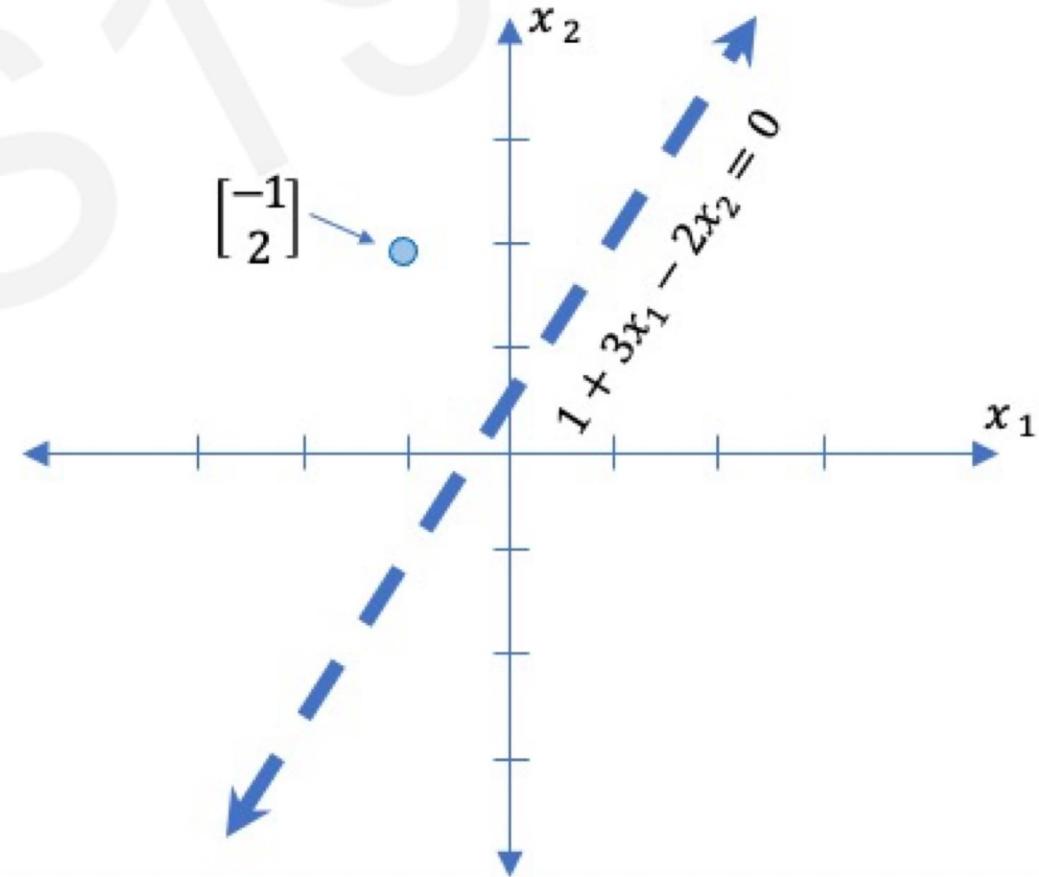
# The Perceptron: Example



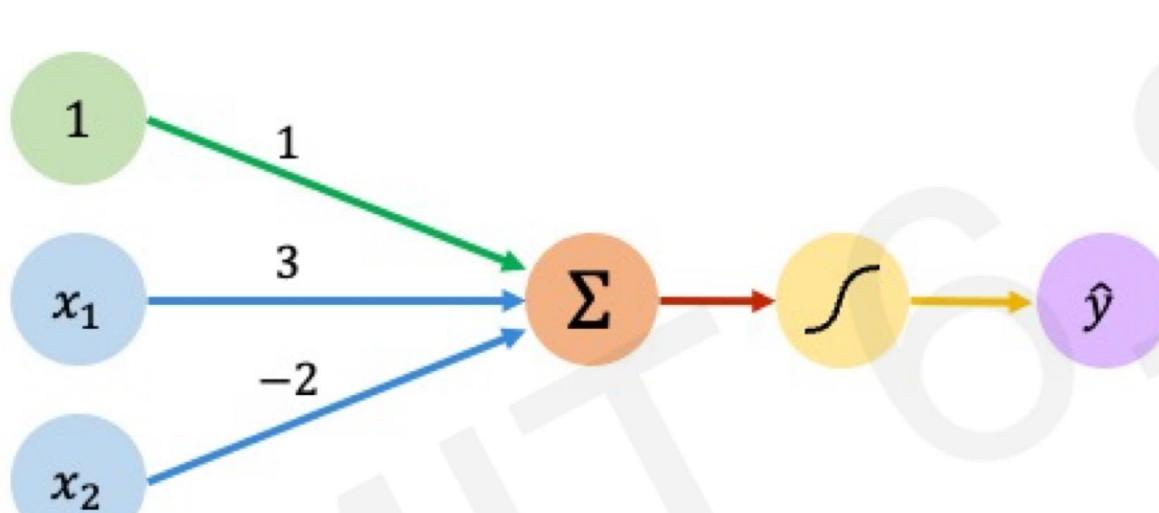
Assume we have input:  $\mathbf{x} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$

$$\begin{aligned}\hat{y} &= g(1 + (3 * -1) - (2 * 2)) \\ &= g(-6) \approx 0.002\end{aligned}$$

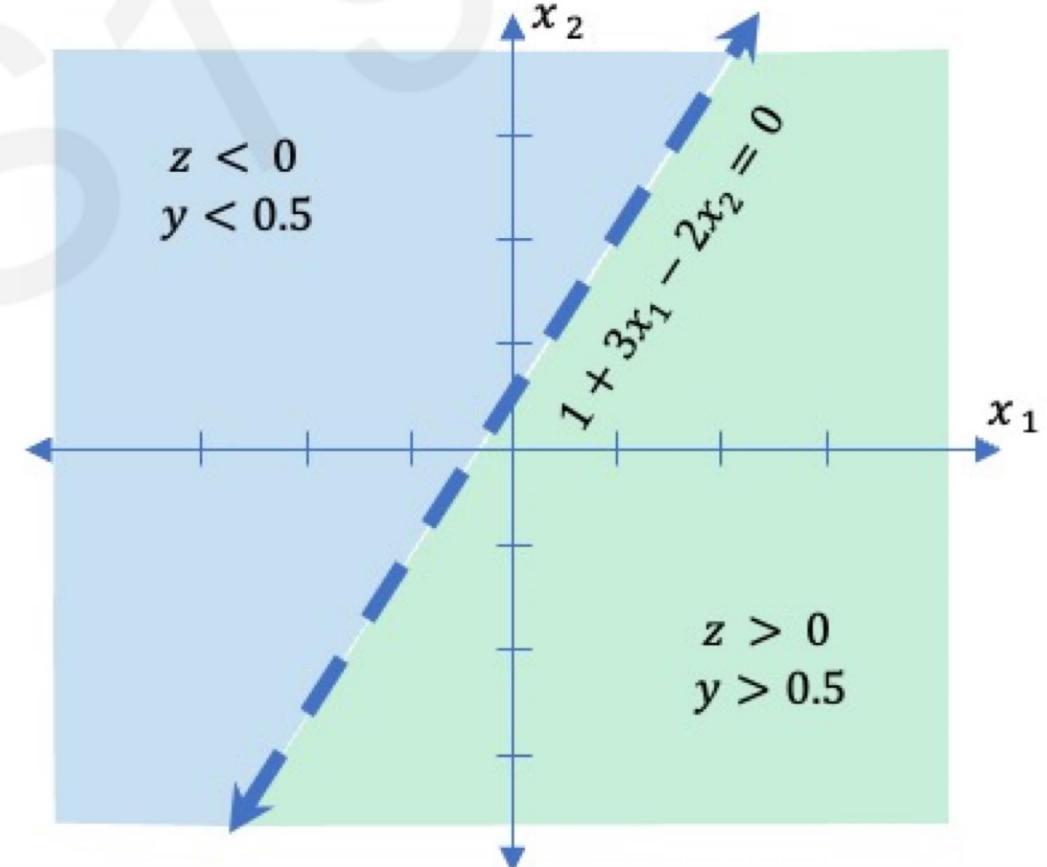
$$\hat{y} = g(1 + 3x_1 - 2x_2)$$



# The Perceptron: Example

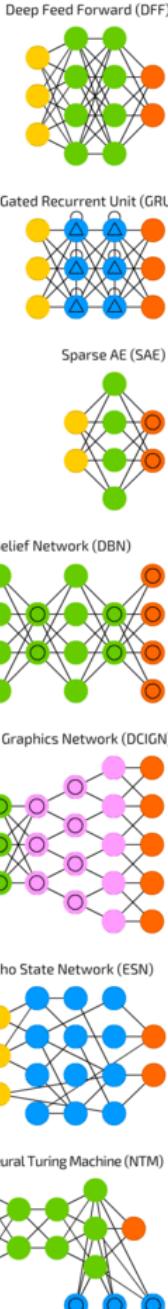


$$\hat{y} = g(1 + 3x_1 - 2x_2)$$

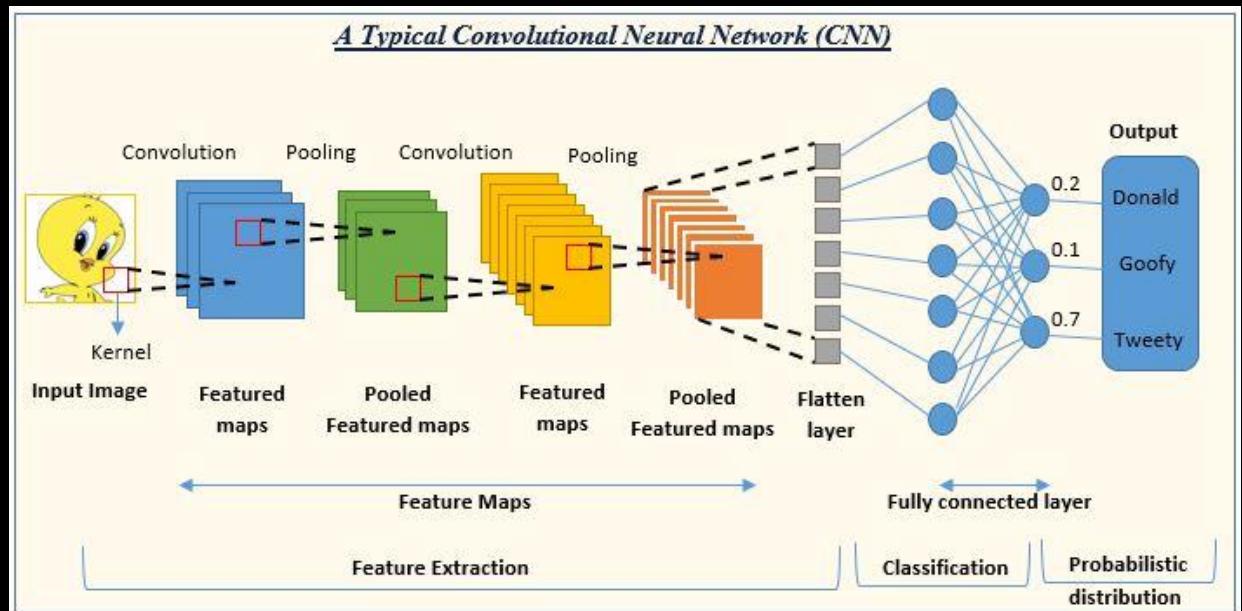
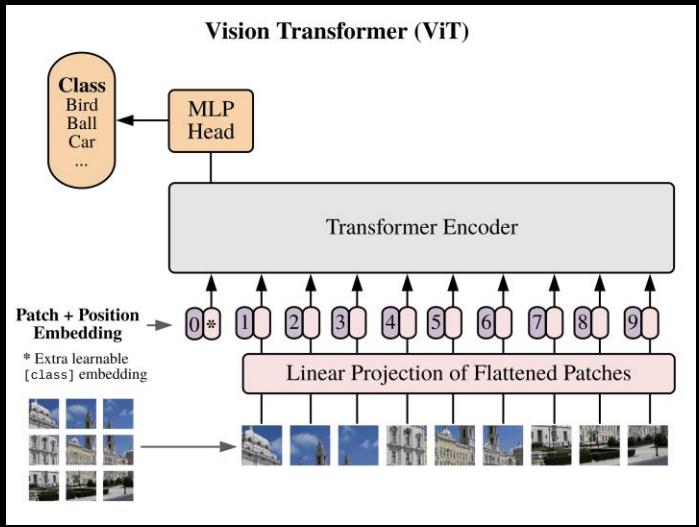
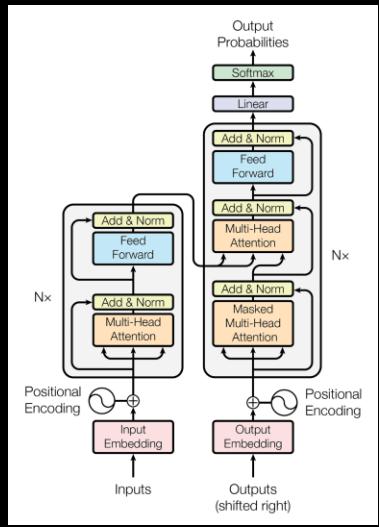


