Acknowledgement of Country

The University of Queensland (UQ) acknowledges the Traditional Owners and their custodianship of the lands on which we meet.

We pay our respects to their Ancestors and their descendants, who continue cultural and spiritual connections to Country.

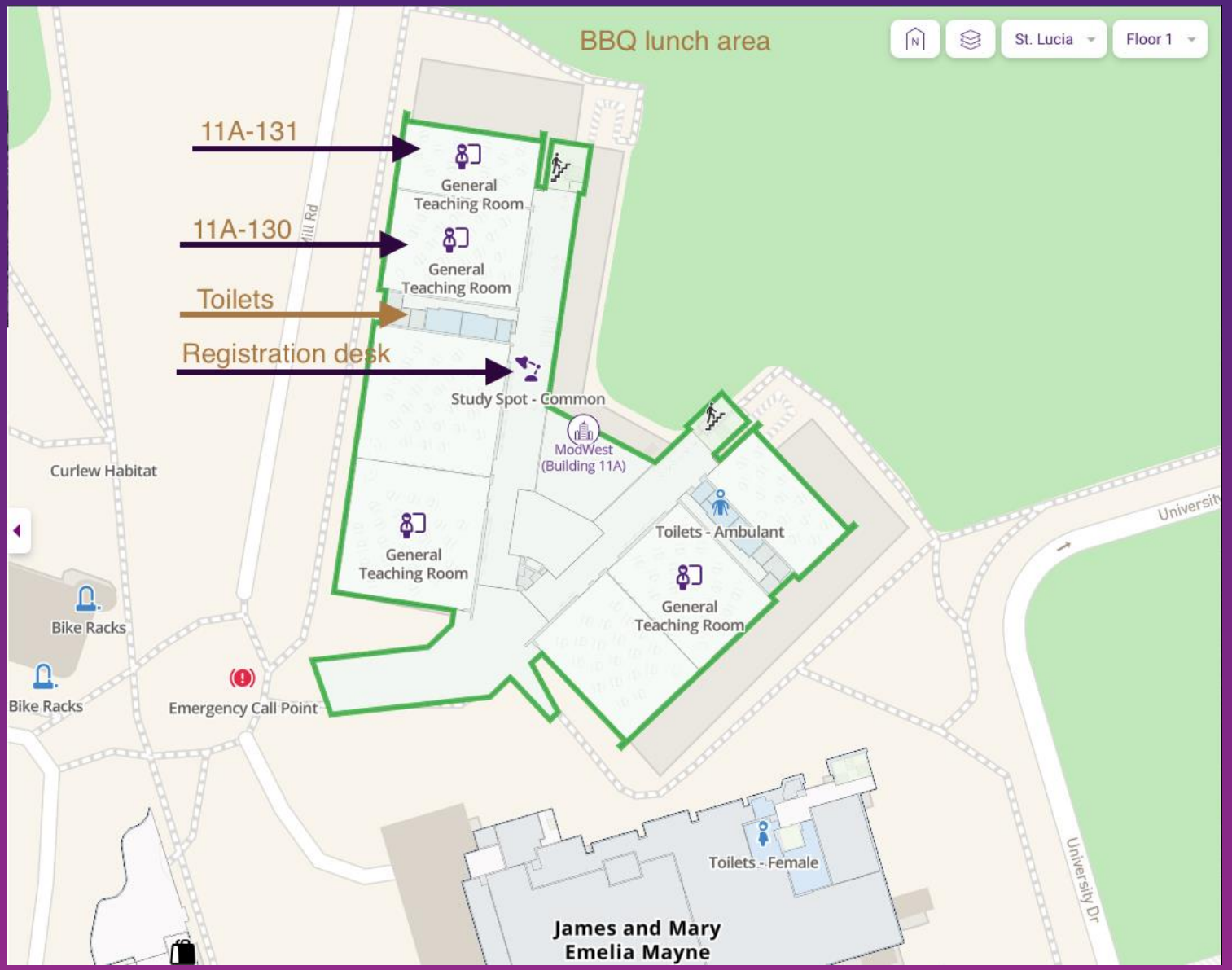
We recognise their valuable contributions to Australian and global society.

Image: Digital reproduction of *A guidance through time* by Casey Coolwell and Kyra Mancktelow

General Information:

- We are currently located in Building 11A MODWEST
  - Bathrooms
  - Vending machines
- Food court and other bathrooms are located in Building 63 or Building 21B
- If you are experiencing cold/flu symptoms or have had COVID in the last 7 days please ensure you are wearing a mask for the duration of the module

EMERGENCY CONTACT/UQ SECURITY: 3365 3333



Learning materials

Instructions to access WiFi/desktop/server:

https://cnsgenomics.com/data/teaching/GNGWS26/module2/

The winter school server is available until **24th July 2026** (2 weeks after the course)

Slides and practical notes for this module:

https://cnsgenomics.com/data/teaching/GNGWS26/module2/

Module 2 Cellular Omics

Room 315/316, Building 69

Lecturers/Instructors: Quan Nguyen, Xiao Tan, Prakrithi Pavithra

Module 2 Cellular Omics – Leaning Objectives

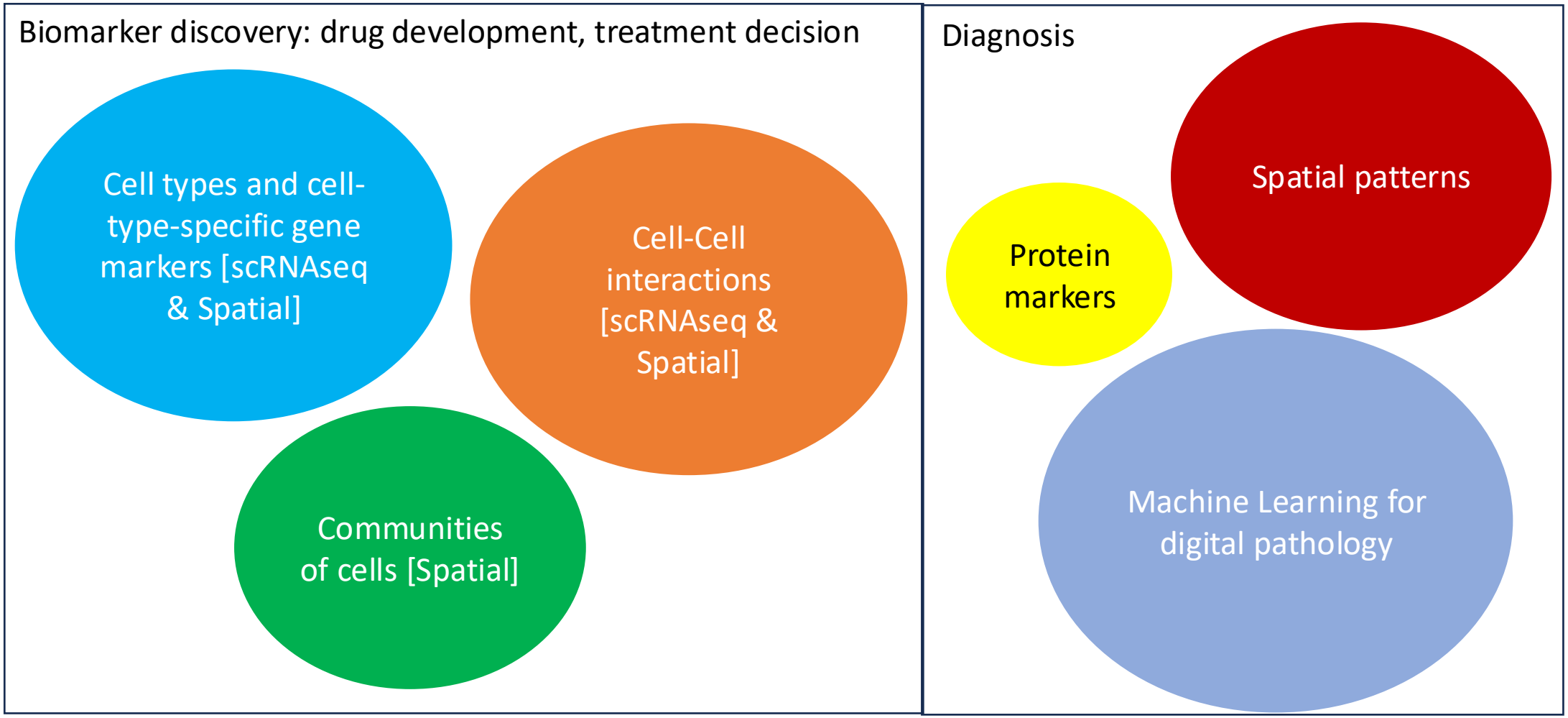
- [Day 1] How the data get generated experimentally – latest sequencing and imaging technologies:
  - Technologies for generating single-cell and spatial transcriptomics data
  - Technologies for other spatial omics, focusing on proteomics

- [Day 1] Exploratory visualisation to understand the data

- [Days 1, 2] Statistical analyses to discover new biological processes and biomarkers associated with disease, including cells, genes and groups of cells within the tissue. This includes:
  - Identifying cell types
  - Finding gene markers
  - Mapping cell neighbourhoods (cell communities)
  - Multiomics integration
    - Spatial transcriptomics + proteomics
    - gsMAP: incorporating GWAS with Spatial transcriptomics

- [Day 2] Machine learning analysis of sequencing and imaging data
  - Key concepts
  - Practical hands-on analyses
  - Transformer models for gene expression
  - Vision transformer for H&E images

Module 2 Cellular Omics – Leaning Objectives



Module 2 Cellular Omics – Schedule

- Afternoon 6th July: 1 pm – 4 pm
  - ✓ Session 1: Single cell and spatial technologies (1:00-2:00)
  - ✓ Session 2: Practical - (2:00-2:45)
    - ✓ Single cell analysis (2:00-2:30)

  Break (2:45-3:00)
  - ✓ Session 3: spatial analysis (3:00-3:30)
  - ✓ Session 4: Visualise spatial single cell data (3:30-3:50)
  - ✓ Closing: 3:50-4

- Morning 7th July (9 am – 12 pm) Practical for single cell and spatial analysis
  - ✓ Session 1: Cell typing (9-10)
  - ✓ Session 2: Community analysis (10-10:15)

  Break: 10:15-10:30 am
  - ✓ Session 3: Multiomics Integration lecture (10:30-11:30am)
  - ✓ Practical: gsMap (11.30-12 am)

- Morning 7th July (1 pm – 4 pm) Practical for machine learning
  - ✓ Session 1: Deep learning for spatial data (1:00-1:30)
  - ✓ Session 2: Autoencoder for gene expression (1:30 -2:45)

  Break (2:45-3:00)
  - ✓ Session 3: transformer for genes and images(3:00-3:50)
  - ✓ Closing: 3:50-4:00

Lecture 4: Multiomics Integration

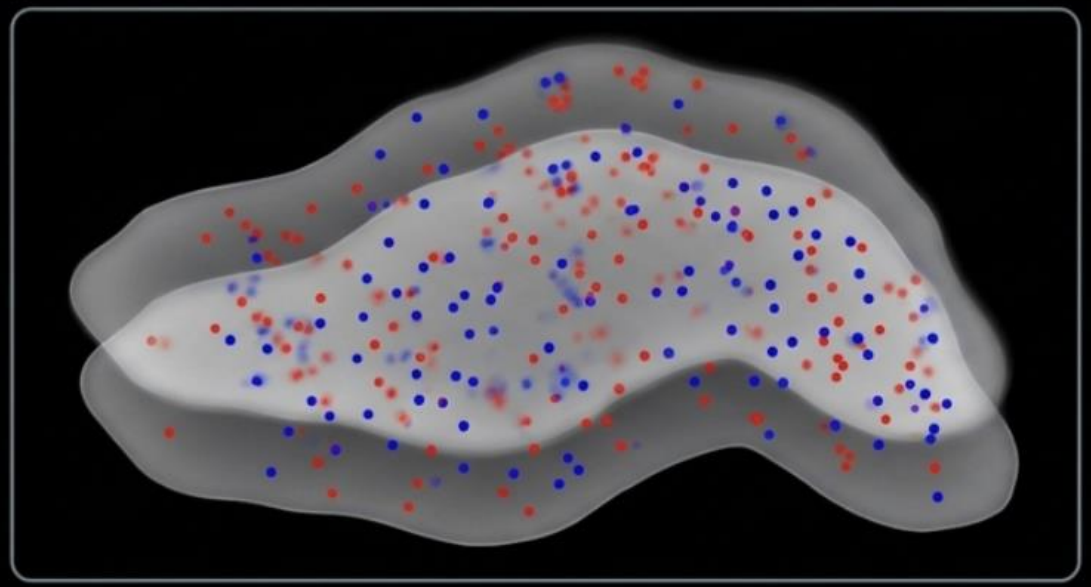
Lecture 4.1: Transcriptomics + Proteomics Spatial

Why Integrate Spatial transcriptomics and proteomicsc

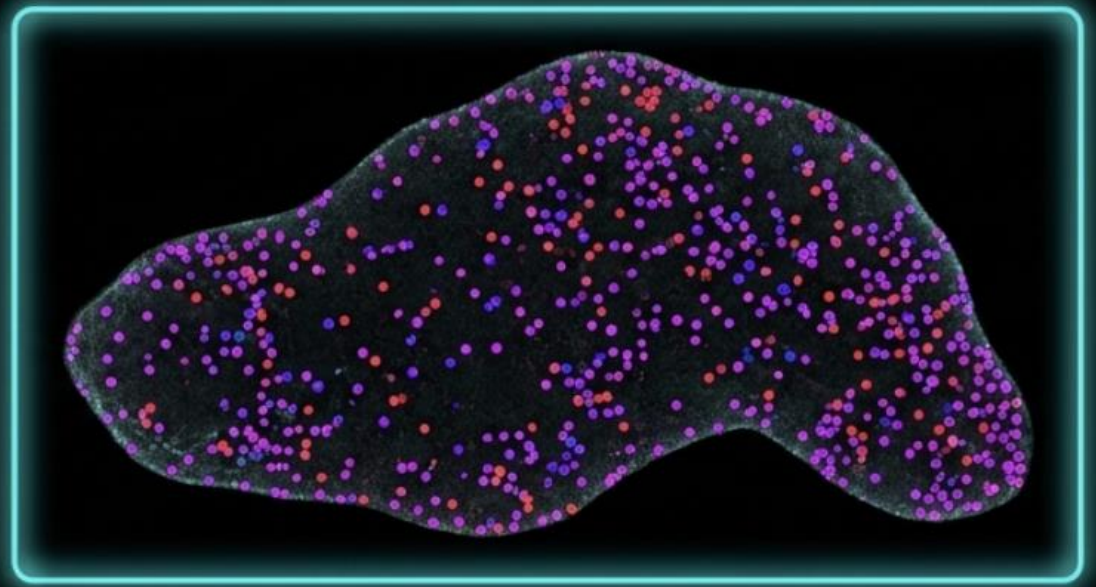
- **Capture complementary biology** by combining spatial transcriptomics (Xenium) with CODEX/ Pheno-cycler fusion (PCF).
- **Measure RNA and protein simultaneously** within the same tissue architecture.
- **Validate transcriptional signals** using corresponding protein expression.
- **Improve cell type and cell state identification** by leveraging both gene and protein markers.
- **Characterize cellular neighborhoods** and tissue microenvironments with greater confidence.
- **Reveal regulatory mechanisms** where RNA and protein abundance differ.
- **Generate a comprehensive spatial atlas** for downstream biological and disease analyses.
- **Take-home message:** *Xenium + PCF provides a more complete view of tissue biology than either modality alone.*

The Misalignment Bottleneck in Dual-Slide Methods

Current Standard: Adjacent Slides

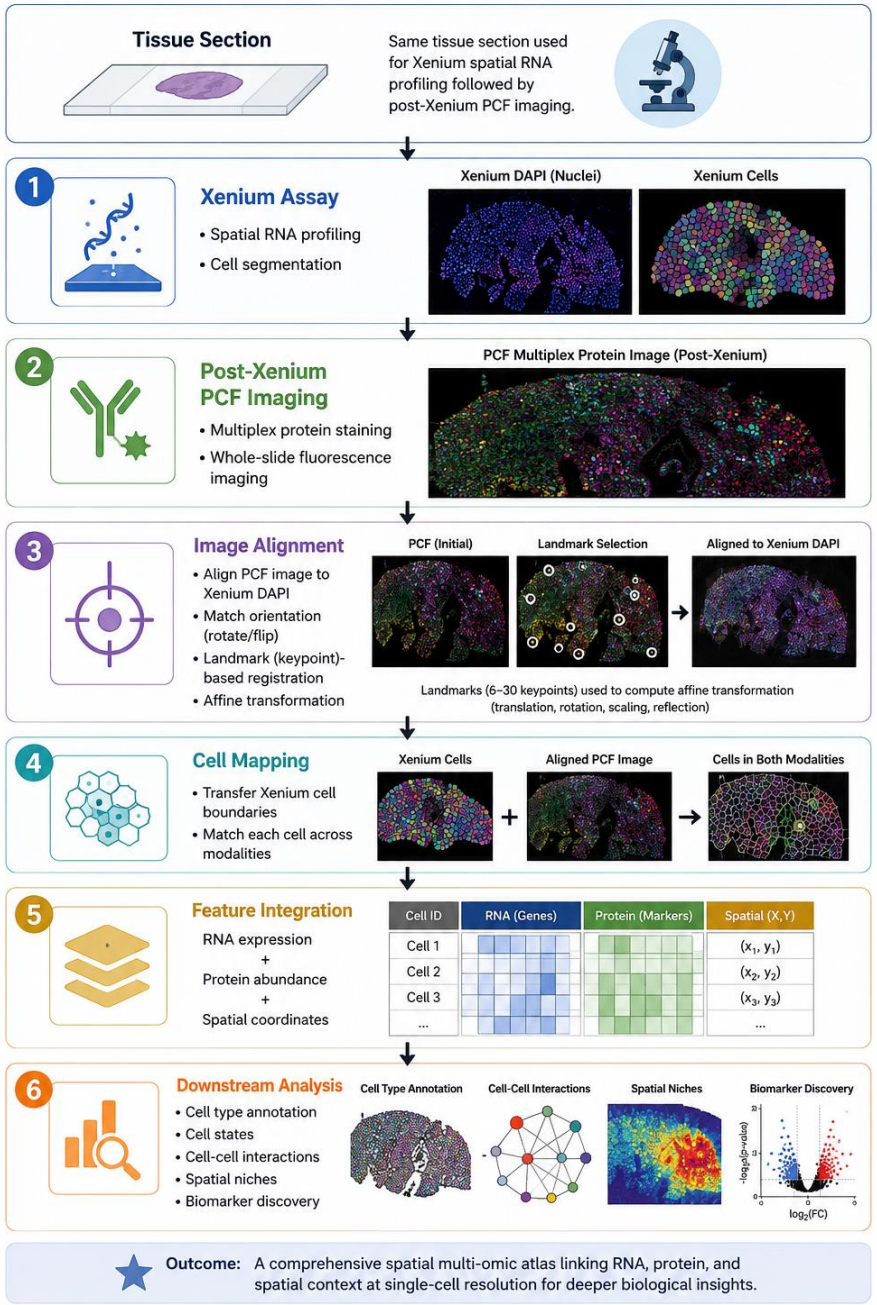


The Innovation: Single-Slide Multi-Omics



Spatial Alignment	Misaligned & Distorted	Perfect 1:1 Correlation
Cellular Context	Variable across 5µm slices	Absolute functional context
Workflow Effort	Redundant prep & complex software alignment	Streamlined physical workflow

Multimodal Integration Workflow (Xenium + PCF)



1. Spatial RNA Profiling

1. Xenium assay on tissue section
2. Cell segmentation & transcript detection

2.Protein Imaging

1. Post-Xenium PCF staining
2. Multiplex protein imaging

3.Image Registration

1. Align PCF to Xenium DAPI
2. Landmark-based affine transformation

4.Cell Matching

1. Transfer Xenium cell boundaries
2. Match corresponding cells across modalities

5.Data Integration

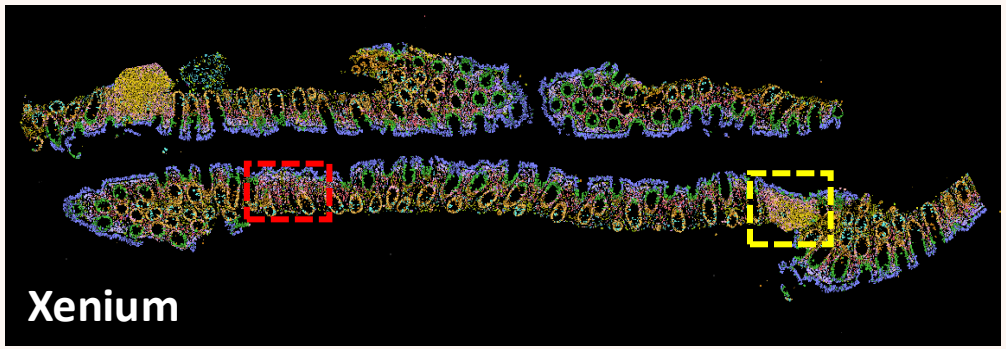
1. Combine RNA, protein, and spatial coordinates
2. Generate a unified single-cell dataset

6.Biological Insights

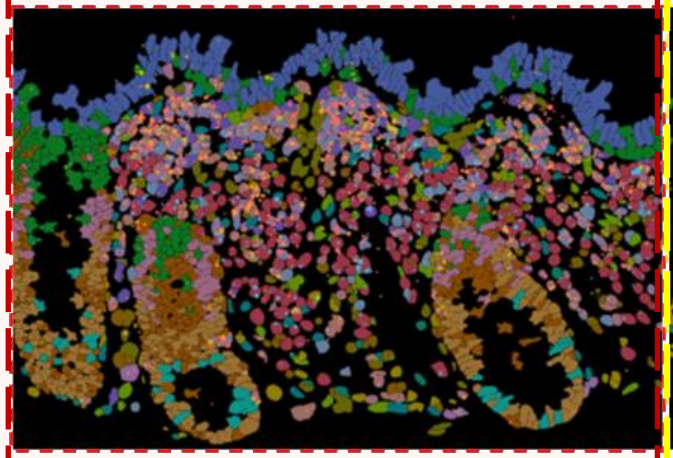
1. Cell type annotation
2. Spatial niches & cell–cell interactions
3. Biomarker and disease mechanism discovery

Aligning cross-platform images with Xenium explorer to overlay multi-omics data

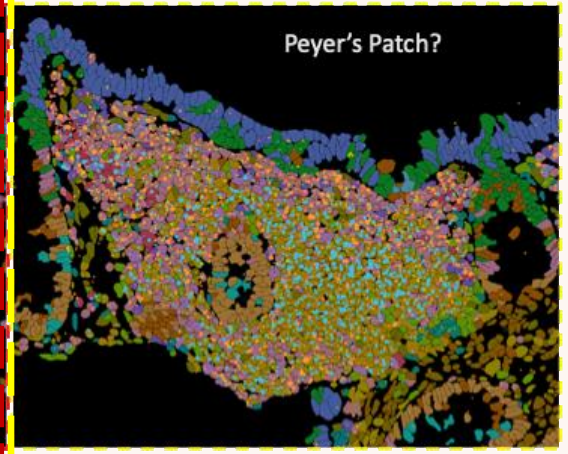
Transcriptomics and Proteomics Overlay

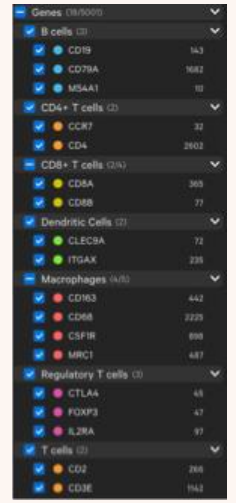


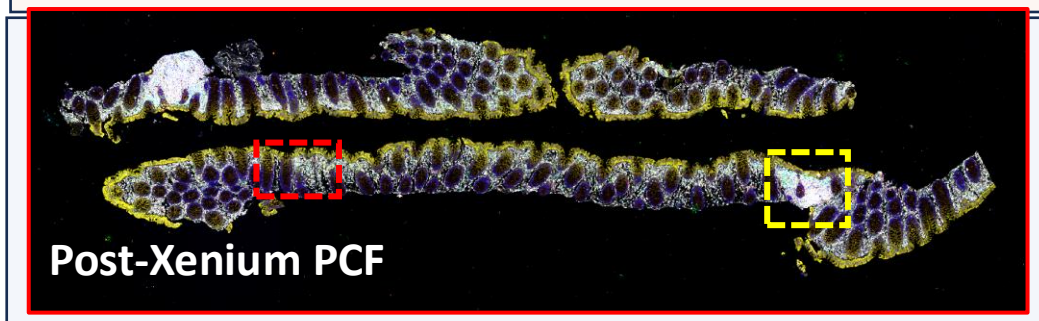
Epithelium and Crypts

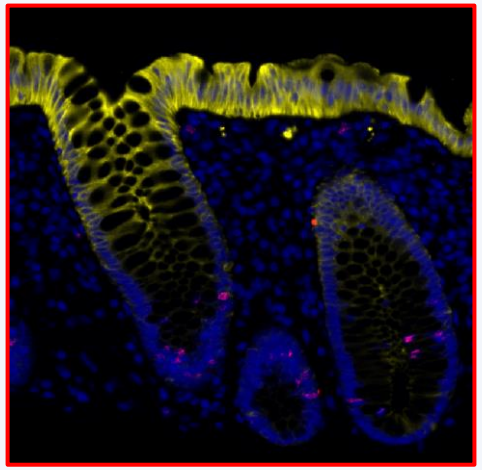


Peyer's Patch



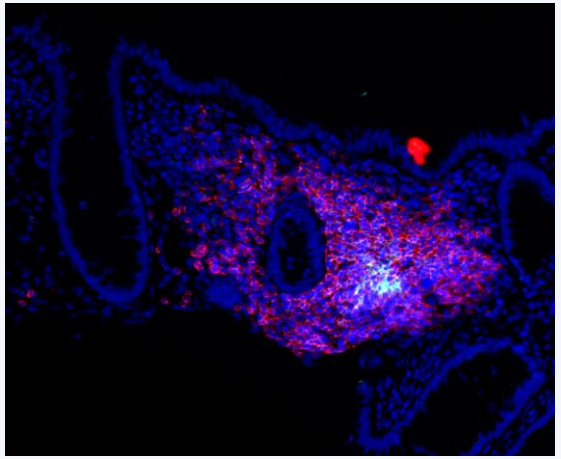






DAPI
Ki-67
Pan-CK

Ki-67 (Stem cells in the Cryp)
Pan-CK (diff. enterocytes on epithelium)



DAPI
CD20
CD21

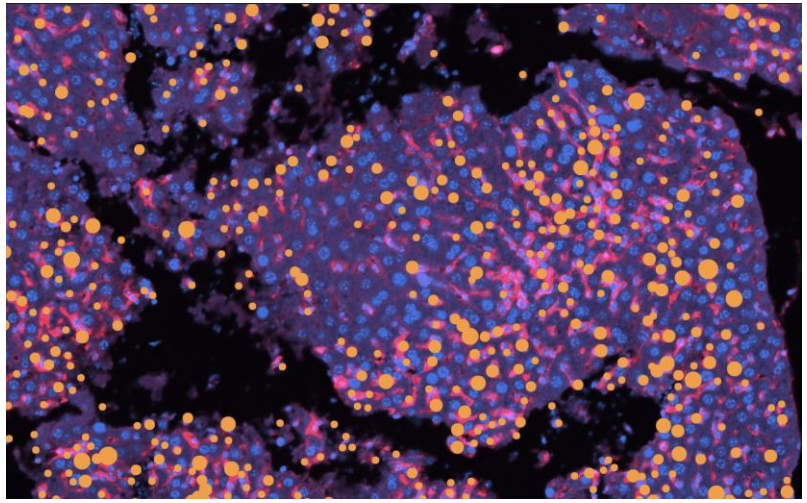
CD20 Mature B cells marker
CD21 Follicular DC/mature B cells

PCF
Xenium

Xenium – PCF Alignment

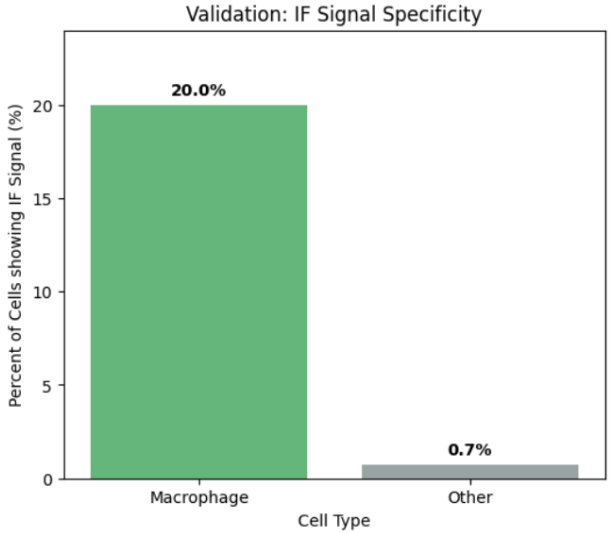
Xenium – Immuno-Fluorescense overlay

Confident identification of **macrophages** using gene and protein markers



- Cd68 (xenium)
- DAPI (IF)
- F4/80 (IF)

