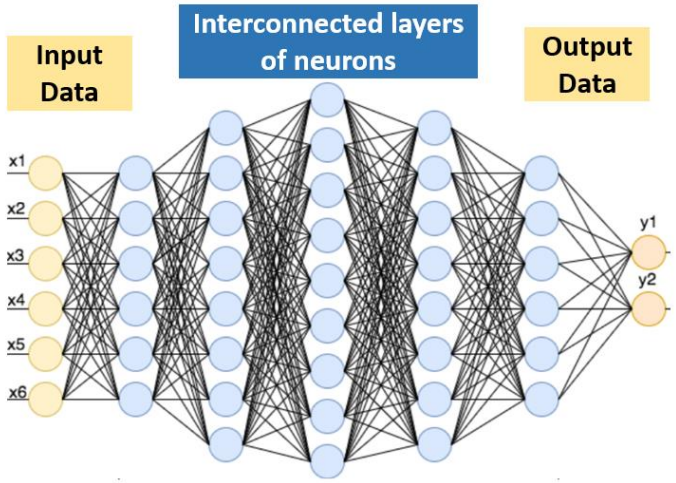
Foundation models for single-cell multi-omics using generative AI

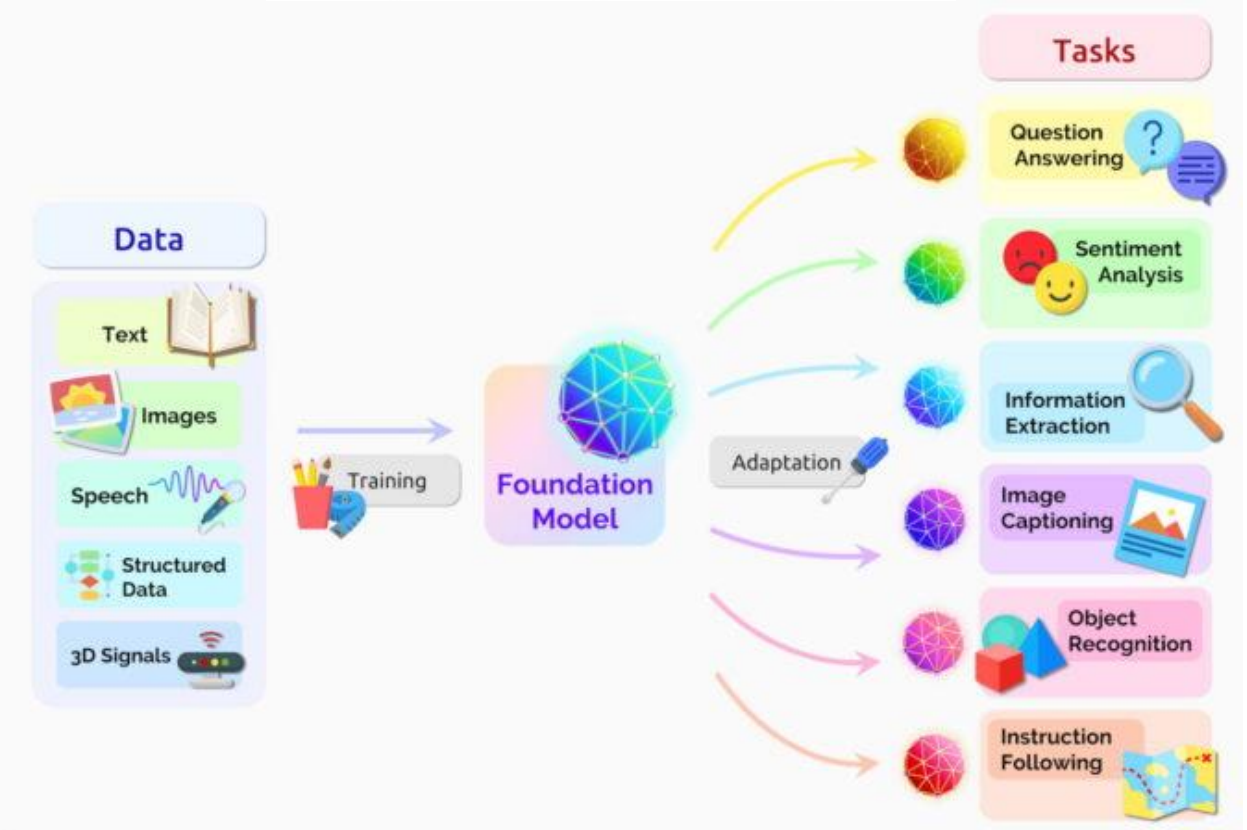
2024.03.12.

LLM/Foundation model

https://blogs.nvidia.com/blog/what-are-foundation-models/

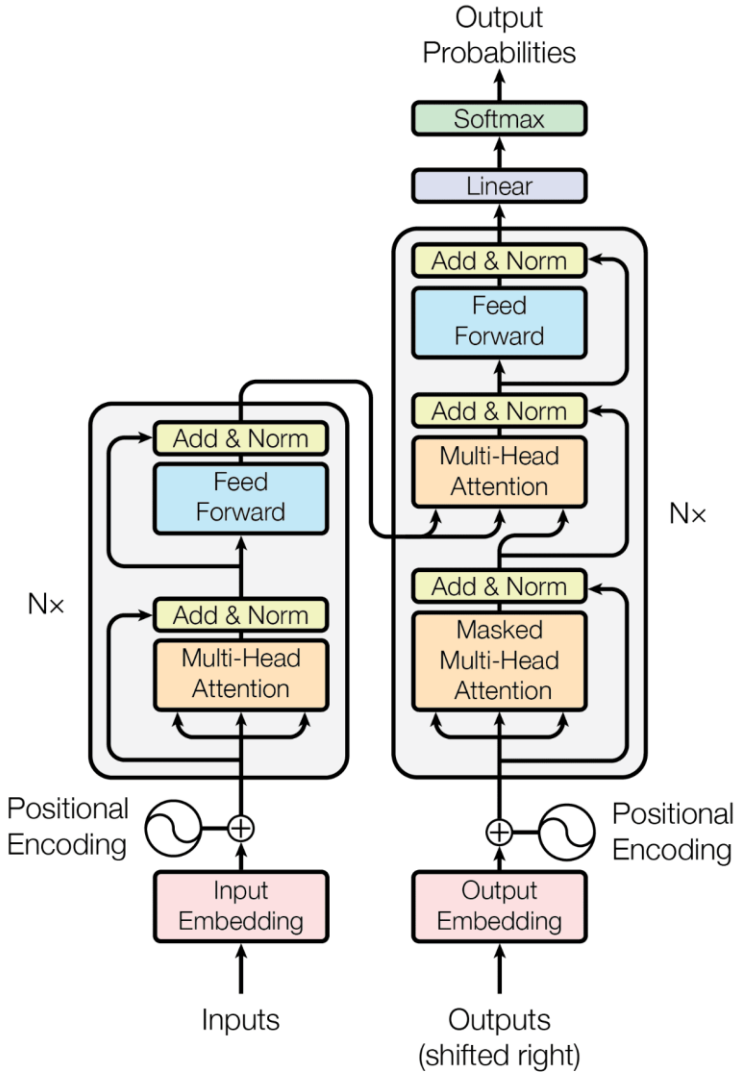
- A foundation model is an AI neural network — trained on mountains of raw data, generally with **unsupervised learning** — that can be adapted to accomplish a broad range of tasks
- **Finetuning** for diverse downstream tasks





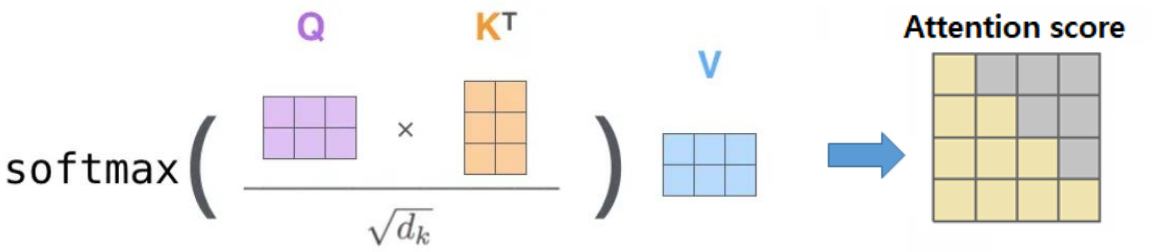
Transformers

- Attention is all you need!
- Dynamic representation for context-specific information
- For example:
  - Eat the sandwich
  - Don’t eat the sandwich
- Connections and dependecies among the words (biology)
- Self-attention mechanism
- Encoder and decoder
- Multihead attention