# A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org



# Types of Neural Networks



# Use cases of Deep Learning



- Natural Language Processing (NLP)
  - Machine Translation: Google Translate
  - Chatbots and virtual assistants: Siri, ChatGPT

<https://thispersondoesnotexist.com/>

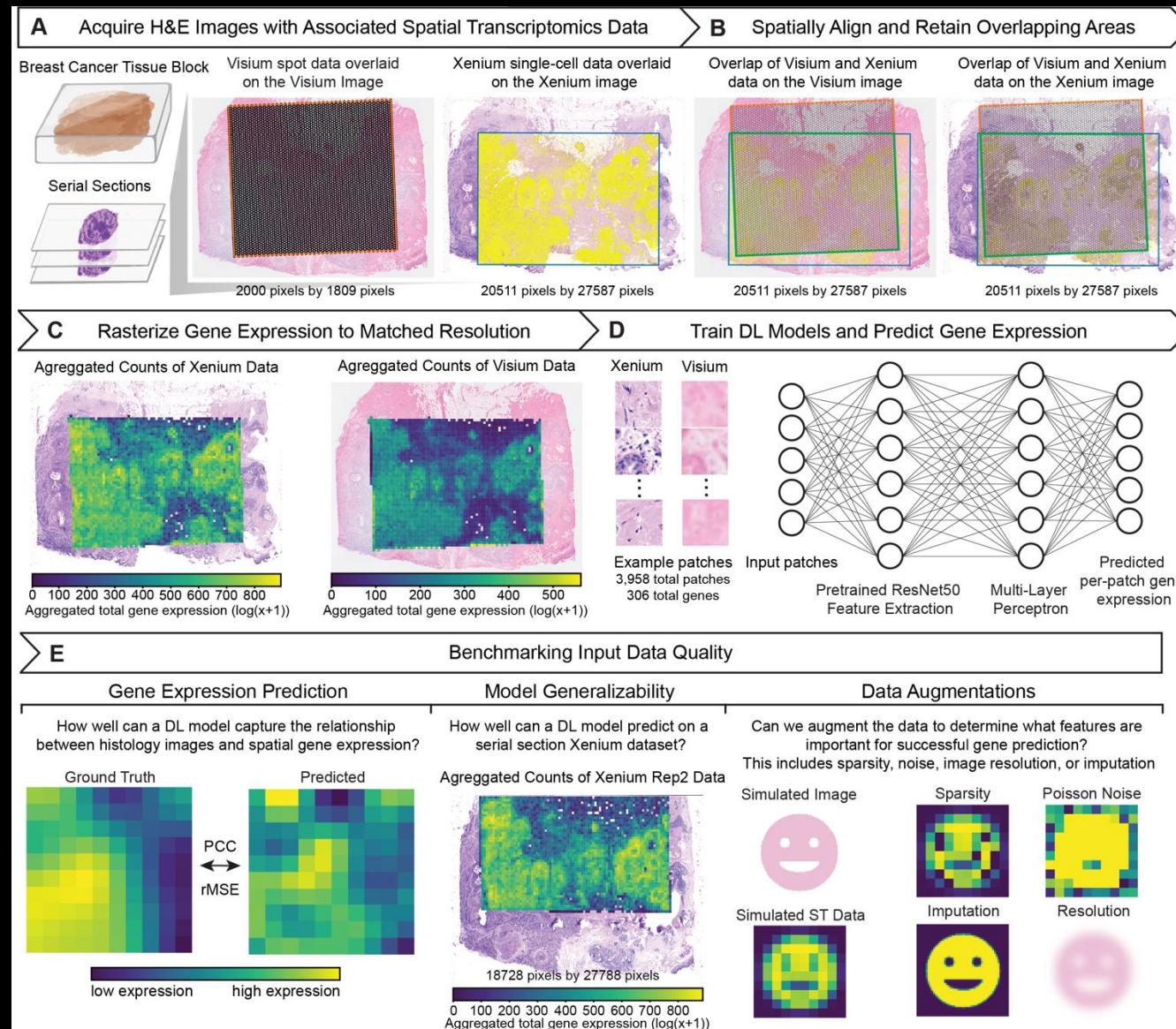
- Generative AI
  - Image generation
  - Music and audio generation

- Computer Vision
  - Image classification
  - Autonomous vehicles



Many, MANY more

# My Research – gene prediction from histology images



# Spatial Transcriptomics Intro

# Questions

- Who here has heard of spatial transcriptomics before?
  - How?

A lot of these slides were borrowed  
from my PI, Dr. Fan, when she  
teaches her class Genomic Data  
Visualization

Highly recommend this class! I have taken it and TA-ed it ☺

<https://jef.works/genomic-data-visualization-2024/>

Spatially resolved transcriptomic technologies enable us to profile what genes are expressed by cells in tissues

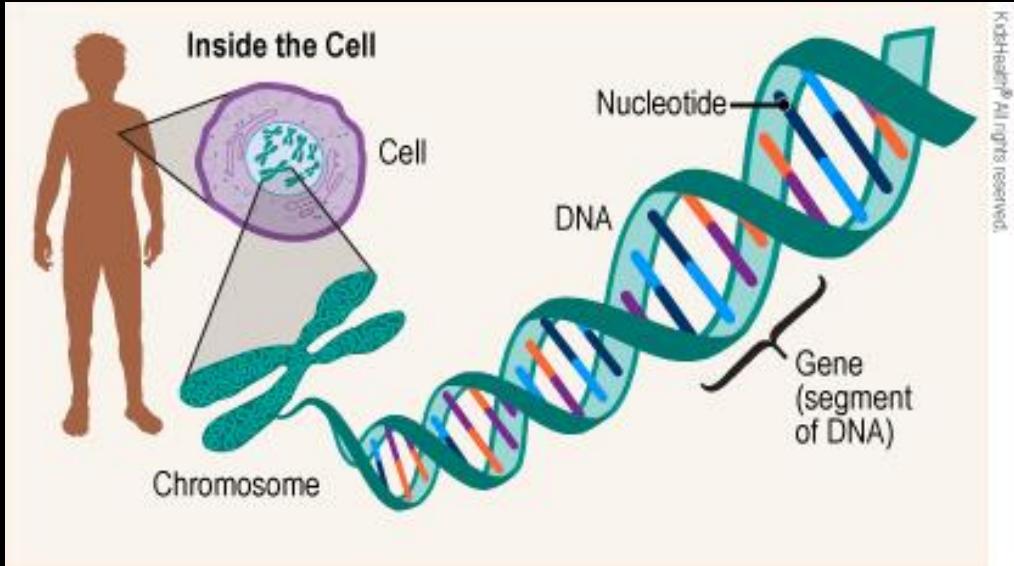


Lyla Atta



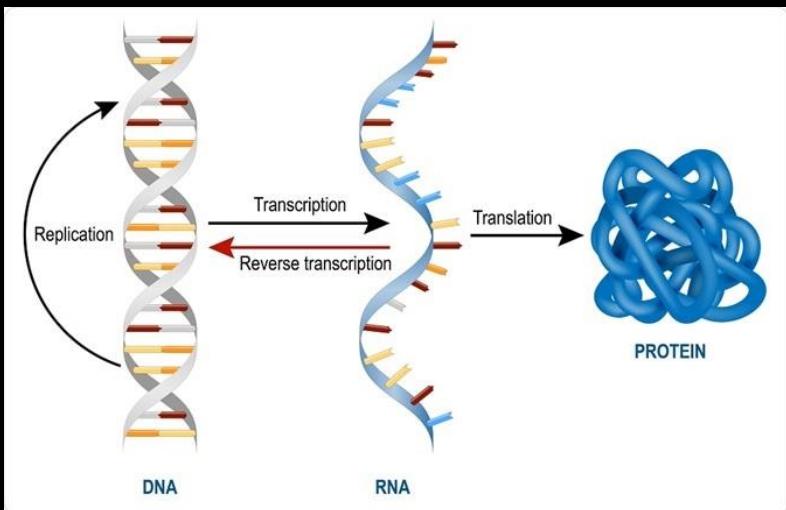
(Atta and Fan, Nature Communications, Sept 2021)

# What is a gene tho?



A gene is the basic unit of heredity and a functional segment of DNA that contains the instructions for building a specific protein or performing a function in the body

About how many genes do we have?

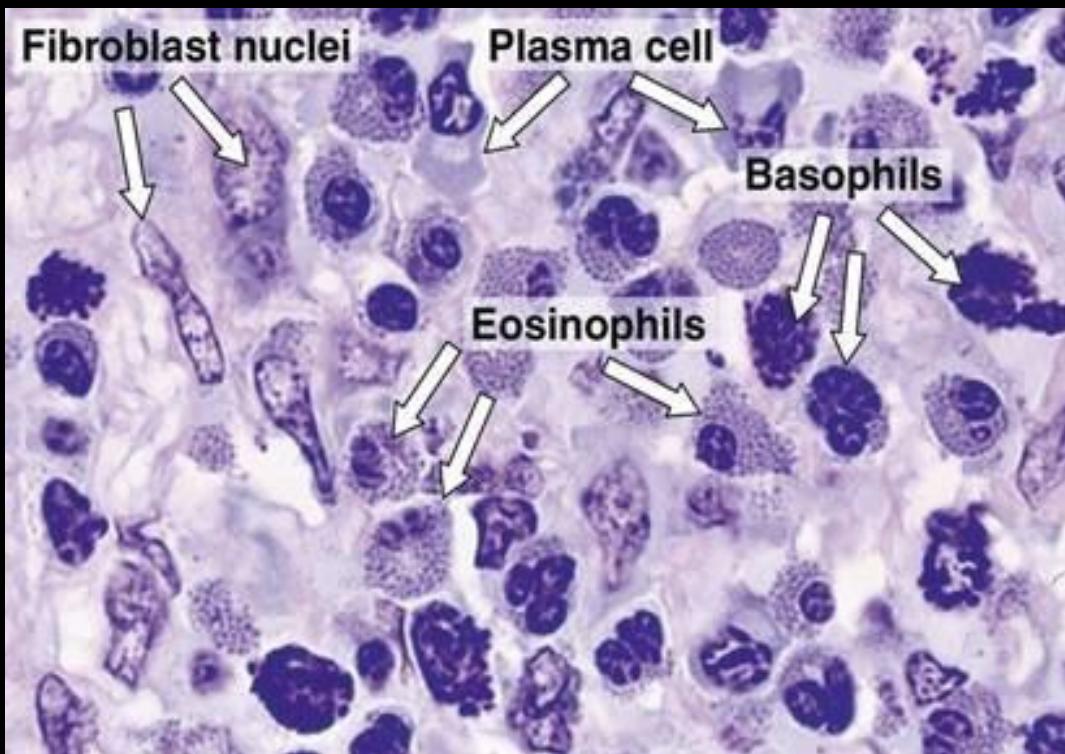


RNA, or ribonucleic acid, is a molecule that carries instructions from DNA to make proteins, acting as a messenger for the genetic code