96.4% of IF high cells are annotated as Macrophages in xenium



Lecture 4.2: Spatial Transcriptomics + Population Genetics

Spatially resolved mapping of cells associated with human complex traits

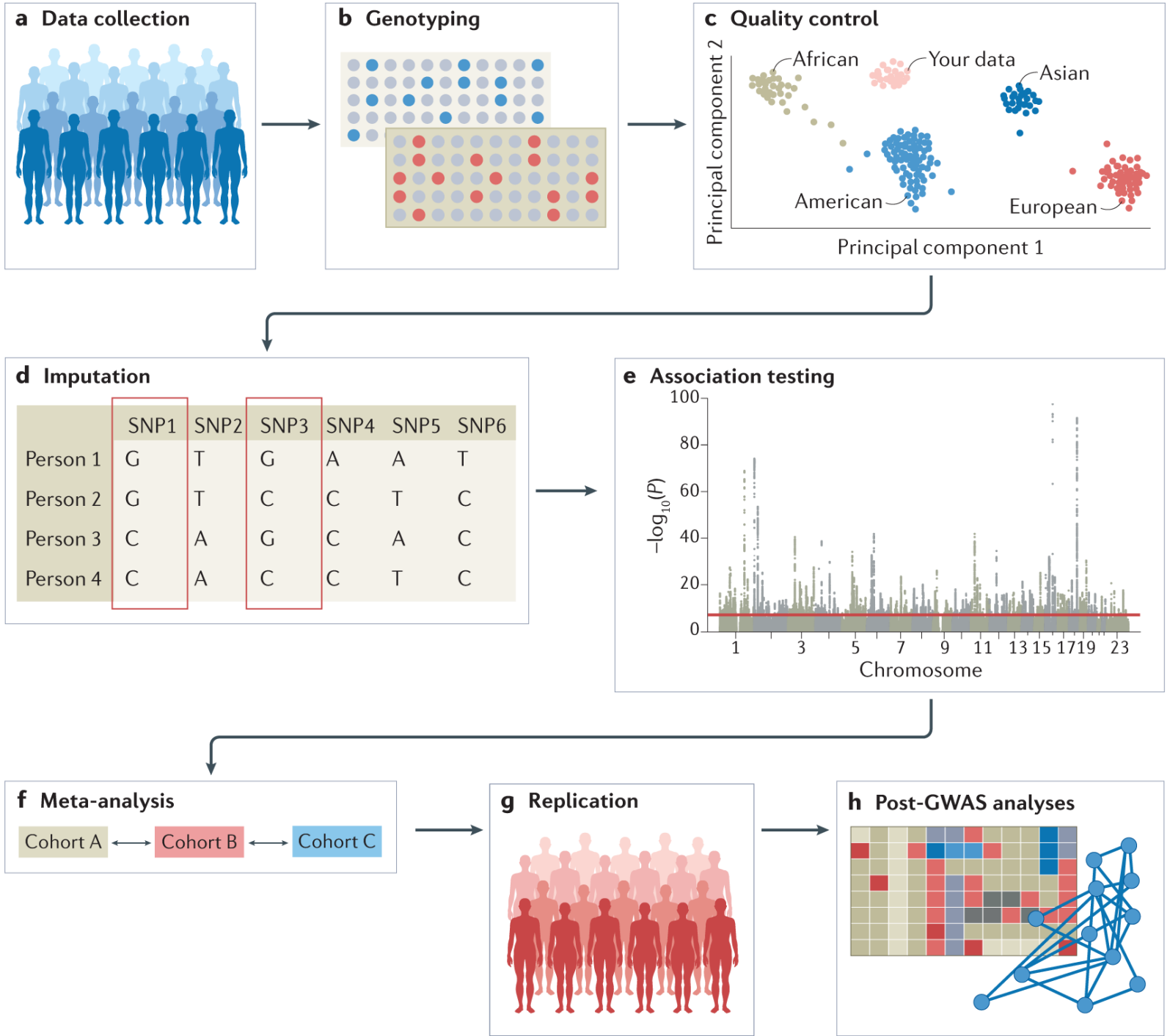
Liyang Song, Wenhao Chen, Junren Hou, Minmin Guo, Jian Yang

doi: https://doi.org/10.1101/2024.10.31.24316538

Why Integrate Genetics and Spatial transcriptomics

- **Connecting Genetic Risk to Tissue Biology**
- **Prioritize disease-associated genes** identified through GWAS.
- **Localize genetic risk** to specific cell types and spatial regions within tissue.
- **Identify disease-relevant cellular niches** enriched for genetic susceptibility.
- **Link genetic variants to biological pathways** in their native tissue context.
- **Bridge population genetics with spatial molecular phenotypes** to uncover mechanisms of SNP-gene regulation
- **Discover candidate therapeutic targets** by identifying where risk genes are active.
- **Improve interpretation of complex diseases** through spatially resolved genetic architecture.
- **Take-home message:** *Spatial mapping of GWAS signals transforms genetic associations into biologically meaningful, tissue-specific insights.*

GWAS?


a Data collection
b Genotyping
c Quality control
African
Your data
Asian
American
European
Principal component 2
Principal component 1
d Imputation
SNP1 SNP2 SNP3 SNP4 SNP5 SNP6
Person 1 G T G A A T
Person 2 G T C C T C
Person 3 C A G C A C
Person 4 C A C C T C
e Association testing
100
80
60
40
20
0
−log10(P)
1 3 5 7 9 11 13 15 17 19 23
Chromosome
f Meta-analysis
Cohort A
Cohort B
Cohort C
g Replication
h Post-GWAS analyses

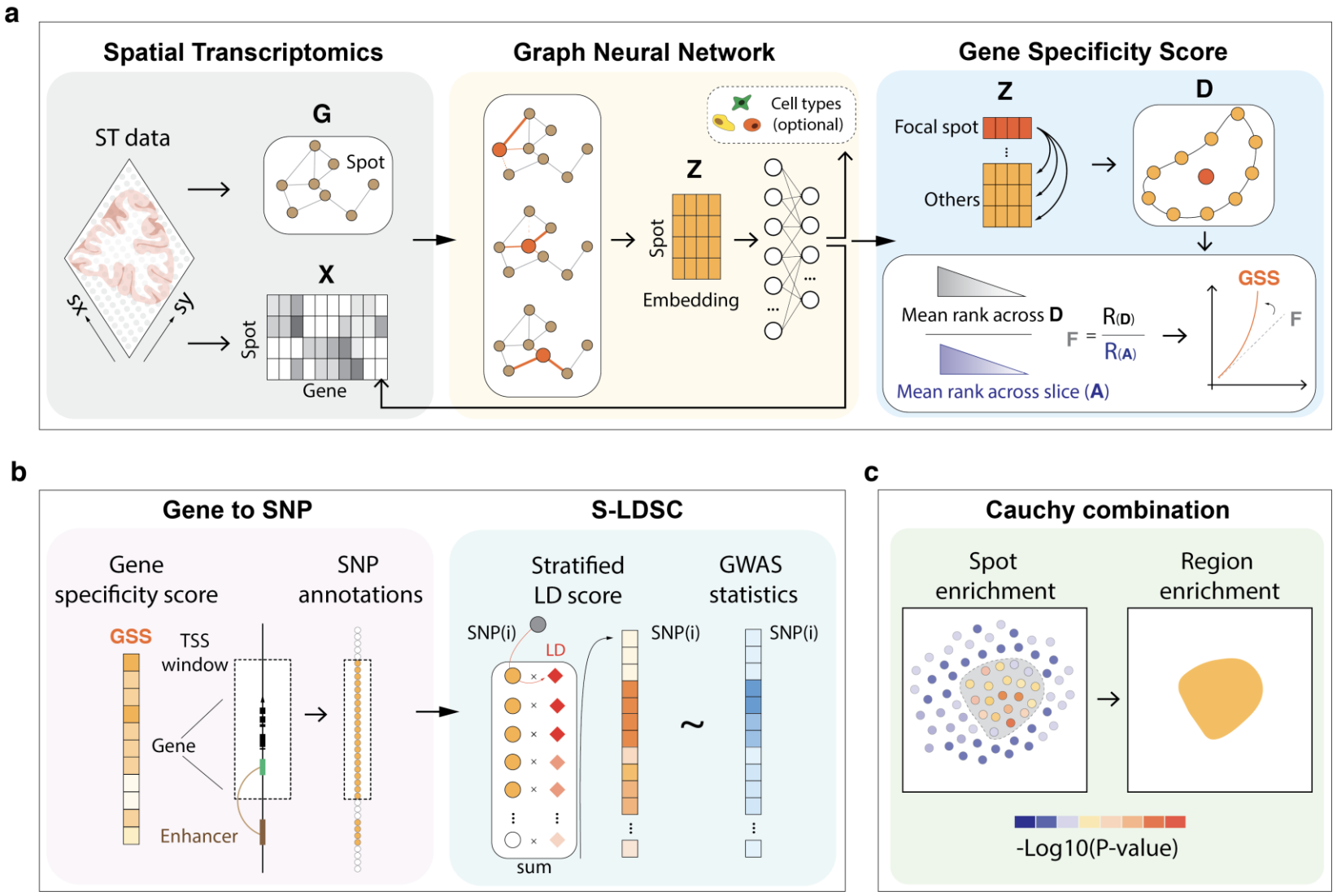
Summary Stats

SNP	rsID	CHR	BP	Allele1	Allele2	FRQ	BETA	SE	P	Direction	HetISq	HetChiSq	HetDf	HetPVal	TotalEffective_N
1:23308:G:C	1:23308:G:C	1	23308	C	G	0.0153	0.0479	0.1078	0.6569	??????????????????????????+???+?????	0	0.91	1	0.34	38896
1:24963:GT:G	1:24963:GT:G	1	24963	G	GT	0.0174	0.0965	0.0999	0.3339	??????????????????????????+???+?????	0	0.198	1	0.6563	38896
1:54490:G:A	rs141149254	1	54490	A	G	0.1694	-0.0423	0.023	0.0656	??????????????????-????????-?+?+?????	19.6	3.73	3	0.2922	75595
1:60351:A:G	rs62637817	1	60351	A	G	0.9194	-0.0088	0.0418	0.8329	??????????????????-???????????+?+?????	36.2	3.136	2	0.2085	38871
1:64931:G:A	rs62639104	1	64931	A	G	0.0792	0.0171	0.0423	0.6863	??????????????????+????????????-?-?????	36.6	3.156	2	0.2064	38871
1:98667:T:C	rs377391031	1	98667	T	C	0.9278	-0.0389	0.0509	0.4448	??????????????????????????-???+?????	54	2.175	1	0.1403	38896
1:106691:G:A	rs879731066	1	106691	A	G	0.0246	-0.0451	0.0844	0.5928	??????????????????????????-???-?????	59.4	2.461	1	0.1167	38896
1:534321:G:A	rs536226227	1	534321	A	G	0.0103	0.0794	0.1592	0.6178	??????????????????????????+?+???????	0	0.205	1	0.6508	48175
1:569933:G:A	rs368347679	1	569933	A	G	0.0952	-0.0214	0.038	0.5726	??????????????????-???????????-?+?????	0	0.269	2	0.8741	38871

SNP — Genetic variant representing a single nucleotide change in DNA.
rsID — Unique reference ID for a SNP from dbSNP.
CHR — Chromosome where the SNP is located.
BP — Base pair position of the SNP on the chromosome.
Allele1 — Effect allele used to calculate association statistics.
Allele2 — Reference/non-effect allele used for comparison.
FRQ — Frequency of the effect allele in the study population.
BETA — Estimated effect size of the effect allele on the trait.

SE — Standard error indicating precision of the effect estimate.
P — P-value measuring statistical significance of association.
TotalEffective_N — Effective sample size used in the GWAS analysis.

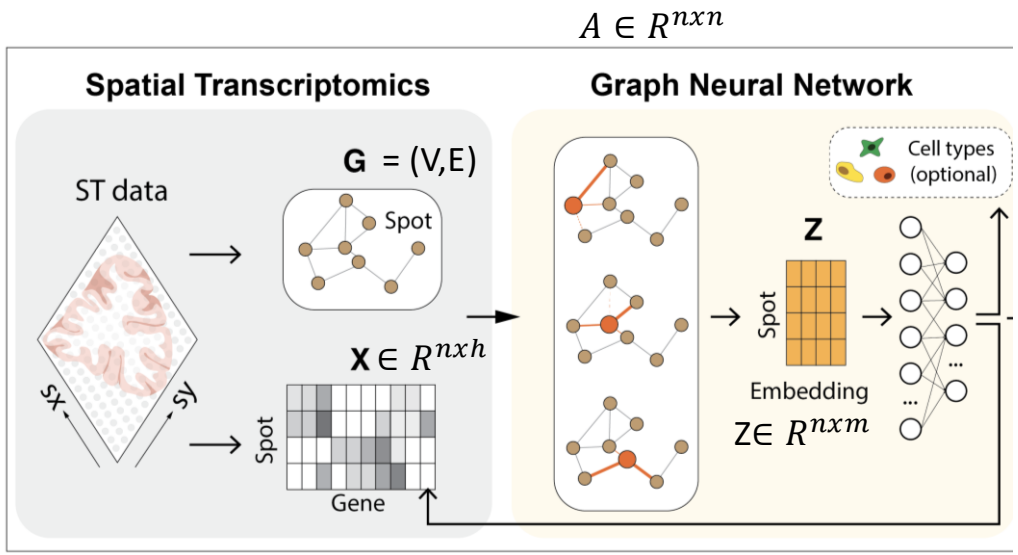
Genetically informed spatial mapping of cells for complex traits: gsMap



Input Data

- GWAS summary stats
- ST data (h5ad with raw counts and cell-type or tissue-region annotations)
- LD reference data (provided)

Identifying Homogenous spots



$$L = \gamma\left(\sum_{i=1}^{n}\sum_{g=1}^{h}(x_{ig} - \hat{x}_{ig})^2\right) + (1-\gamma)\left(\sum_{i=1}^{n}\sum_{k=1}^{c} -p_{ik}\log(q_{ik})\right)$$

$$\cos(\theta_{ij}) = \frac{\mathbf{Z}_i\mathbf{Z}_j}{\|\mathbf{Z}_i\|\|\mathbf{Z}_j\|}$$

where $\mathbf{Z}_i \in \mathbb{R}^{mx1}$ is the embedding vector of spot i

x_{ig} : normalized expression value of spot i for gene g

x^{ig}: reconstructed value

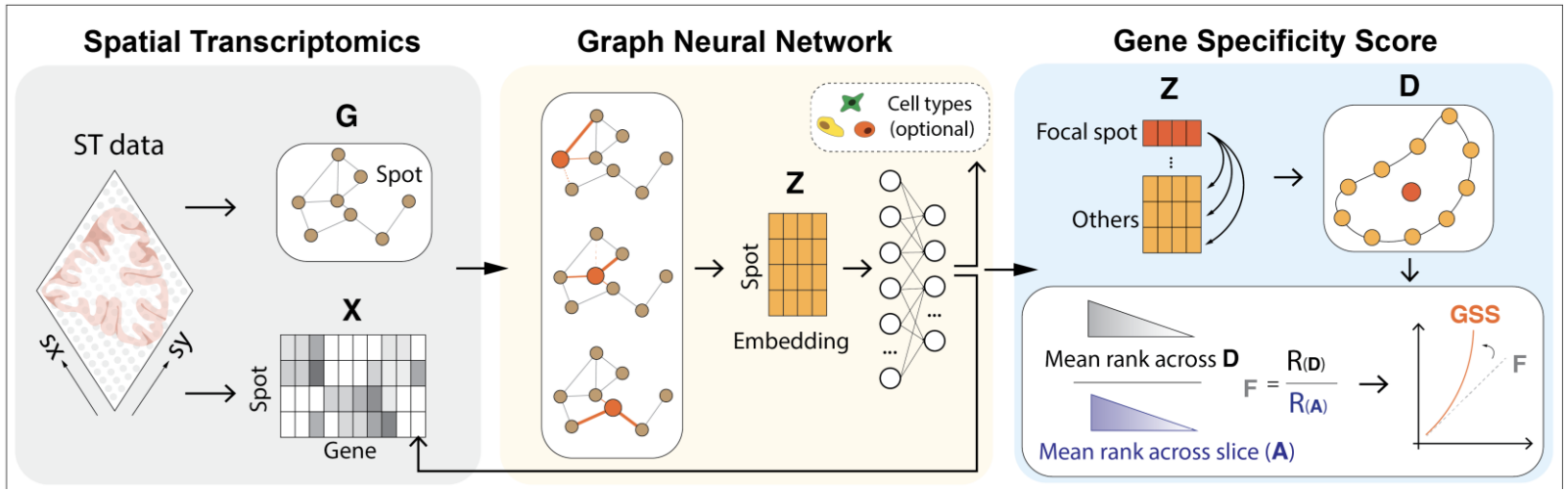
pik : probability of spot i belonging to cell type k

qik : predicted probability

c : number of cell types,

γ :hyperparameter that balances the reconstruction loss and cross-entropy loss.

Gene Specificity Score (GSS)

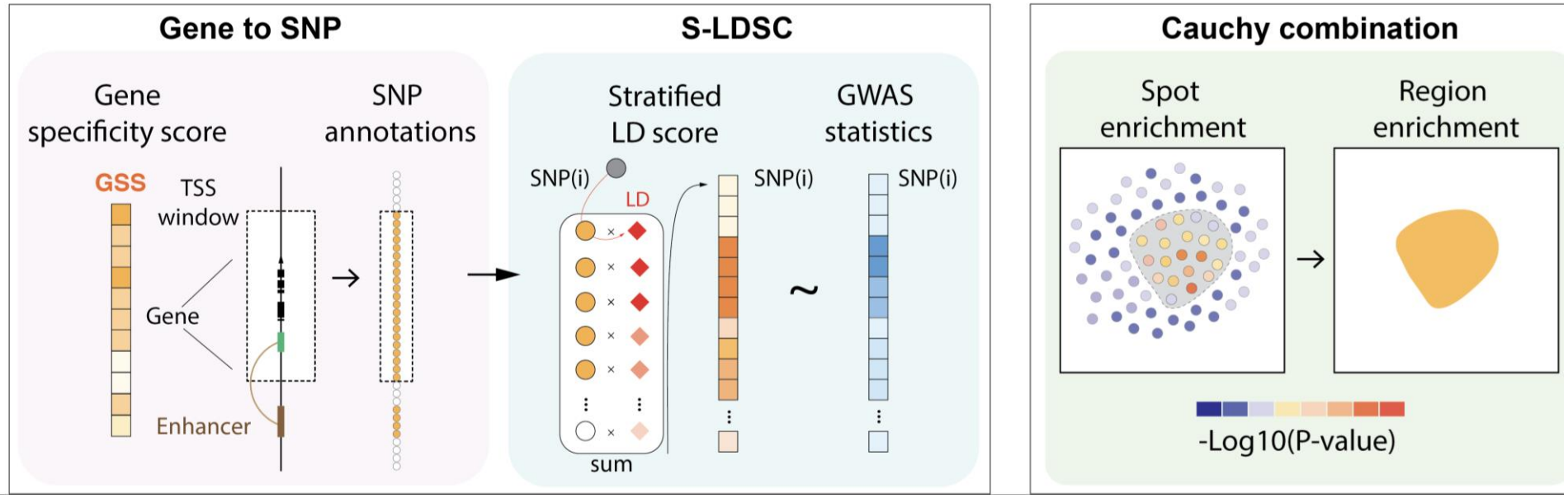


$$F_{ig} = \frac{\sqrt[d+1]{R_{ig}\left(\prod_{k \in D_i} R_{kg}\right)}}{\sqrt[n]{\prod_{j=1}^{n} R_{jg}}}$$

- Rank genes by expression within each spatial spot.
- Compare local rank (micro-domain) against global rank (all spots).
- Higher scores indicate spatially enriched domain-specific genes.
- Noise and unreliable signals are filtered out.
- Gaussian transformation enhances strong spatial signals.

$$S_{ig} = \exp\left\{F_{ig}^2\right\} - 1$$

Mapping gene specificity score to SNPs

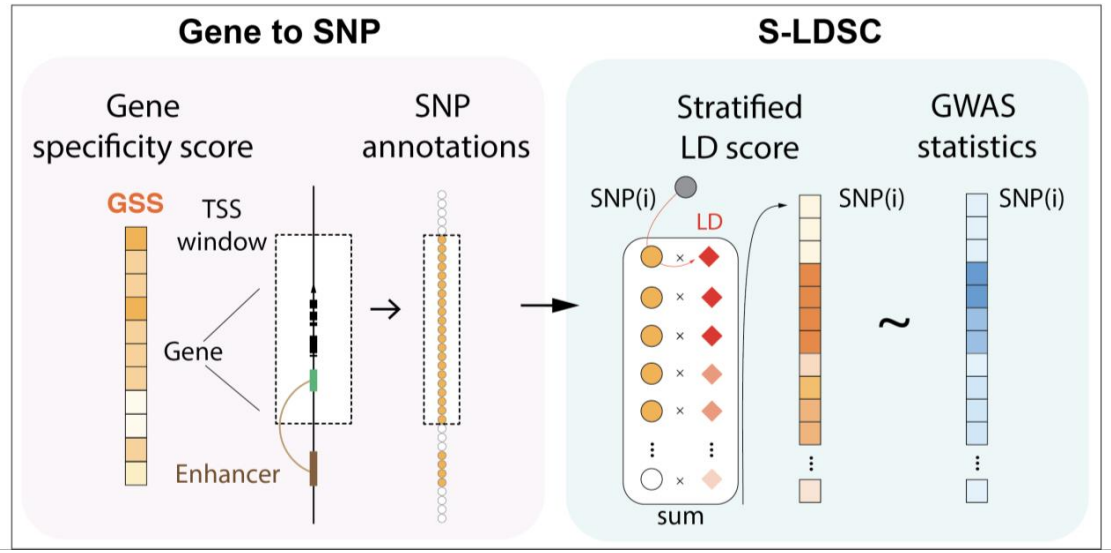


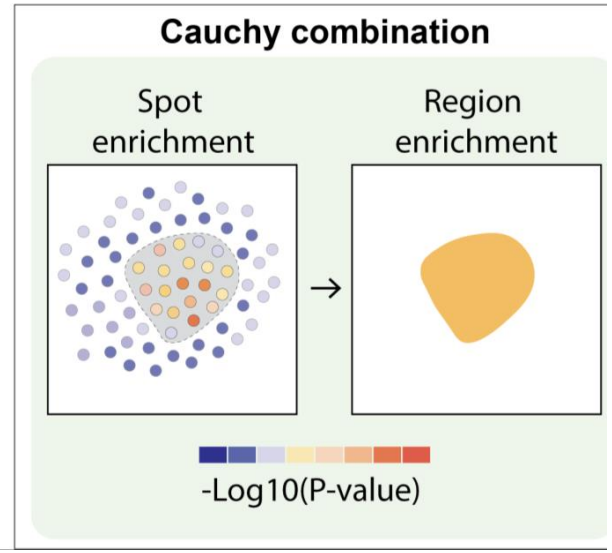
This is typically done within a **100 kb window** around a gene's transcribed region.

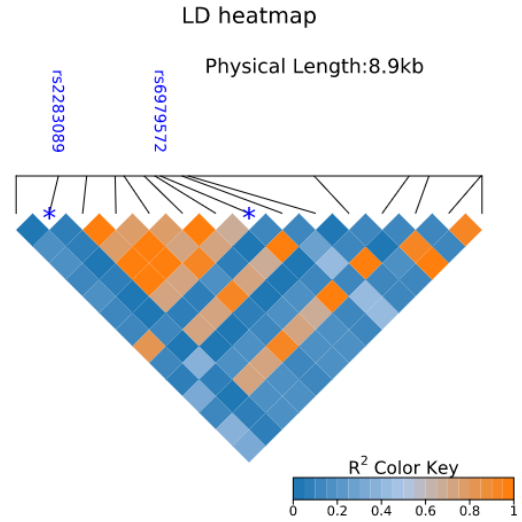
GSS of a gene is assigned to the co-localizing SNPs

These annotations are then used to compute stratified linkage disequilibrium (LD) scores.

Linking genomic annotations with GWAS data







$$E(\chi_j^2) = N\left(l(j,s)\beta_s + \sum_c l(j,c)\beta_c\right) + 1$$

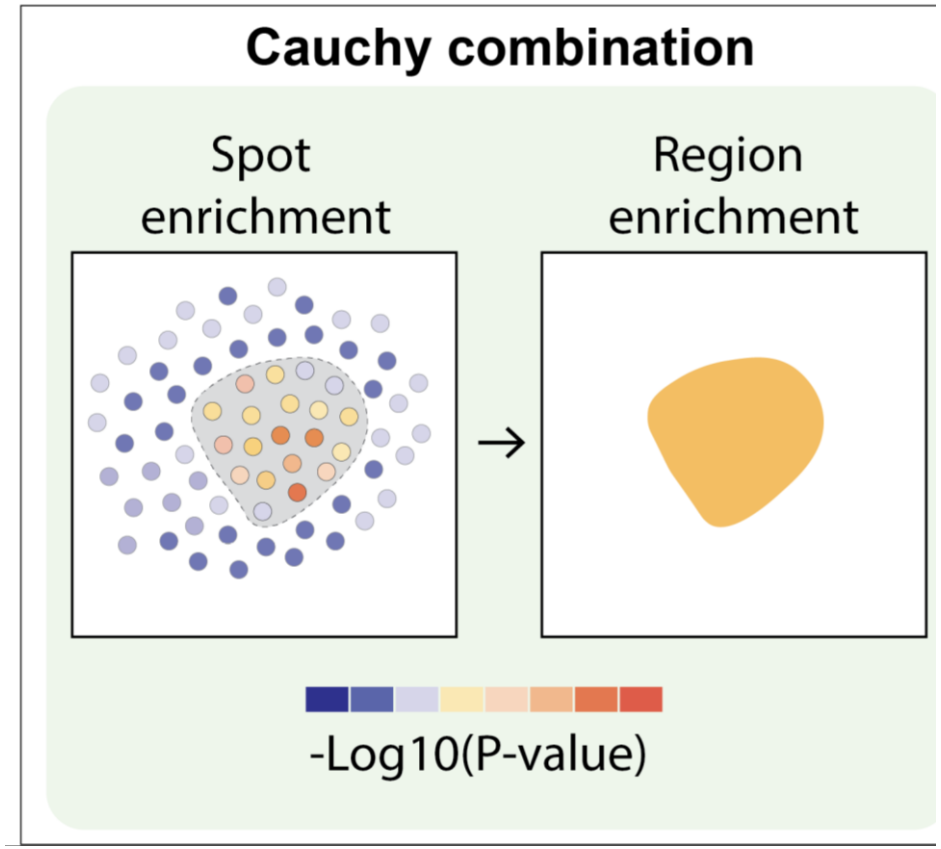
χ2 denotes the association statistic of SNP j with the trait

N – GWAS sample size

$l(j,s) = \sum_k a_{sk} r_{jk}^2$ *is the stratified LD score of SNP j with respect to SNP annotation as. (i.e., GSS of spot s) with βs being the coefficient and r2jk being the squared LD correlation between SNPs j and k*

$l(j,c) = \sum_k a_{ck} r_{jk}^2$ *is the stratified LD score of SNP j with respect to SNP baseline annotation ac. βc : the coefficient*

Estimating the strength of enrichment for a spatial region



$$T_{Cauchy} = \sum_{i=1}^{\phi} \tan\{(0.5 - P_i)\pi\}$$

where P' represents the P value of spot i belonging to the spatial region.

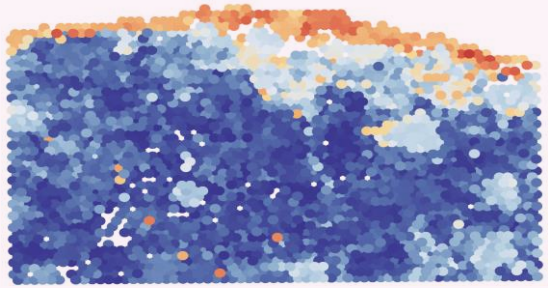
The aggregated P value for the spatial region is approximated as:

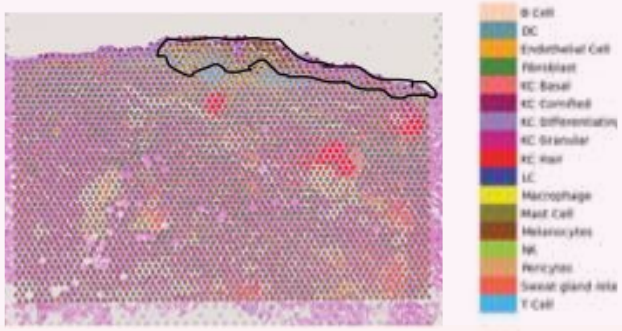
$$P_{region} \approx \frac{1}{2} - \pi\{\arctan\,(T_{Cauchy})\}$$

To aggregate P-values of individual spots within the spatial region

Results for sample: mel48974

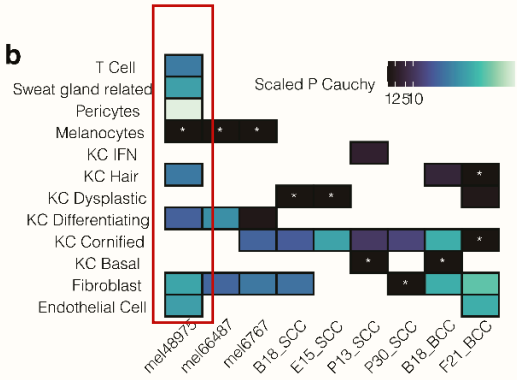
GSS



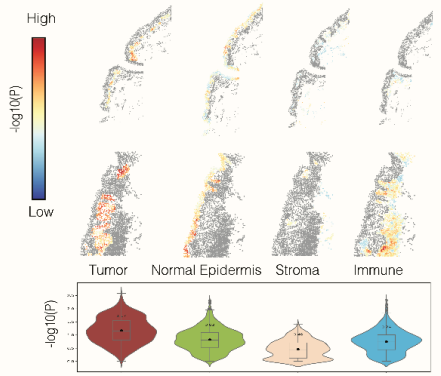


Cauchy combination

Annotation	P Cauchy	P Median
Melanocytes	1.5713e-06	2.4243e-06
KC Differentiating	1.2629e-04	1.7172e-01
KC Hair	3.4541e-04	1.0288e-03
T Cell	3.8224e-04	3.9801e-04
Endothelial Cell	1.6519e-03	2.3536e-03
Sweat gland related	1.9025e-03	3.1033e-03
Fibroblast	2.2130e-03	1.8163e-01
Pericytes	1.4624e-01	2.2540e-01