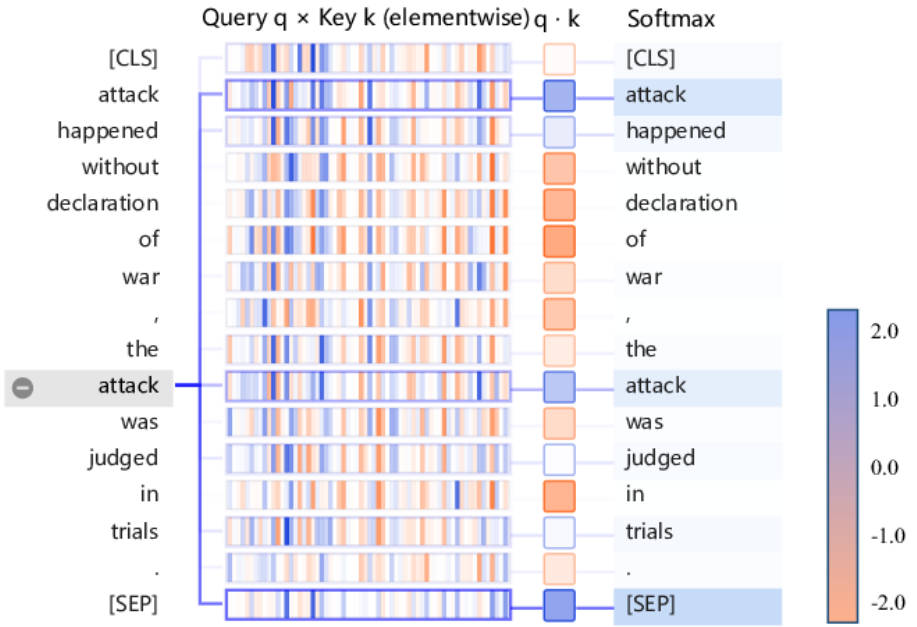


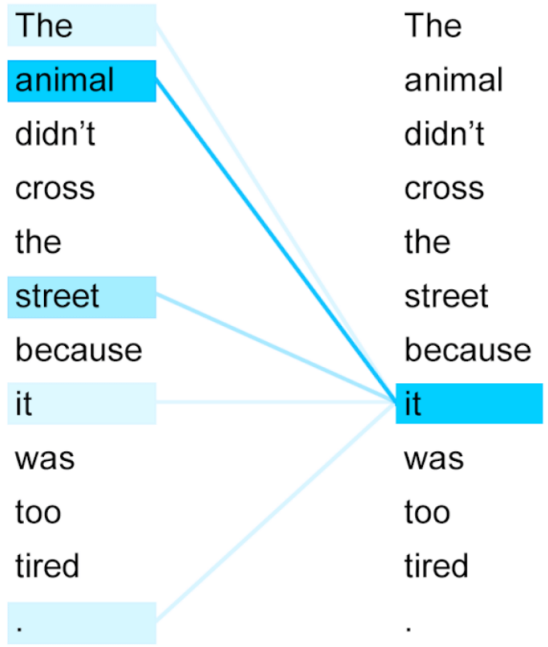
Attention

- Attention allows models to focus on different parts of input sequence
- Scaled dot-product attention
- Mapping between keys and querries between tokens



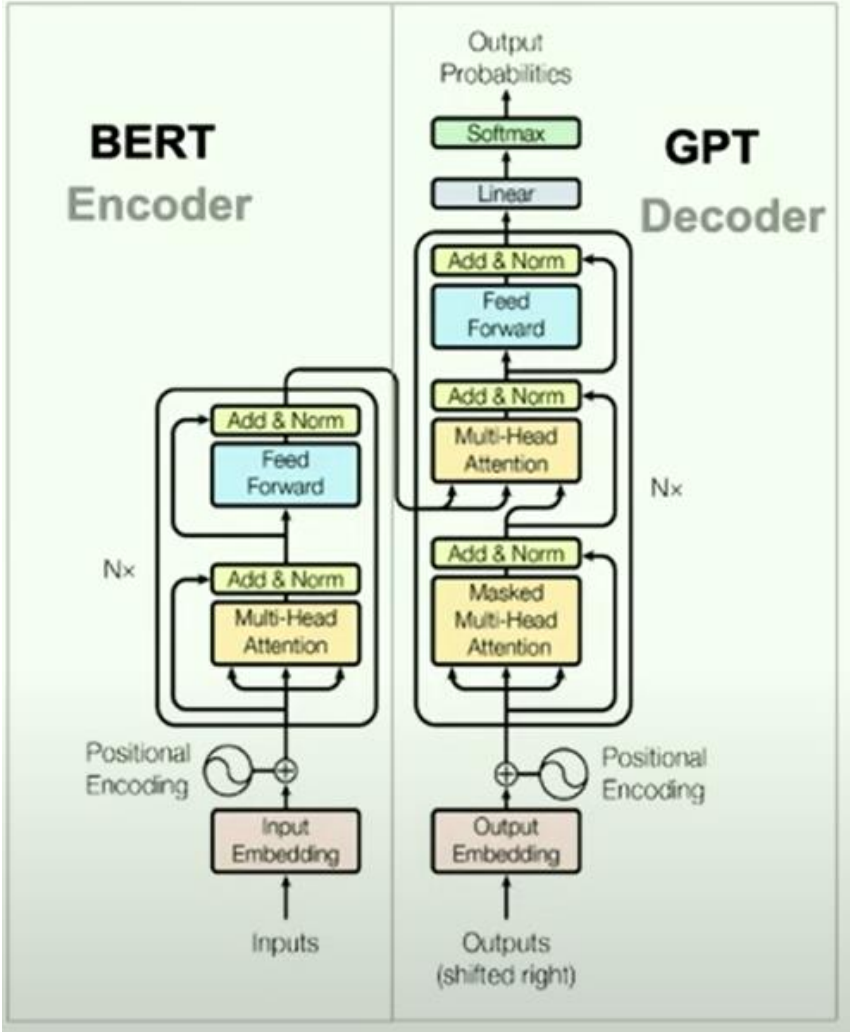
- Q: current positions in input seeking context from other positions
- K: captures the information that Q attends to
- V: actual content assotiated with positions in input





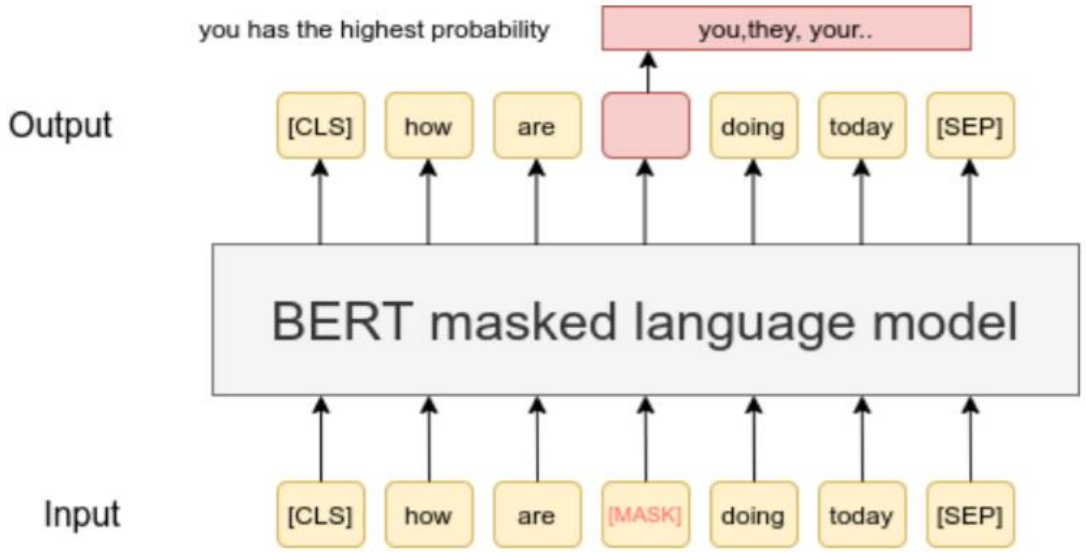
Transformer architecture in BERT and GPT

- BERT-style
- (Encoder-only or Encoder-Decoder)
- Training: Masked Language Model
- Pretrain task is to predict the masked word

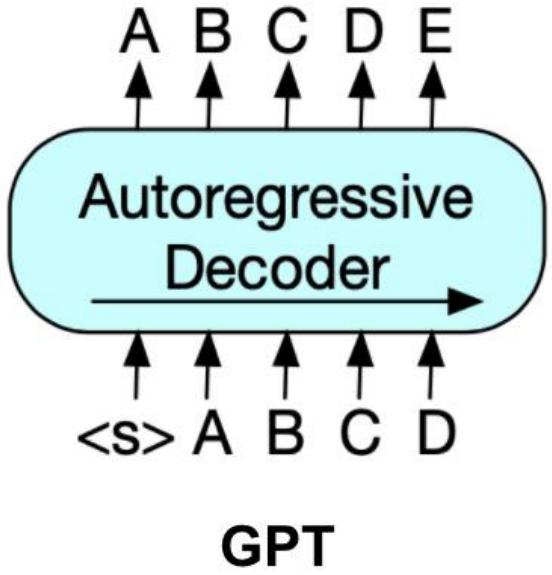


- GPT-style
- (Decoder only)
- Autoregressive model (predict the next word based on prior words)
- Pretrain task is to predict the next word

Bert


you has the highest probability
you,they, your..
Output
[CLS]
how
are
doing
today
[SEP]
BERT masked language model
Input
[CLS]
how
are
[MASK]
doing
today
[SEP]

GPT


A B C D E
Autoregressive Decoder
<s> A B C D
GPT

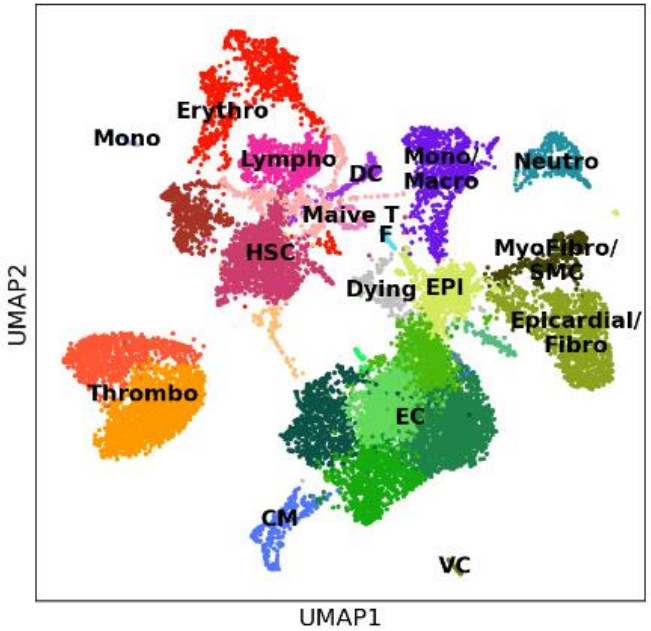
Application in genomics

- Large pretrained model for finetuning on downstream tasks with limited training data
- Input:
  - DNA sequence (DNABERT)
  - Single cell data (scGPT)