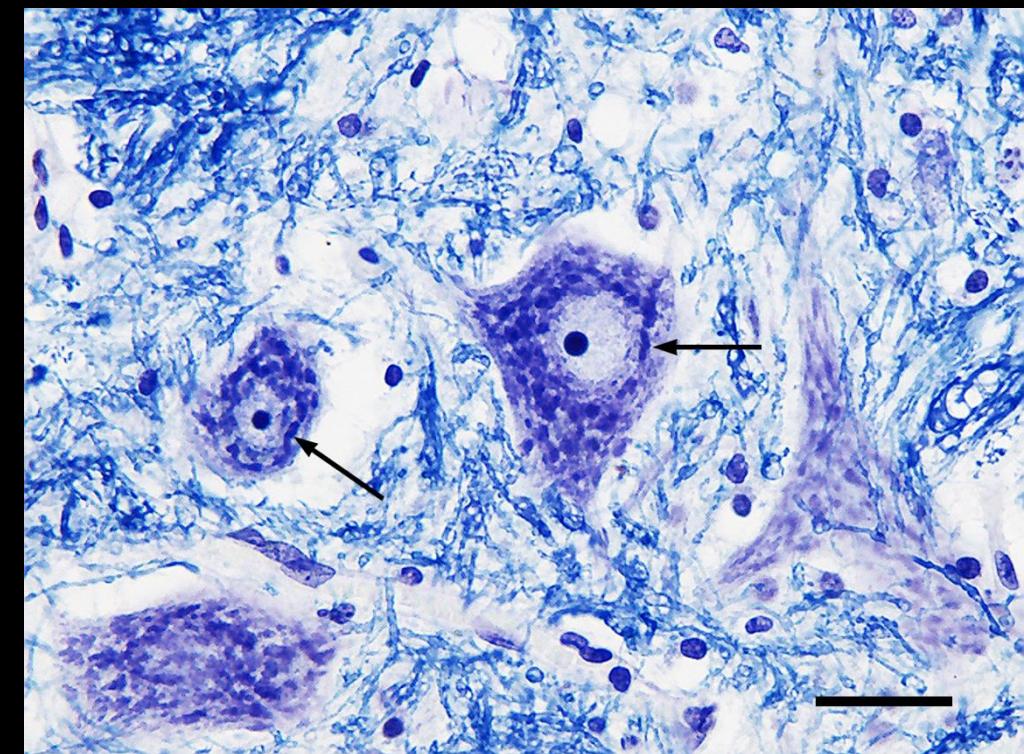
What are some different cell-types?

# What are some different cell-types?

H&E



Nissl



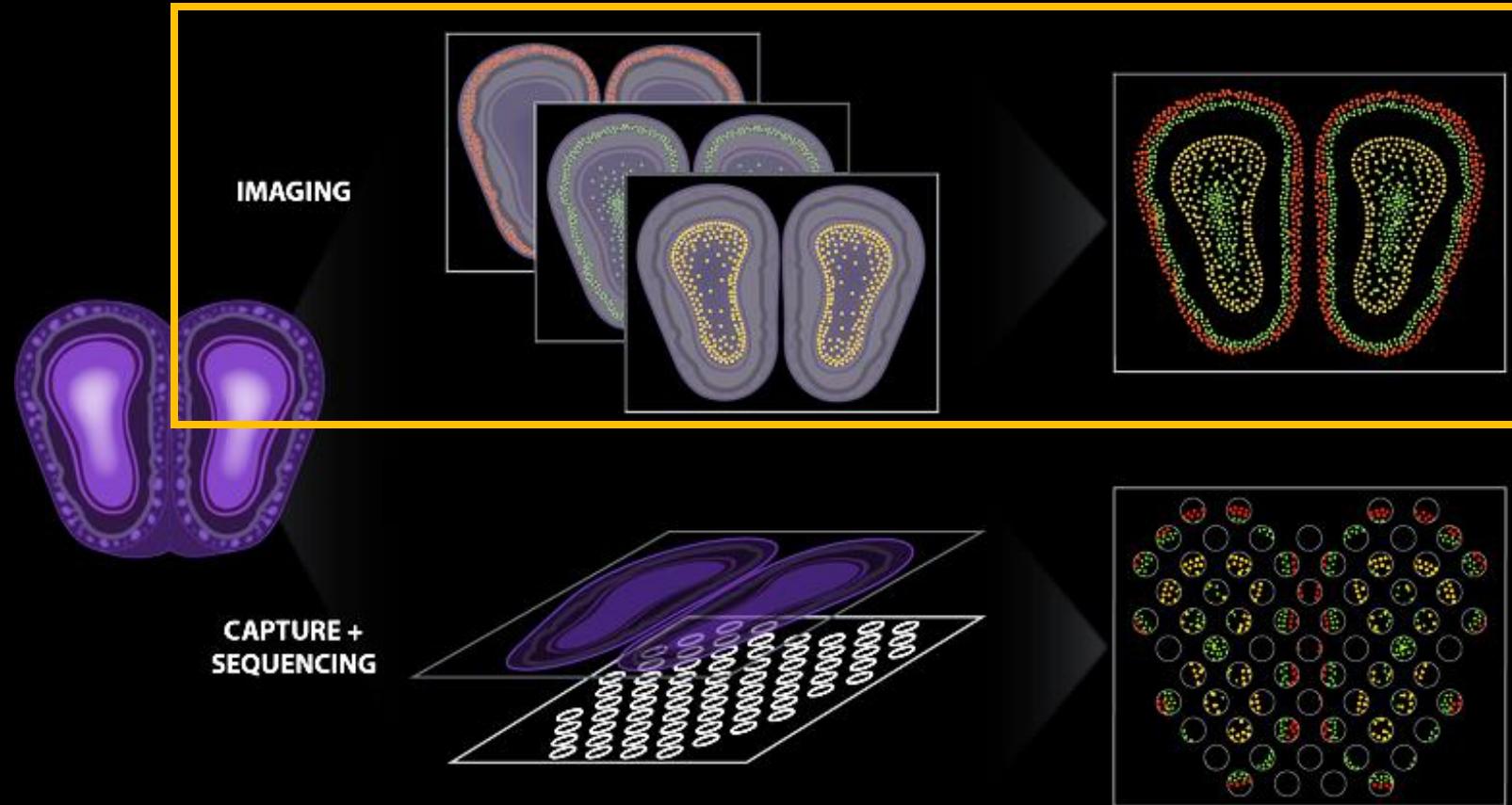
How can we distinguish these cell-types (and cell-states) from each other?

Why might having spatial information be helpful?

Spatially resolved transcriptomic technologies enable us to profile what genes are expressed by cells in tissues

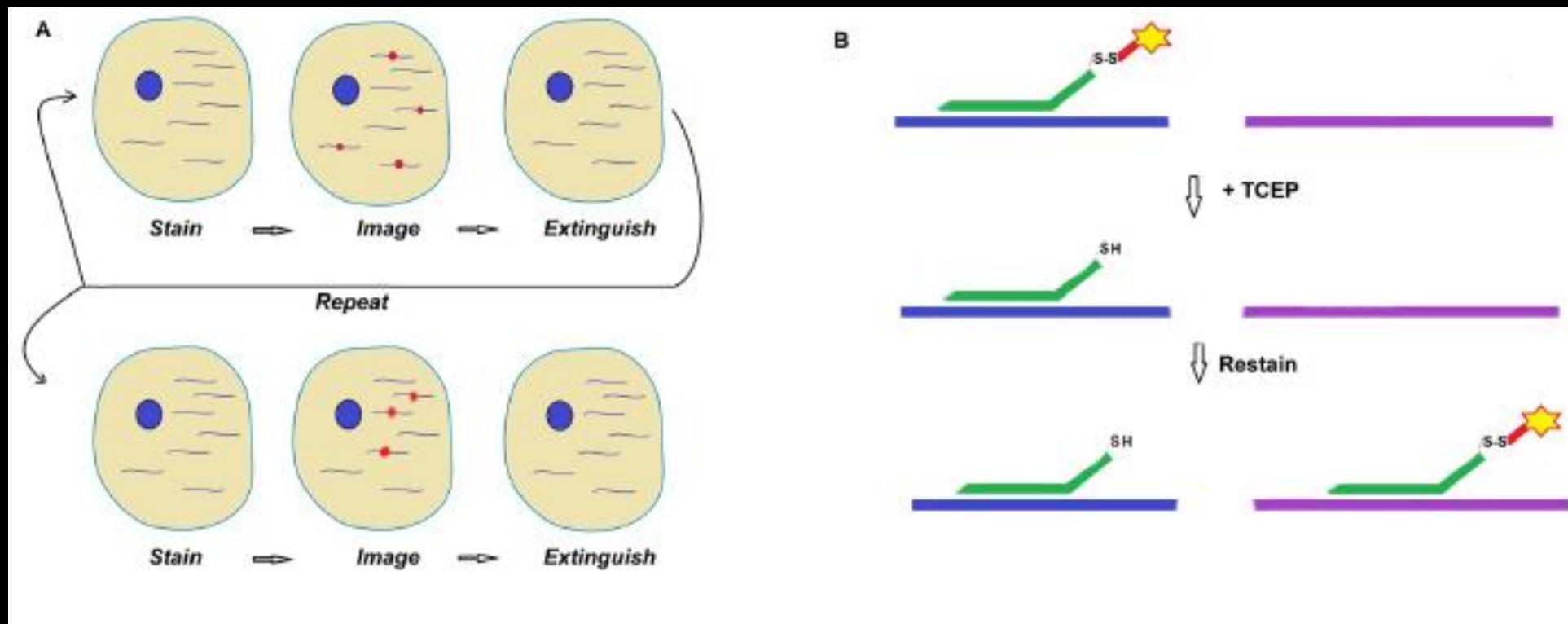


Lyla Atta

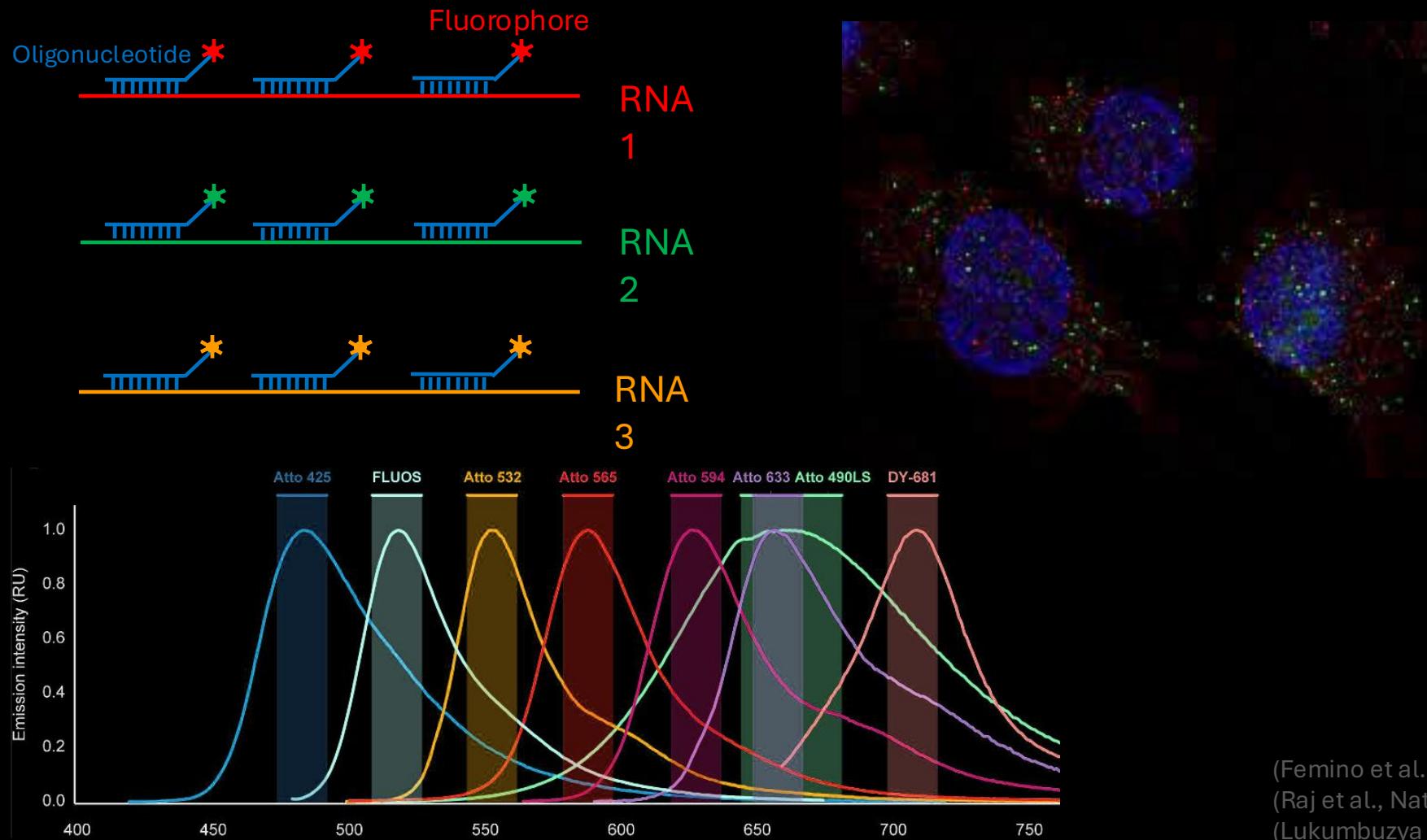


(Atta and Fan, Nature Communications, Sept 2021)