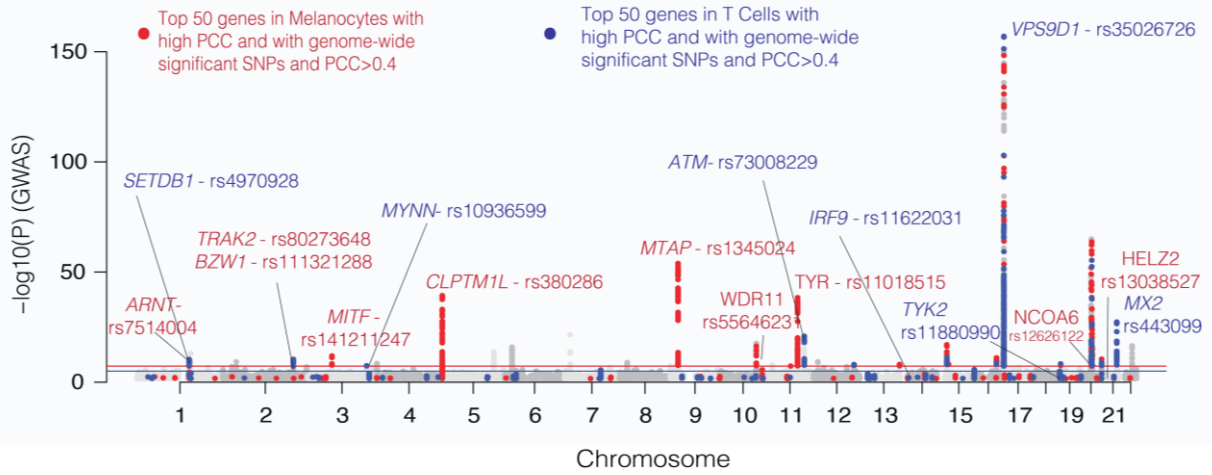
Visium



mel48974 CosMx

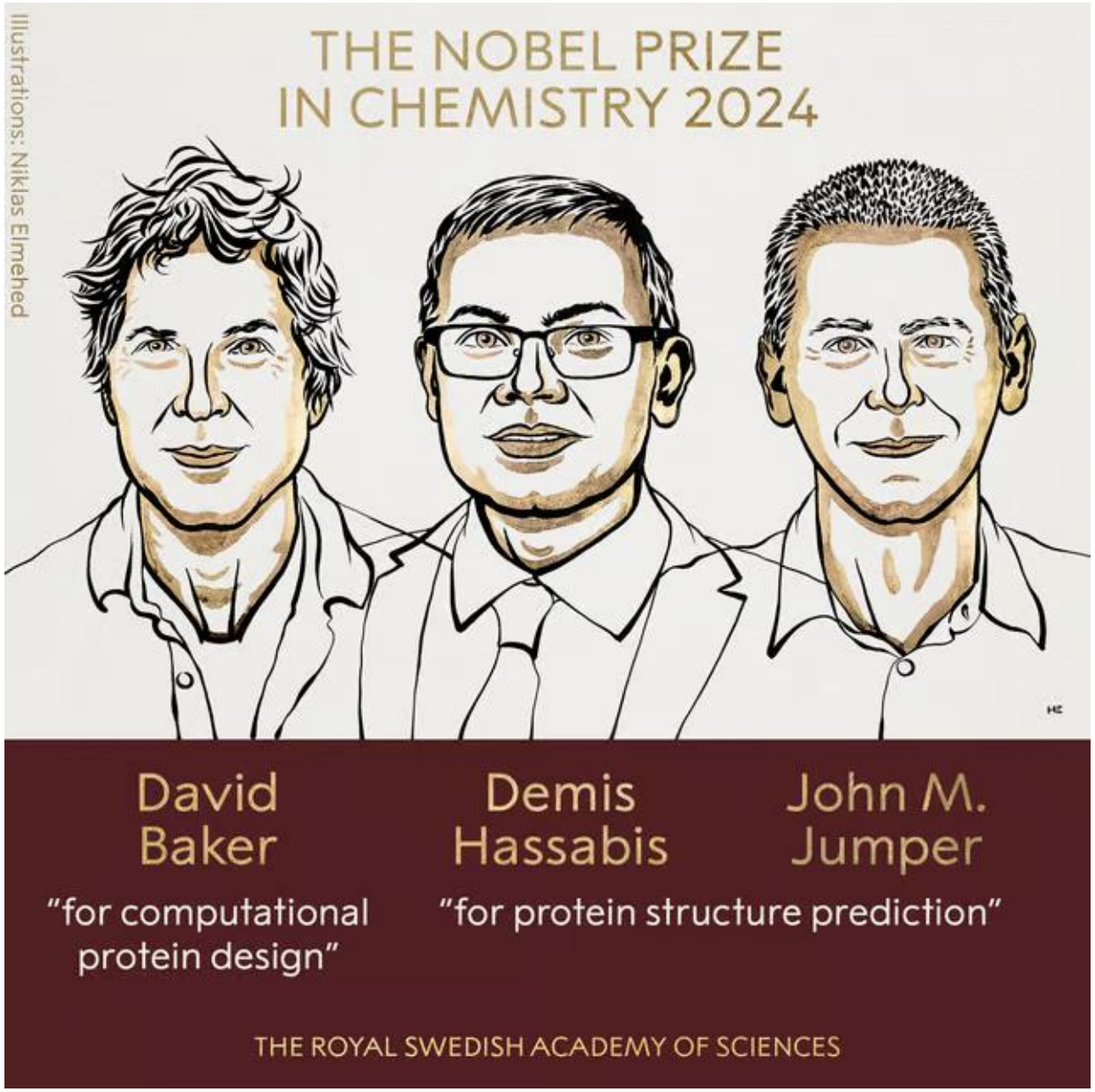
Top Genes-SNPs

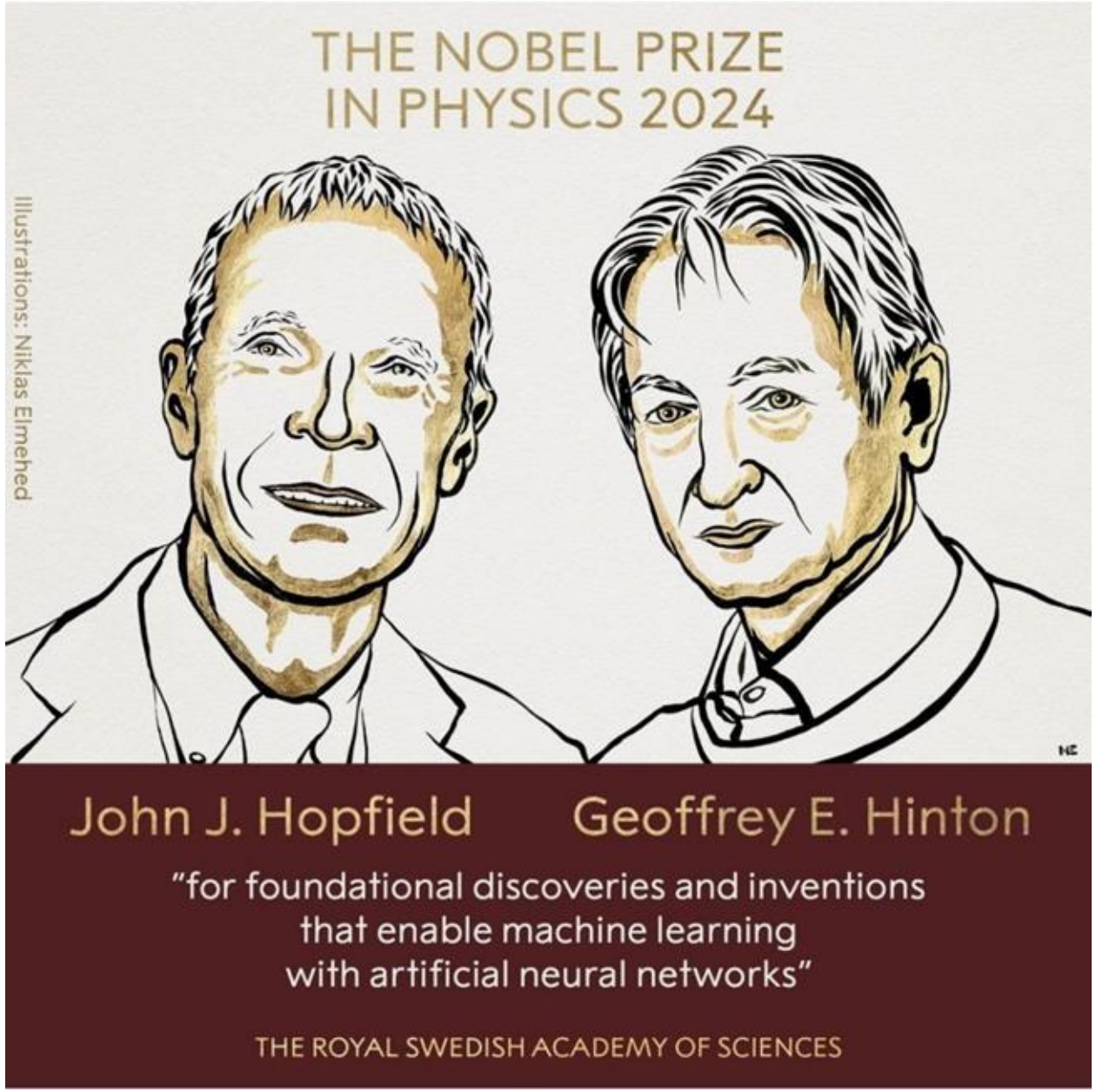
Gene	Annotation
GTF2F2	Melanocytes
PSME2	T Cell
WDFY1	Melanocytes
PBRM1	Melanocytes
BRD4	Melanocytes
HGS	Melanocytes
WDR82	Melanocytes
DPP9	Melanocytes
CBX3	Melanocytes
PML	T Cell



Lecture 5: Machine Learning

The next revolution - Generative AI





Deep Learning

Deep learning for spatial transcriptomics data

1. Basic Concept

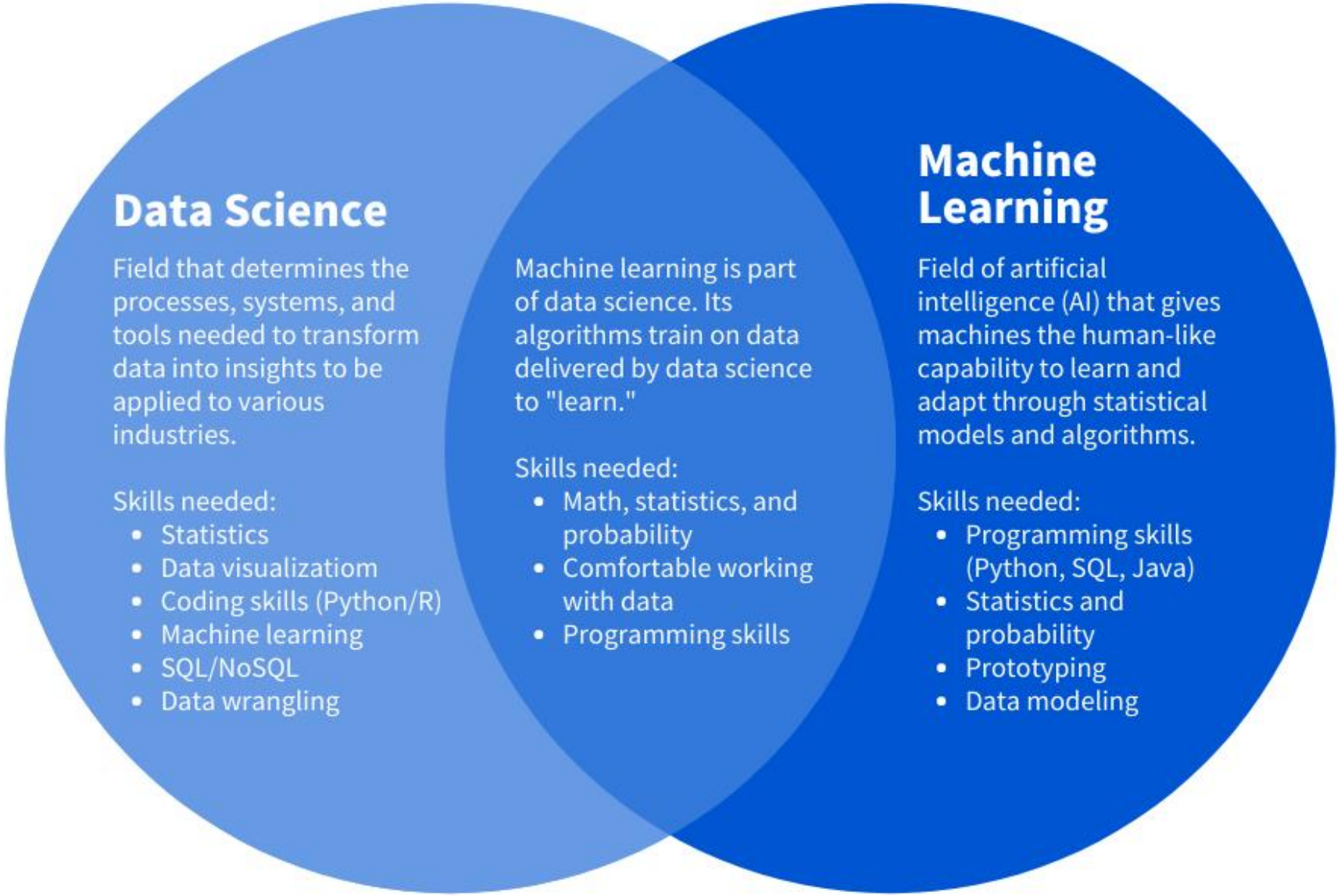
2. Neural Network model (NN)

- Key elements
- Model architectures

3. NN in spatial omics

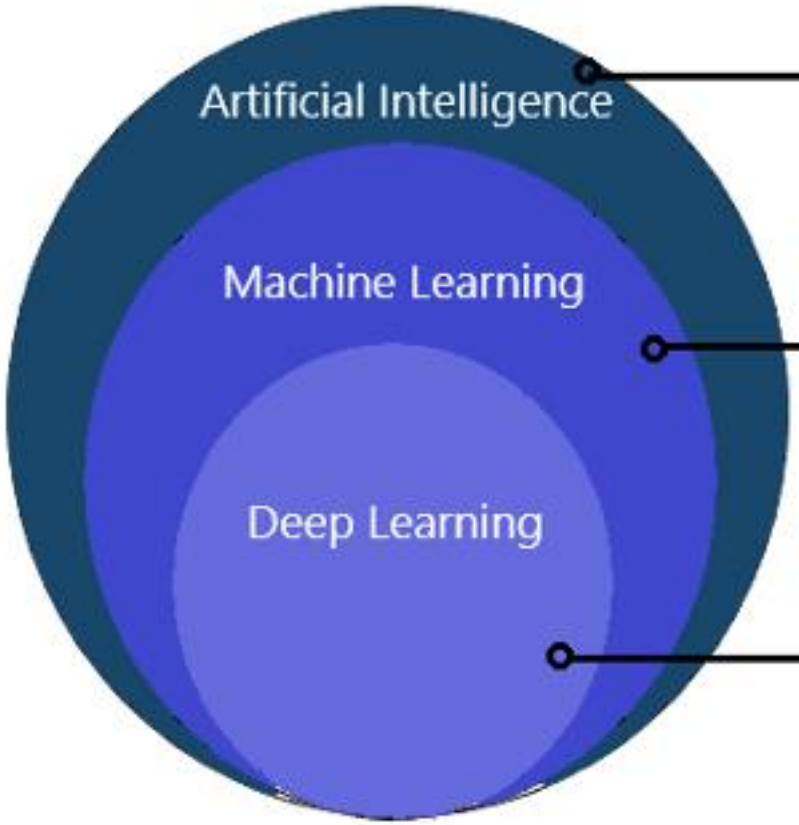
- Convolutional neural network (CNN)
- Autoencoder

Data science and machine learning



(Coursera, 2022)

Machine learning, statistical learning, deep learning



Artificial Intelligence

AI enables machines to think without any human intervention.

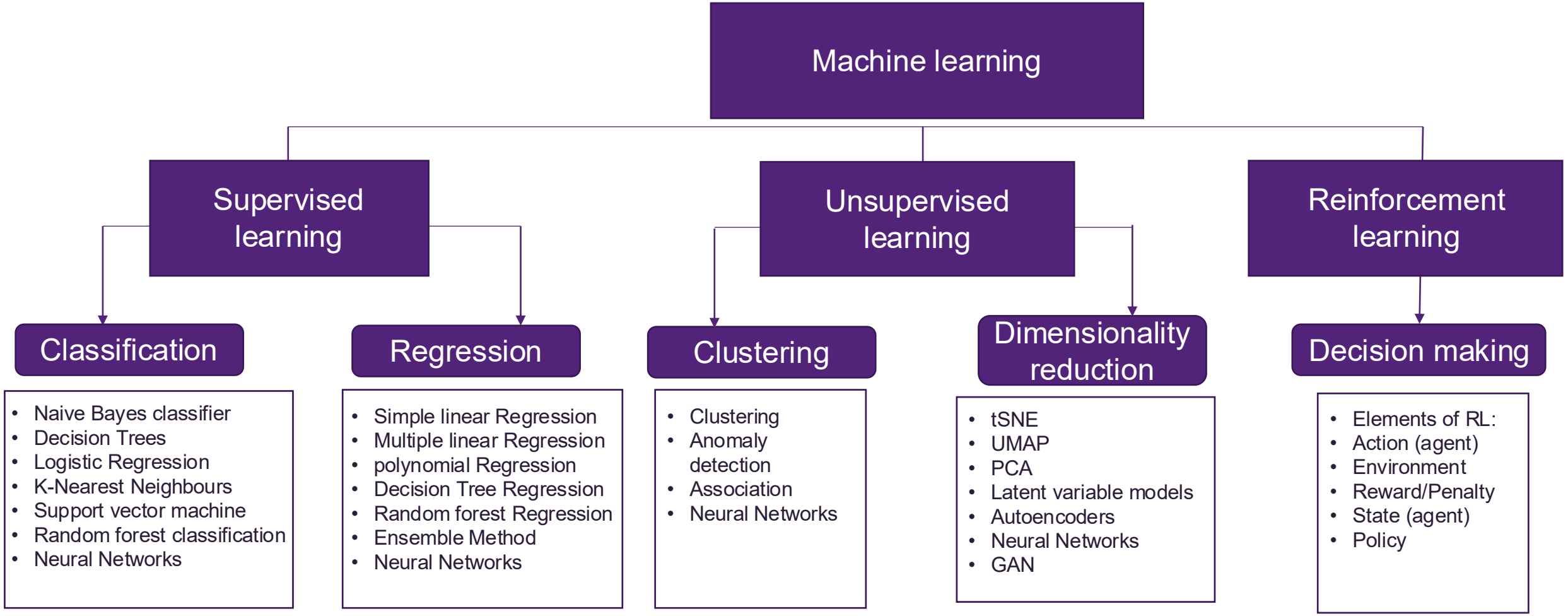
Machine Learning

Subset of AI that uses statistical learning algorithms that learn pattern in data over time

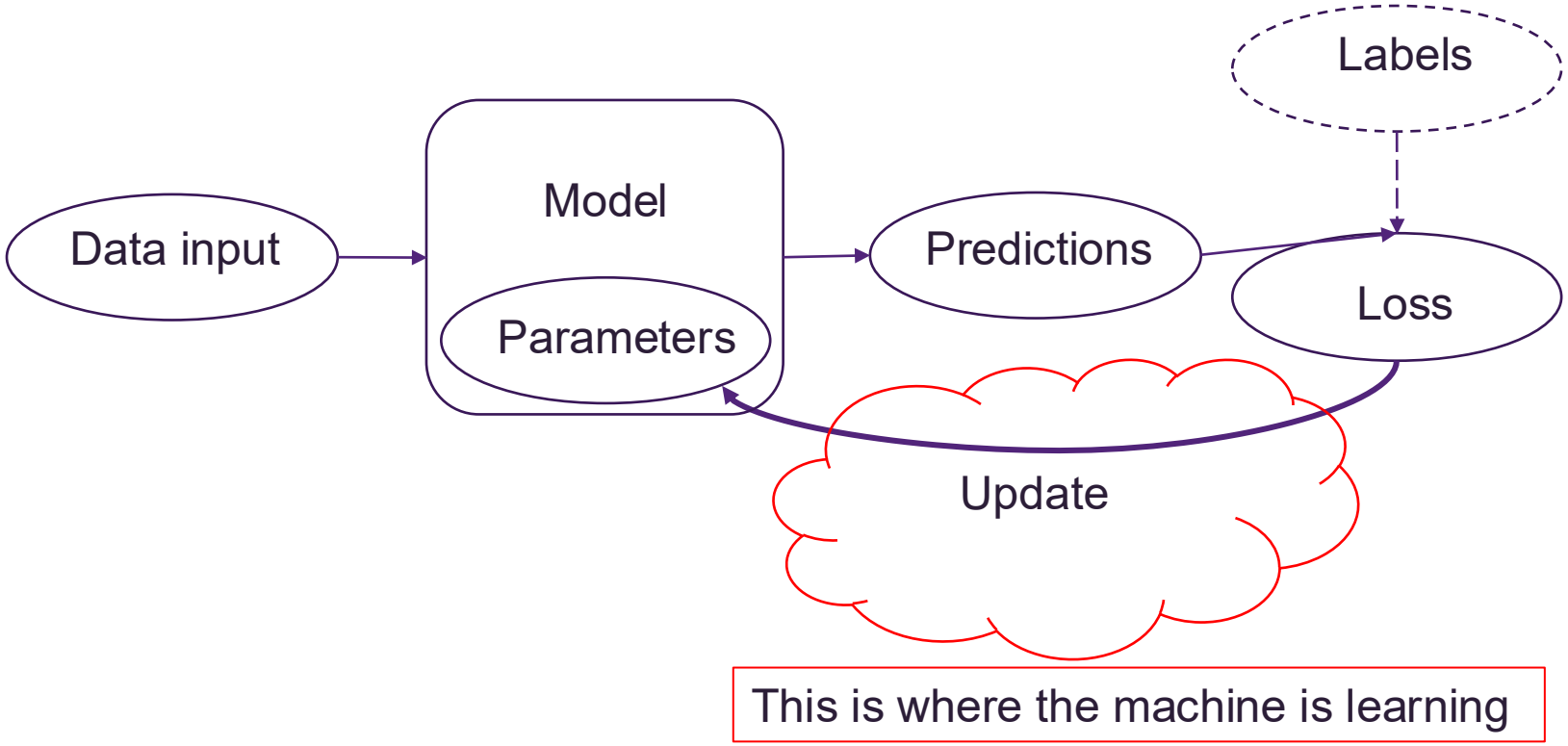
Deep Learning

Subset of ML that filters the data through multiple layers

Machine learning categories and tasks

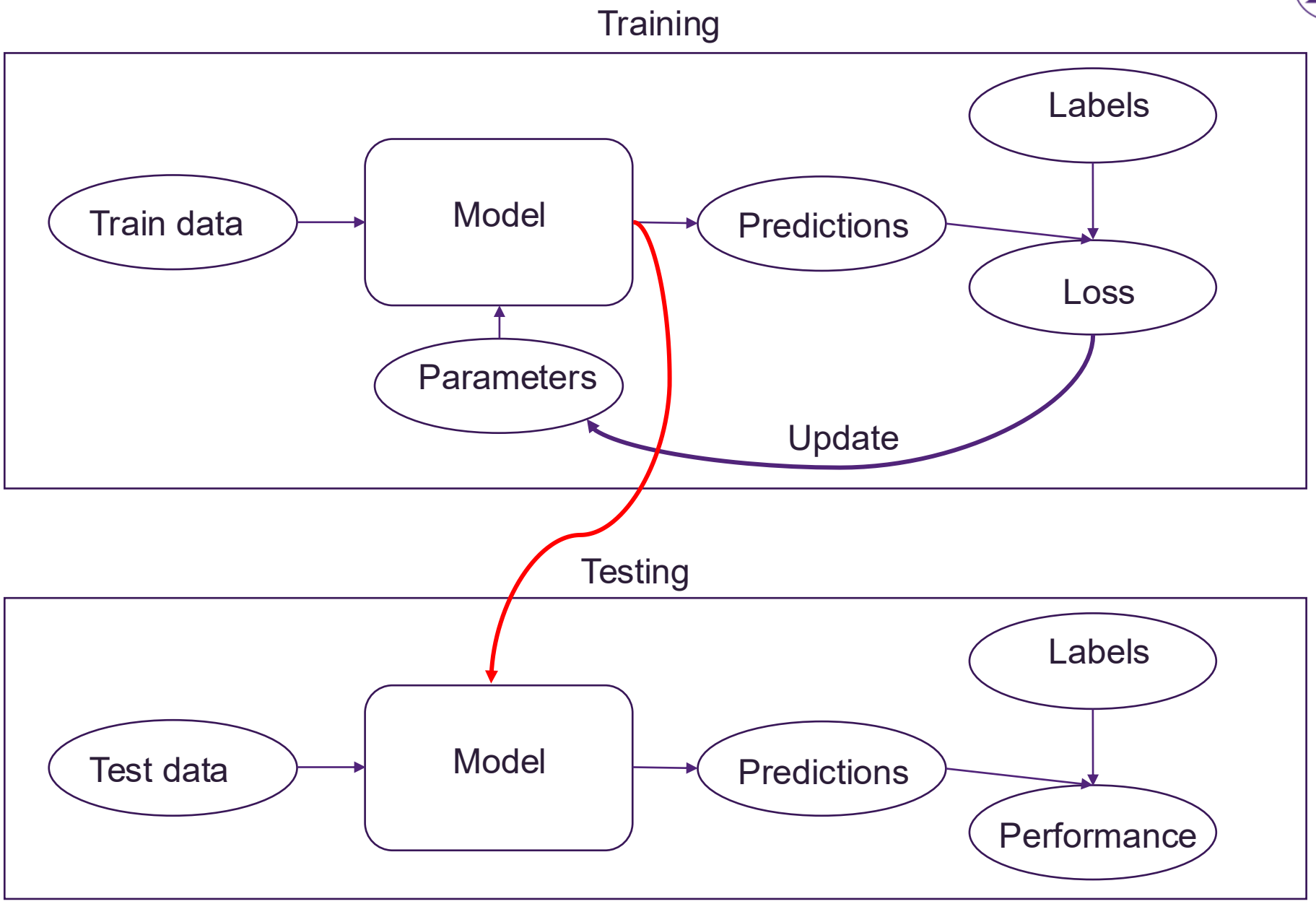


Machine learning – Loss function

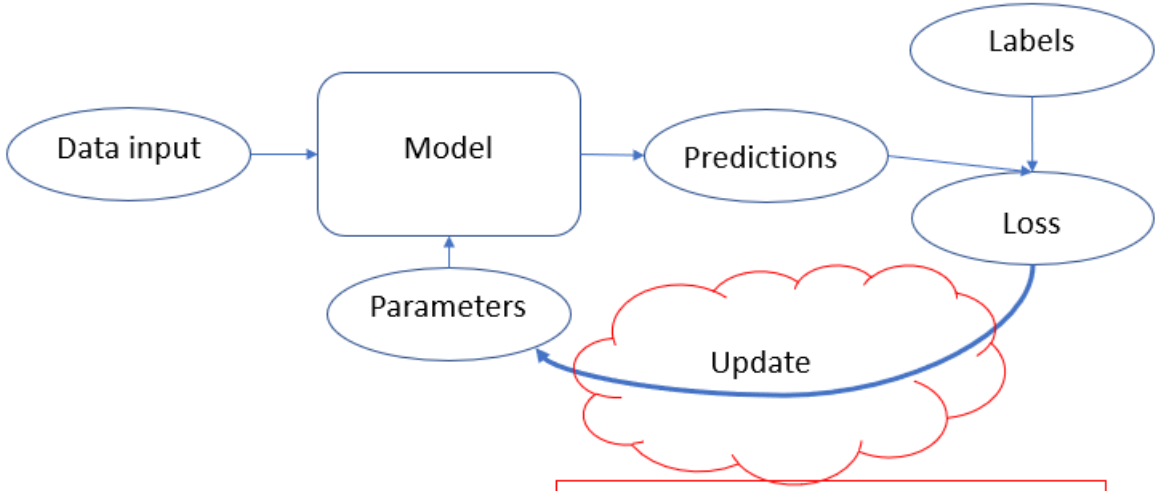


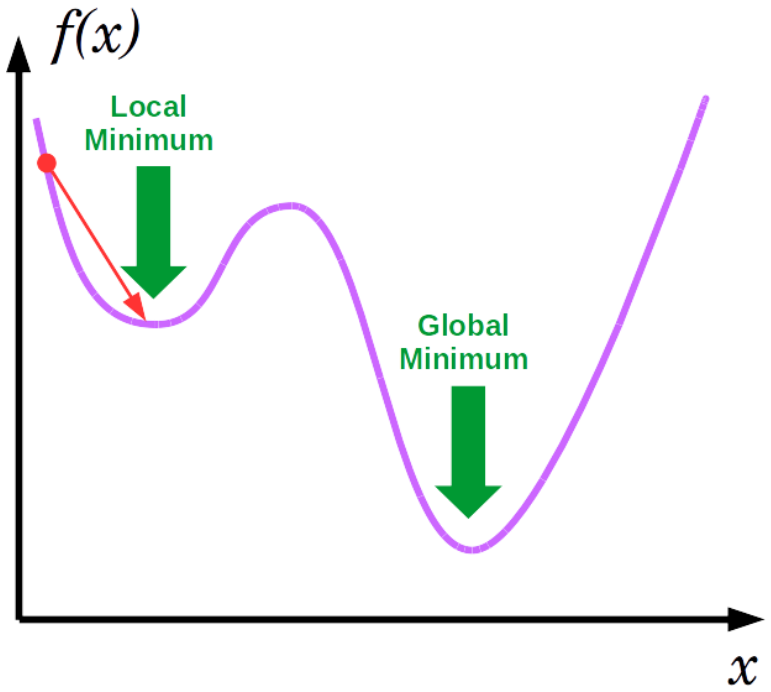
- ML: The training of programs developed by allowing a computer to learn from its experience (rather than through manually coding the individual steps)
- Loss function is where ML meets statistical models
- (hyper)Parameters are where machine learning deviate from statistical models

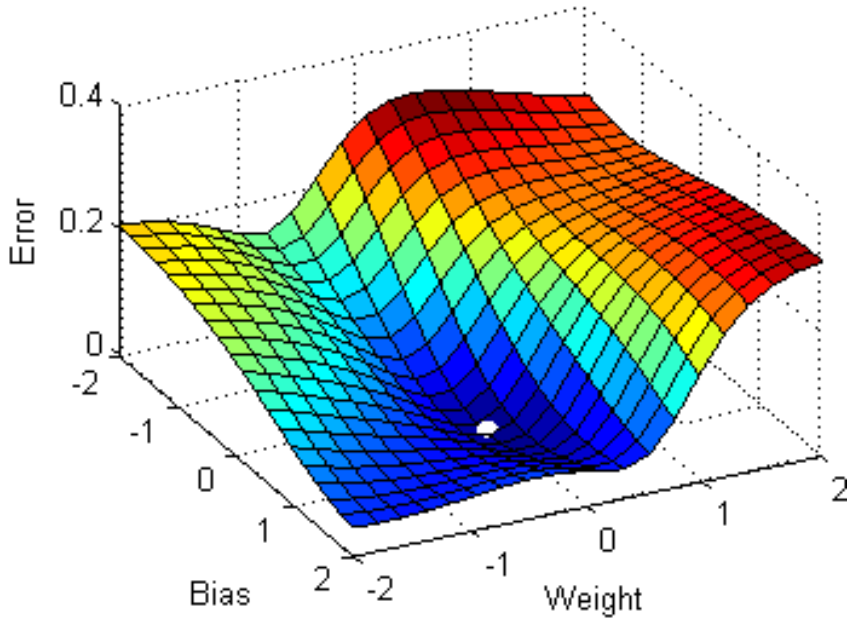
Machine learning – Training and testing datasets



Machine learning – Loss function



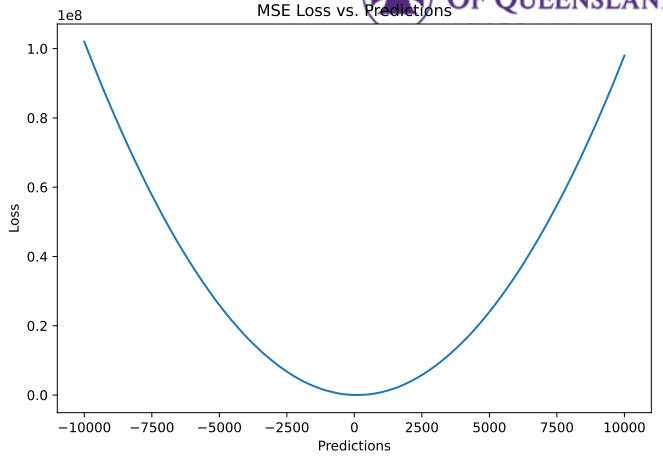




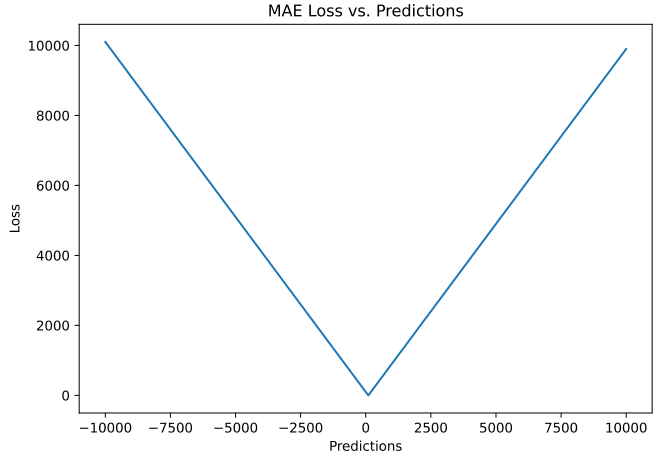
Different loss functions

Regression:

Mean Square Error/Quadratic Loss/L2 Loss: $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$



Mean Absolute Error/L1 Loss: $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$



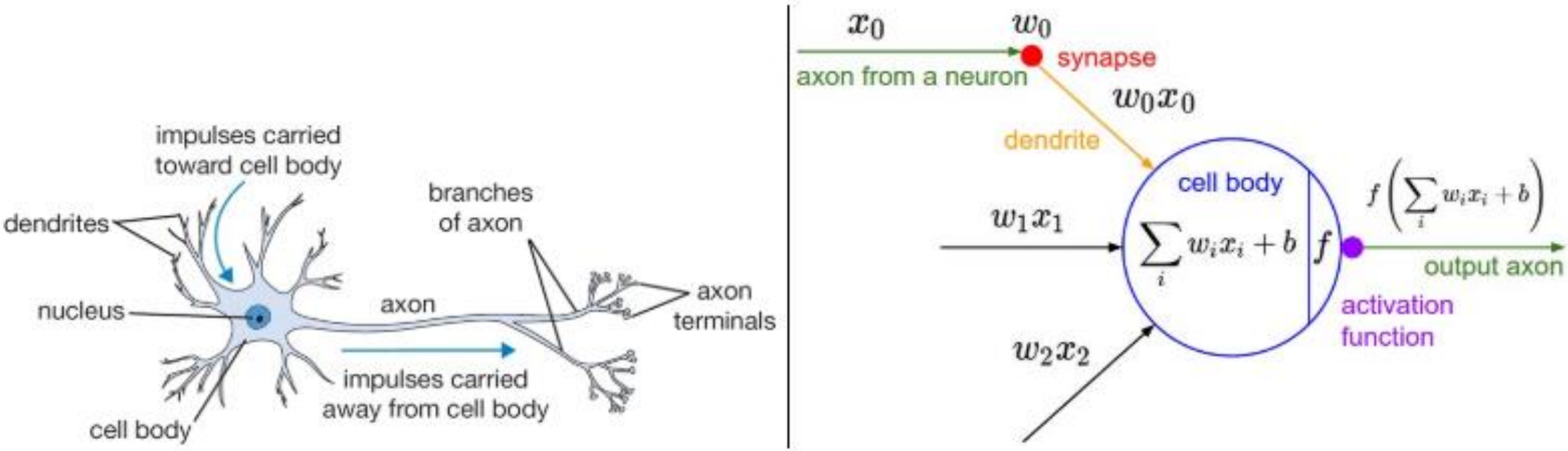
Mean Bias Error: $MBE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)$

Classification:

Cross Entropy Loss/Negative Log Likelihood: $-(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i))$

https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

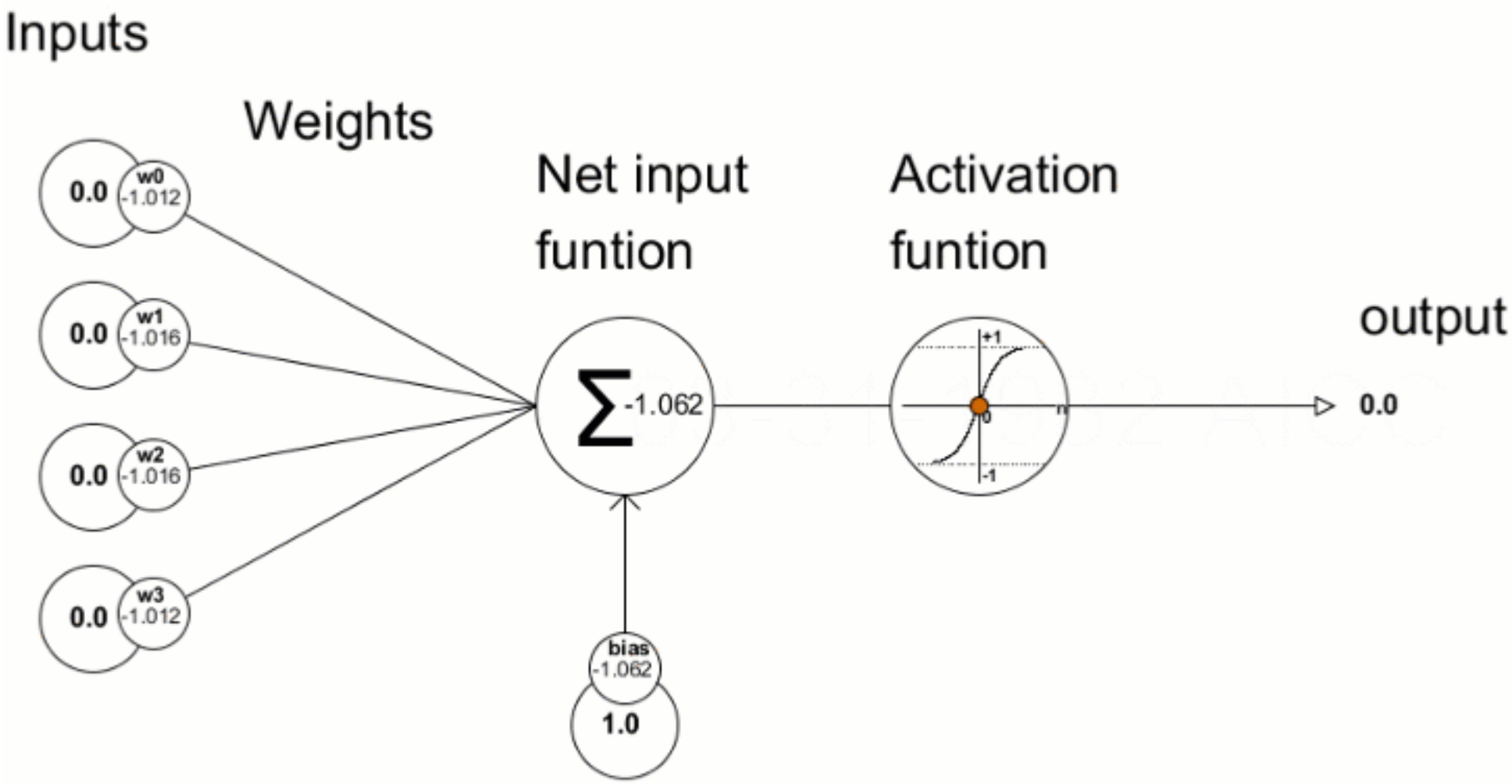
Deep learning – Neural network



(Source: cs231n, Stanford)

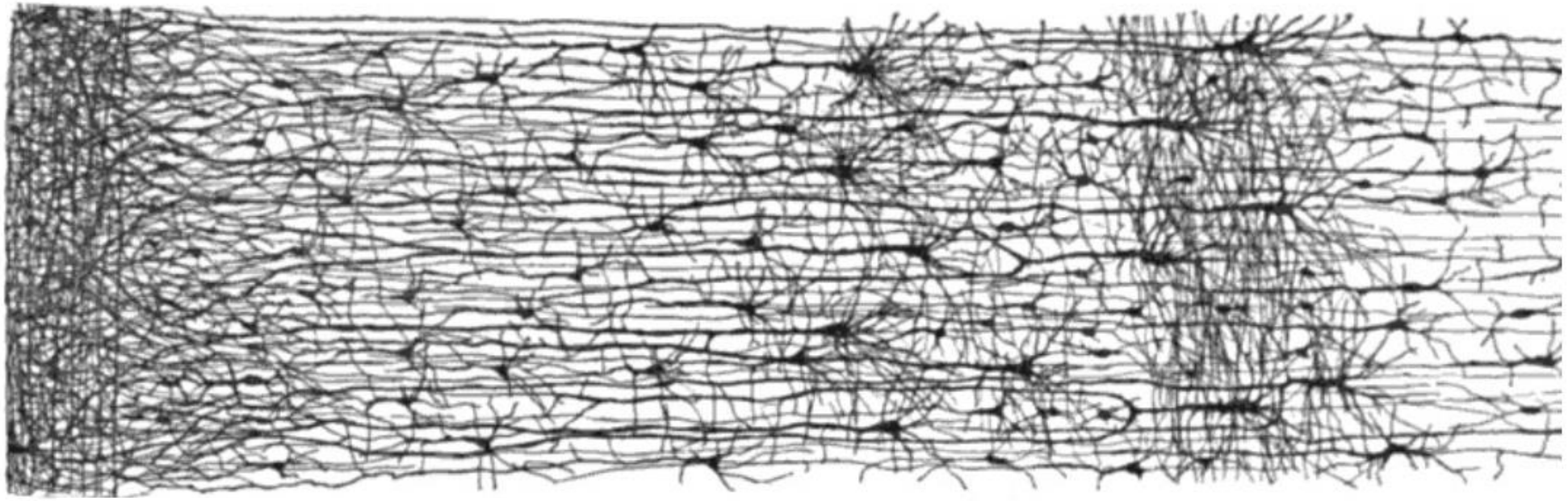
$$Y = \sum (weight * input) + bias$$

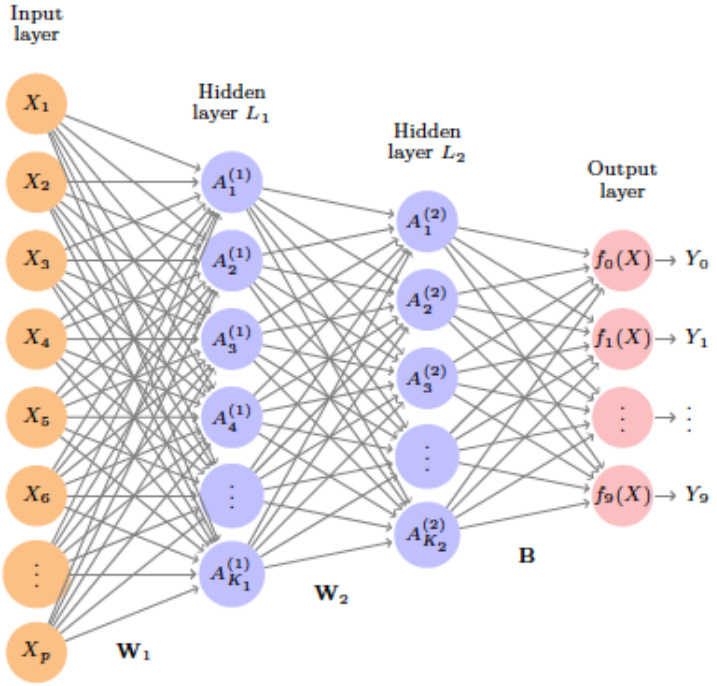
Single neuron in action – activation function

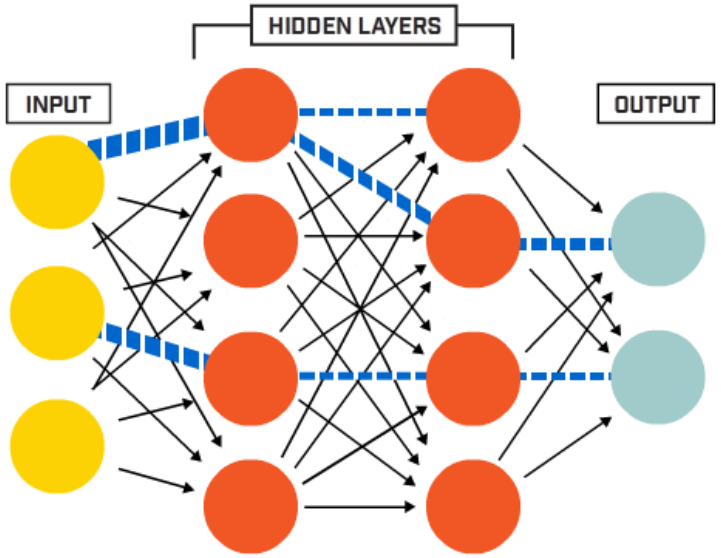


(Towards AI, 2019)

Neural network and multilayer perceptrons

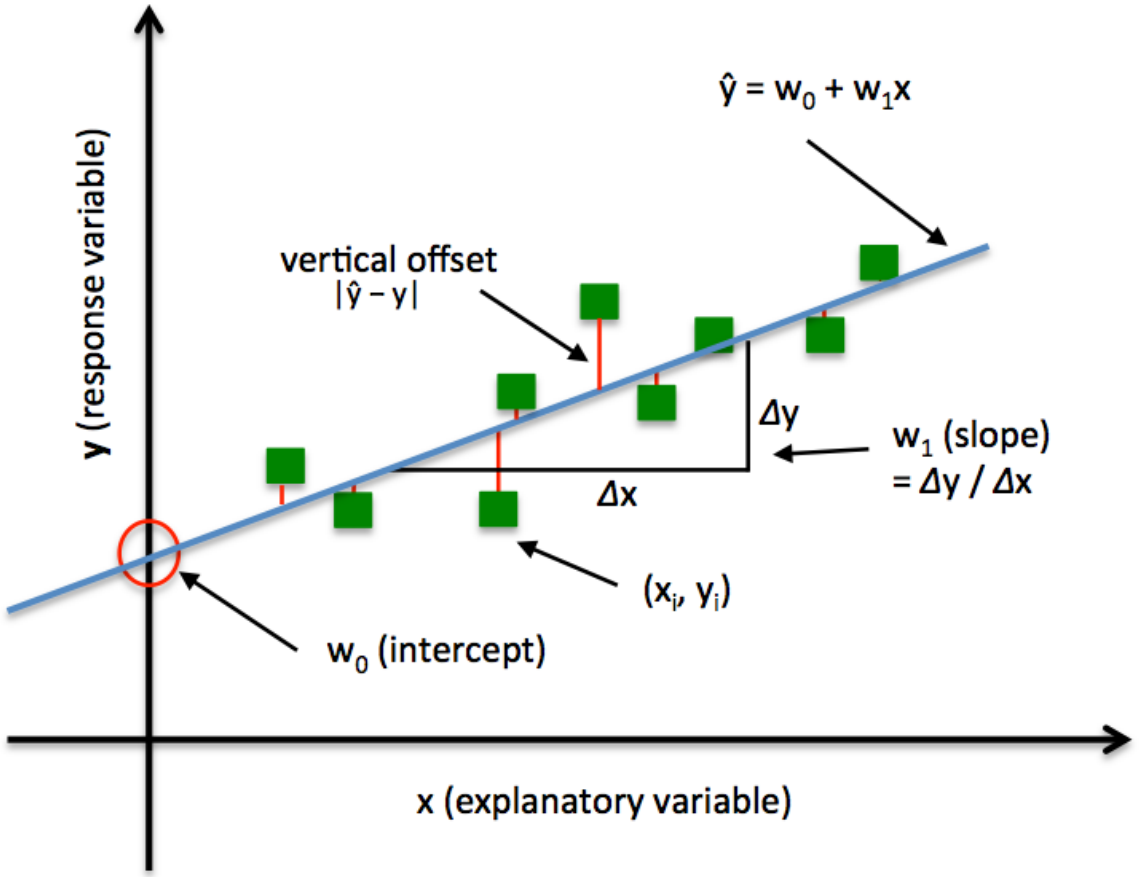






(Towards AI, 2019)

From linear regression to machine learning



$$Error = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

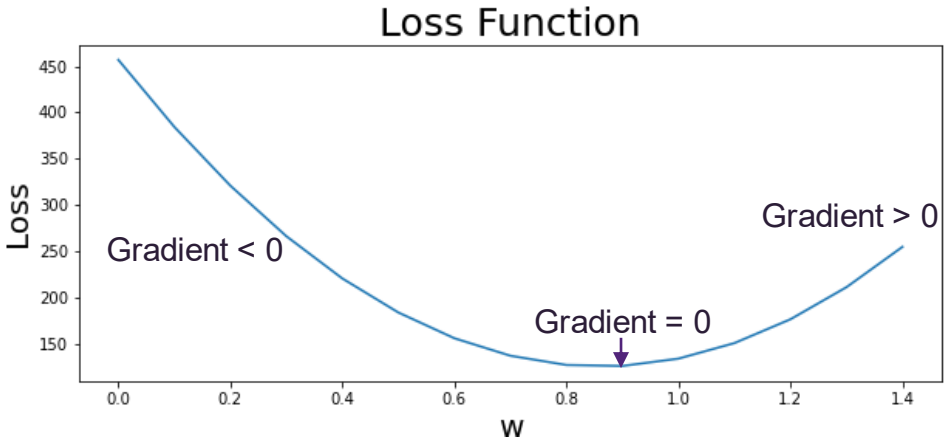
$$= \frac{1}{N}\sum_{i=1}^{N}(y_i - w_0 - w_1 X_i)^2$$

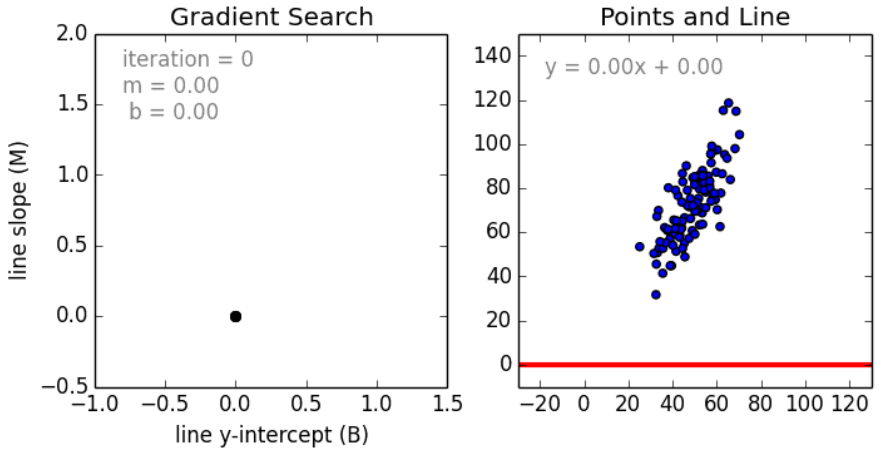
= Objective function
= Loss function
= J(w_0, w_1)

Minimize wrt w_0 and w_1 by gradient descent

http://rasbt.github.io/mlxtend/user_guide/regressor/LinearRegression/

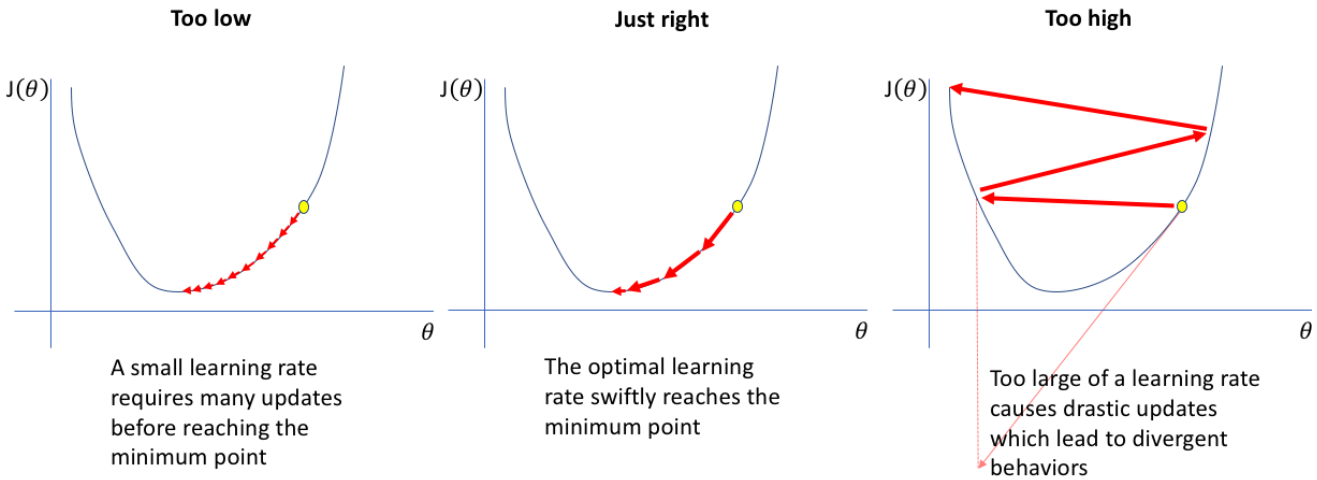
General terms: Gradient Descent Example for Linear Regression





$$w^t = w^{t-1} - \alpha \nabla J(w^{t-1})$$

α is the learning rate (step length)
Effect of learning rate →



https://www.jeremyjordan.me/nn-learning-rate/

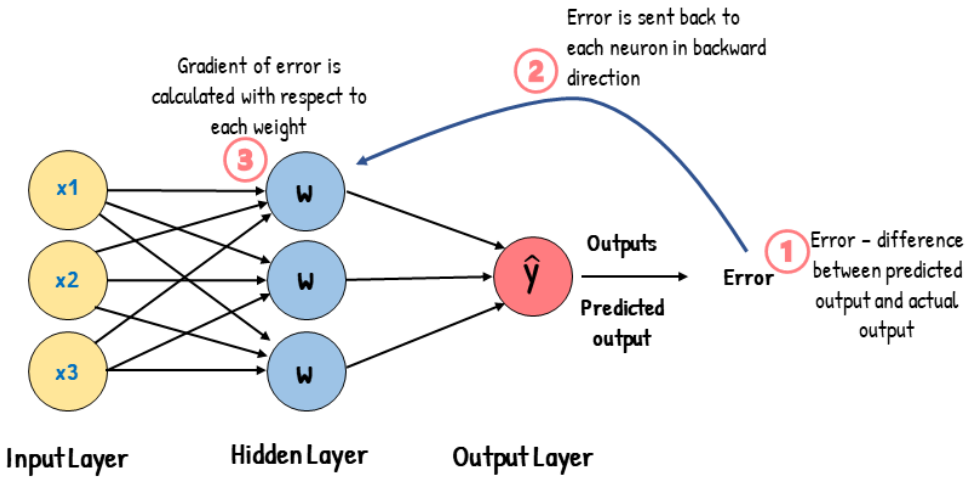
https://github.com/mattnedrich/GradientDescentExample

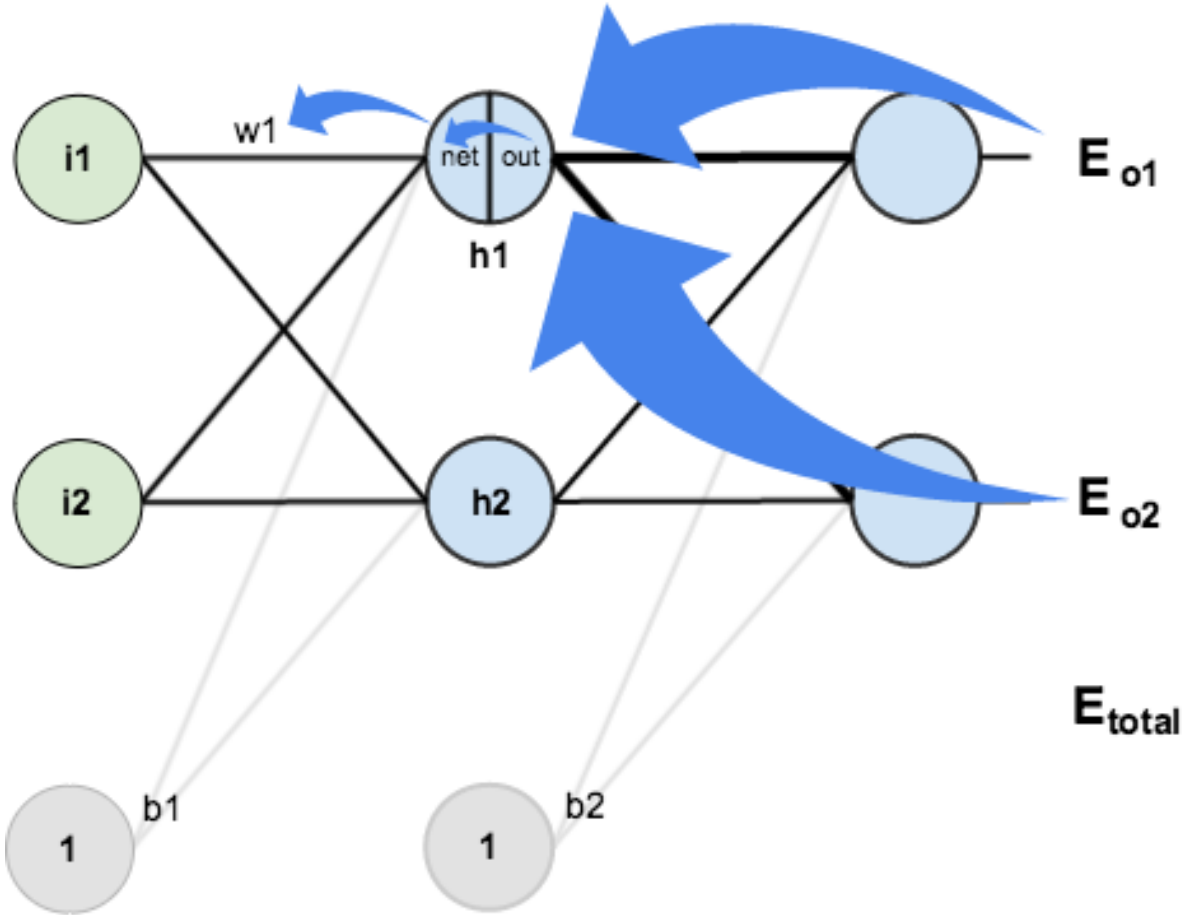
Multilayer perceptron – backward propagation

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

Chain rule

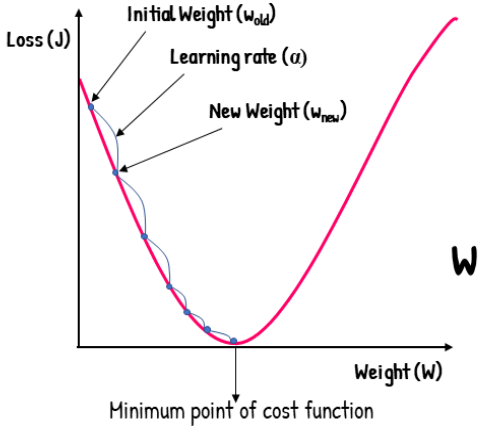




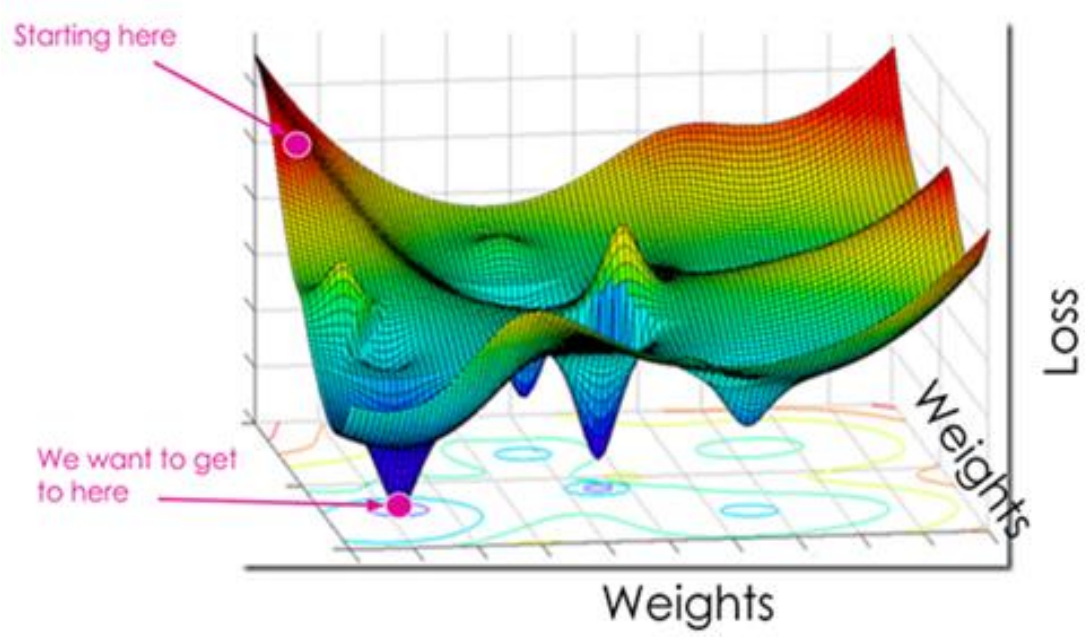
$$E_{total} = E_{o1} + E_{o2}$$

	Backpropagation	Gradient Descent
Definition	An algorithm for calculating the gradients of the cost function	Optimization algorithm used to find the weights that minimize the cost function
Requirements	Differentiation via the chain rule	•Gradient via Backpropagation •Learning rate
Process	Propagating the error backwards and calculating the gradient of the error function with respect to the weights	Descending down the cost function until the minimum point and find the corresponding weights

Gradient Descent



$$w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$$

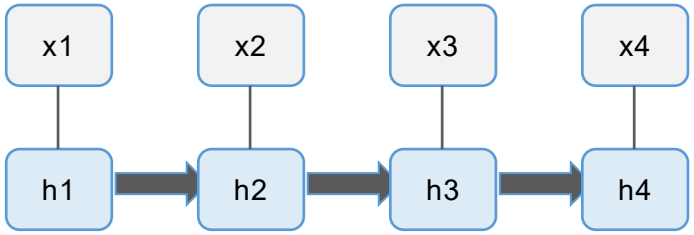


Transformers

A neural network architecture built around attention

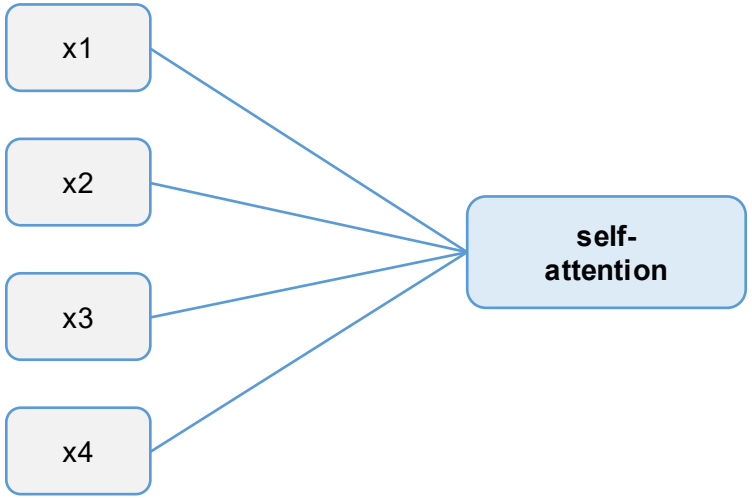
Why Transformers?