Single cell transcriptomics

- RNA counts of genes in cells

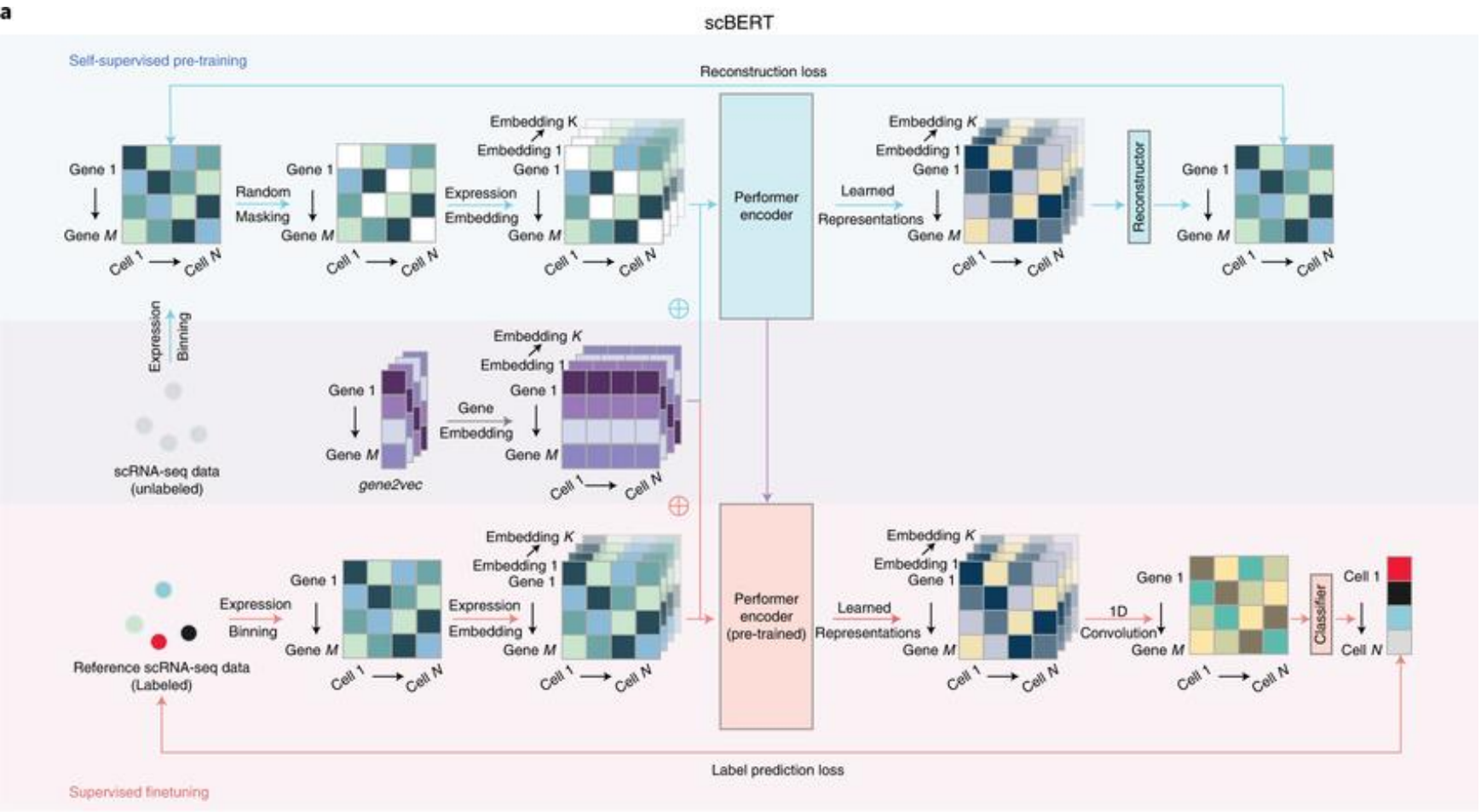
	Cell1	Cell2	...	CellN
Gene1	3	2	.	13
Gene2	2	3	.	1
Gene3	1	14	.	18
...	.	.	.	.
...	.	.	.	.
...	.	.	.	.
GeneM	25	0	.	0



Models based on single cell data

- scBERT
- Geneformer
- scGPT

scBERT

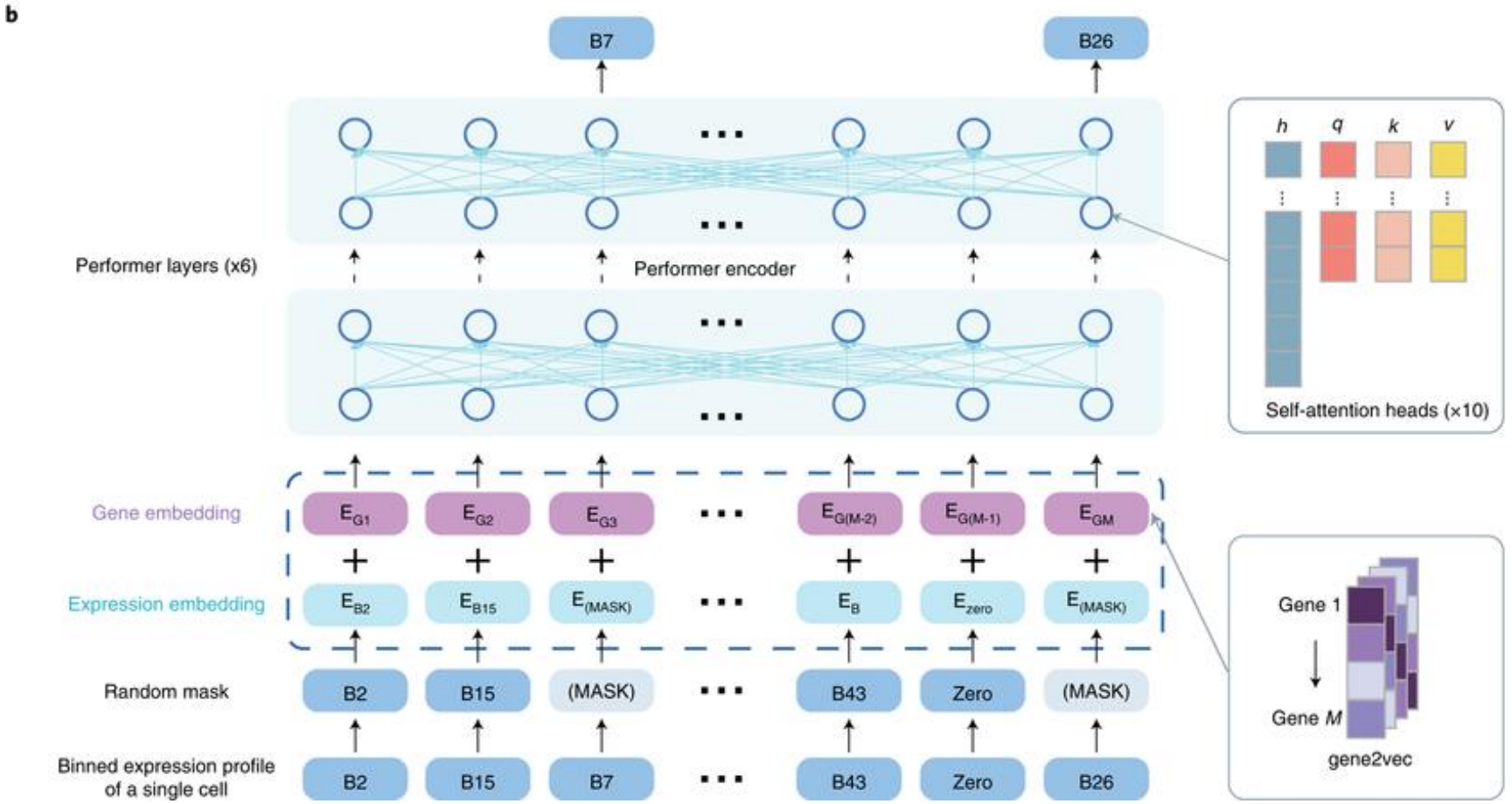


- Expression embedding
- Capture intrinsic properties of gene-gene dependencies (gene2vec)
- Pretrained model finetuning: for example cell type classification

- Tokenization: cell as sentence, genes as words
- Represent expression? : binning expression values to for example very high, medium…