# Sequential smFISH



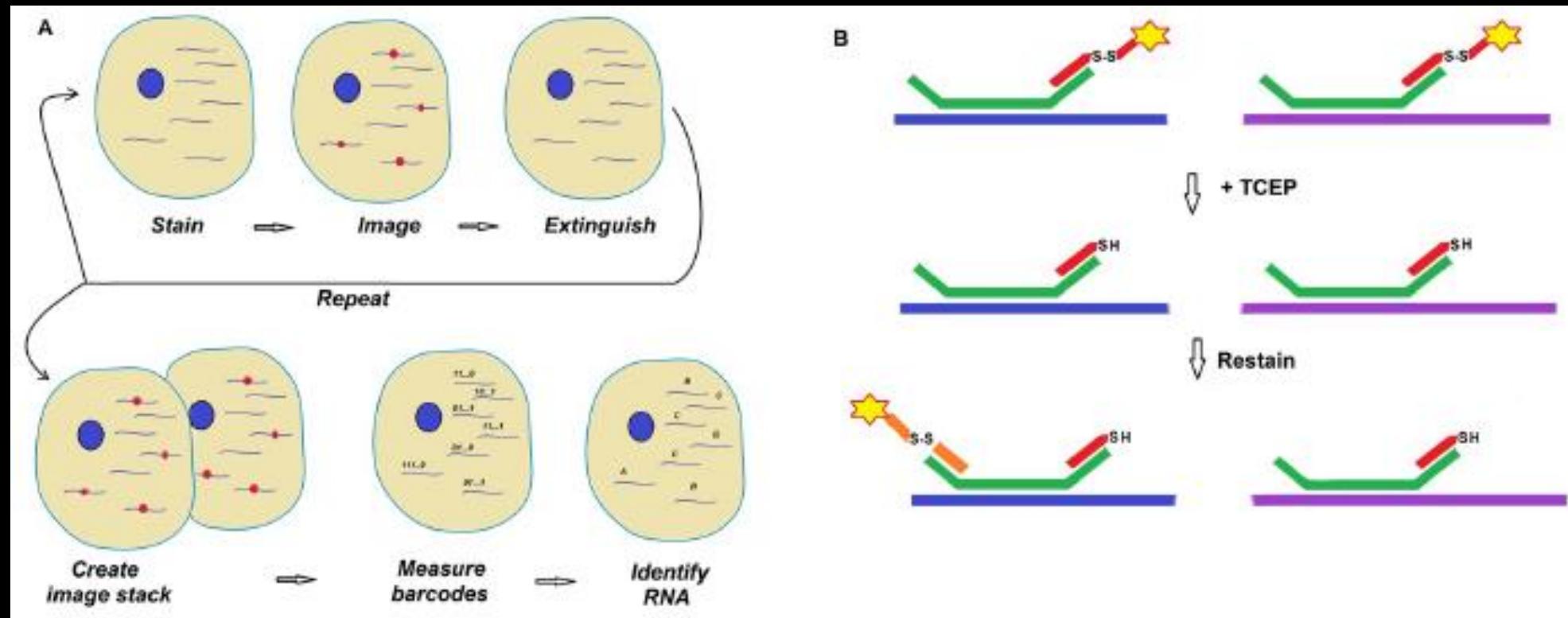
# smFISH enables imaging of single mRNA molecules



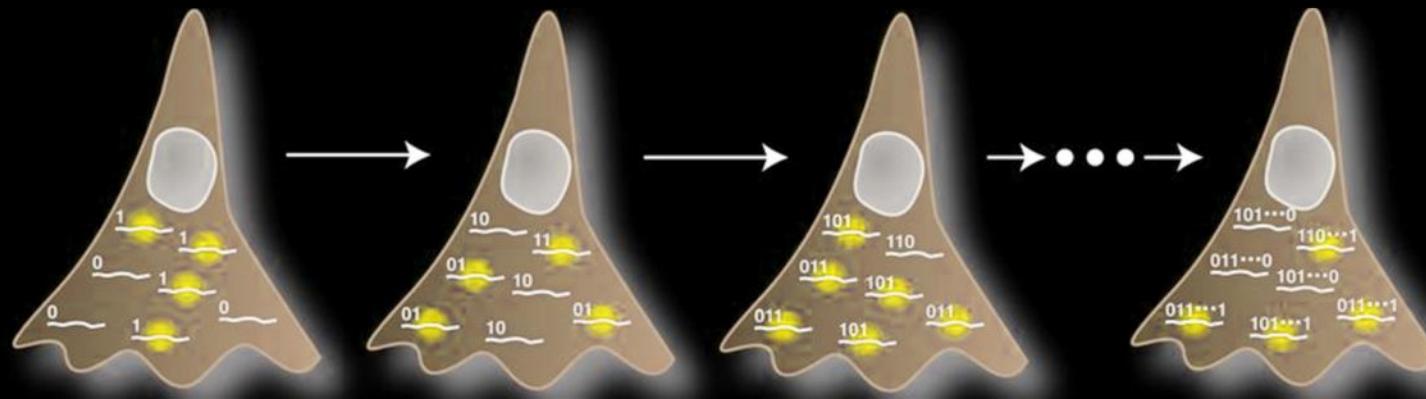
# Sequential smFISH

- Consider I am imaging 1 fluorescence probe color. I can do 10 rounds of imaging.
- Using sequential smFISH, how many different gene species can I detect?

# Multiplexed smFISH



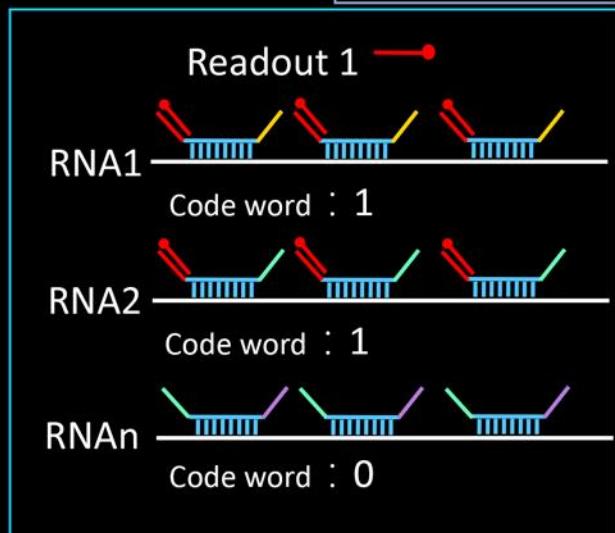
# Multiplexed error-robust fluorescence in situ hybridization (MERFISH) enables simultaneous transcriptome-scale measurements



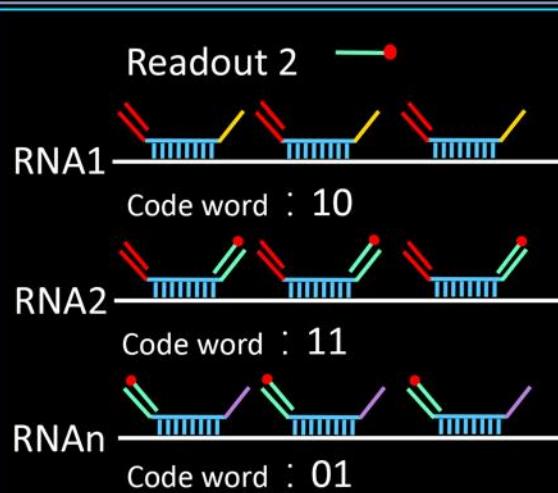
- Combinatorial labeling
- Error-robust barcoding
- Sequential imaging

(Chen et al., Science, 2015)

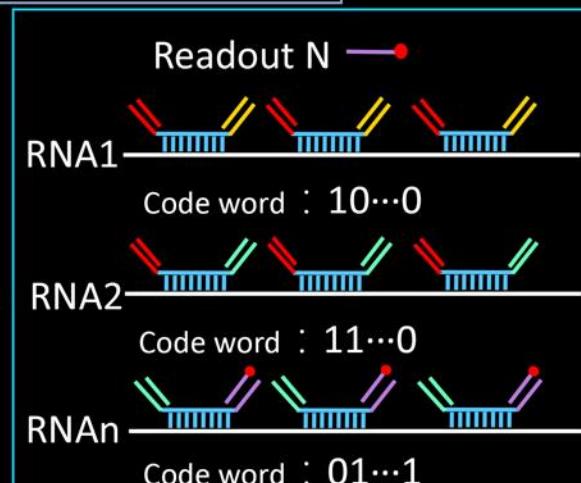
Computationally assign: RNA1:101...0, RNA2:110...0 , ..., RNAn:010...1



chemical cleavage

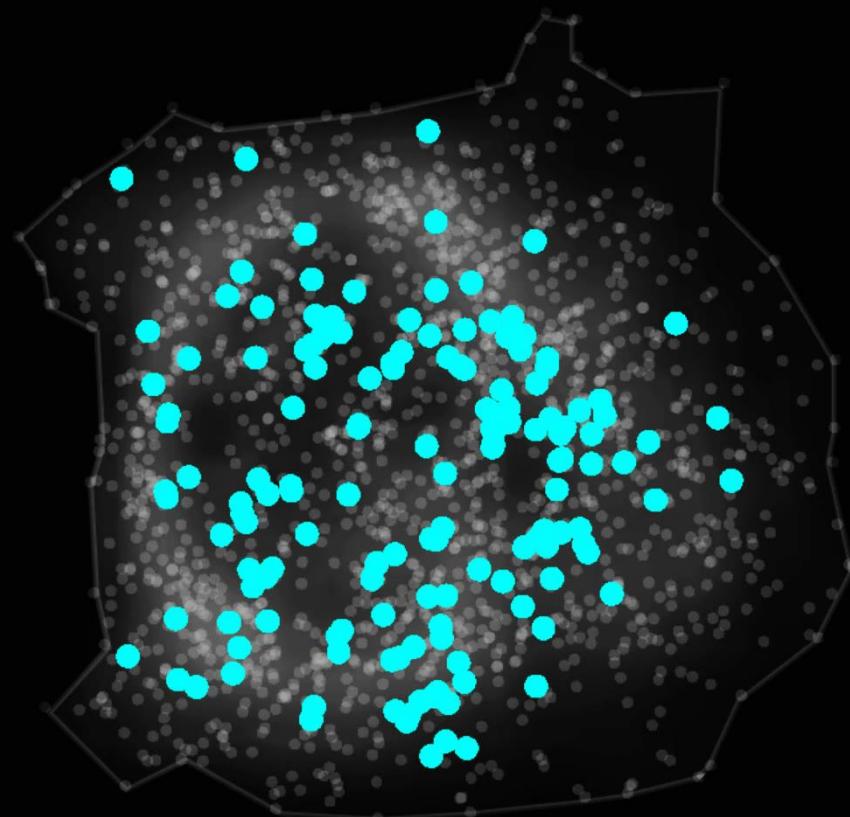


chemical cleavage



(Slide courtesy of Chenglong Xia)