Before: sequence models process tokens step by step



Long paths make distant information hard to use

Transformer: every token can attend to every other token



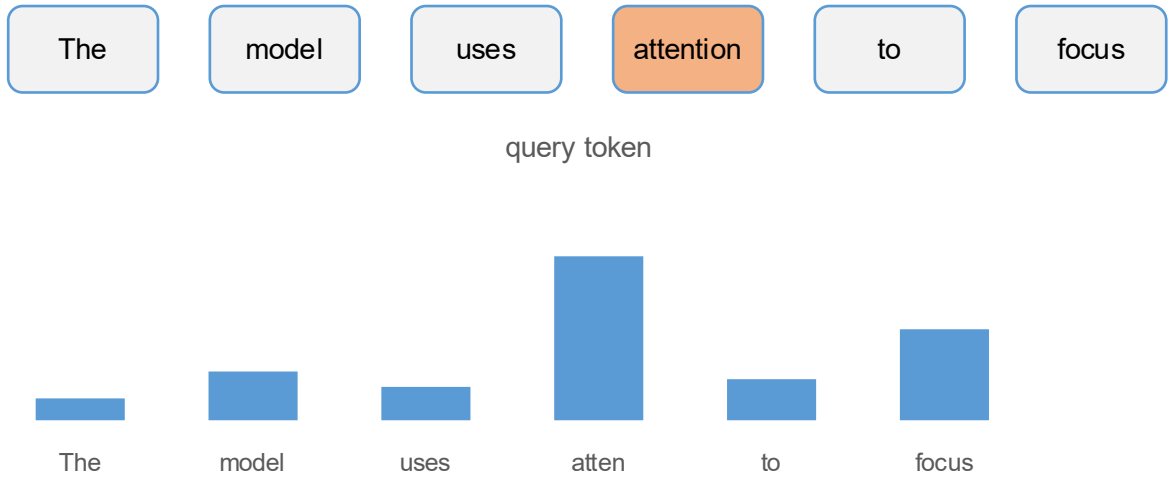
- Parallel training on GPUs
- Short path between any two tokens
- Scales to large foundation models

Take-home: attention is an information routing mechanism.

Ref: Vaswani et al. 2017, Attention Is All You Need

Core idea: attention

For each token, attention learns which other tokens should influence its representation.



attention weights from one token to all tokens

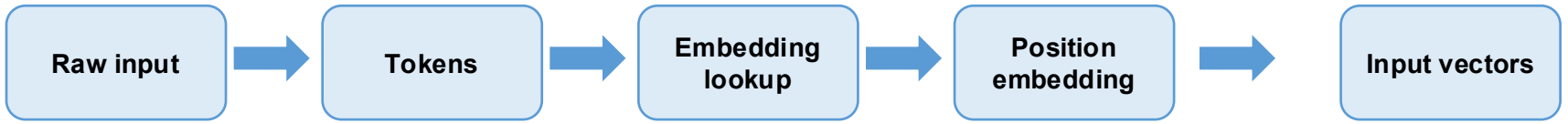
How it works

- Compute similarity scores
- Convert scores to weights with softmax
- Average value vectors using these weights

context_i = sum_j alpha_ij value_j

alpha_ij is the learned attention weight from token i to token j

Tokens, embeddings and position



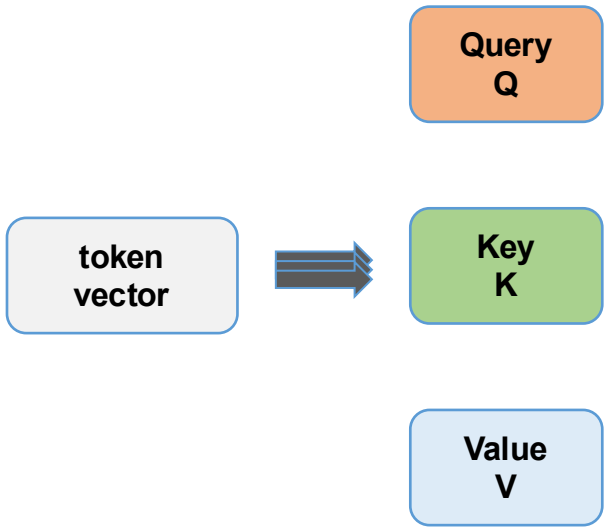
Example text tokenization:
[The] [gene] [is] [expressed]

$$x_i = token\_embedding_i + position\_embedding_i$$

- Transformers do not naturally know the order of tokens
- Position information is added to every token vector
- The same idea will be reused for image patches in ViT

Self-attention: Query, Key and Value

Each token makes three learned projections. These are not hand-designed features.



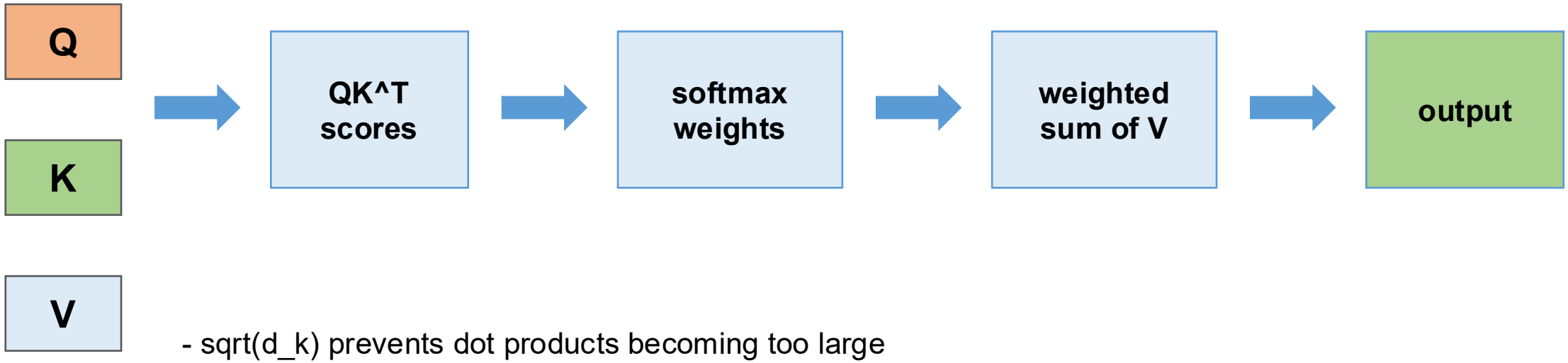
Vector	Question	Intuition
Q	What am I looking for?	A search query
K	What do I contain?	A label / address
V	What information do I pass on?	The message

Attention score between token i and token j = dot product of Q_i and K_j

Scaled dot-product attention

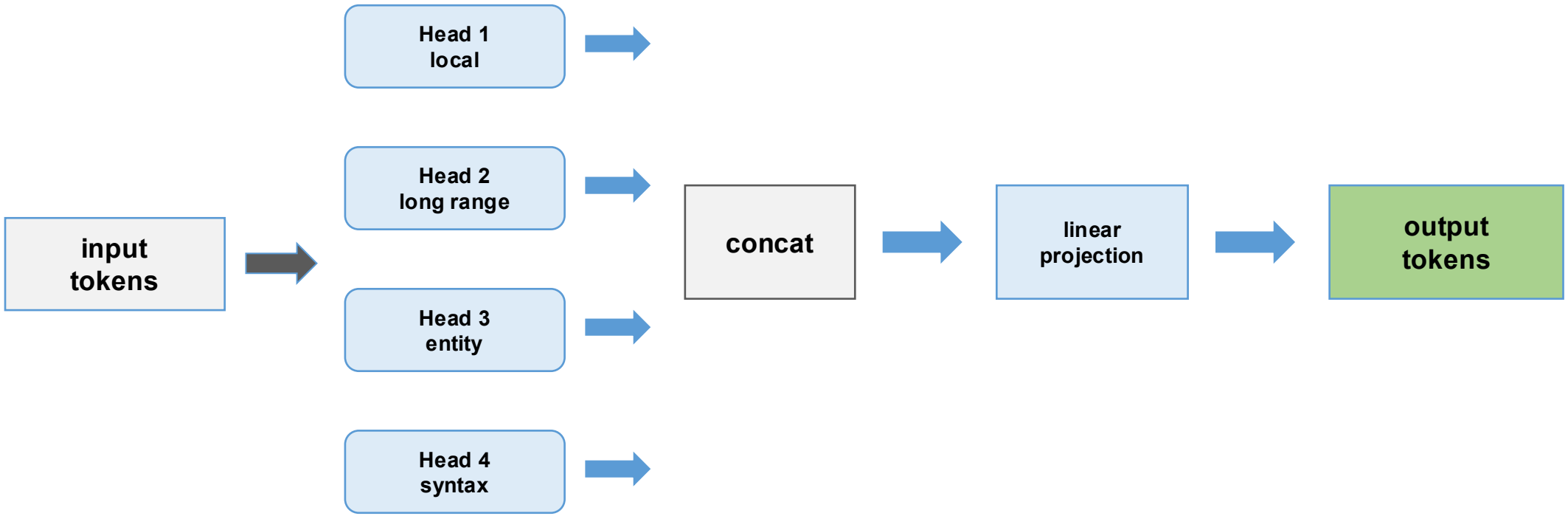
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Attention(Q, K, V) = softmax(Q K^T / sqrt(d_k)) V



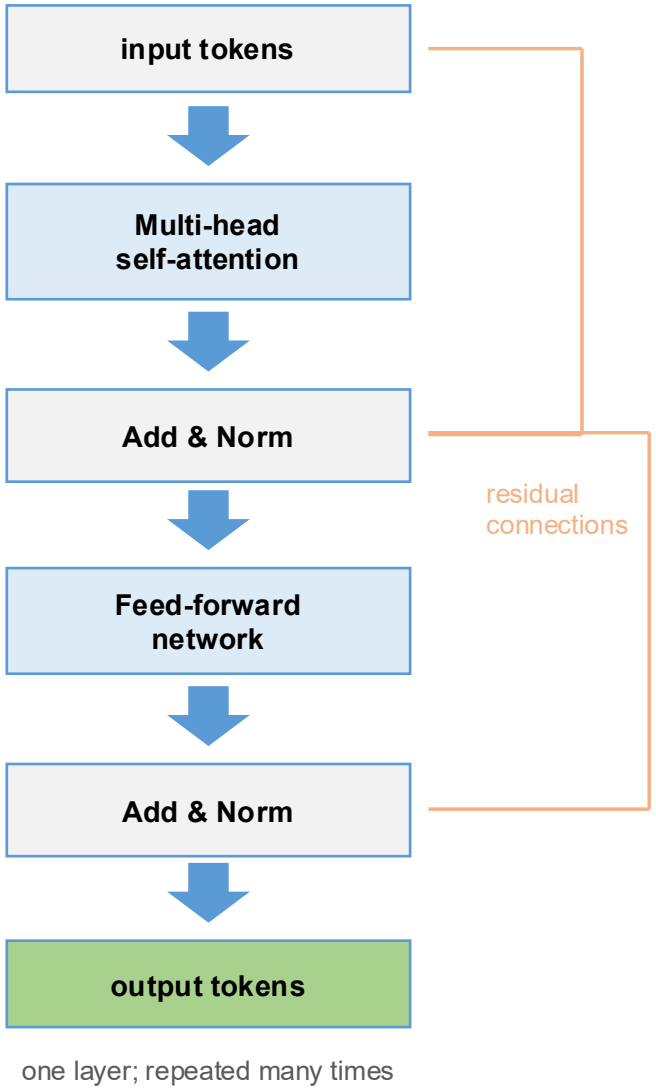
- sqrt(d_k) prevents dot products becoming too large
- softmax makes weights sum to 1
- Every output token becomes a context-aware token

Multi-head attention



- Multiple heads let the model learn several relationship patterns at once
- The heads are concatenated and projected back to the model dimension

Transformer encoder block

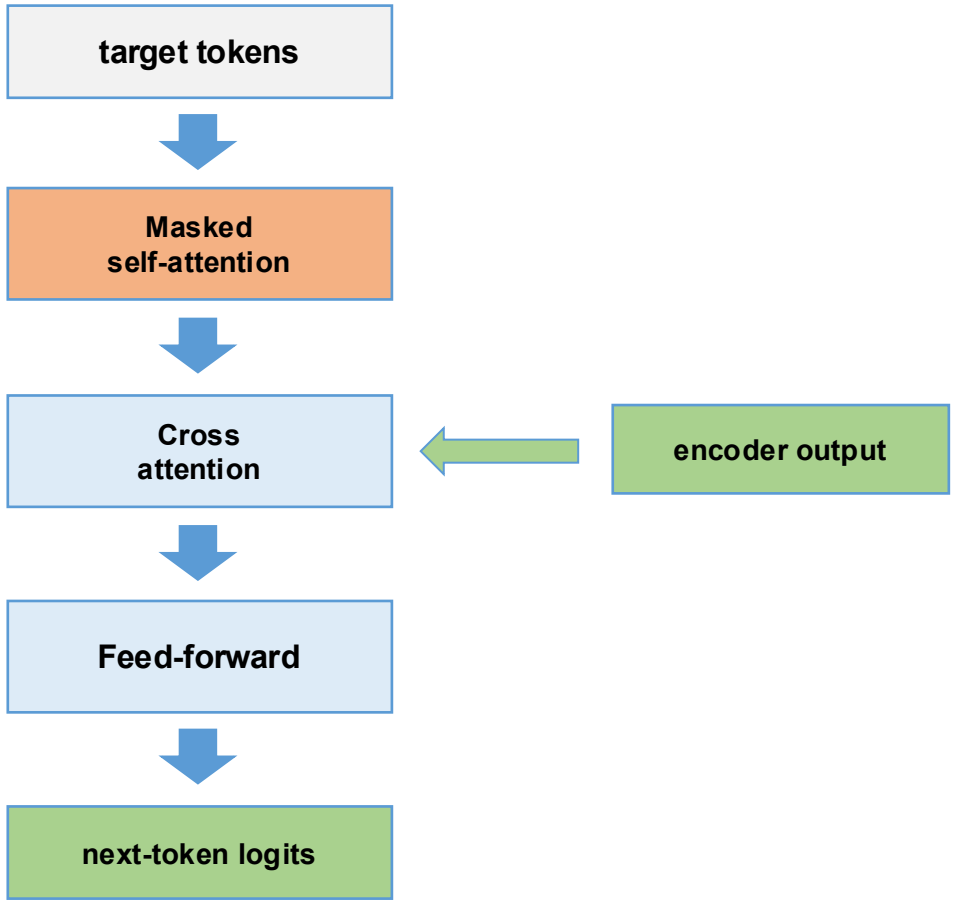


Inside each encoder block

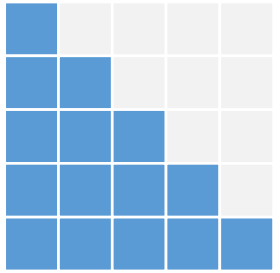
- Self-attention mixes information across tokens
- Feed-forward network transforms each token independently
- Residual connections help gradients pass through deep stacks
- Layer normalization stabilizes training

Encoder-only models are useful for representation learning and classification.

Transformer decoder block



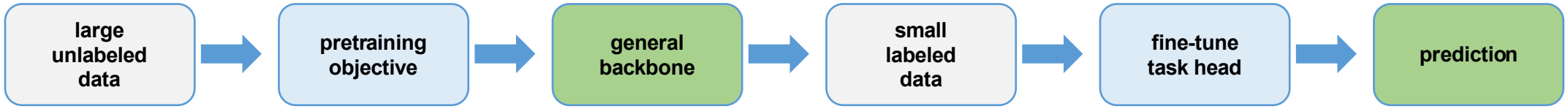
Masked self-attention prevents looking at future tokens



lower triangle = visible

Family	Uses	Example task
Encoder	bidirectional context	classification
Decoder	predict next token	generation
Enc-dec	input to output	translation

Training and transfer learning



Transfer learning

- Pretraining learns general patterns from large data
- Fine-tuning adapts the model to a specific task
- Same backbone can support many downstream tasks

Common objectives

- Next-token prediction
- Masked-token prediction
- Self-supervised / contrastive objectives

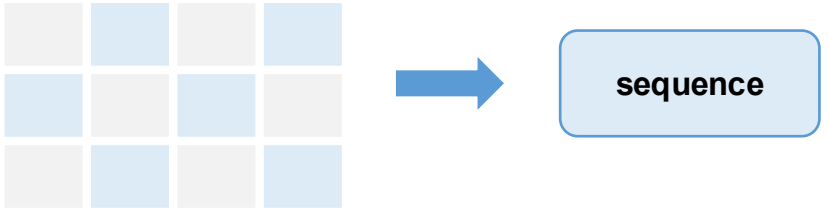
What Transformers are good at - and what to watch out for

Strengths	Limitations
Global context: every token can interact	Attention memory grows roughly with sequence length squared
Parallel training on modern hardware	Large models often need large data and compute
Flexible: text, image patches, proteins, genomics	Can be sensitive to data quality and domain shift
Pretrained backbones transfer well	Interpretability requires careful validation

Rule of thumb: use a Transformer when long-range dependencies or strong pretrained models matter.

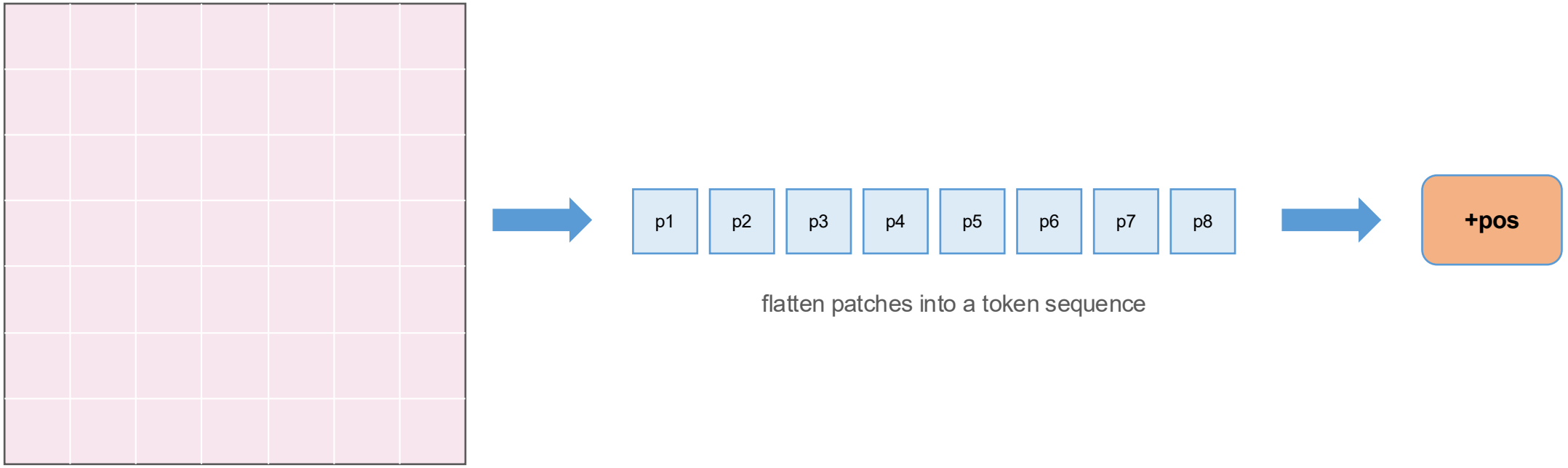
Vision Transformers (ViT)

Applying a Transformer to image patches



Ref: Dosovitskiy et al. 2020, An Image is Worth 16x16 Words

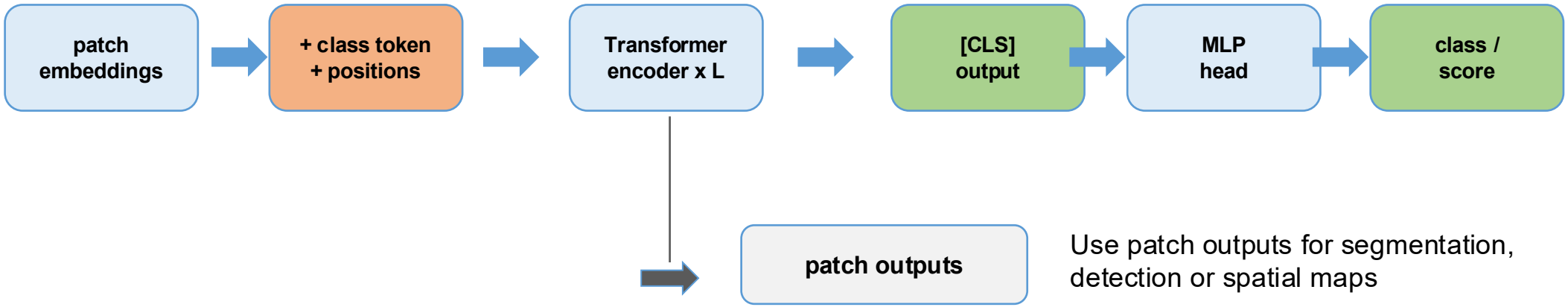
From image to patch tokens



Example: 224x224 image with 16x16 patches -> 14x14 = 196 patch tokens

Ref: Dosovitskiy et al. 2020

Vision Transformer architecture

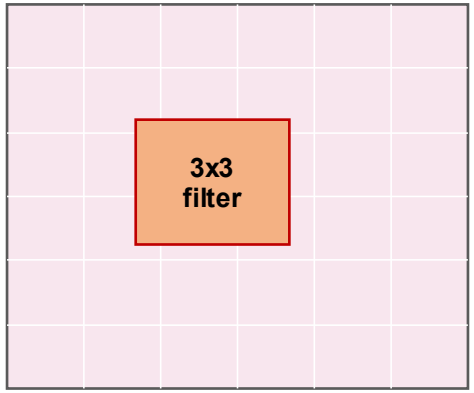


- For image classification, the class token summarizes the whole image
- For pathology/spatial tasks, patch tokens can preserve local information

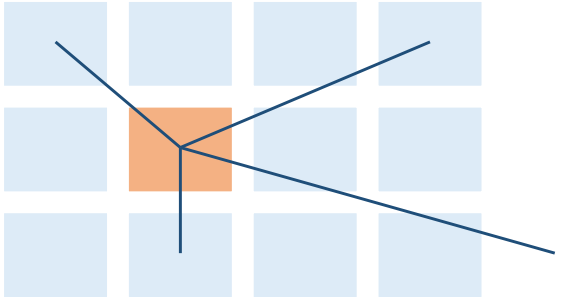
CNN versus ViT

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Convolutional neural network

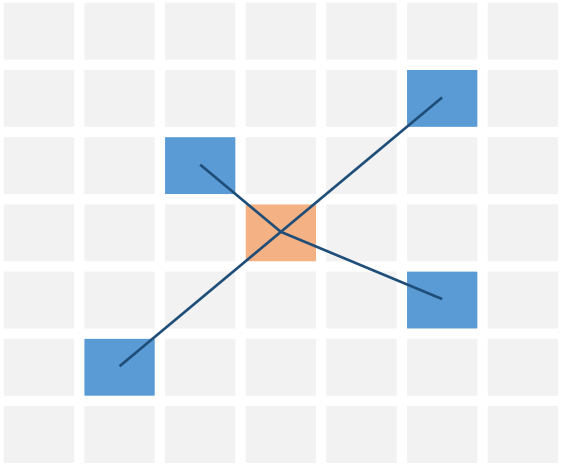


Vision Transformer



Question	CNN	ViT
Basic unit	small convolution filters	image patch tokens
Context	local -> larger through depth	global attention between patches
Data bias	strong image inductive bias	benefits from large pretraining

Interpreting ViT attention



one patch attends to several image regions

What it tells us

- Attention maps show how patch tokens exchange information
- They are useful for sanity checking what the model focuses on
- They are not a complete explanation by themselves

This leads naturally into explainable machine learning methods such as saliency, occlusion and LIME.

Training ViT in practice

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

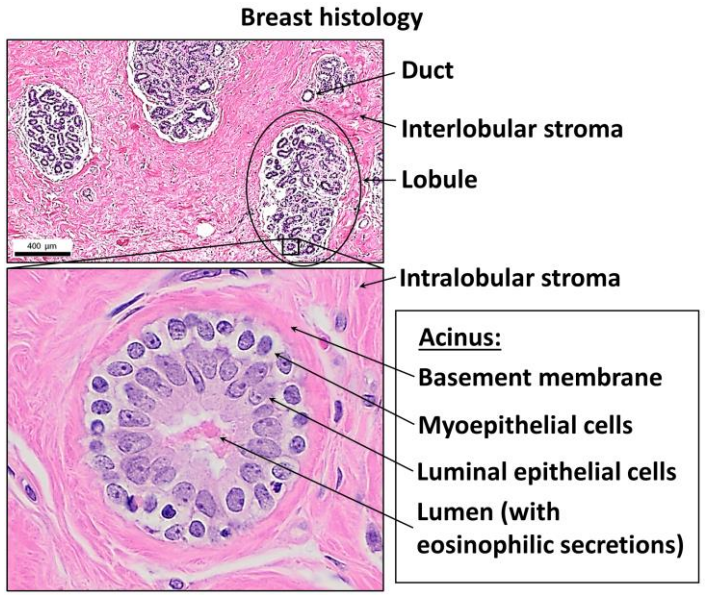
Practical workflow

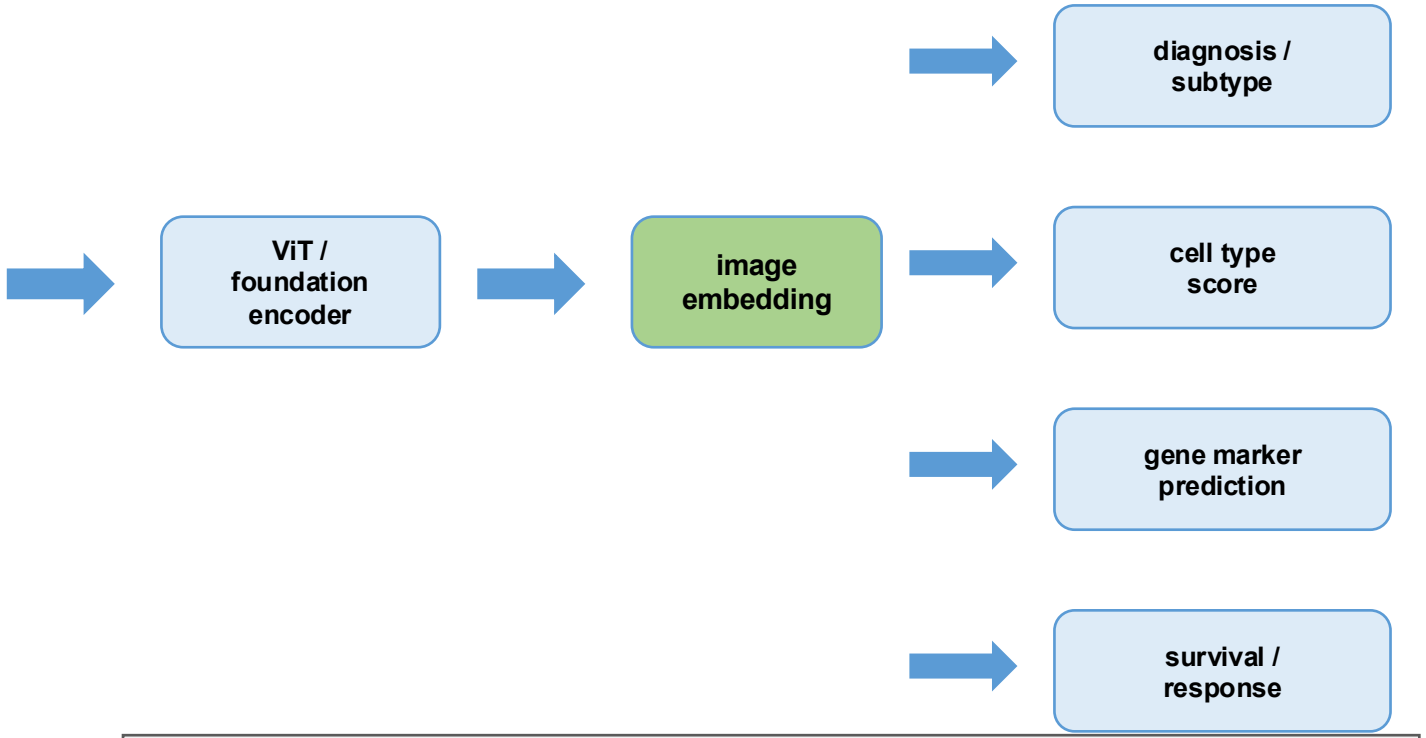
- Choose image resolution and patch size
- Start from pretrained weights when possible
- Use augmentation and domain-specific preprocessing
- Fine-tune the head only, then full model if data allows
- Evaluate on independent and out-of-distribution data

Hyperparameter	Why it matters
Patch size	smaller = more tokens and memory
Resolution	controls biological detail
Model size	capacity versus overfitting
Learning rate	most common fine-tuning failure

For small biomedical datasets, pretrained ViT/foundation model embeddings are often the strongest baseline.

ViT in pathology and spatial omics





Spatial omics can provide labels and biological validation for what the image encoder learns.