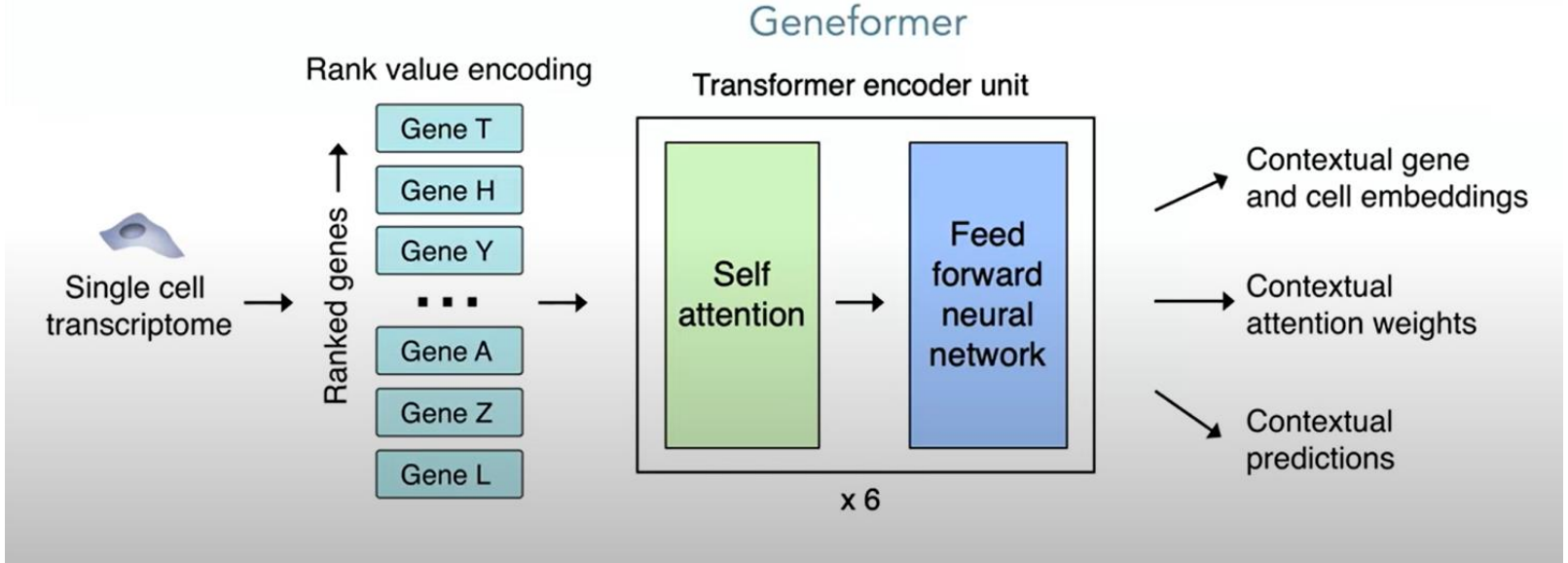
scBERT training

- Training: try to predict expression of masked genes based on the rest expressions of genes



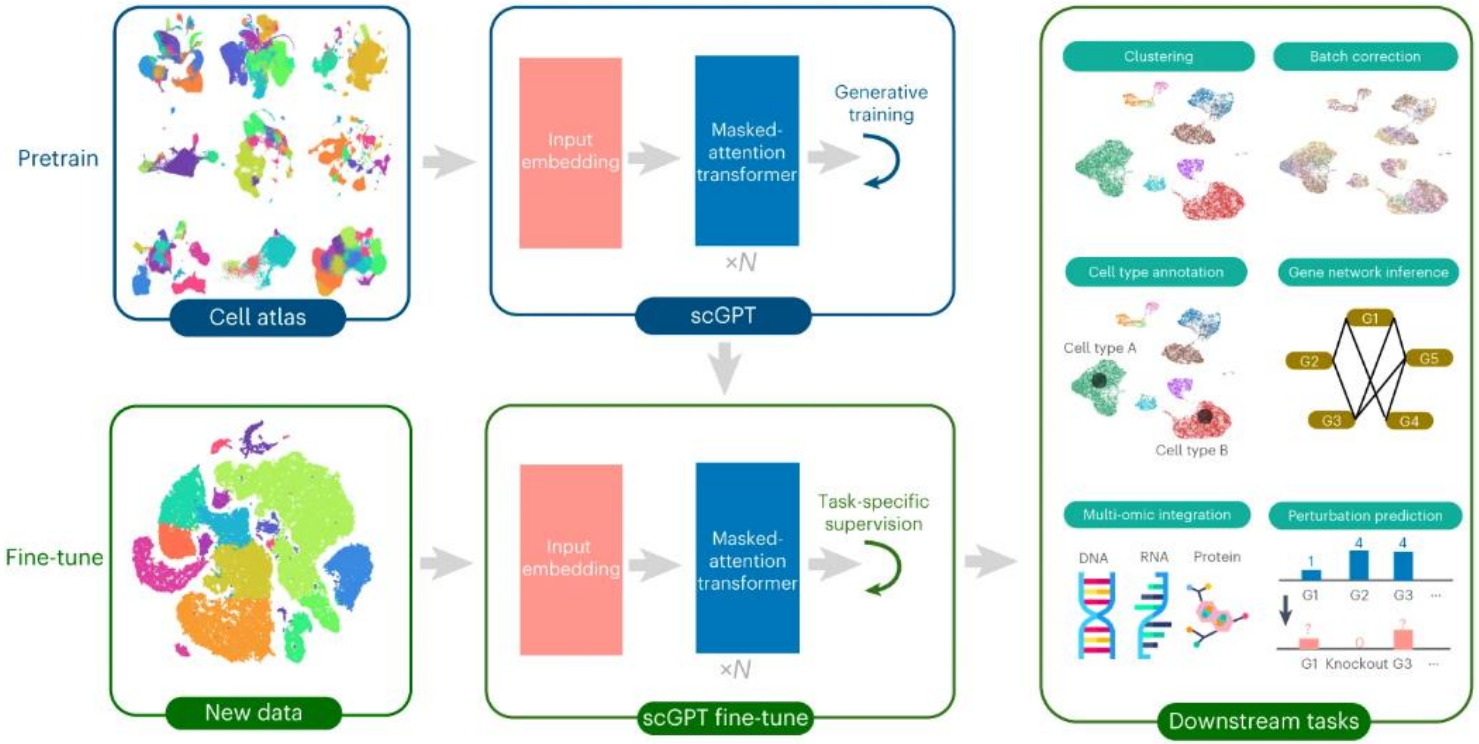
Geneformer

- Pretrained 30 million scRNA.seq
- Tokenization
  - Gene exression: discretized by ranking genes according to their expression
  - Normalized by gene rank of other cells
  - Particular single cell: comparing gene expression compared to the rest of the single cells – relative ranking high (opposite house keeping genes)
- Attention weights: central genes in gene regulatory networks higher weights
- In silico perturbation: remove gene and compare cell and gene embedding (drastically different – high effect)



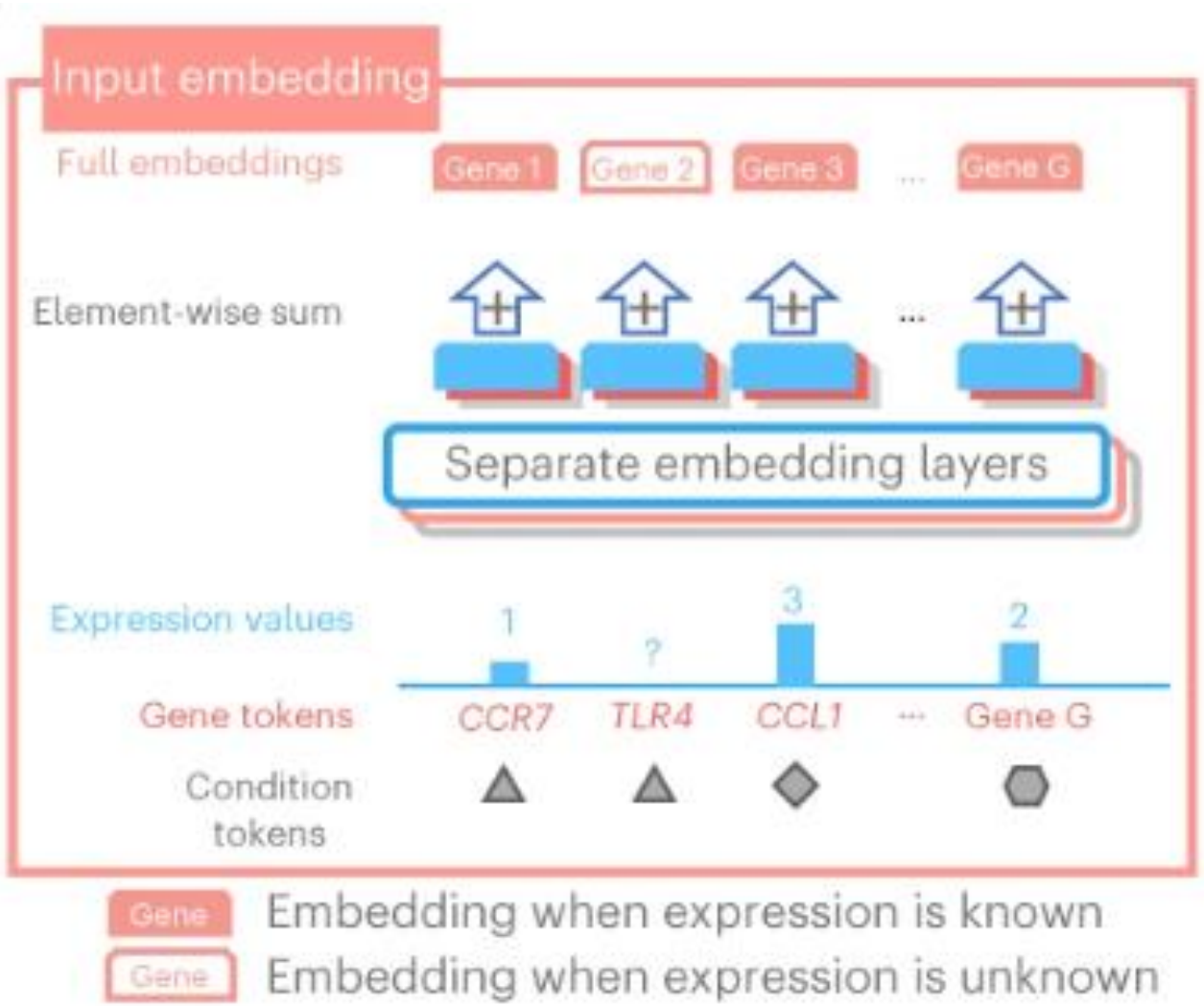
scGPT

- CZ CELLxGENE: 33M millio normal human cells drom 50+ tissue
- It can be adapted to specific tasks:
  - Cell-type annotation
  - Multi-batch correction
  - Preturbation prediction