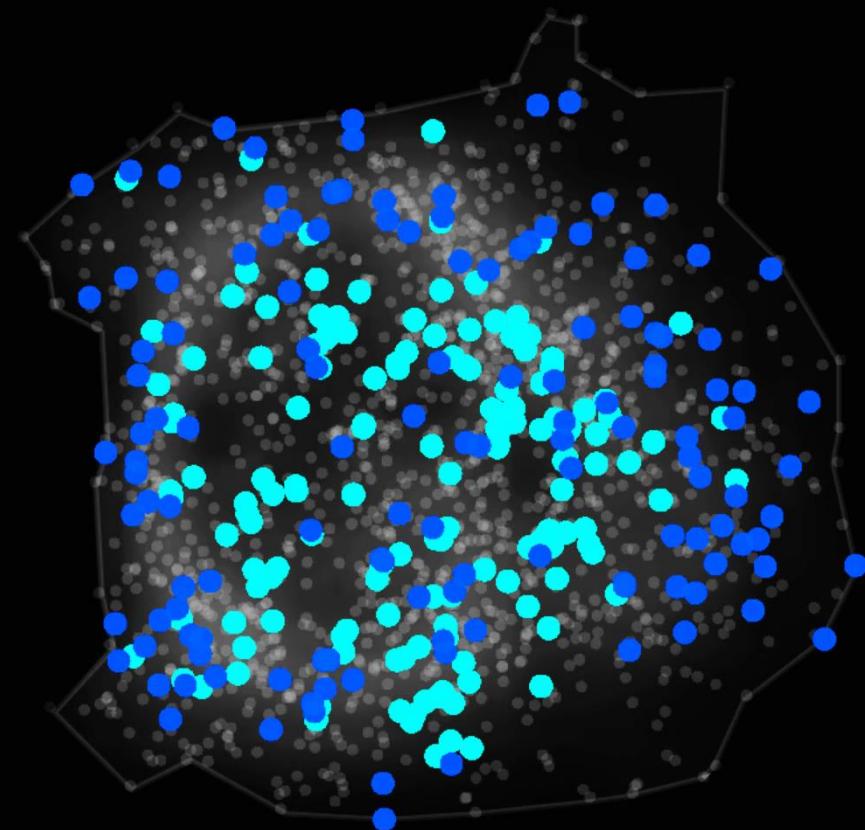
## Gene A



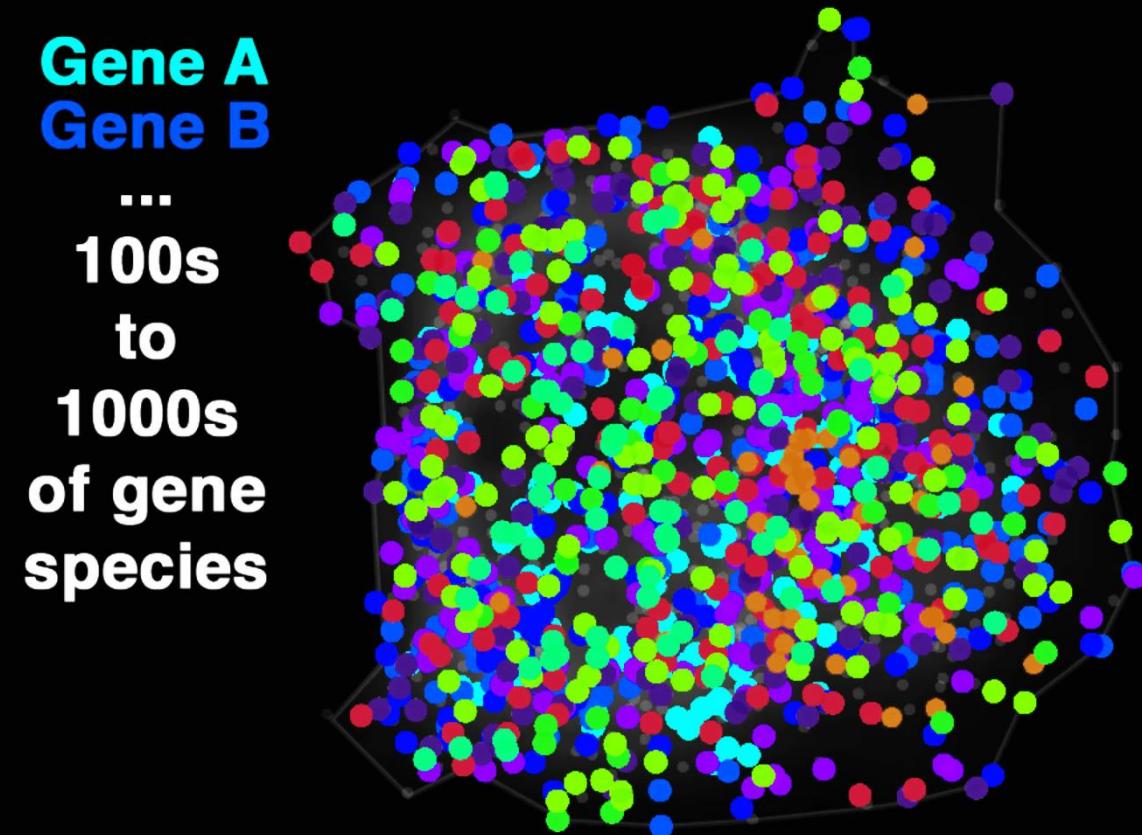
(Xia\*, Fan\*, Emanuel\*, et al PNAS, Sept 2019)

(MERFISH: Chen et al, Science 2015)  
(SeqFISH: Lubeck et al, Nature Methods, 2014)  
(osmFISH: Codeluppi et al, Nature Methods 2018)  
(STARmap: Wang et al, Science 2018)

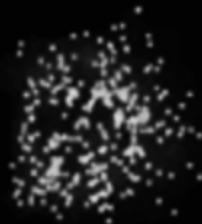
Gene A  
Gene B

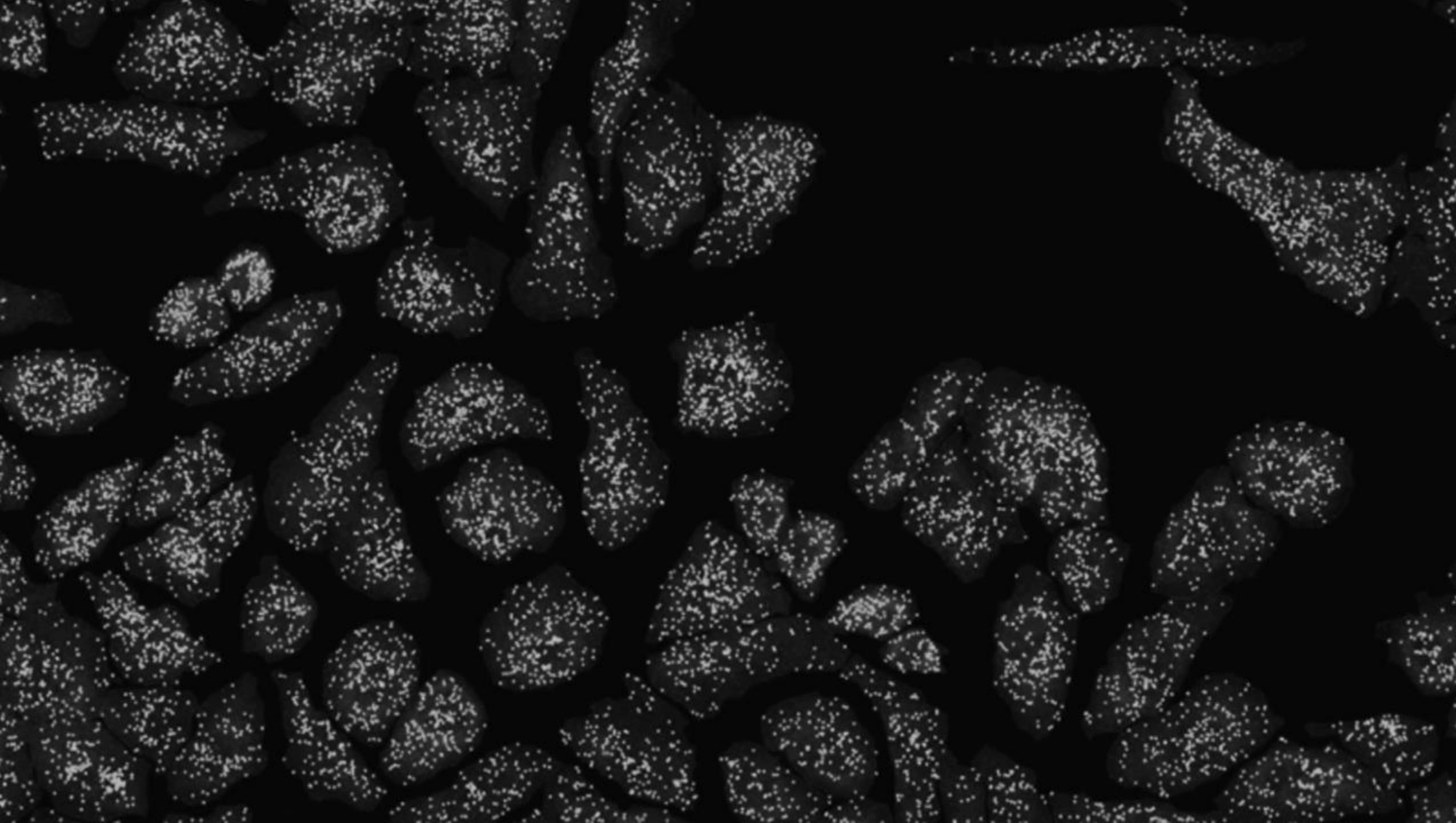


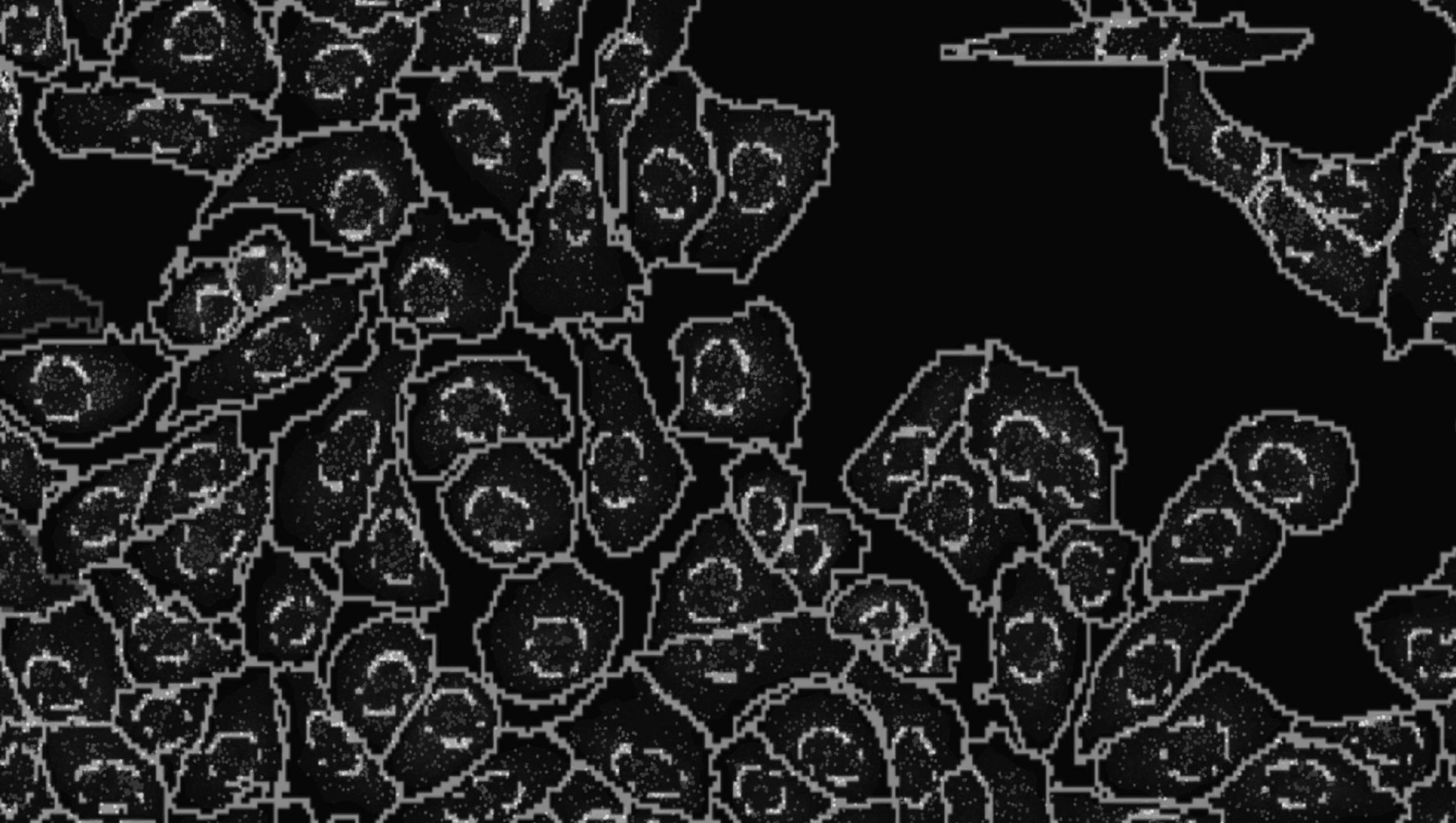
(MERFISH: Chen et al, Science 2015)  
(SeqFISH: Lubeck et al, Nature Methods, 2014)  
(osmFISH: Codeluppi et al, Nature Methods 2018)  
(STARmap: Wang et al, Science 2018)