Explainable Machine Learning

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Why

1. Accountability: To know about the conditions/factors that caused that decision.
2. Trust: To understand what the model is doing
3. Compliance: decision-making must provide meaningful information. From model extracts useful information for discovery rather than performance (accuracy vs interpretability tradeoff)
4. Performance: If we know how a model work then we fine-tune and optimize. Bug fixing and model optimization
5. Enhanced control: As we can understand the flaws and vulnerabilities.
6. Credibility/reliability of the model

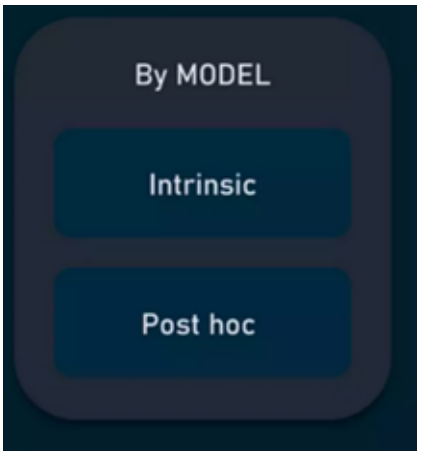
How

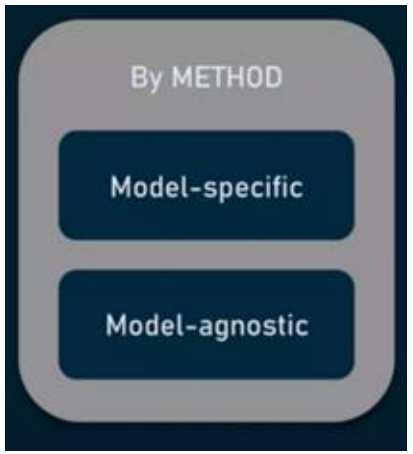
1) Interpreting outputs: with saliency maps, with occlusion sensitivity, and with class activation maps (Global Average Pooling)
2) Visualisation of the model training steps: with gradient ascent (class model visualization), with dataset search, and deconvolution
3) Deep dream (going deeper in NNs) or LIME (Local interpretable model-agnostic explanations)

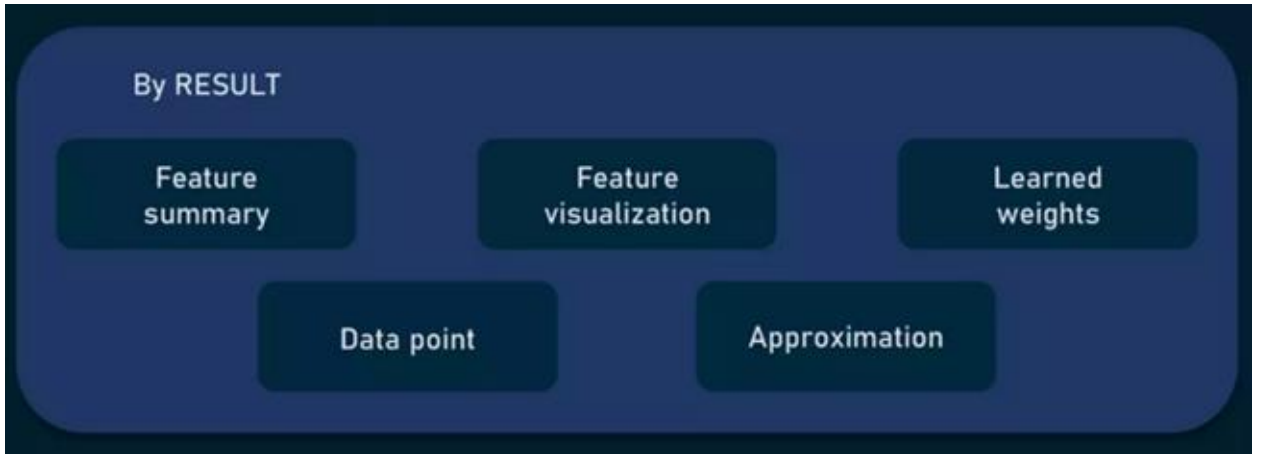
e.g. Saliency map compute the gradient of output category with respect to input image:

$$\frac{\partial output}{\partial input}$$

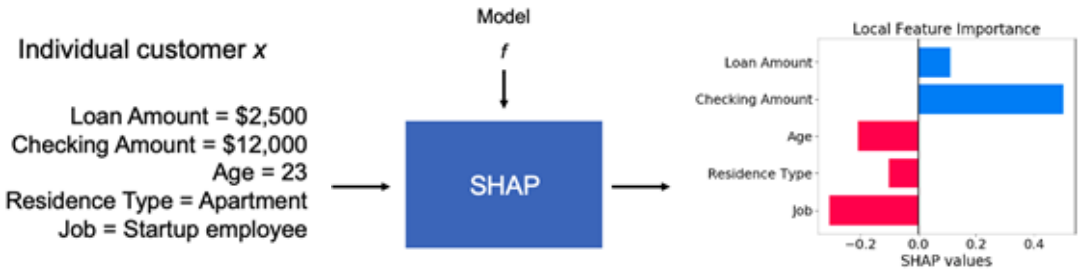
Different Methods to Interpret Deep Learning Models

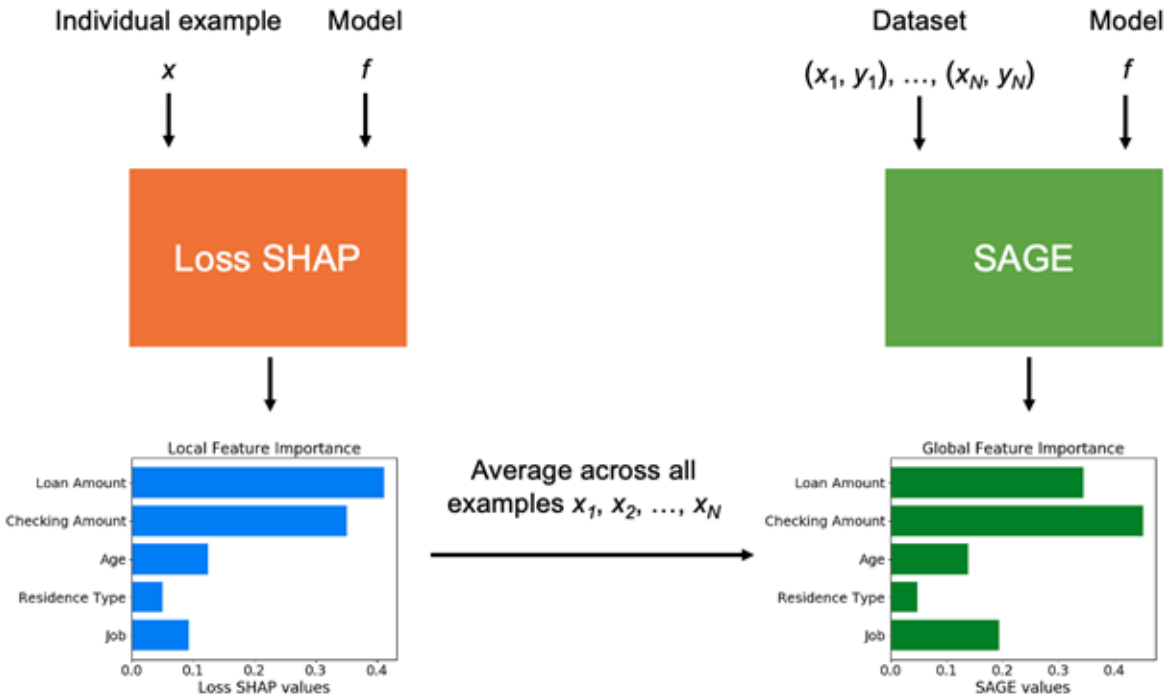




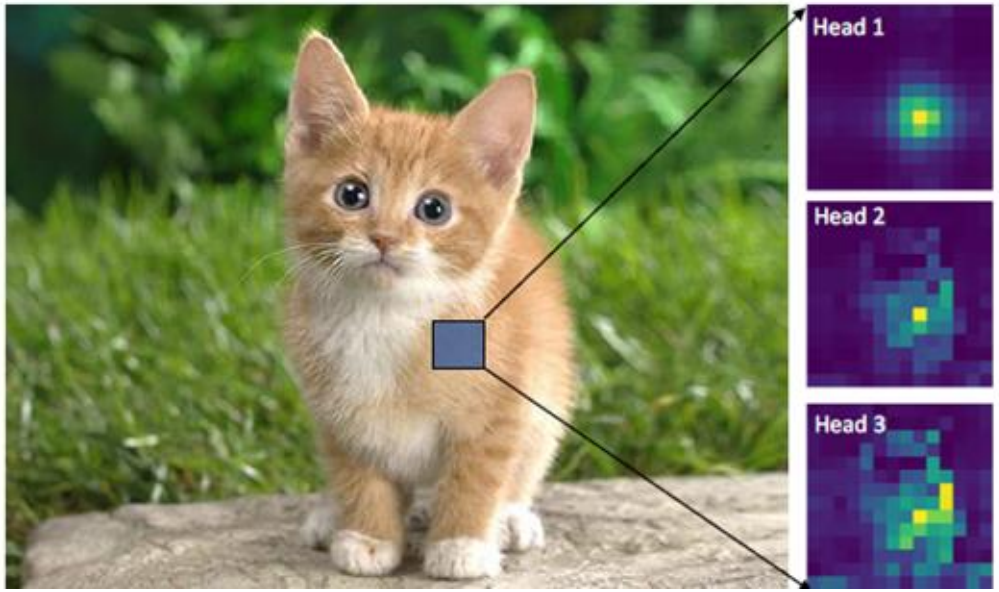


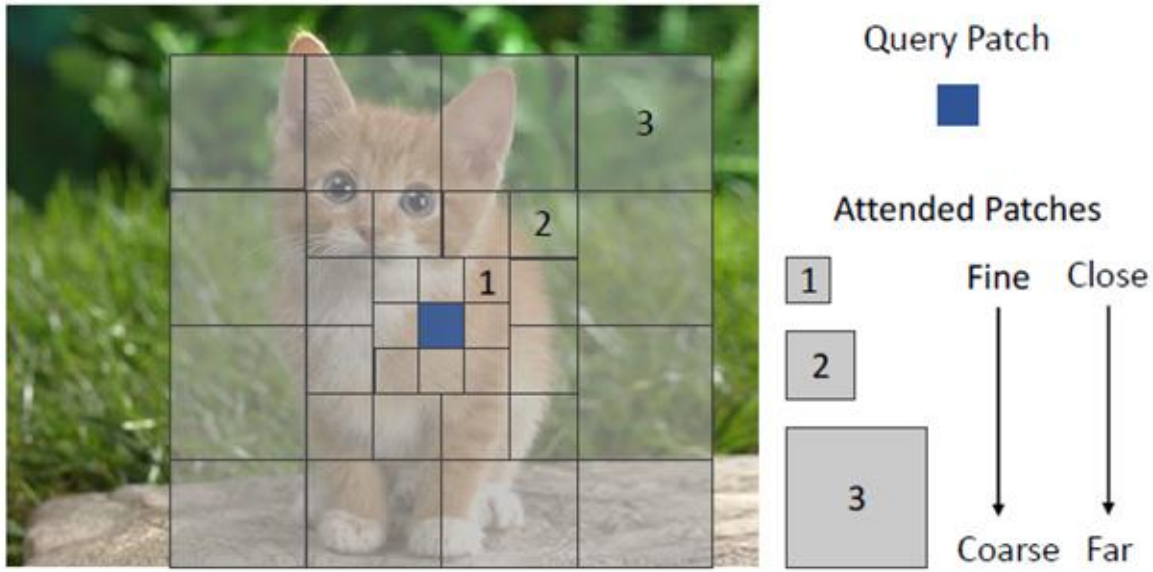
	SHAP	LossSHAP	SAGE
Type of explanation	Local	Local	Global
Cooperative game	$v_{f,x}$	$v_{f,x,y}$	v_f
Feature values	$\phi_i(v_{f,x})$	$\phi_i(v_{f,x,y})$	$\phi_i(v_f)$
Meaning	Contribution to prediction $f(x)$	Contribution to accuracy of $f(x)$	Contribution to f's performance
Approximation	IME [4] KernelSHAP [1] TreeSHAP [5]	LossSHAP sampling [2] TreeSHAP [5]	Mean LossSHAP value [2] SAGE sampling [2]





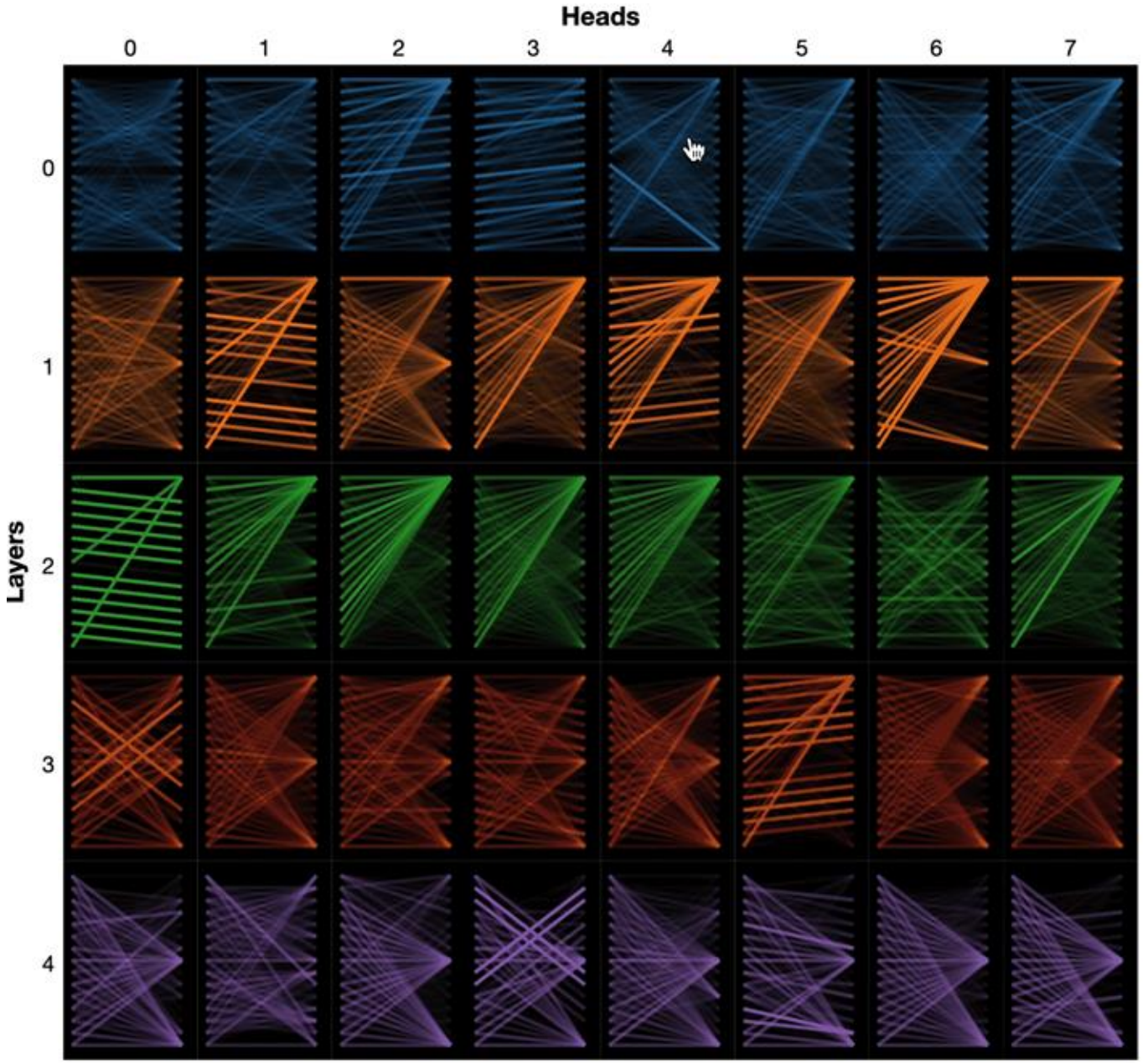
A visual example of explainability in a transformer





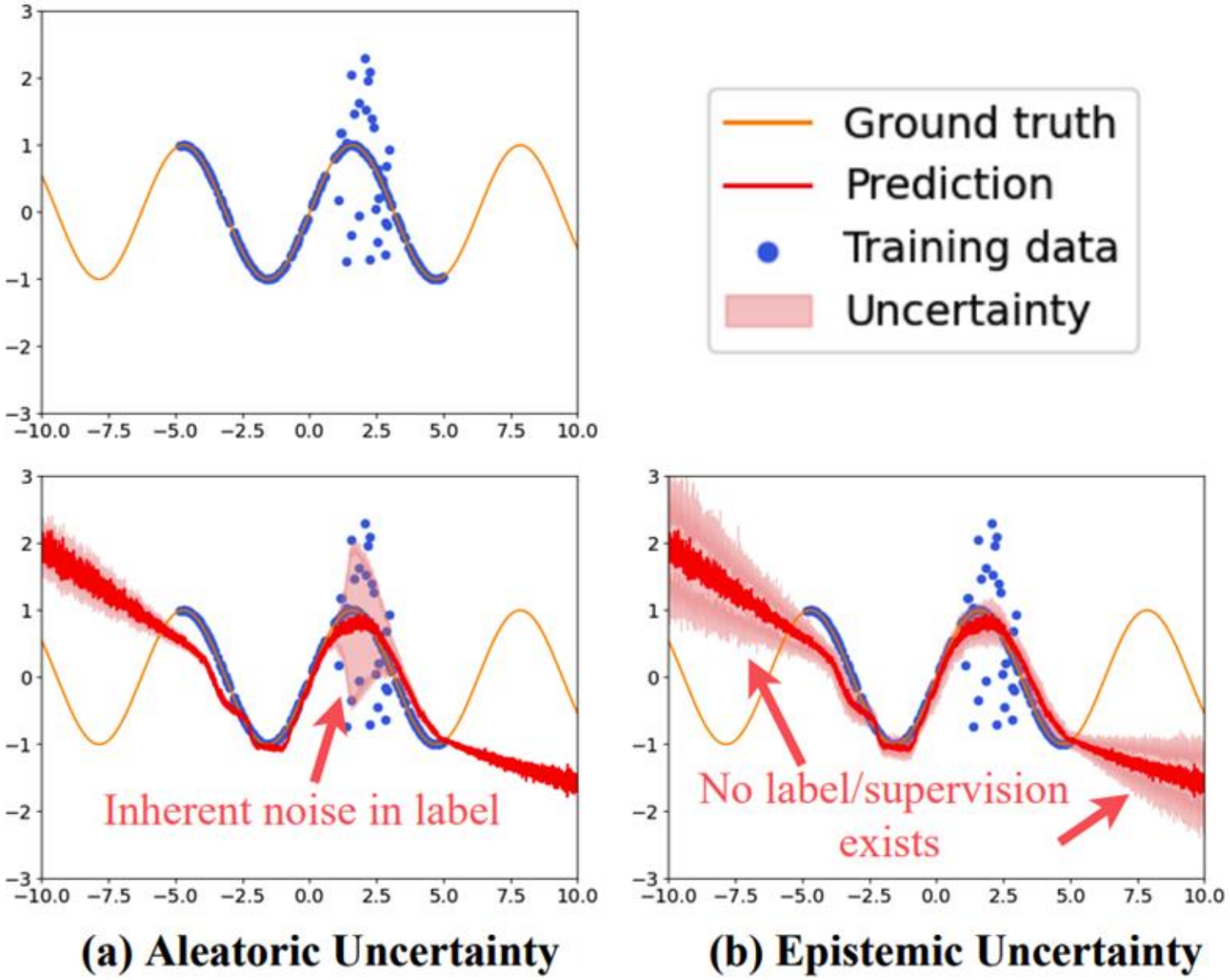
Each token attends its closest surrounding tokens at fine granularity and the tokens far away at coarse granularity, and thus can capture both short- and long-range visual dependencies efficiently and effectively.

A visual example of explainability in a transformer

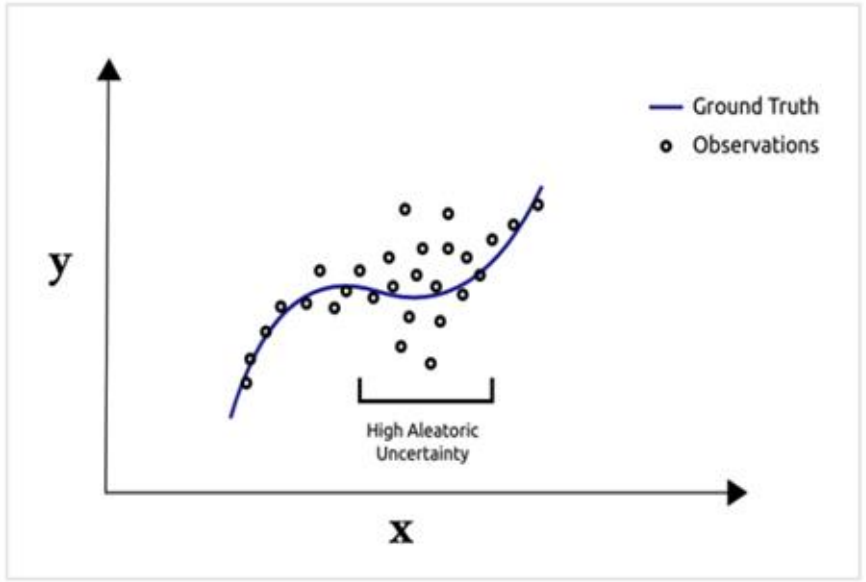


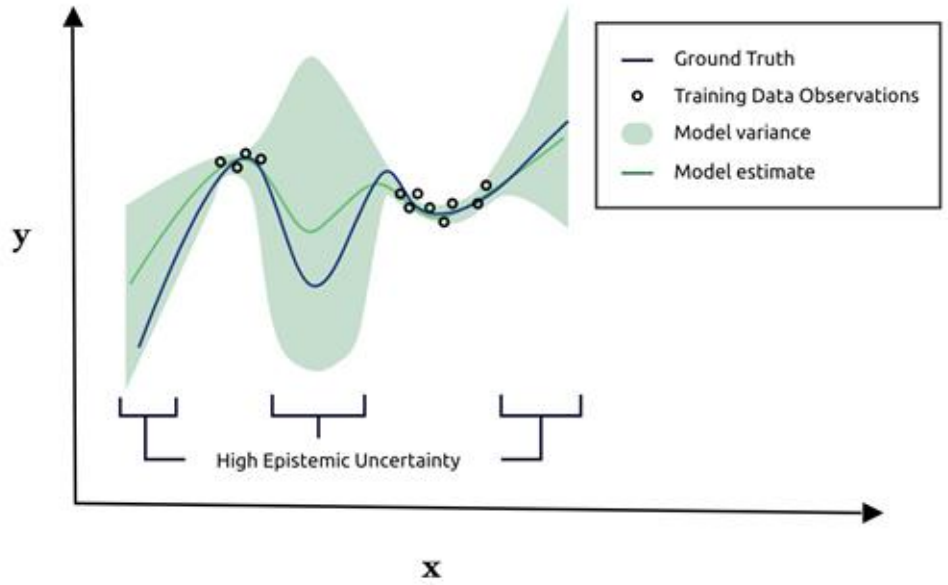
https://github.com/jessevig/bertviz

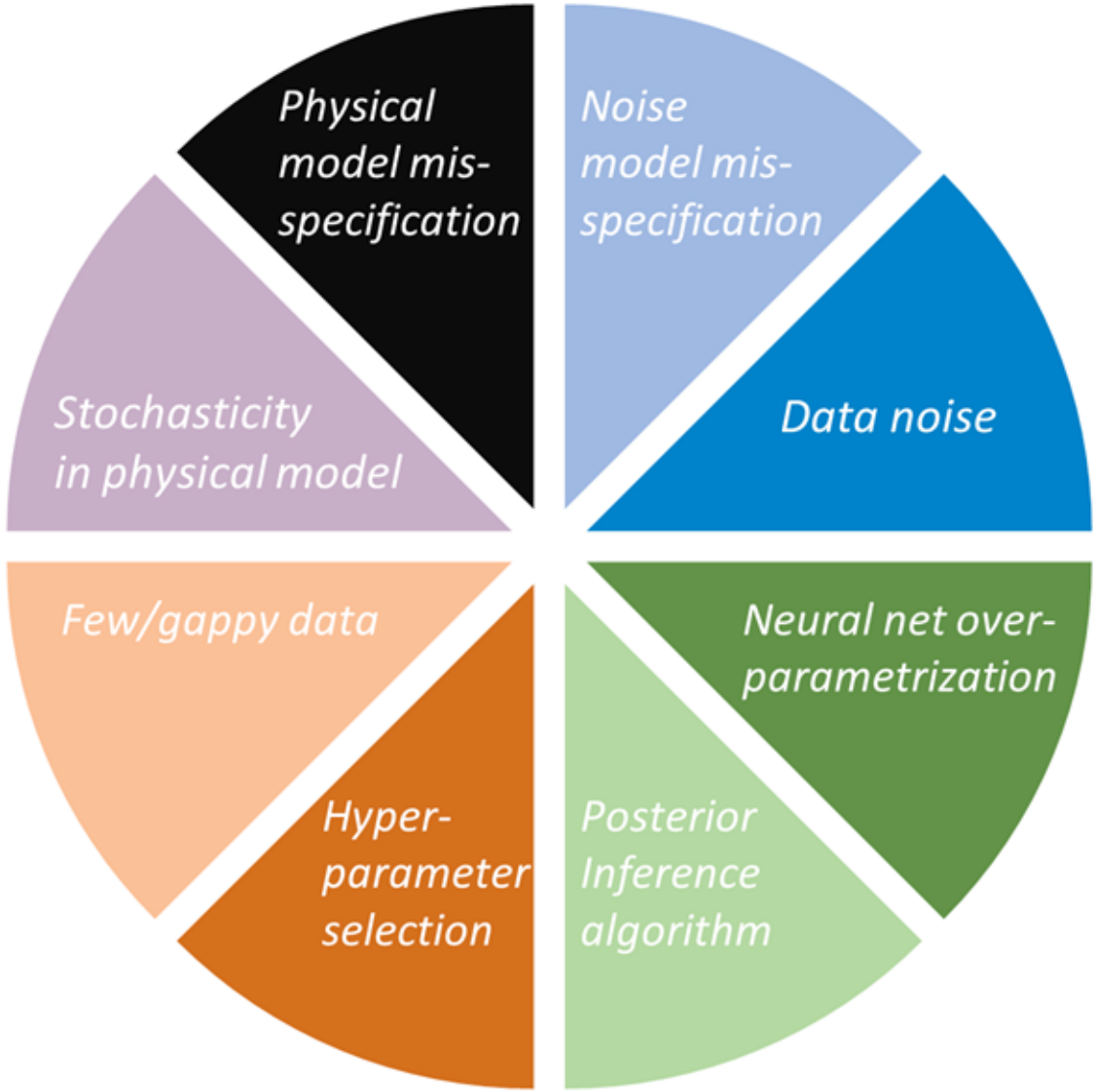
Importance of knowing what we don't know - Two types of uncertainty



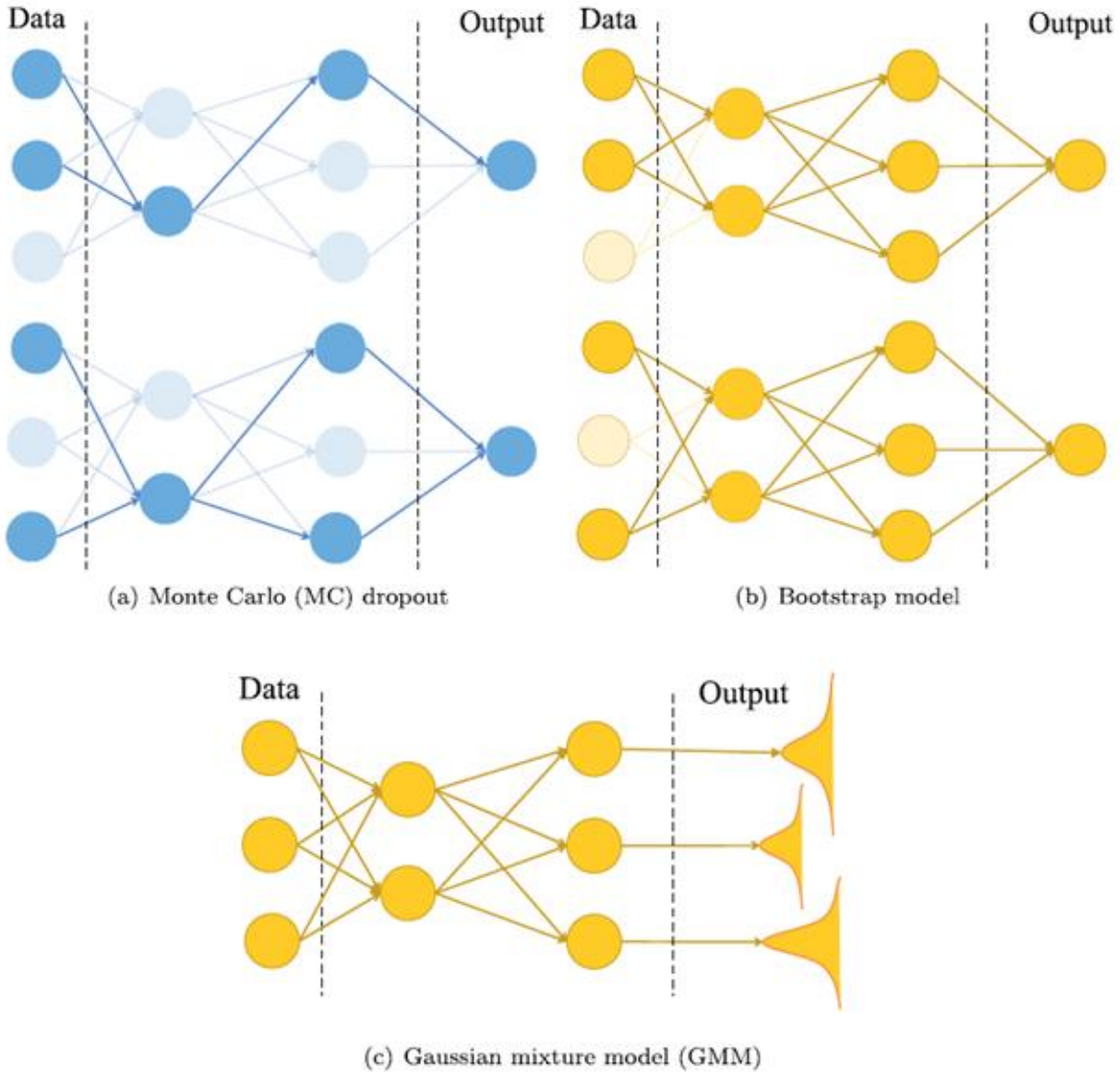
Types and sources of uncertainty







Quantifying uncertainty



(a) Monte Carlo (MC) dropout

(b) Bootstrap model

(c) Gaussian mixture model (GMM)

- Assume observed data is normally distributed

$$\hat{\mu}, \hat{\sigma}^2 = f_\theta(\mathbf{x})$$

$$p_\theta(y|\mathbf{x}) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$$

- Minimize difference of predictive distribution to the target distribution

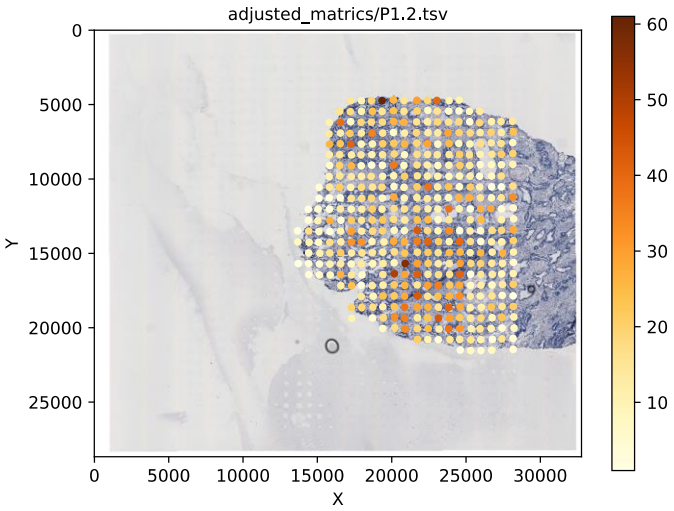
$$-\log p_\theta(y|\mathbf{x}) = \frac{\log \hat{\sigma}^2}{2} + \frac{(y-\hat{\mu})^2}{2\hat{\sigma}^2}$$

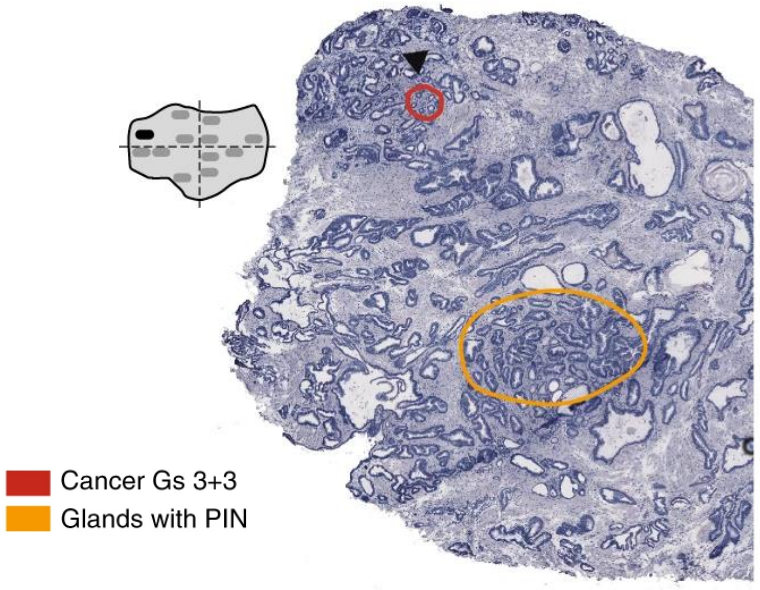
- multiple forward passes of the same data through NN network while applying different dropout masks
- “combine” predictions to obtain an Expectation-based prediction and uncertainty estimate