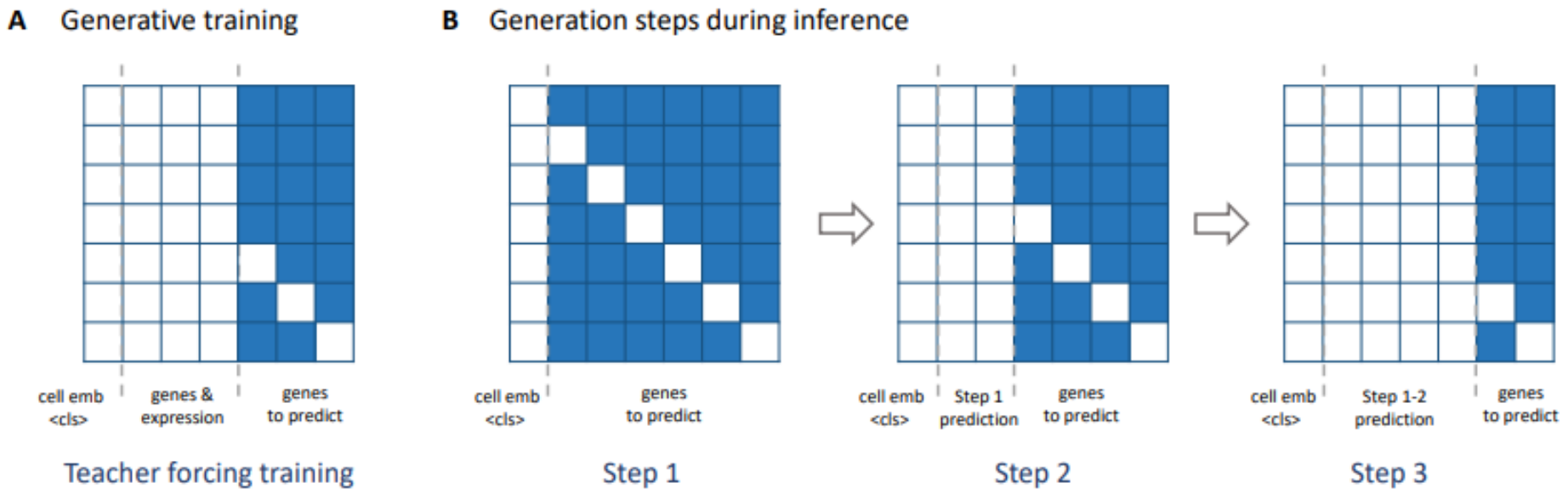
How to represent each gene?

- Tokenization: additional tokens for meta information (e.g., perturbation)
- Gene expression discretized
- Gene token
- Condition token



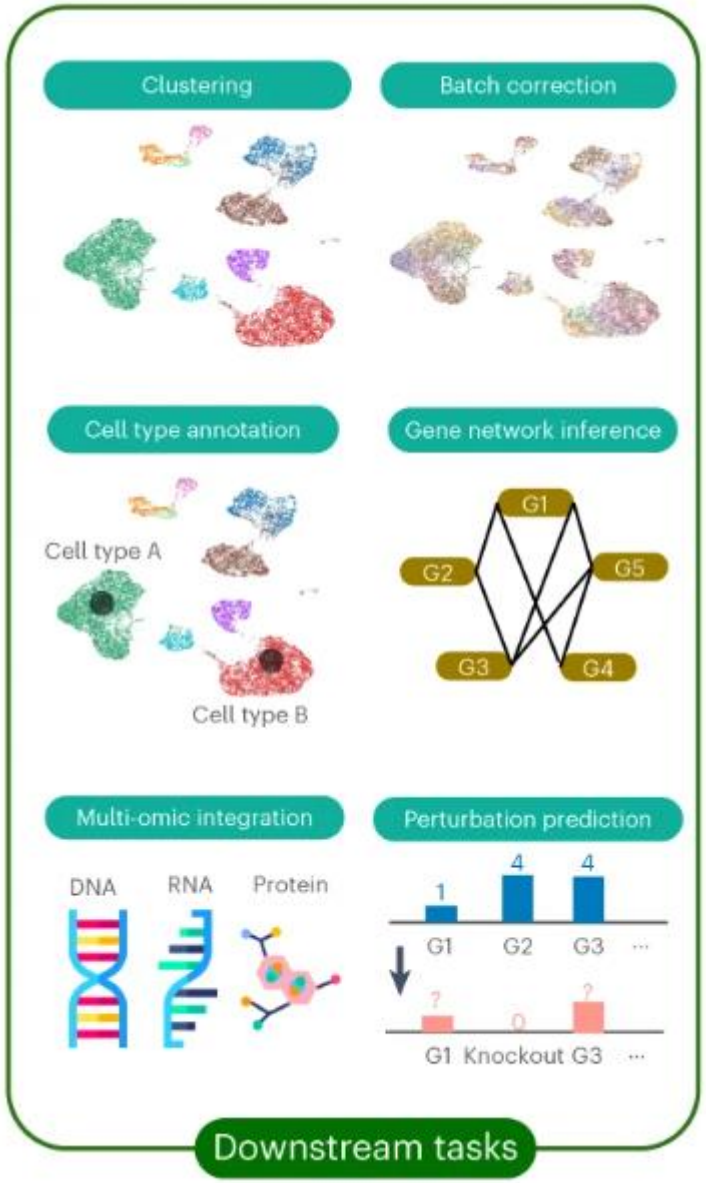
Generative pretraining

- Autoregressive, try to use the information which were seen before to predict the next „word”
- Try to predict unkown genes from the known genes iteratively – capture gene dependencies
- Genes were predicted with high confidence will be added to the known gene list



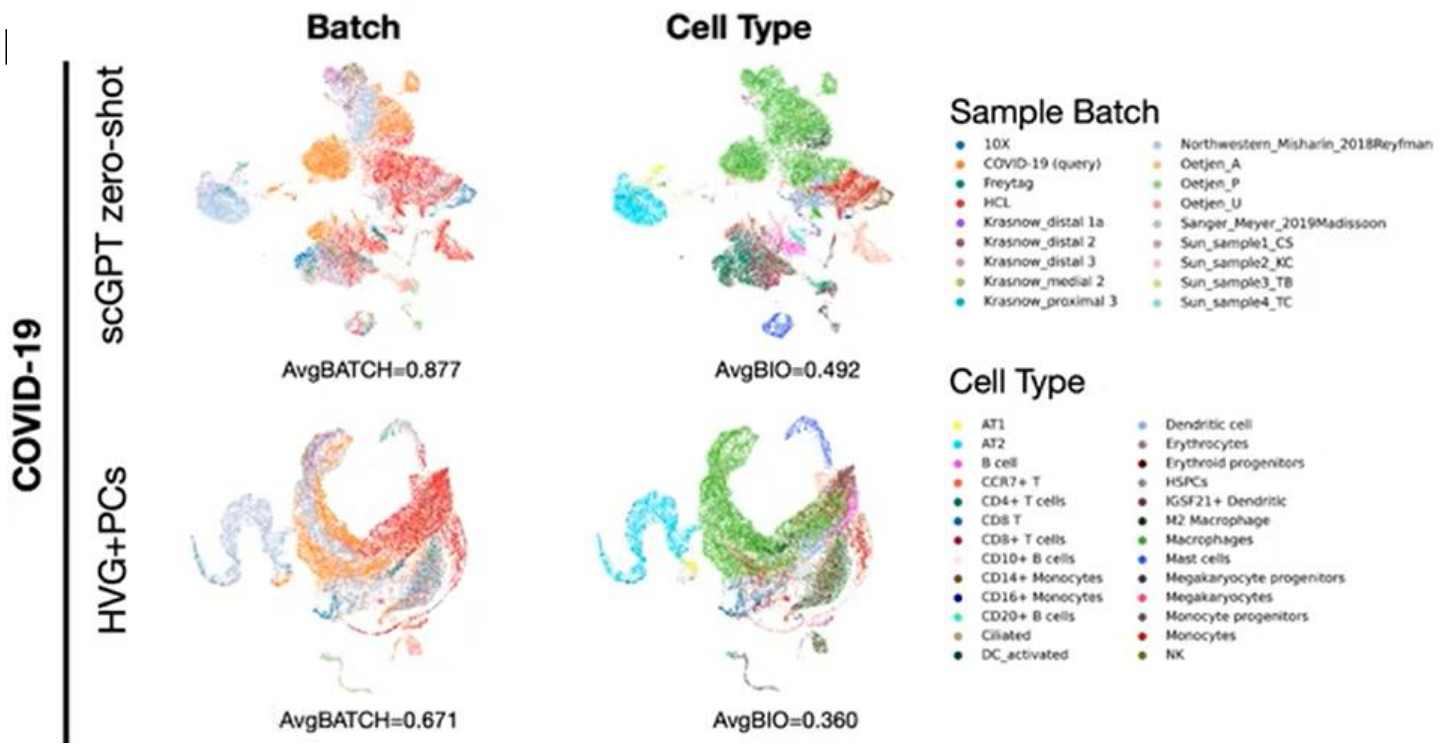
Finetuning

- Fine-tuning: learning of meaningful cell and gene representation
- Self-supervised:
  - Gene expression prediction
  - Data integration
- Supervised:
  - Cell annotation