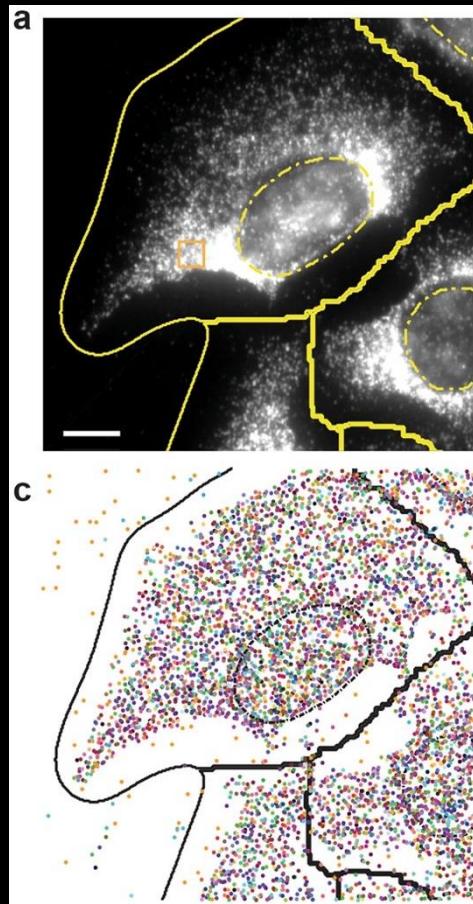
(MERFISH: Chen et al, Science 2015)  
(SeqFISH: Lubeck et al, Nature Methods, 2014)  
(osmFISH: Codeluppi et al, Nature Methods 2018)  
(STARmap: Wang et al, Science 2018)







Cell segmentation + spot detection + gene counting within cells = cell x gene count matrix

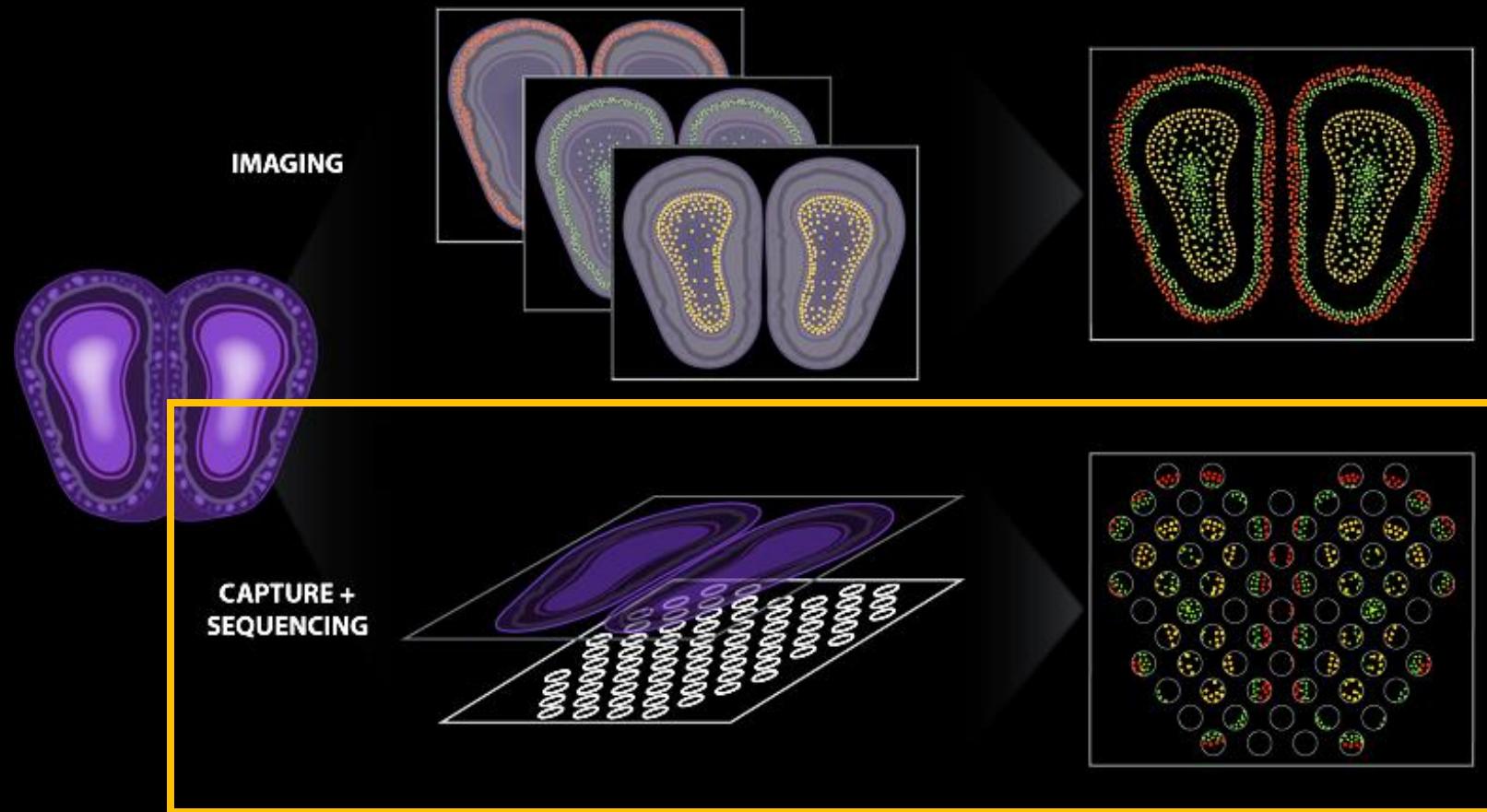


name	cell1	cell2	cell3	cell4	cell5	cell6	cell7	cell8	cell9	cell10	cell11	cell12	cell13	cell14
Plp1	2	3	0	0	0	0	0	0	1	0	63	143	0	0
Atp1b1	0	0	0	0	1	0	1	1	0	3	1	9	3	0
Snap25	0	2	0	1	0	0	0	1	0	1	4	48	0	1
Meg3	0	0	0	0	2	12	0	0	0	0	9	22	1	0
Apoe	42	87	55	107	90	126	104	71	74	124	82	64	259	140
Syt1	0	1	0	1	1	6	0	0	1	0	5	19	2	0
Rtn1	1	0	0	1	0	4	1	2	3	0	3	10	2	0
Snhg11	0	0	0	0	0	0	0	0	0	0	5	14	0	0
Nrgn	0	0	0	6	0	3	0	1	1	0	14	7	0	0
Cpe	53	66	96	84	110	243	108	65	108	127	101	38	235	168
Npy	0	0	0	0	0	0	0	0	0	0	2	115	1	0
Stmn3	5	0	0	4	0	0	0	0	0	0	6	12	0	0
Atp1a2	80	144	58	77	32	80	49	75	151	135	82	36	201	138
Mbp	0	0	0	0	0	0	0	0	0	0	1	19	0	0
Slc1a2	118	191	163	104	144	135	83	119	114	197	94	70	229	215
Ndrg4	0	0	0	1	0	0	0	0	0	1	0	8	0	2
Scg5	0	0	0	0	0	4	0	7	0	0	2	38	3	0
Cst3	185	38	211	134	54	157	141	184	57	133	132	57	100	248
Sparcl1	126	159	113	89	87	128	81	125	53	141	107	91	264	219
Calm2	3	0	0	11	3	0	0	0	1	1	3	52	0	0
Basp1	0	0	0	3	0	0	0	0	0	0	3	1	0	0
Map1b	2	1	0	5	0	6	0	0	1	0	2	7	0	0
Ofm1	0	0	0	0	0	1	0	0	1	0	2	4	0	3
Ttc3	0	5	3	1	2	6	0	2	4	6	8	20	3	6
Camk2n1	0	1	5	4	0	4	1	3	2	1	1	5	1	1
Atp6vOc-ps2	5	0	3	7	5	1	0	2	4	6	5	8	6	0
Gabbr1	0	0	0	0	0	2	0	0	0	0	0	5	0	0
Gria2	4	1	4	3	1	12	0	4	0	6	6	6	1	4
Vsnl1	0	0	0	1	0	0	0	0	0	0	0	5	0	0
Nsf	0	2	1	2	0	0	0	0	0	0	0	7	0	0
Dnm1	0	0	0	3	0	4	0	1	0	0	4	2	0	0
Chnl	1	2	0	1	0	0	0	0	1	0	6	7	0	5
Gpm6a	18	8	18	18	25	13	15	33	8	24	22	30	15	8

Spatially resolved transcriptomic technologies enable us to profile what genes are expressed by cells in tissues

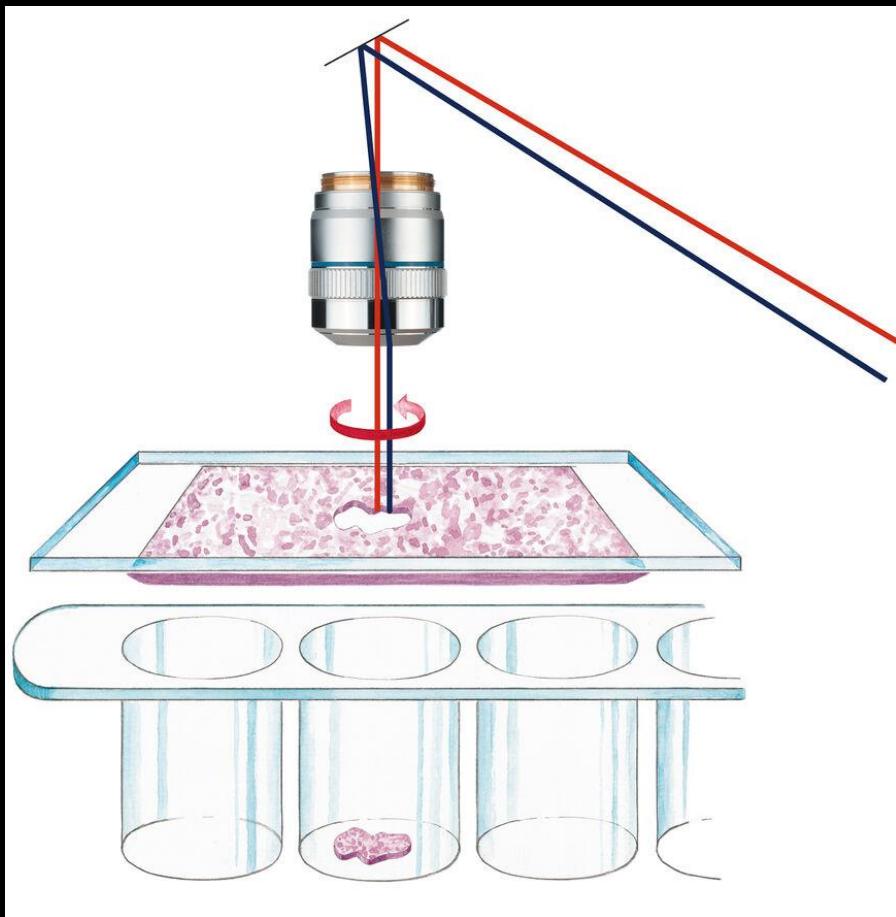


Lyla  
Atta



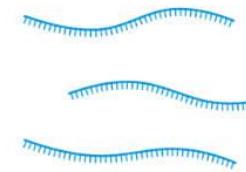
(Atta and Fan, Nature Communications, Sept 2021)