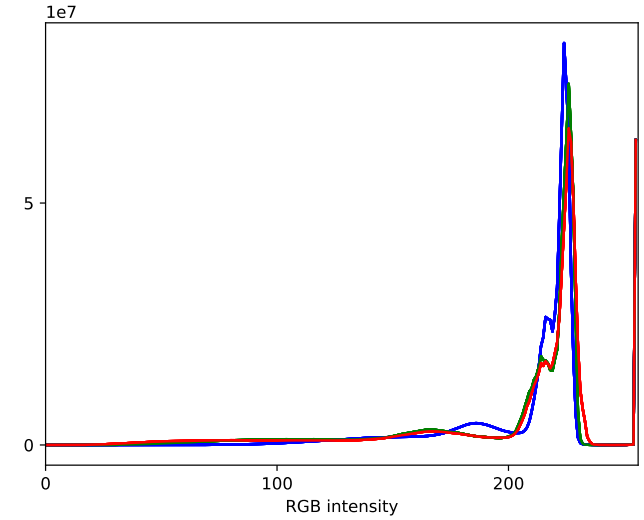
Lecture 5: ML Applications in Spatial Omics

Spatial Transcriptomics: expression + image + location

	14.96x10.06	15.92x10.05	17x10.06	17.89x10.06
STARD7 ENSG00000084090	0	0	1	0
WDR1 ENSG00000071127	1	1	1	1
NDUFB2 ENSG00000090266	2	4	2	1
BAIAP2L1 ENSG00000006453	2	1	8	1







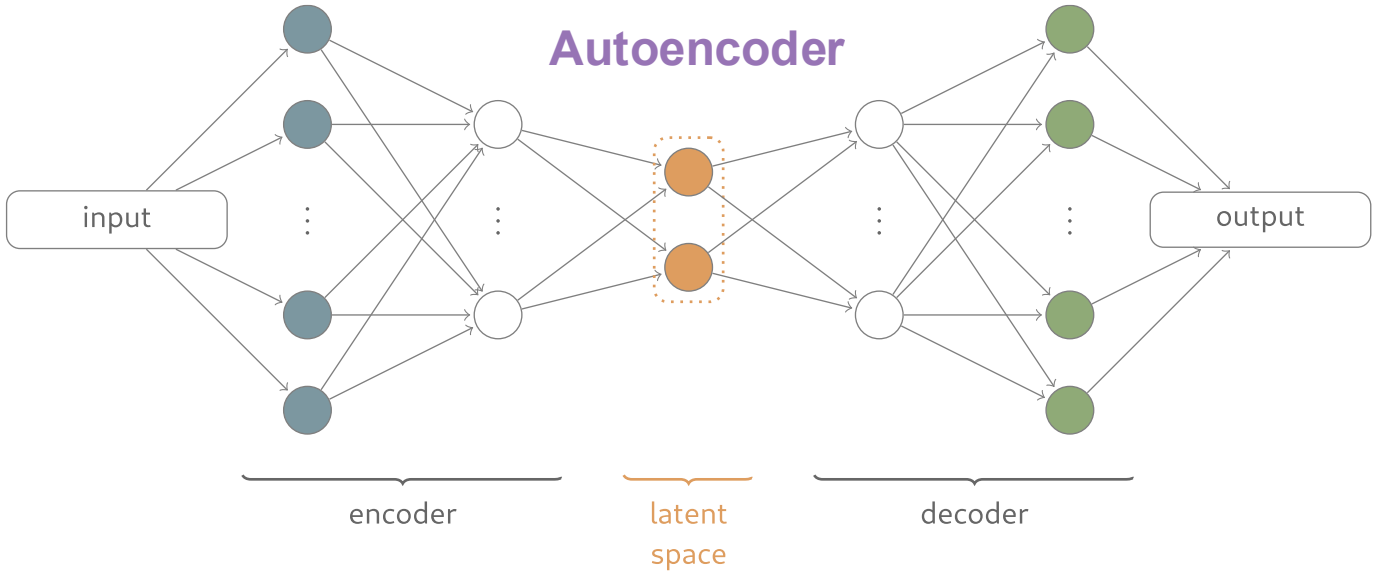
(Berglund et al, 2018)

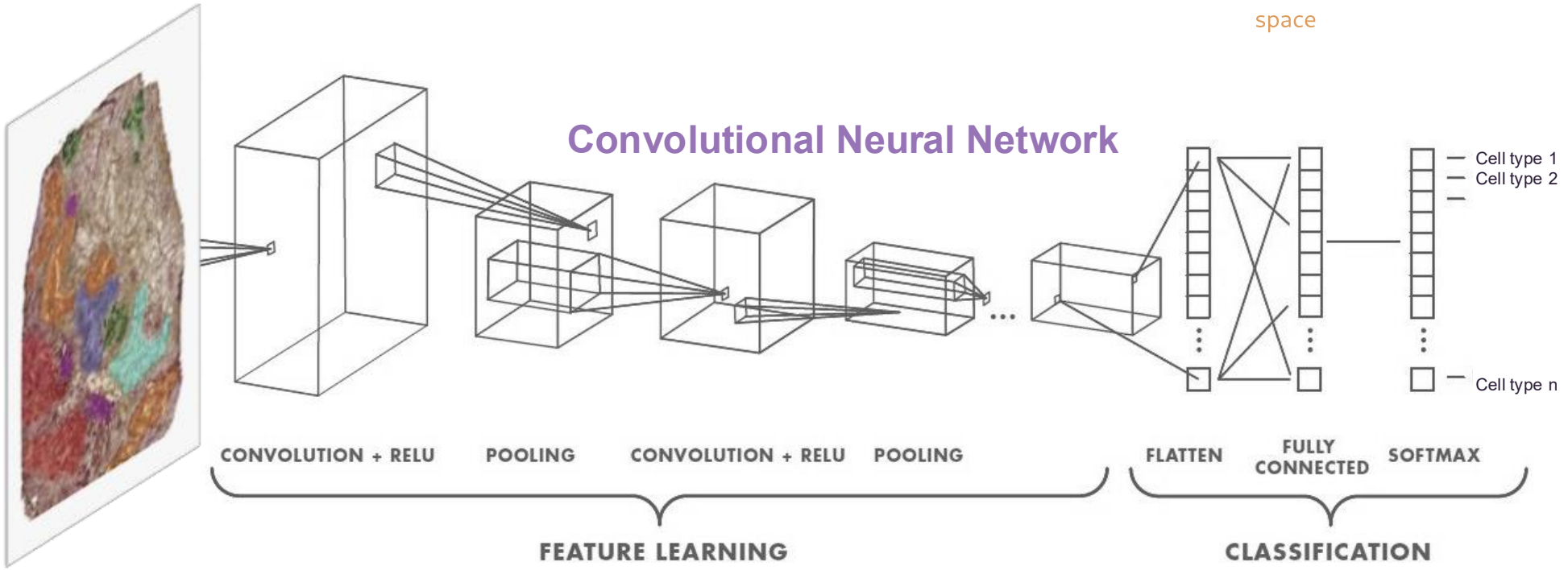
Image mode=RGB, size=32768x28672, (28672, 32768, 3)

Neural Network for Spatial Transcriptomics

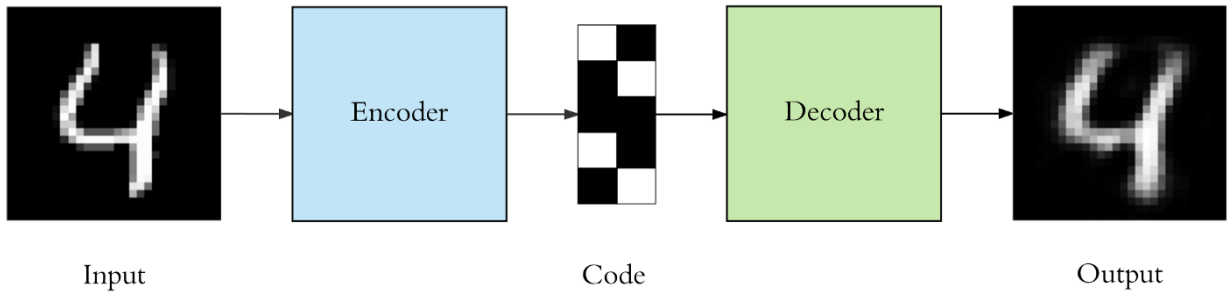
Two neural network (NN) architectures

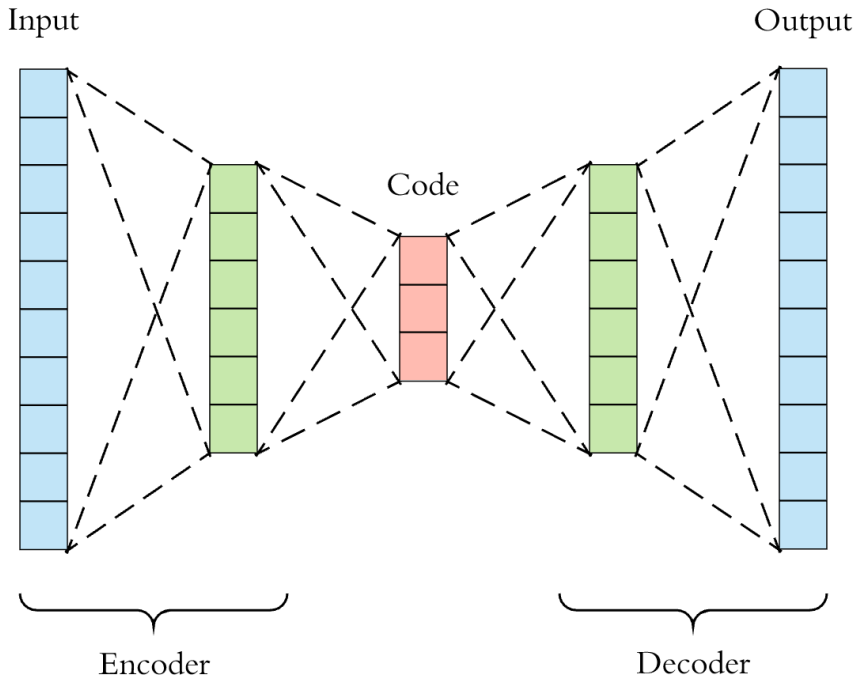
- Convolutional Neural Network (CNN) for feature extraction
  - Designed for spatial imaging data
- Autoencoder (AE) for combining data
  - Find informative shared latent space





Autoencoder





Input is the same shape as the output - compress the input into a lower-dimensional *code (latent-space representation)*

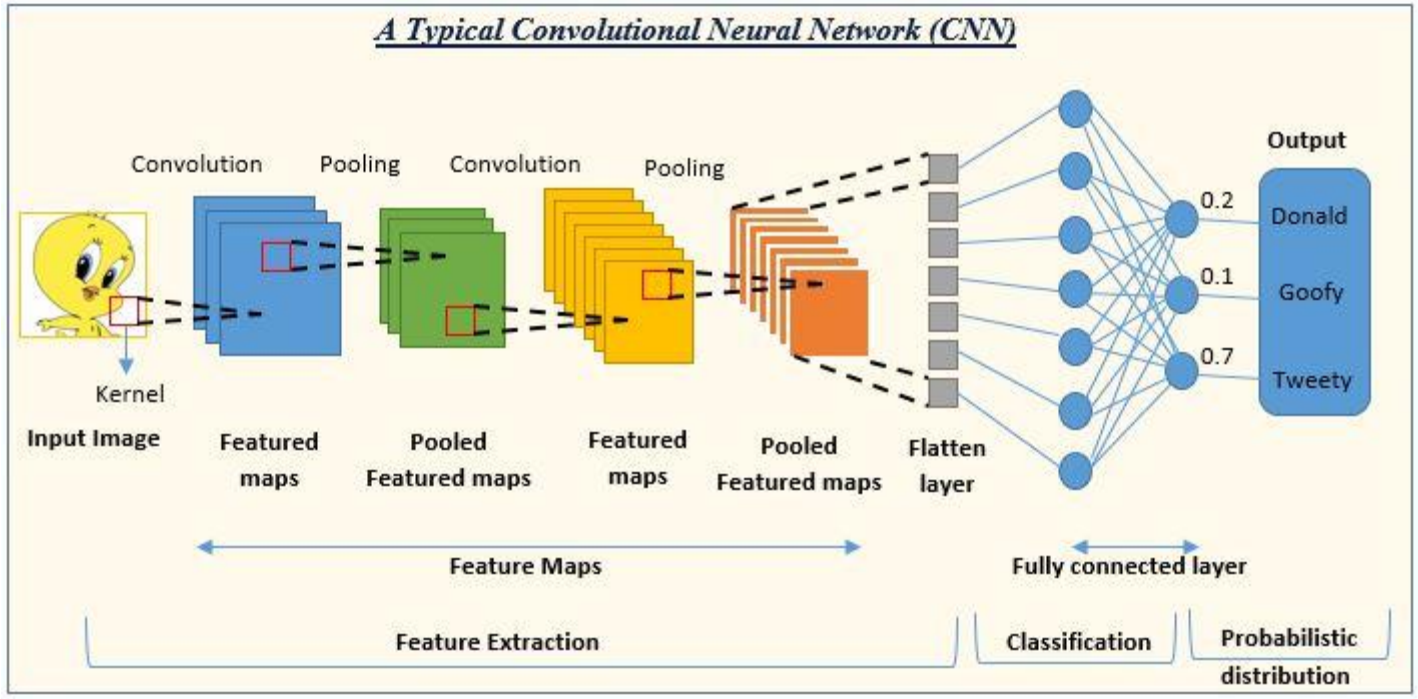
The latent space is determinate

Loss function: KL divergence

Practical 4.1.1

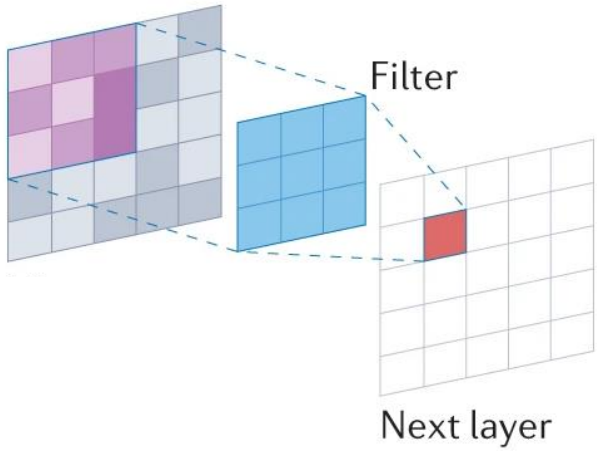
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

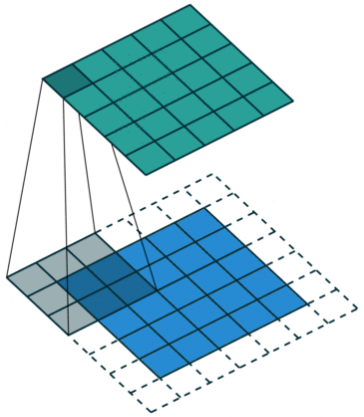
CNN: convolutional neural network



Practical 4.1.2

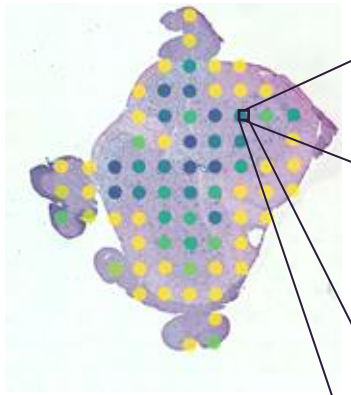
Convolution





https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/
https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Disease Stage Classification Model (fusion model)



Silas et al., Science 2019

Gene Expression Data

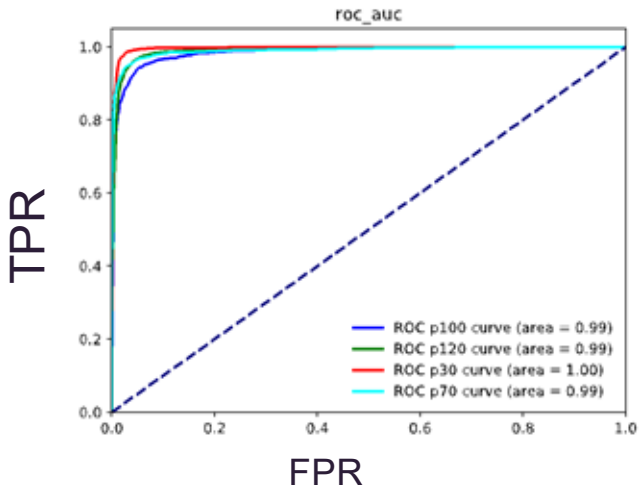
	3.942x25.915	2.977x25.938	1.996x25.956	4.975x26.895	3.947x26.901	2.987x26.921	...
Fam234a	0	0	0	0	0	0	...
Nefl	0	1	0	2	2	10	...
Sema5a	0	0	2	0	1	0	...
Tom1l2	0	0	0	2	0	0	...
Nbea	0	0	0	0	0	0	...
Mif	0	0	1	1	0	2	...
Pcsk1n	2	0	0	0	1	3	...
810021J22Ri	0	0	0	0	0	0	...
Tsfm	0	0	1	0	0	0	...
Zfp706	0	2	0	2	1	0	...
Sfpq	0	0	0	1	0	1	...
Atp1a1	0	0	0	0	1	1	...
Ttc14	0	0	0	0	0	0	...
Fkbp4	0	2	0	0	0	0	...
Mdh1	0	1	0	4	2	8	...
Bub3	0	0	0	0	0	0	...
Rpl13a-ps1	0	1	0	2	3	0	...
Apod	0	2	9	1	0	2	...
Cox7c	1	0	2	2	2	11	...
Gm2237	0	0	0	0	0	0	...
Atp5f1	1	1	0	9	1	4	...
...	...	...	...	...	...	...	...

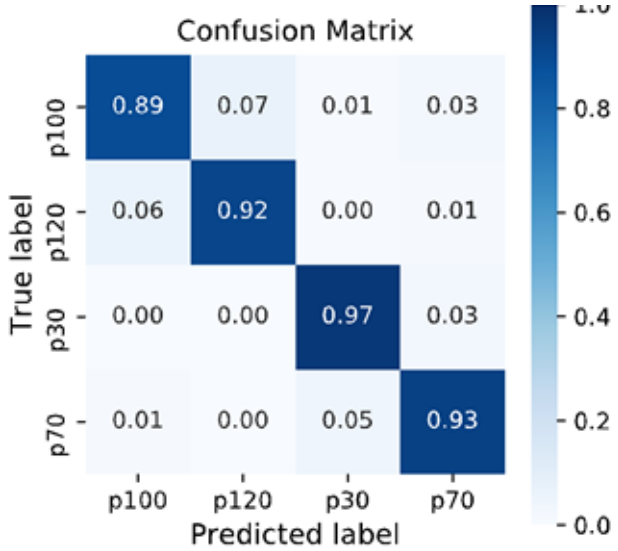
Disease Stage Classification - Performance

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Combined model

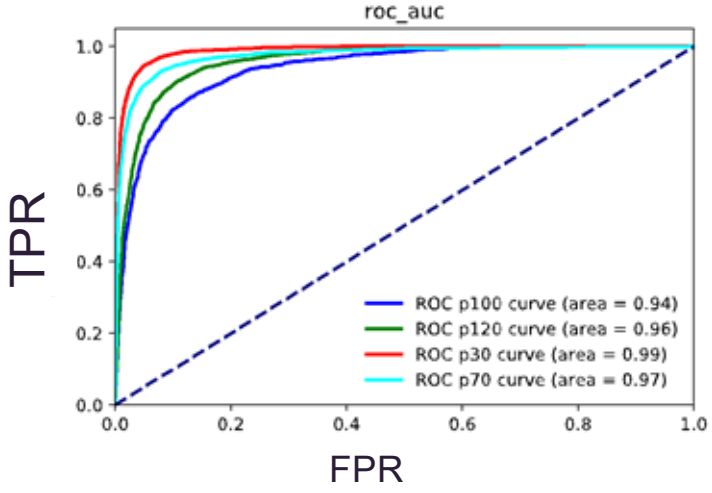
Test accuracy : ~92.75%





Gene count model

Test accuracy: ~84%



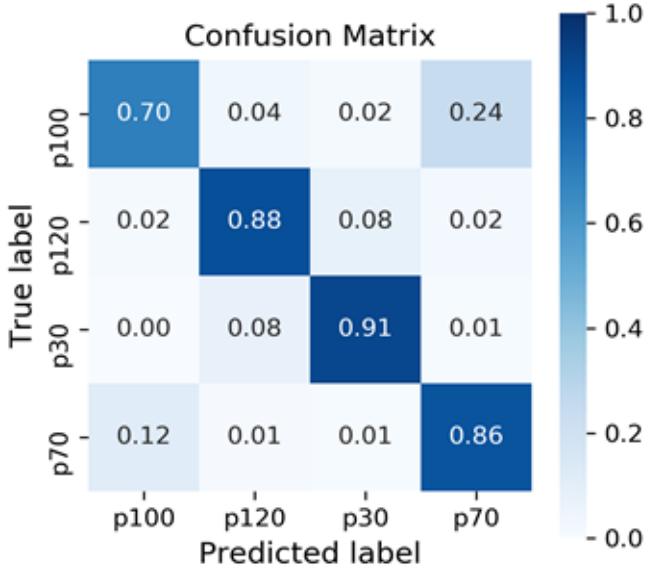
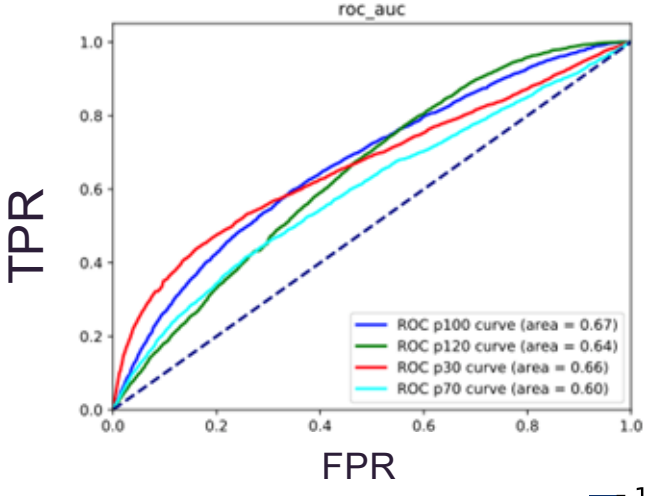
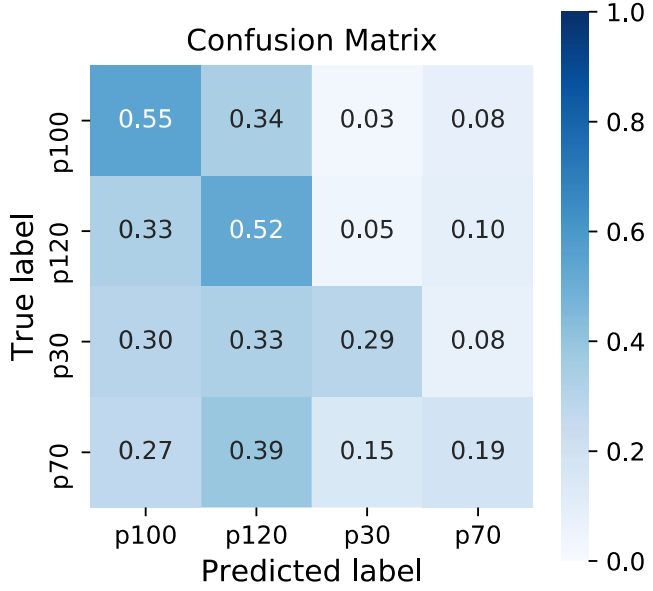


Image model

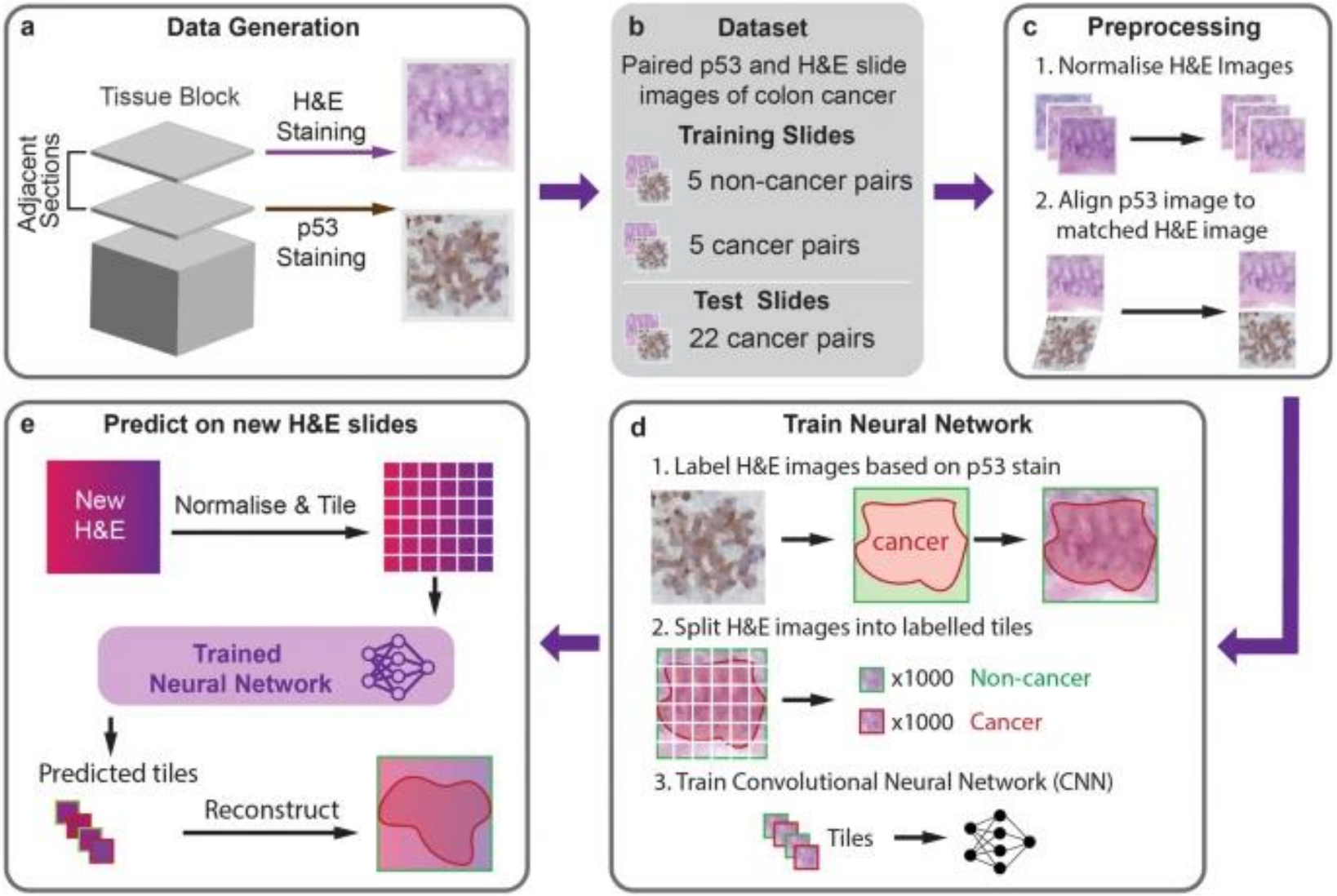
Test accuracy: ~40%



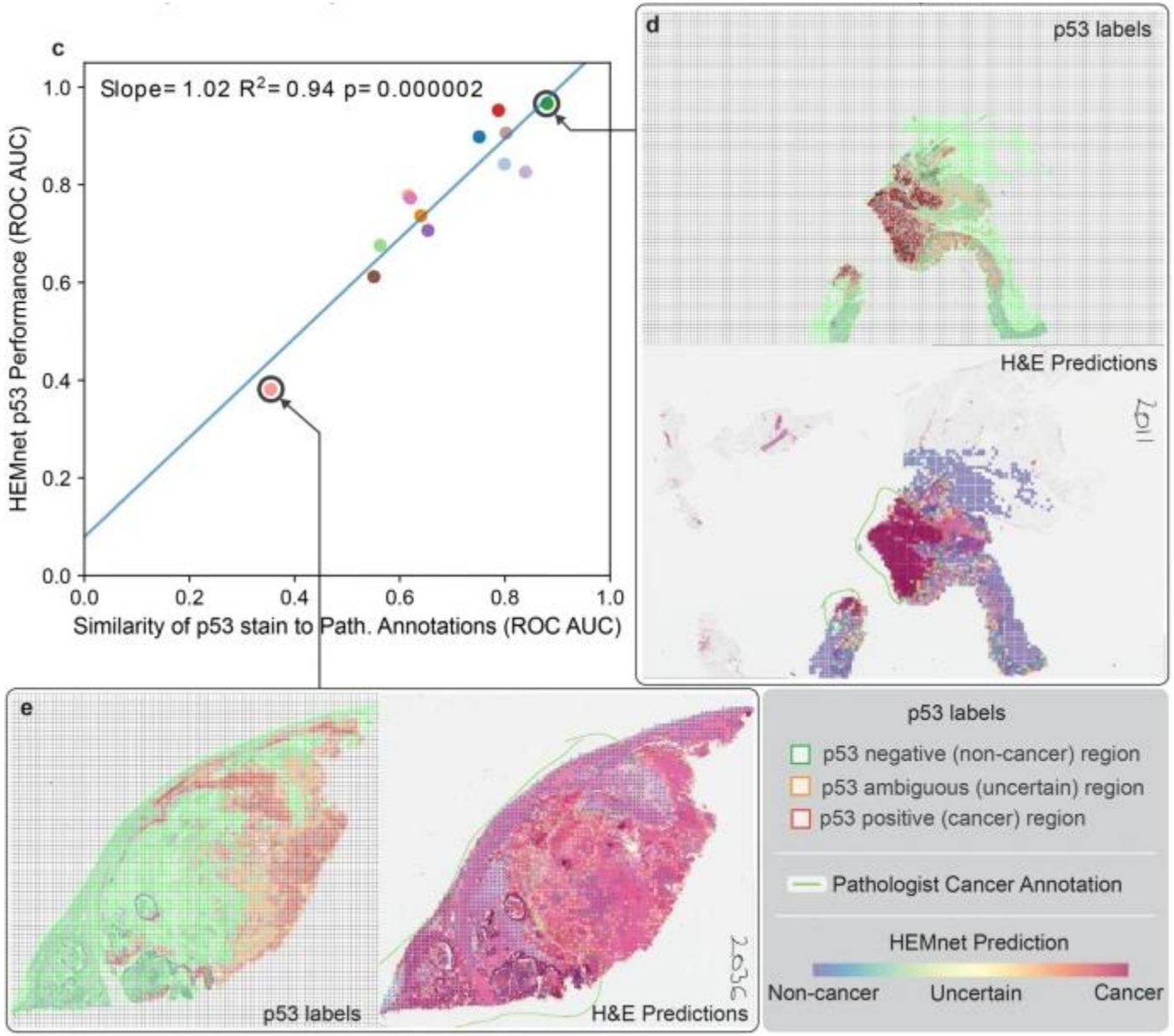


P30: pre-symptomatic
P70: onset
P100: symptomatic
P120: end-stage

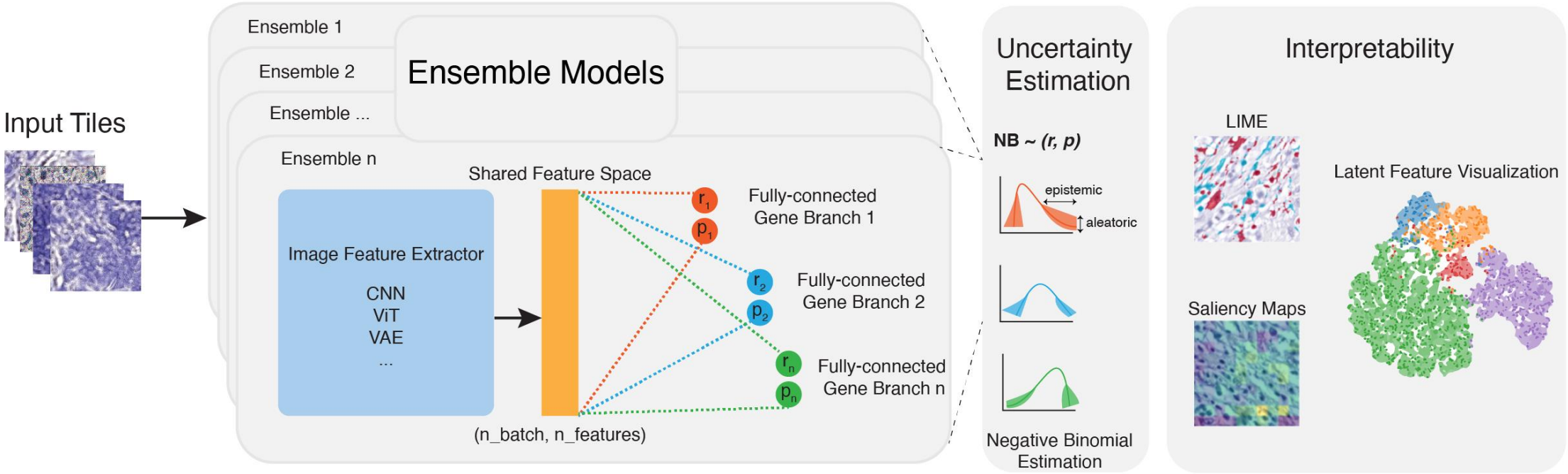
H&E Molecular neural network (HEMnet) workflow overview