Zero-shot application

- Zero-shot: using the pretrained model to generate embedding fo the new data (genes, cells), without further training – fast, accesibble
- No need fine-tuning, lots of GPU
- For example: UMAP



Comparison the model

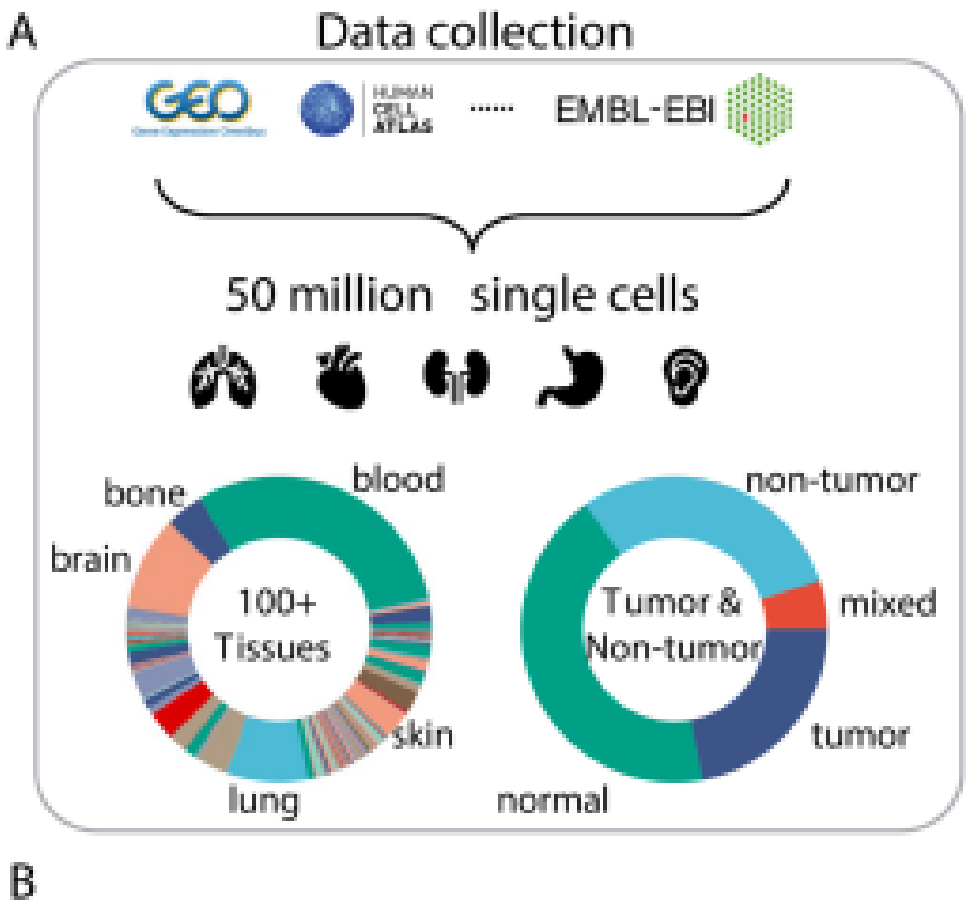
- Similar performance

Dataset	Model	Classification Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
Myeloid	scGPT (fine-tuned)	**0.642**	**0.366**	**0.347**	**0.346**
	scGPT (from-scratch)	0.606	0.304	0.339	0.309
	TOSICA	0.488	0.316	0.276	0.275
	scBert	0.525	0.331	0.323	0.298
Multiple Sclerosis	scGPT (fine-tuned)	**0.856**	**0.729**	**0.720**	**0.703**
	scGPT (from-scratch)	0.798	0.660	0.623	0.600
	scBert	0.785	0.604	0.624	0.599
	TOSICA	0.758	0.664	0.585	0.578
hPancreas	scGPT (fine-tuned)	**0.968**	**0.735**	**0.725**	**0.718**
	scGPT (from-scratch)	0.936	0.665	0.668	0.622
	TOSICA	0.960	0.661	0.681	0.656
	scBert	0.964	0.699	0.689	0.685

- Thank you for the attention!

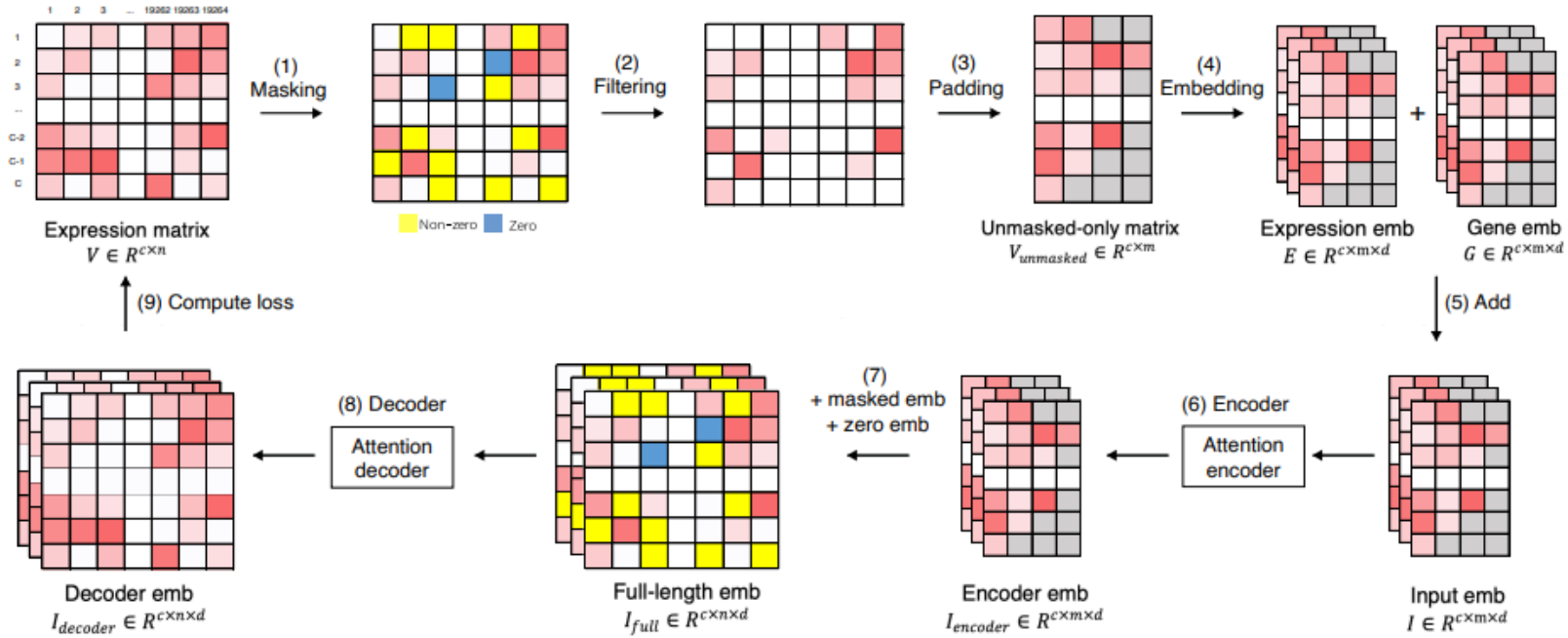
scFoundation

- large-scale pretrained model with 100M parameters
- 50 million human single-cell transcriptomics data
- Downstream tasks, such as gene expression enhancement, tissue drug response prediction, single-cell drug response classification, and single-cell perturbation prediction