# Microdissection allows small chunks of tissues to be sequenced for transcriptome profiling

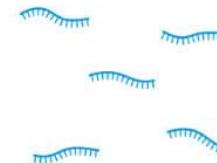


## RNA Sequencing

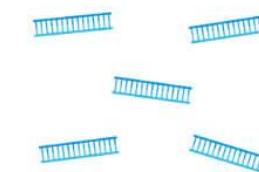
- 1 Isolate RNA from samples



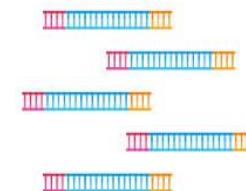
- 2 Fragment RNA into short segments



- 3 Convert RNA fragments into cDNA



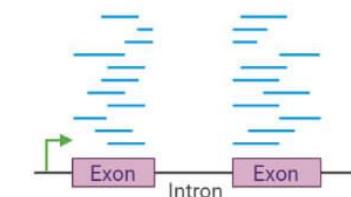
- 4 Ligate sequencing adapters and amplify



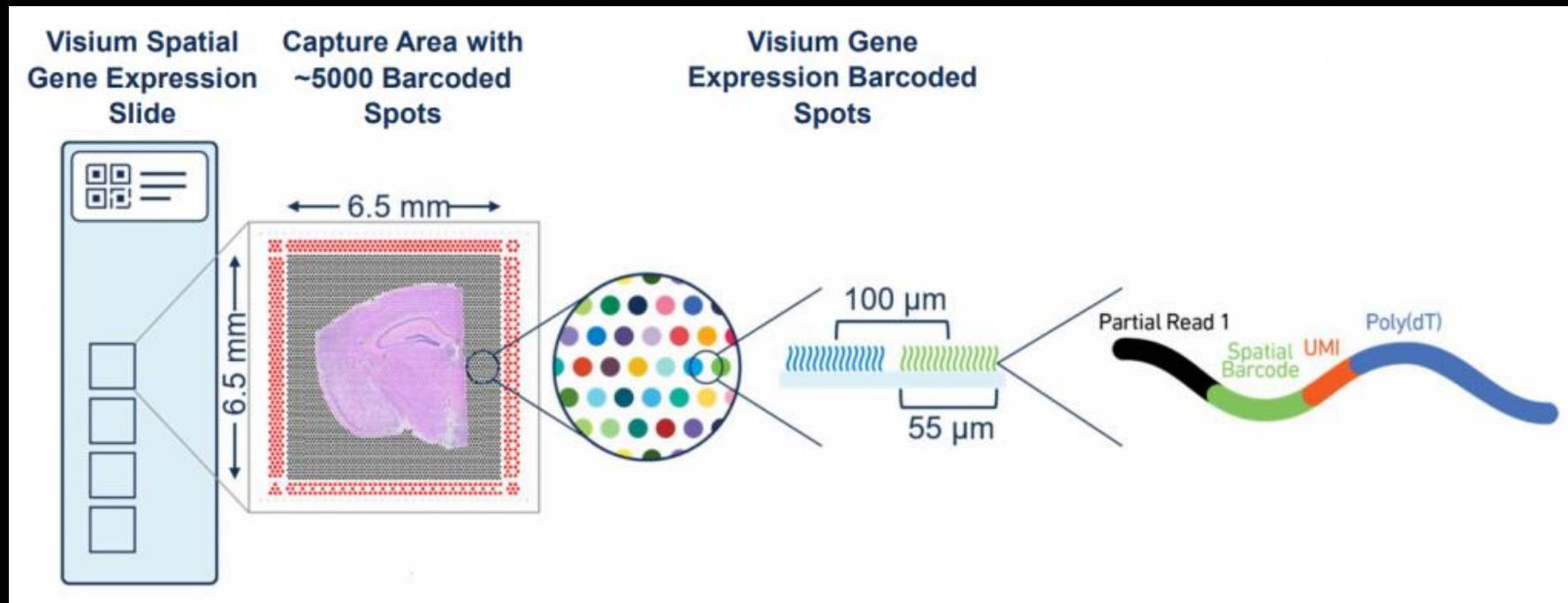
- 5 Perform NGS sequencing



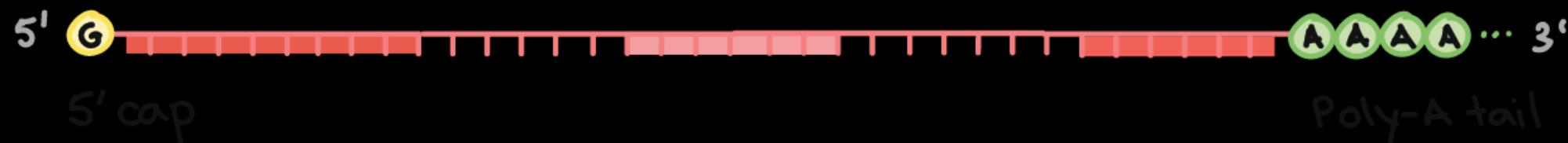
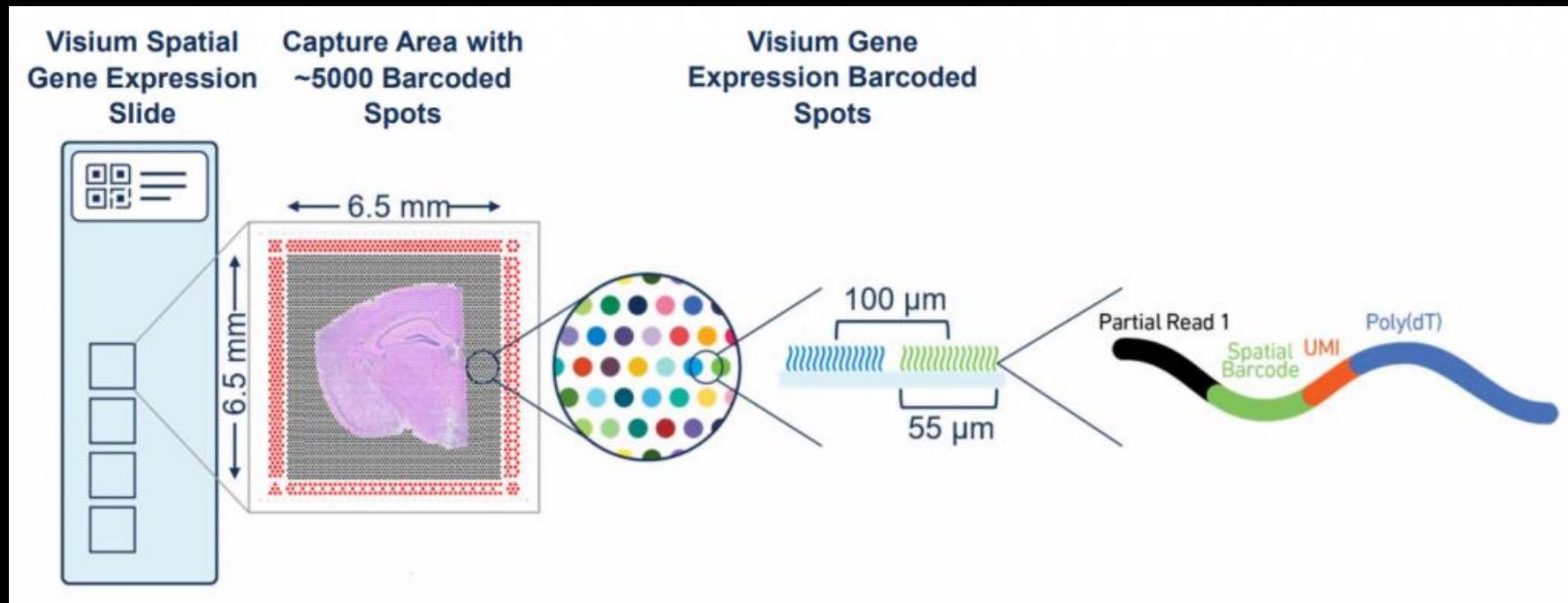
- 6 Map sequencing reads to the transcriptome/genome



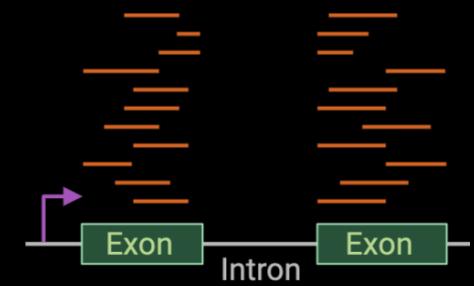
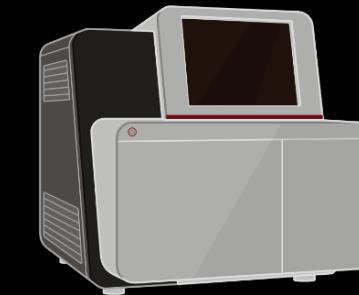
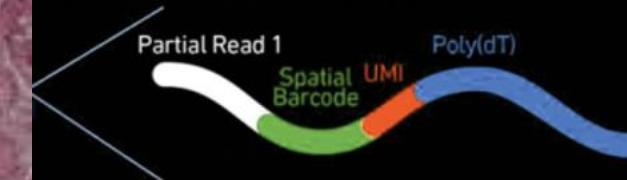
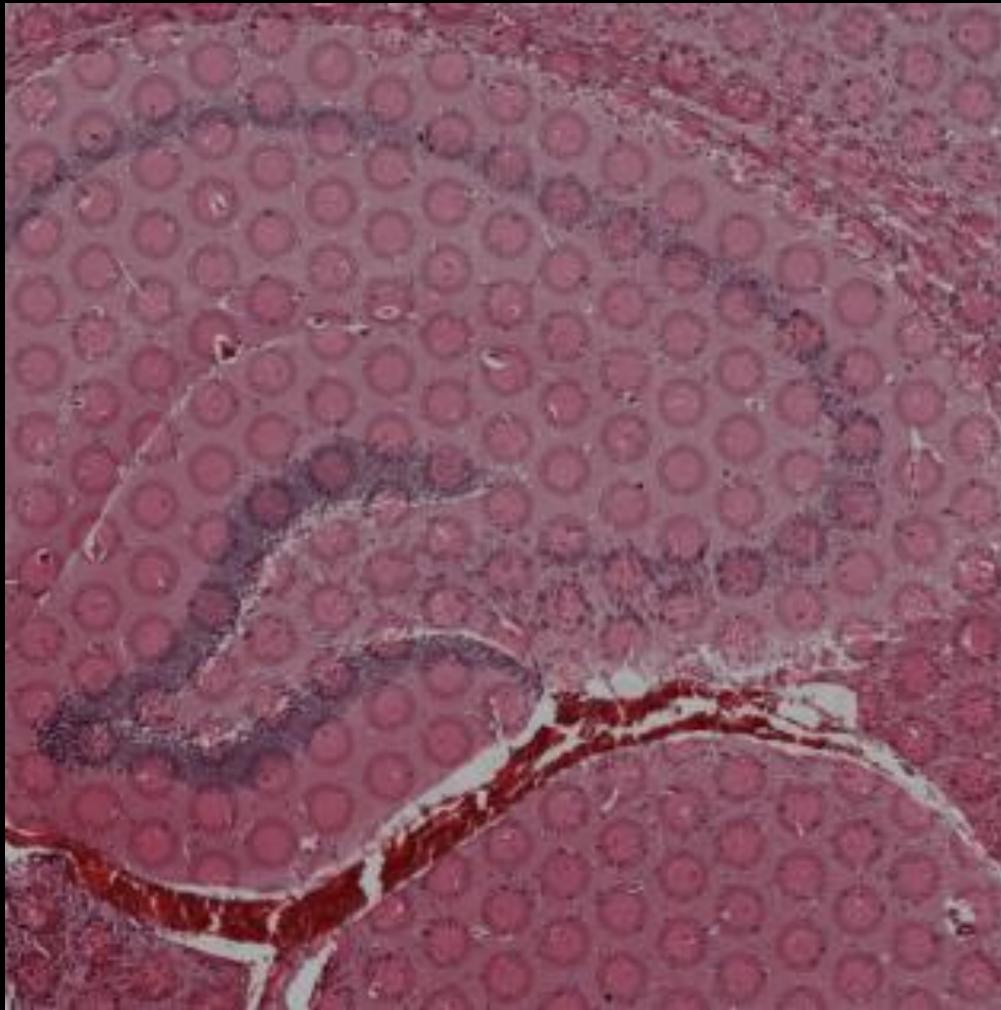
# Spatial-barcode bead capture