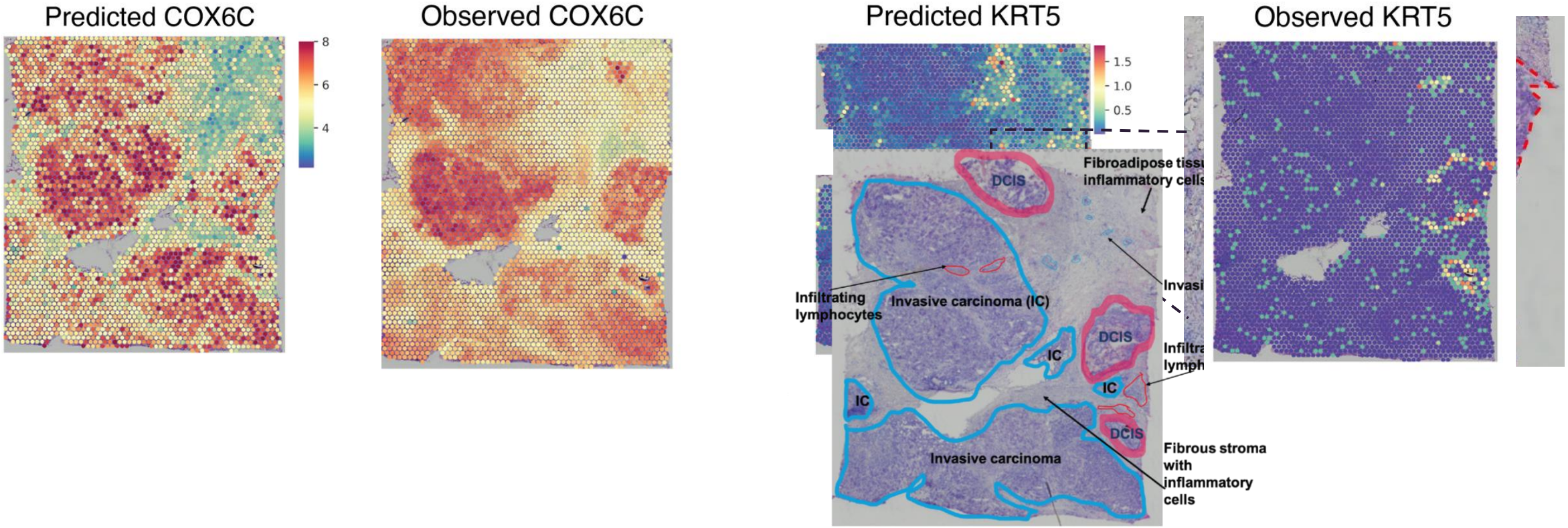
H&E Molecular neural network (HEMnet) workflow overview.



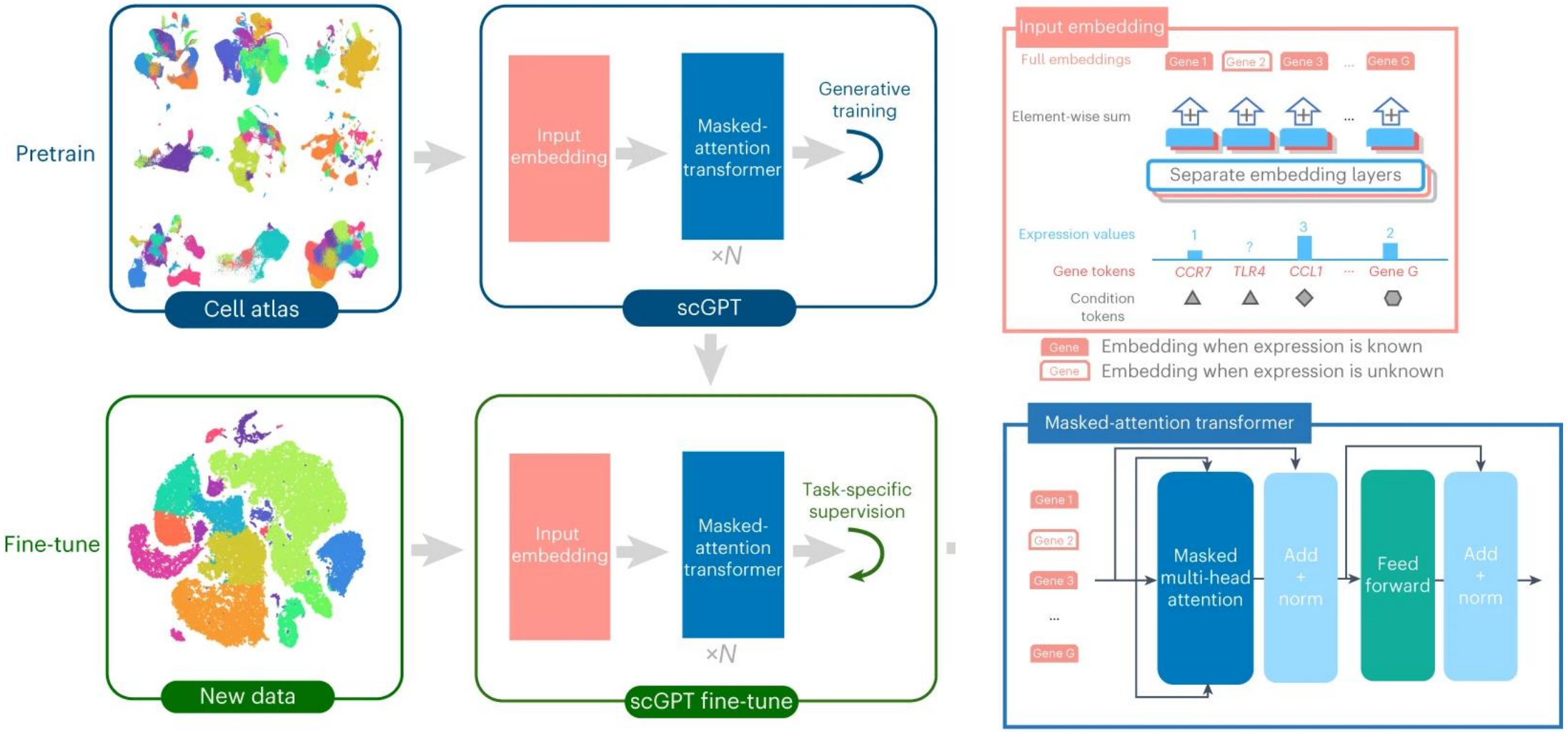
STimage: interpretable prediction of gene markers and cell types



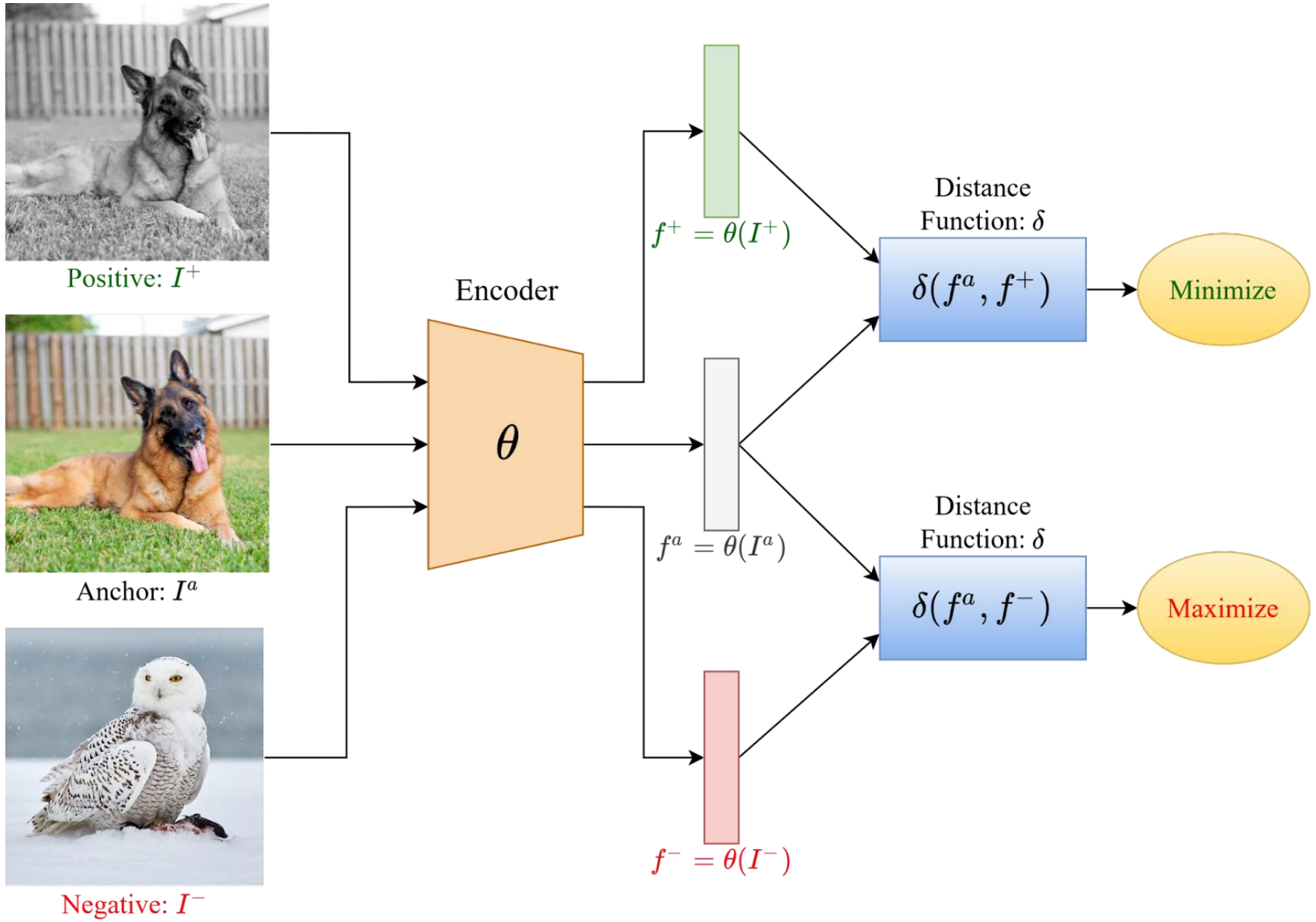
STimage: Model performance on test ST data



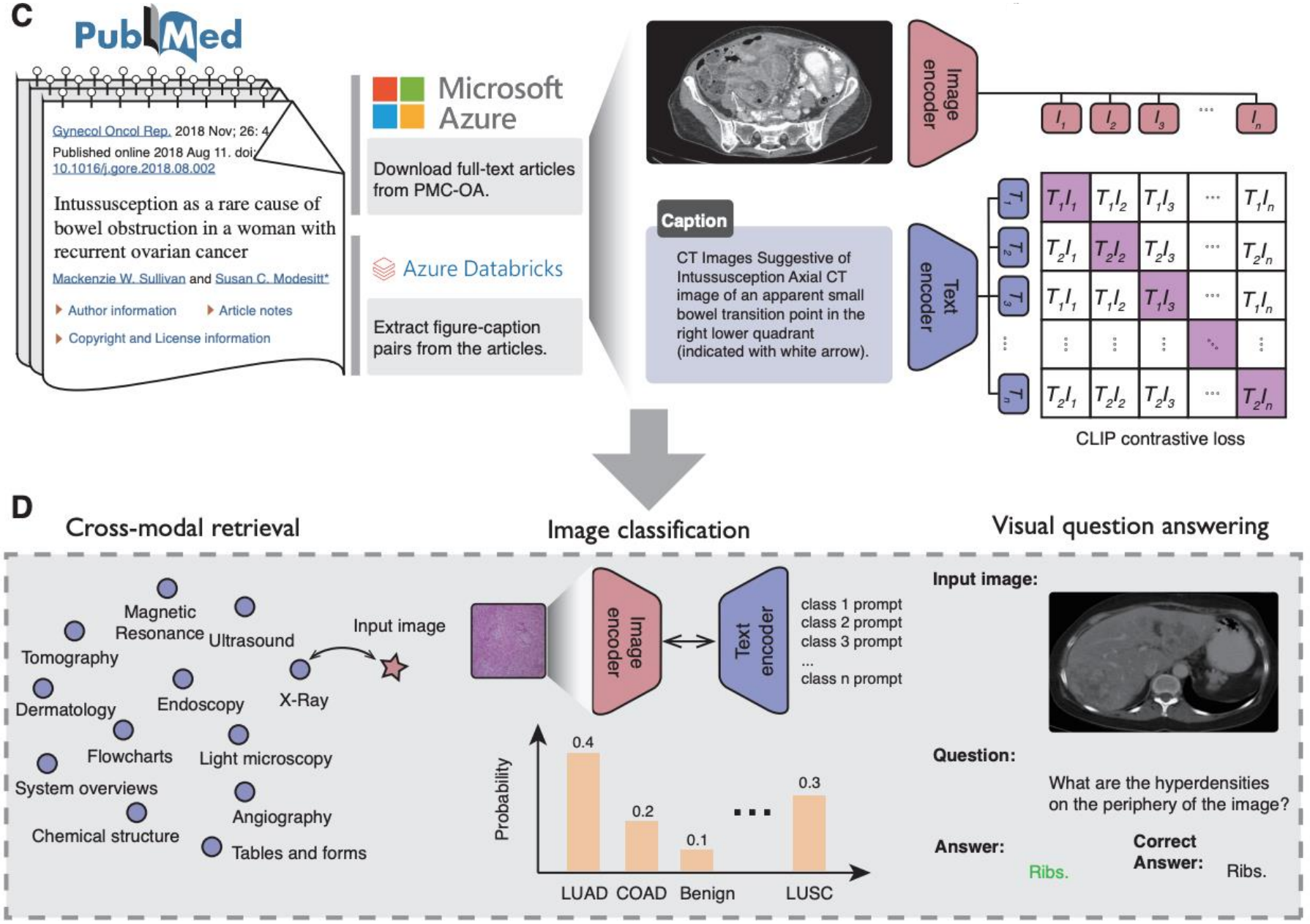
scGPT: single-cell transformer foundation model



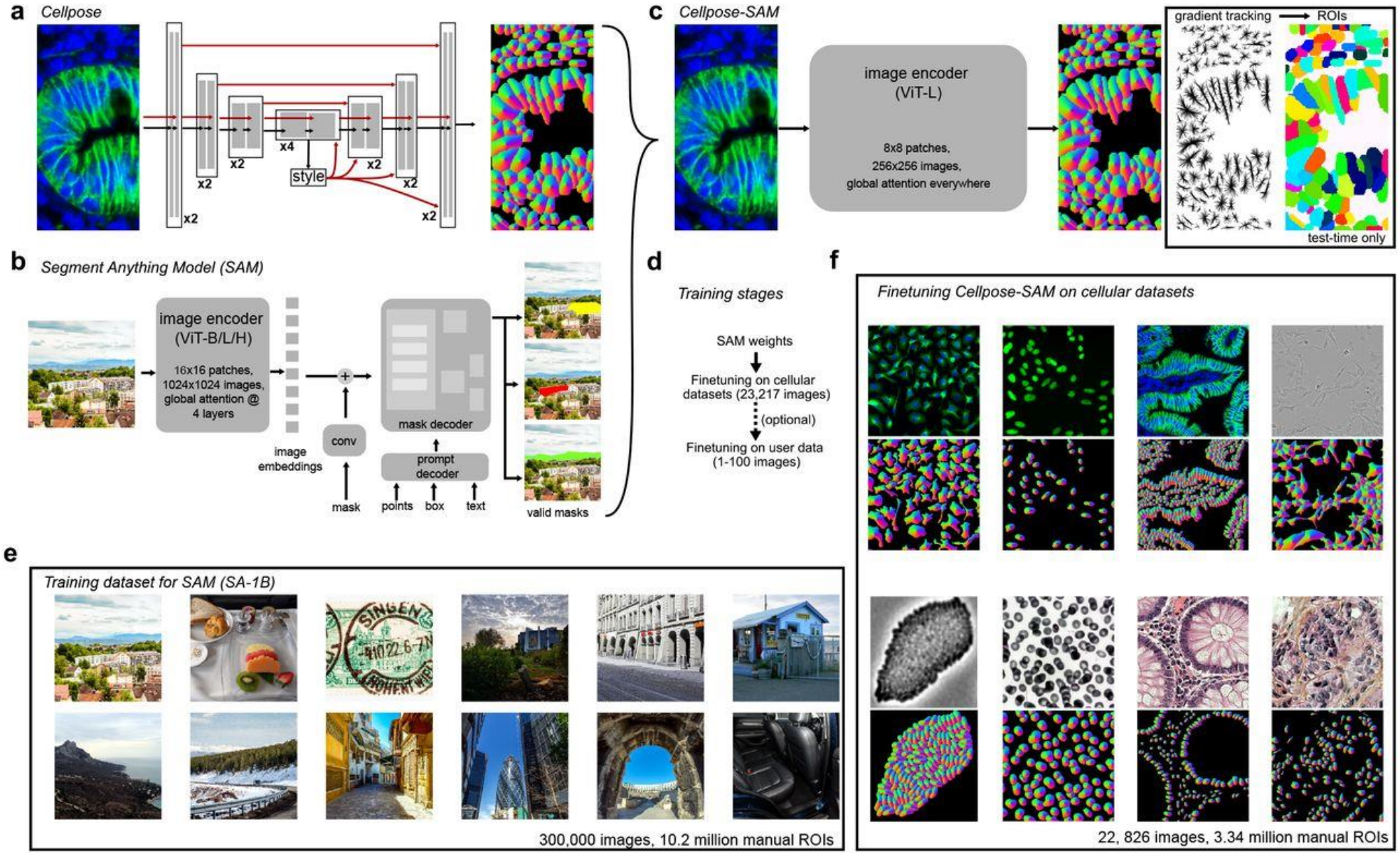
Contrastive learning



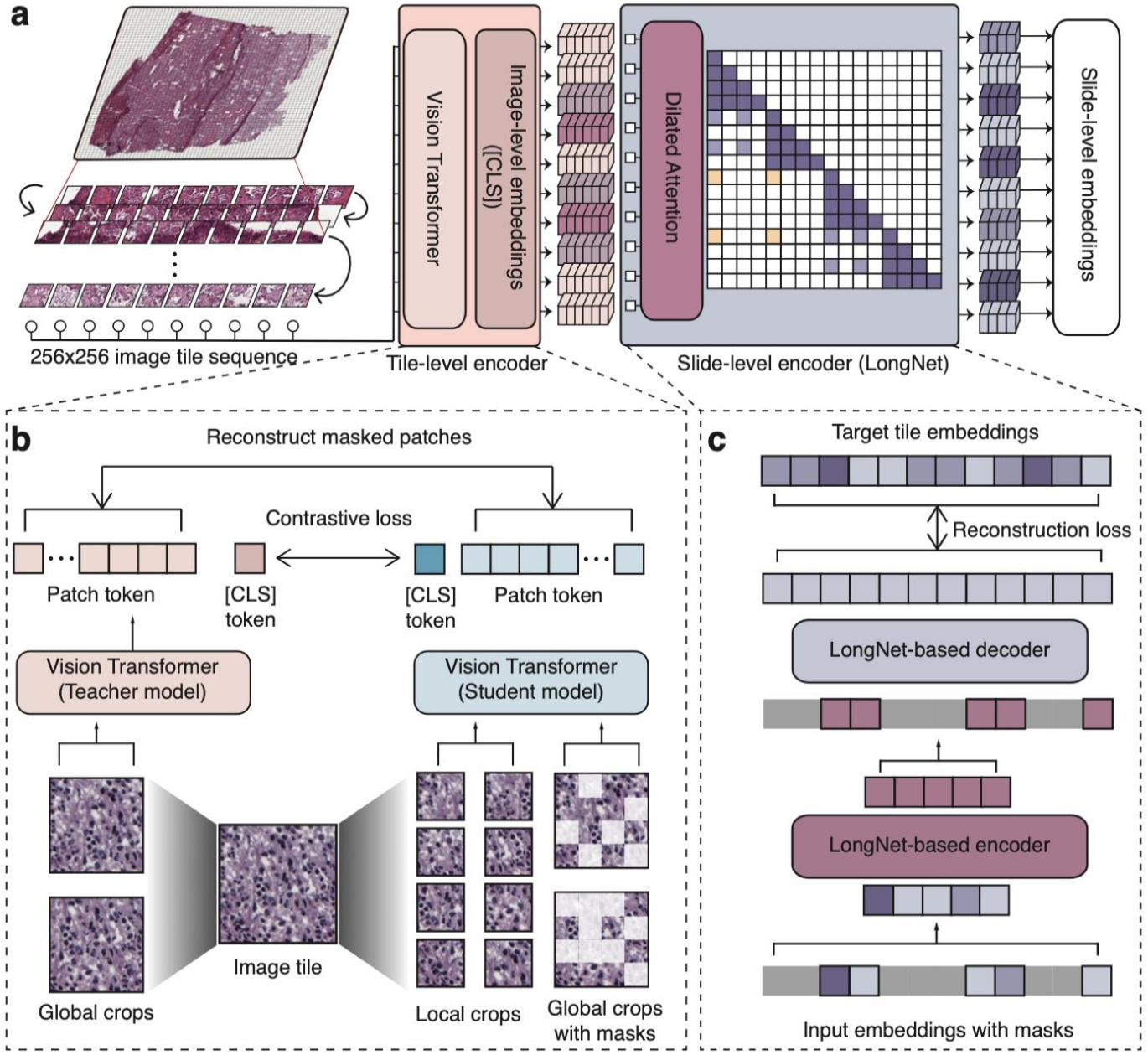
BiomedCLIP: contractive foundation model for image and text



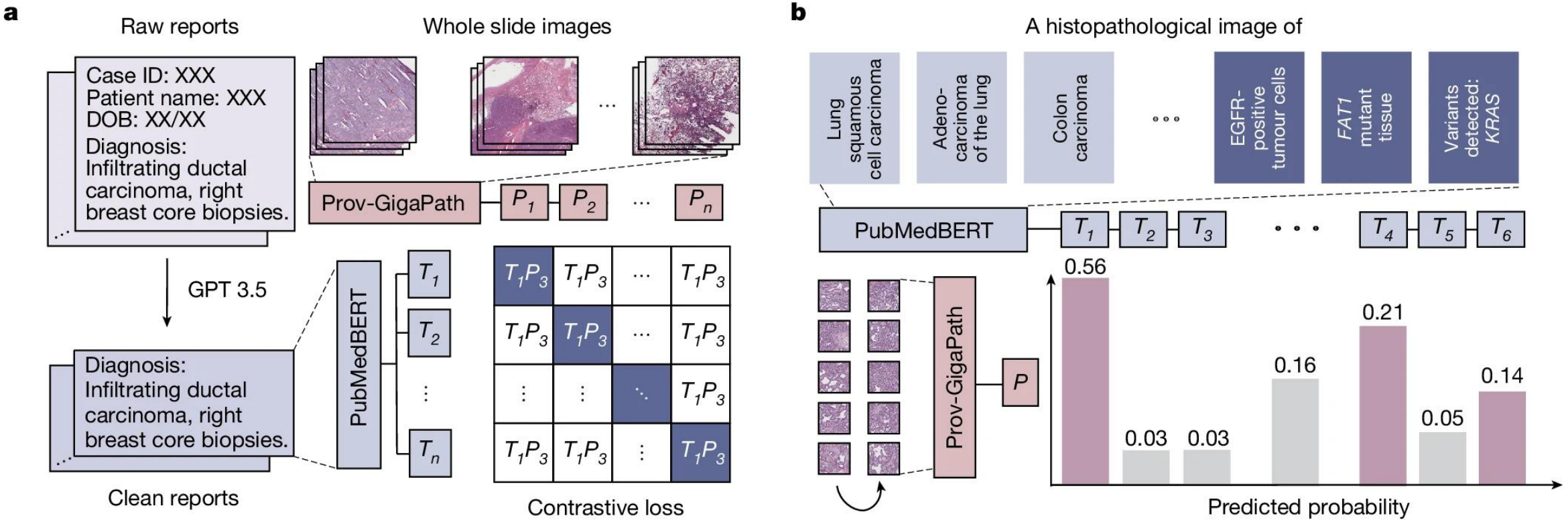
Cellpose-SAM: ViT based cell segmentation model



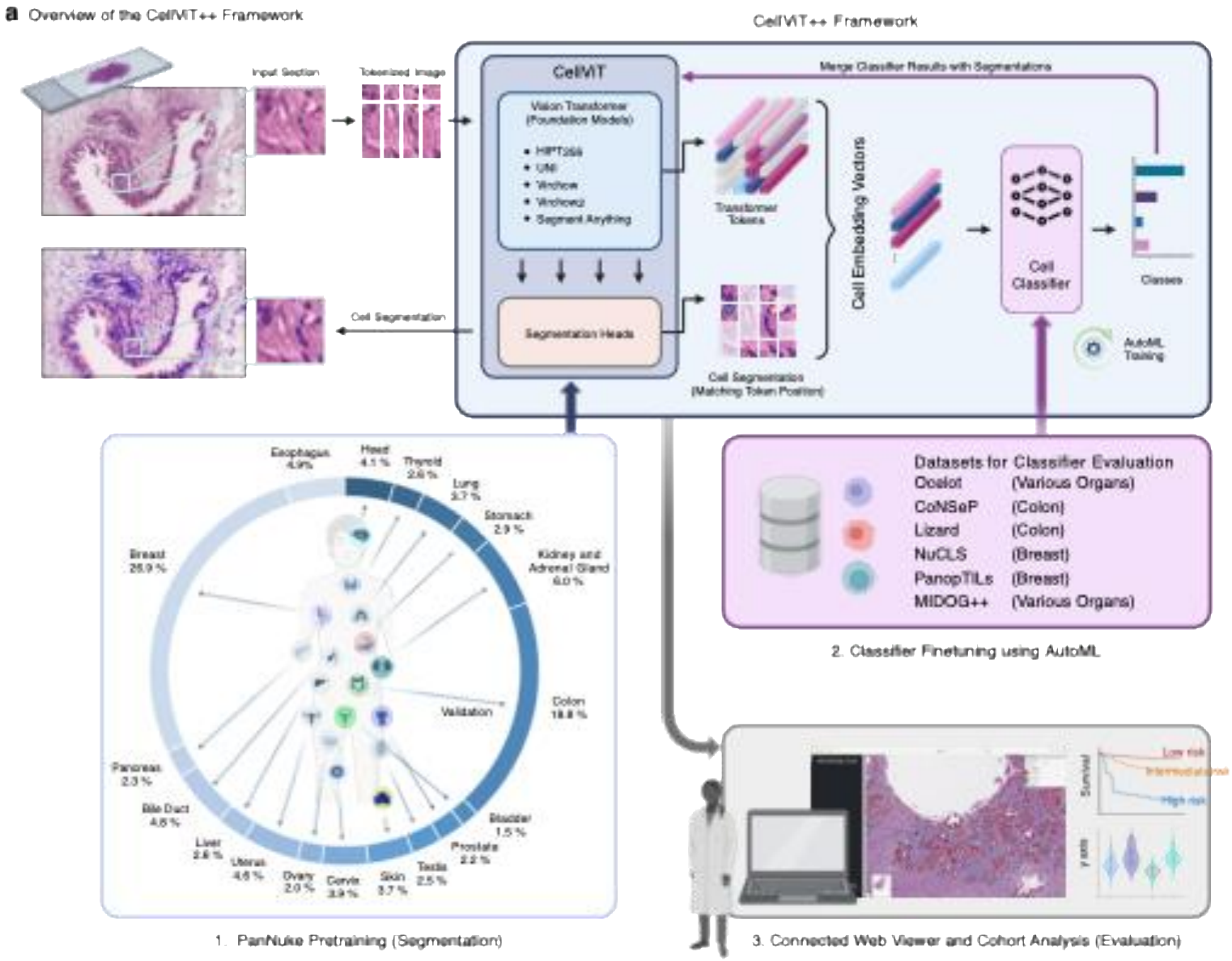
Prov-GigaPath: whole-slide image foundation model



Prov-GigaPath: whole-slide image foundation model



CellVit++: Energy-Efficient and Adaptive Cell Segmentation and Classification



Running the Practical

Terminal

PowerShell

1. Log into your account:

   ssh {username}@203.101.229.126
   username & password from winter school email

2. Follow these commands:

   - /software/bin/micromamba shell init
   - source ~/.bashrc
   - micromamba activate /software/conda-envs/winter_school_2025
   - git clone https://github.com/GenomicsMachineLearning/gml-teaching-2025
   - cd /scratch/$USER/gml-teaching-2025
   - /scratch/$USER/gml-teaching-2025/runme.sh

3. Open JuperterNotebook:

   004-deep-learning/Deep_learning_01.ipynb

Q&A