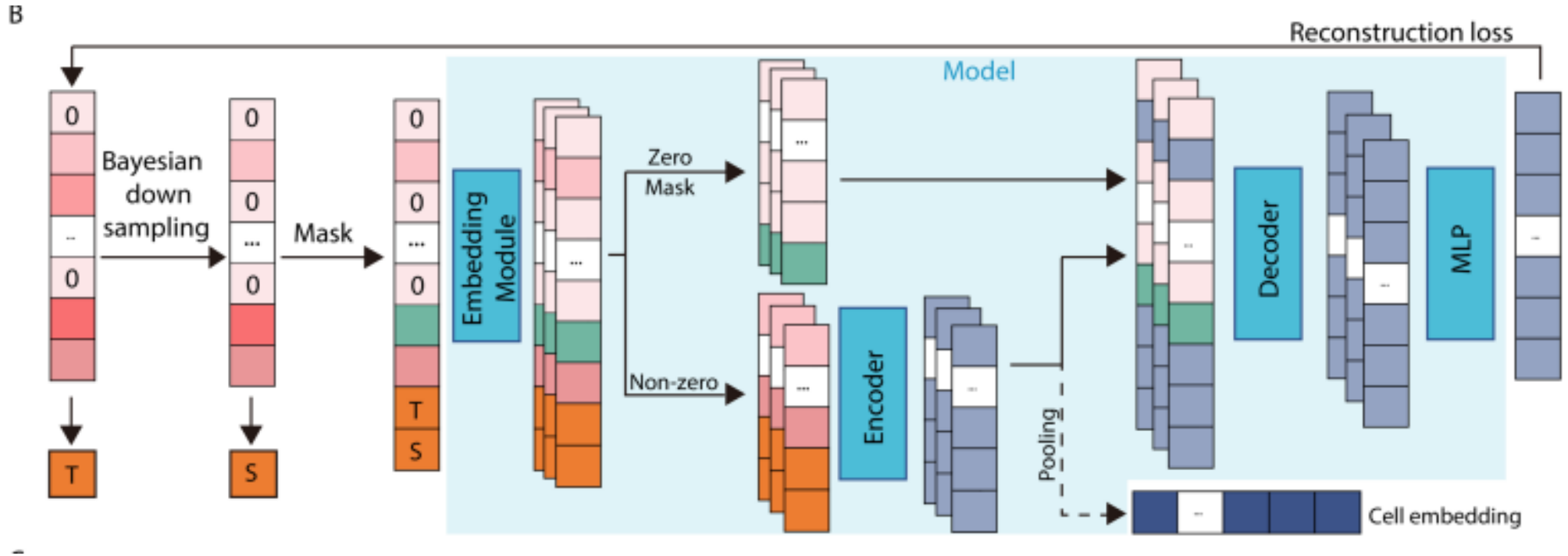
xTrimoGene

- scalable transformer-based model
- an embedding module and an asymmetric encoder-decoder structure
- **embedding module converted continuous gene expression scalars into learnable highdimensional vectors**
- asymmetric encoder-decoder architecture was specifically designed to accommodate the high sparsity characteristics of single-cell gene expression data
- **encoder only accepted embeddings of the non-zero and non-masked expressed genes as input**, and the **decoder accepted all genes' embedding**
- architecture gave **differential attention and computational resources to zero and non-zero values** - the **efficient learning of all gene relationships without any selection**



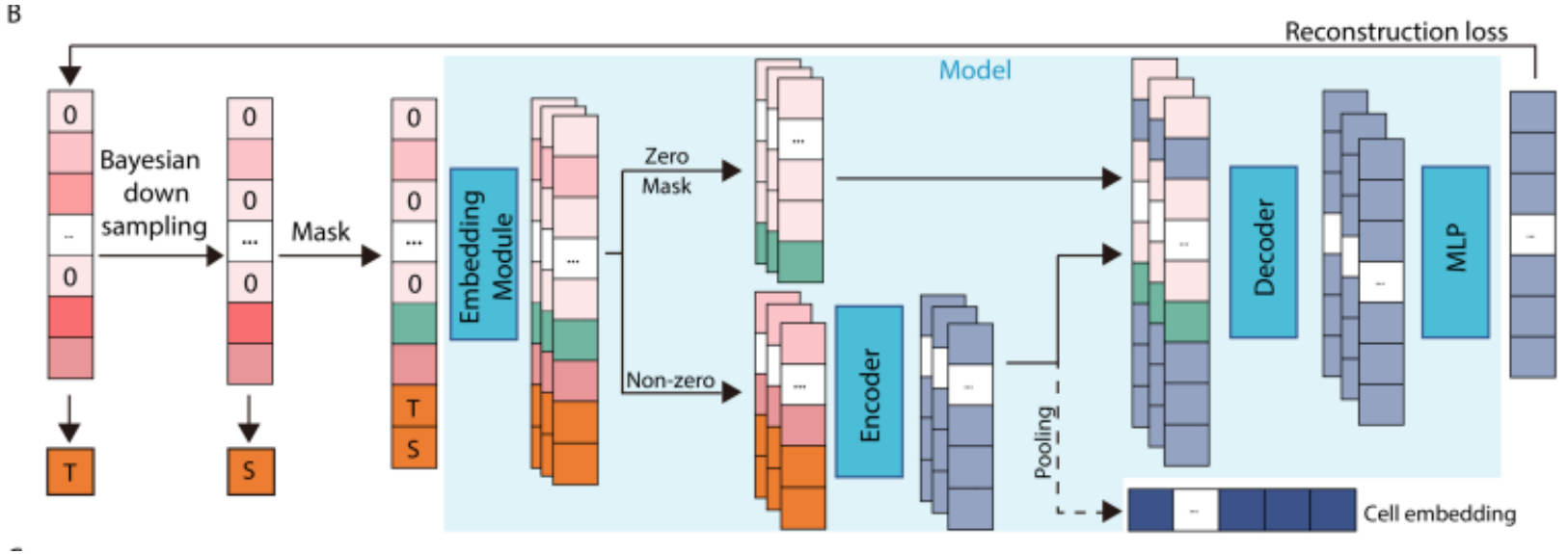
Pre-training

- New pre-training task called the read-depth-aware (RDA) modeling - to predict the masked gene expression of a cell based on other genes' context
- Processed a **raw gene expression training sample** with a **hierarchical Bayesian downsampling strategy to generate an input sample** with an unchanged or altered total count
- Two total count indicators: T (representing 'target') and S (representing 'source'), corresponding to the total counts of the raw and input samples respectively
- Read depth adaptation T to S: downstream analysis is better
- Encoder: just non-zero expression genes

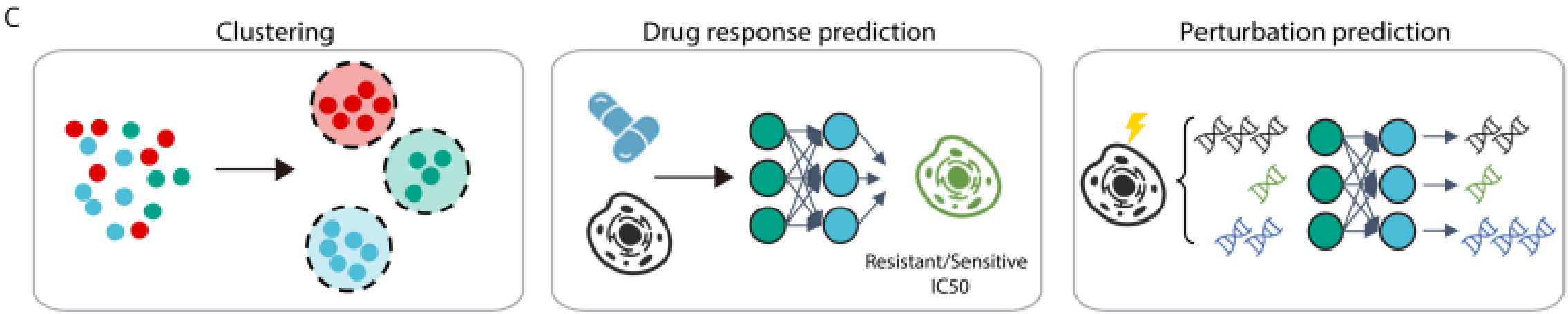


Pre-training

- Values in the input sample are randomly masked
- The scalar values are converted into embeddings - non-zero and non-masked values (including T and S) are fed into the model encoder
- The output embeddings of the encoder are then combined with mask and zero embeddings and fed into the decoder
- The decoder output embeddings are projected to the gene expression value via a shared multilayer perceptron (MLP) layer
- The regression loss between the predicted and raw sample's gene expression values is computed