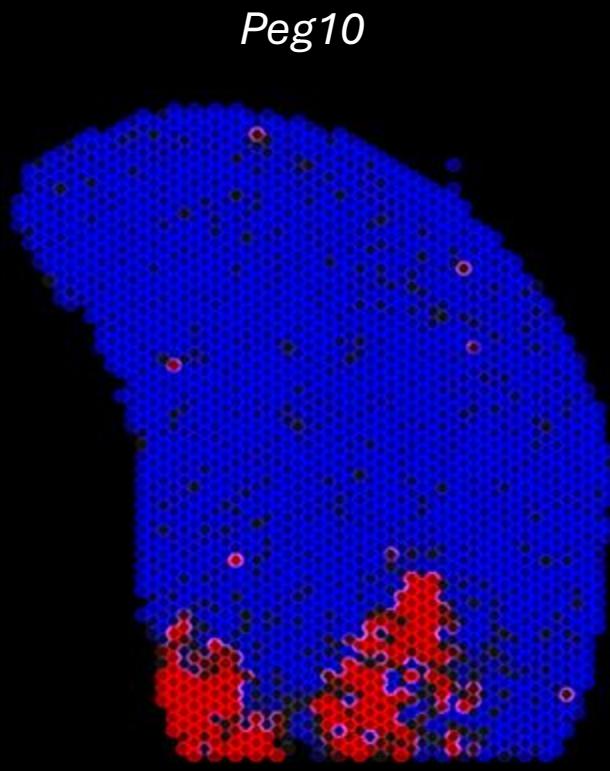
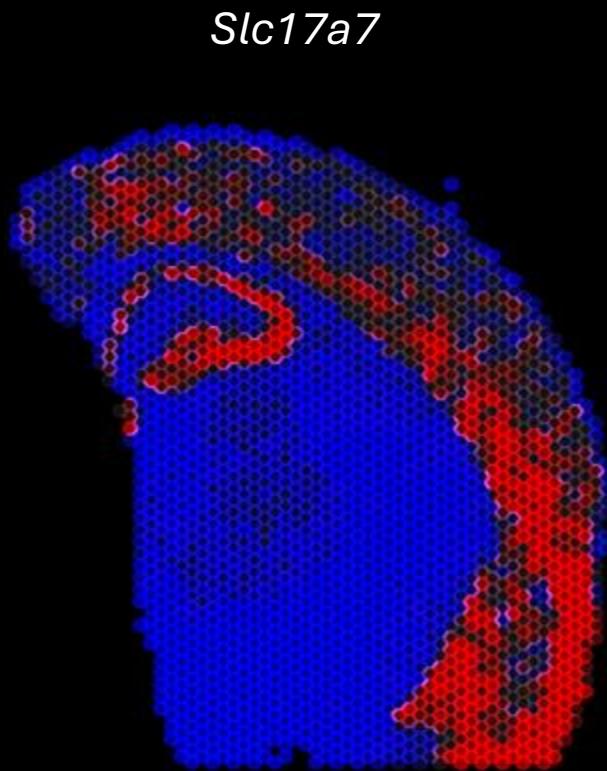
# Recall: mRNAs have poly(A)-tails



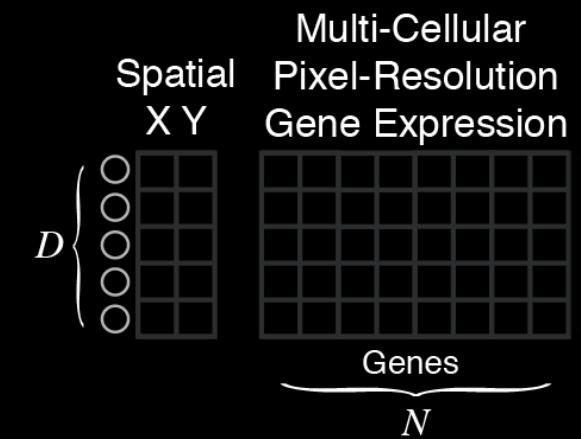
# Spatial-barcode bead capture



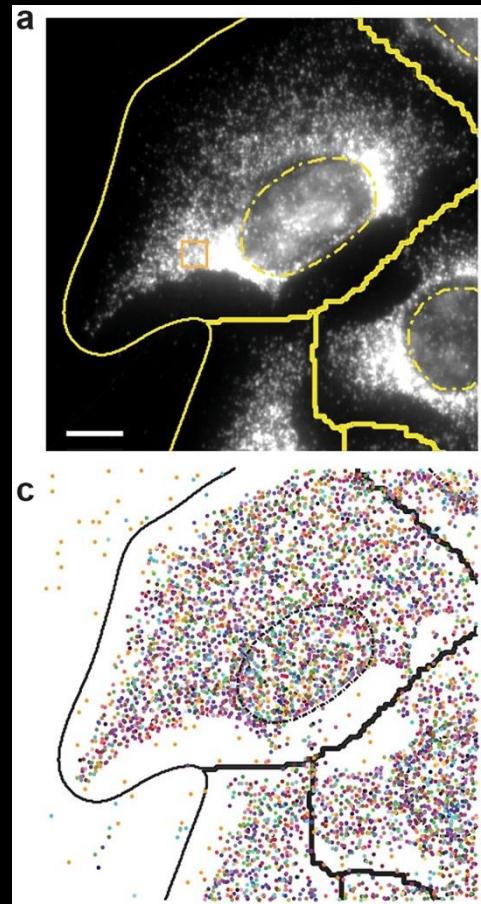
# Spatial-barcode bead capture



...

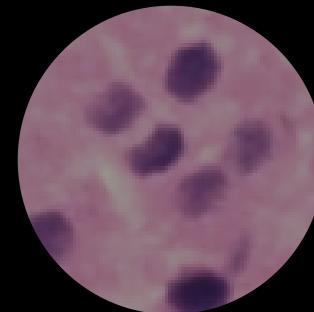
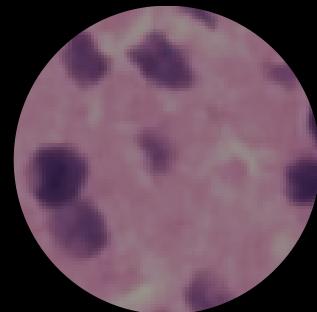


# So, what do these numbers represent here?



name	cell1	cell2	cell3	cell4	cell5	cell6	cell7	cell8	cell9	cell10	cell11	cell12	cell13	cell14	...
Plp1	2	3	0	0	0	0	0	0	1	0	63	143	0	0	0
Atp1b1	0	0	0	0	1	0	1	1	0	3	1	9	3	0	0
Snap25	0	2	0	1	0	0	0	1	0	1	4	48	0	1	0
Meg3	0	0	0	0	2	12	0	0	0	0	9	22	1	0	0
Apoe	42	87	55	107	90	125	104	71	74	124	82	64	259	140	0
Syt1	0	1	0	1	1	5	0	0	1	0	5	19	2	0	0
Rtn1	1	0	0	1	0	4	1	2	3	0	3	10	2	0	0
Snhg11	0	0	0	0	0	0	0	0	0	0	5	14	0	0	0
Ngn	0	0	0	6	0	3	0	1	1	0	14	7	0	0	0
Cpe	53	66	96	84	110	243	108	65	108	127	101	38	235	168	0
Npy	0	0	0	0	0	0	0	0	0	0	2	115	1	0	0
Stmn3	6	0	0	4	0	0	0	0	0	0	6	12	0	0	0
Atp1a2	80	144	58	77	32	80	49	75	151	135	82	36	201	138	0
Mbp	0	0	0	0	0	0	0	0	0	0	1	19	0	0	0
Sle1a2	118	191	163	104	144	135	83	119	114	197	94	70	229	215	0
Ndrg4	0	0	0	1	0	0	0	0	0	1	0	8	0	2	0
Scg5	0	0	0	0	0	4	0	7	0	0	2	38	3	0	0
Cst3	185	38	211	134	54	157	141	184	57	133	132	57	100	248	0
Sparcl1	126	159	113	89	87	128	81	125	53	141	107	91	264	219	0
Calm2	3	0	0	11	3	0	0	0	1	1	3	52	0	0	0
Basp1	0	0	0	3	0	0	0	0	0	0	3	1	0	0	0
Map1b	2	1	0	5	0	6	0	0	1	0	2	7	0	0	0
Ofm1	0	0	0	0	0	1	0	0	1	0	2	4	0	3	0
Ttc3	0	5	3	1	2	6	0	2	4	6	8	20	3	6	0
Camk2n1	0	1	5	4	0	4	1	3	2	1	1	5	1	1	1
Atp6v0c-pe2	5	0	3	7	5	1	0	2	4	6	5	8	6	0	0
Gabra1	0	0	0	0	0	2	0	0	0	0	0	5	0	0	0
Gria2	4	1	4	3	1	12	0	4	0	6	6	6	1	4	0
Vsnl1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0
Nsf	0	2	1	2	0	0	0	0	0	0	0	7	0	0	0
Dnm1	0	0	0	3	0	4	0	1	0	0	4	2	0	0	0
Chn1	1	2	0	1	0	0	0	0	1	0	6	7	0	5	0
Gpm6a	18	8	18	18	25	13	15	33	8	24	22	30	15	8	0

What happens if you have multiple cells in the same spatial spot?



What are the pros and cons of each method?

# Pros and Cons of each method

Feature	smFISH	10x Visium
Spatial Resolution	High (subcellular)	Moderate (multi-cell per spot)
Gene Throughput	Low to Moderate	High (whole transcriptome)
Quantification	Absolute (molecule counting)	Relative (RNA-seq based)
Ease of Use	Complex and labor-intensive	Streamlined and user-friendly
Cost	Lower (no sequencing needed)	Higher (sequencing required)
Customization	High (design specific probes)	Limited (fixed panel for mRNA)

- **smFISH:** Ideal for studies requiring single-molecule precision, subcellular resolution, or targeted analysis of specific genes.
- **10x Visium:** Suitable for genome-wide profiling in a spatial context, especially when large tissue areas are of interest.

# Reflection Cards

## Reflection Card

Please reflect on today's lesson in Deep Learning for Spatial Transcriptomics 2025.

Reflection cards are not graded for content. However, the contents of these reflection cards may help identify potential common areas of confusion that can be addressed in the next class along with helping me make the class better :)

- Essentially a means to help me make this class better!
1. What is something that you learned in today's lecture?
  2. What is something that you are still confused about from today's lecture?
  3. Do you have any other comments/feedback/thoughts/suggestions/concerns?

<https://forms.gle/STLGPtcg6NUdnbmw8>

# Possible Routes for class

- Go more in depth on how neural networks work/function
  - Discuss mathematics behind it
  - Walk through from ground up
  - Focus on CNNs, or transformers, or any other popular model
- Discuss more on spatial transcriptomics
  - Different types and how/why they work
- Literature Review
  - Read through papers together on various topics and discuss as a group
- Pytorch crash course
  - How to code neural networks in pytorch and understand them
- Anything else that comes to mind!