Downsteam tasks

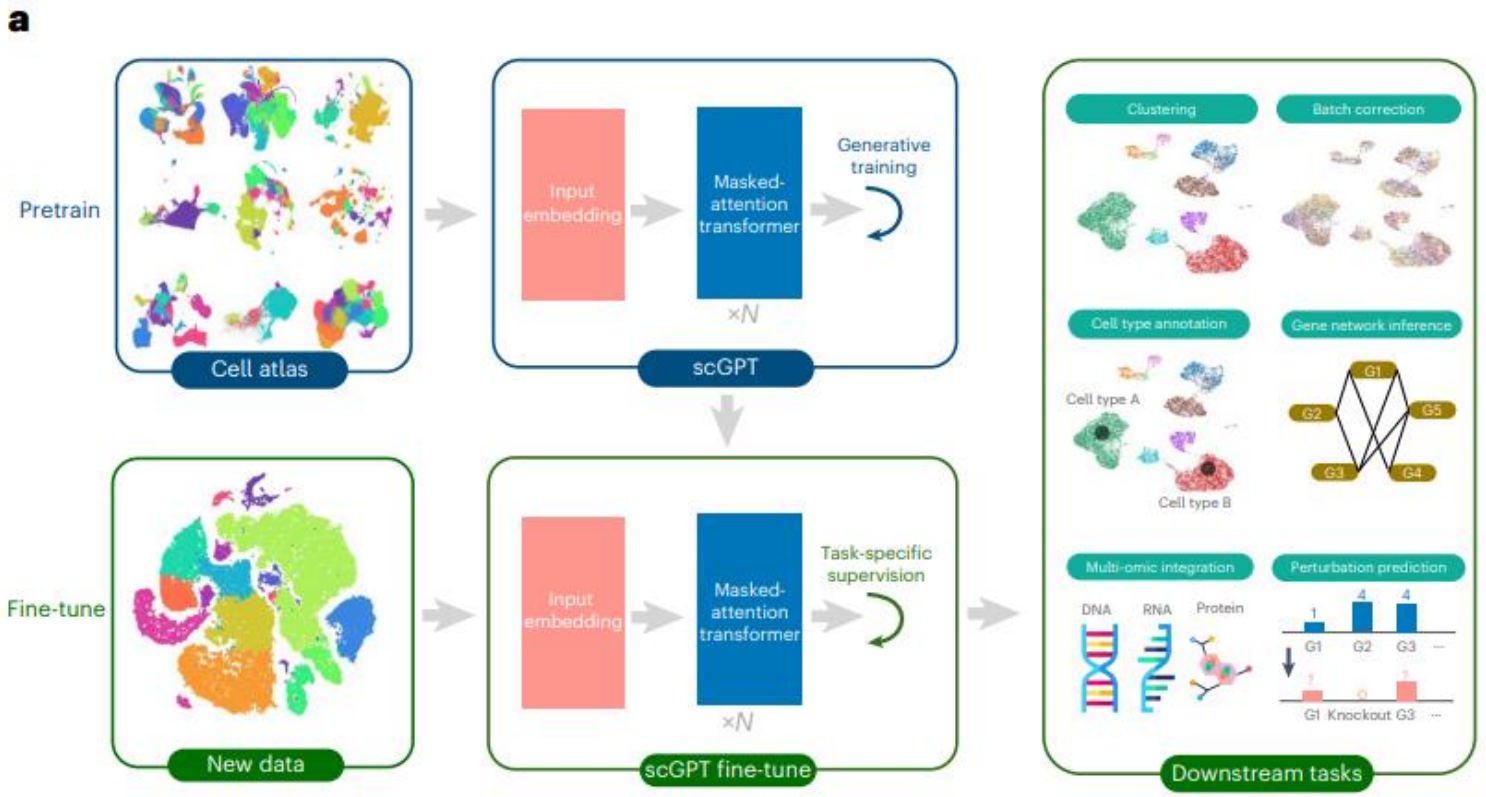


Result comparison

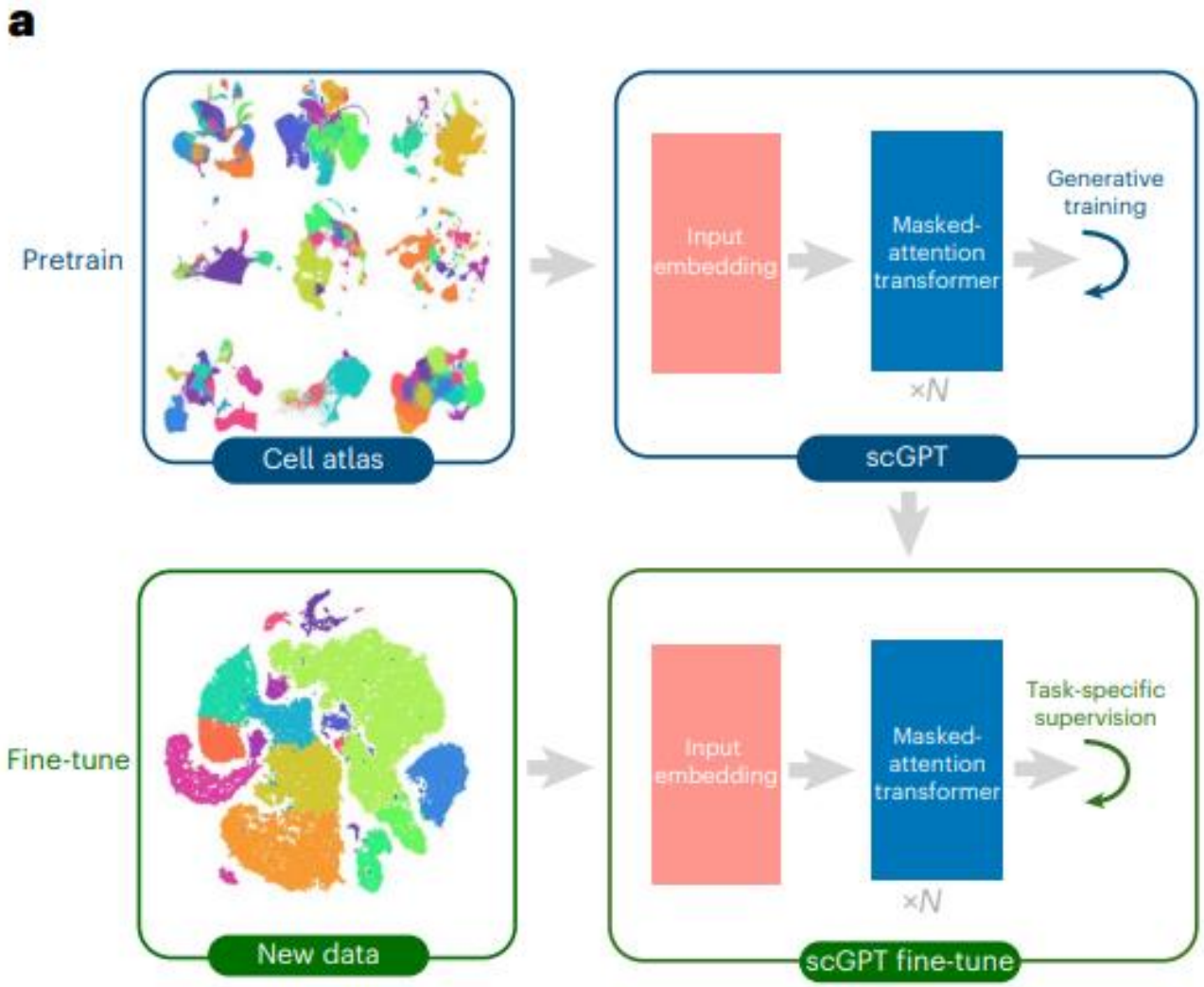
- The improvement over existing models is moderate

Single-cell transformer foundation model overview

- The core model contains stacked transformer layers with multi-head attention **that generate cell and gene embeddings simultaneously**
- scGPT consists of two training stages: **initial general-purpose pretraining on large cell atlases** and **follow-up fine-tuning** on smaller datasets for specific applications

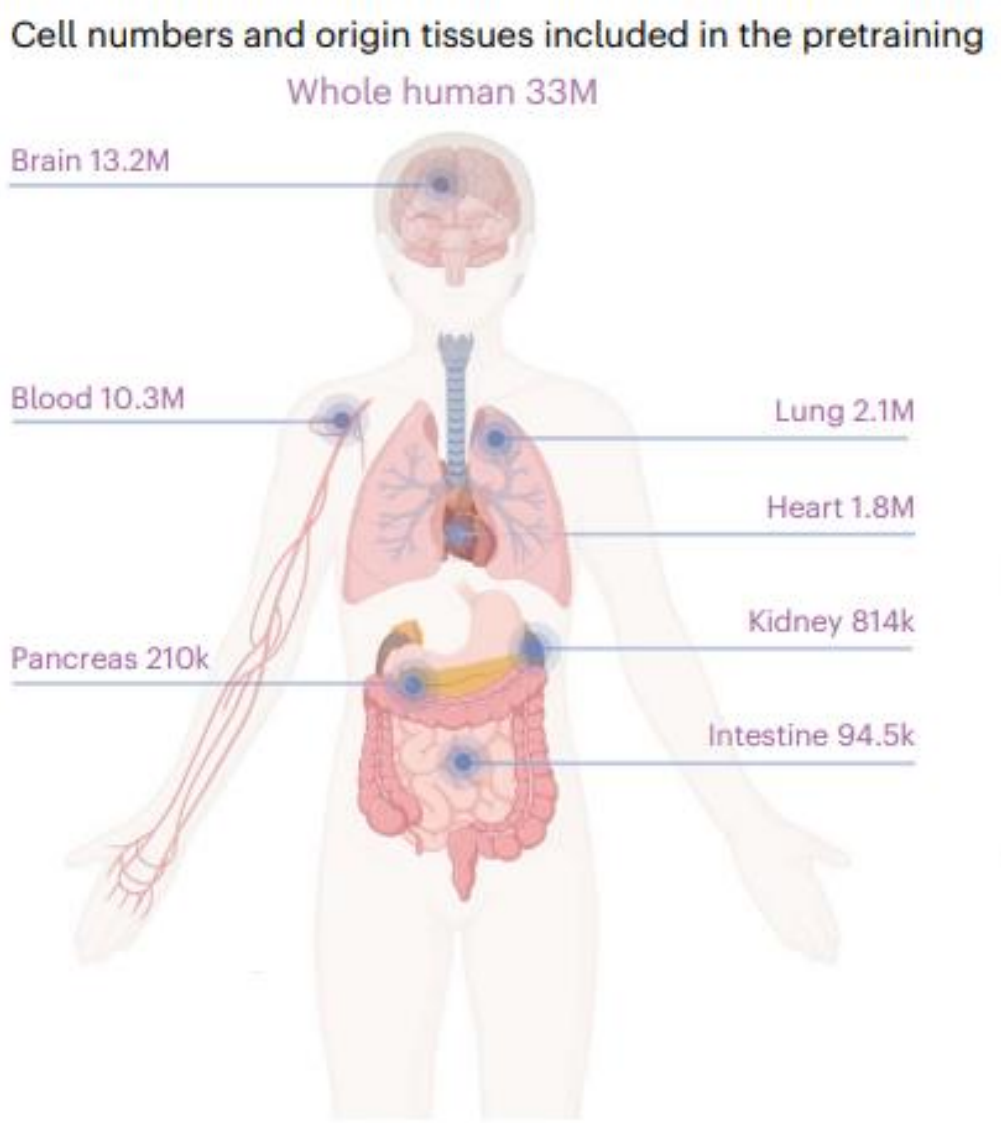


- In the pretraining stage, we introduce a specially designed attention mask and generative training pipeline to train scGPT in a self-supervised manner to jointly optimize cell and gene representations
- During training, the model gradually learns to generate gene expression of cells based on cell states or gene expression cues
- In the fine-tuning stage, the pretrained model can be adapted to new datasets and specific tasks

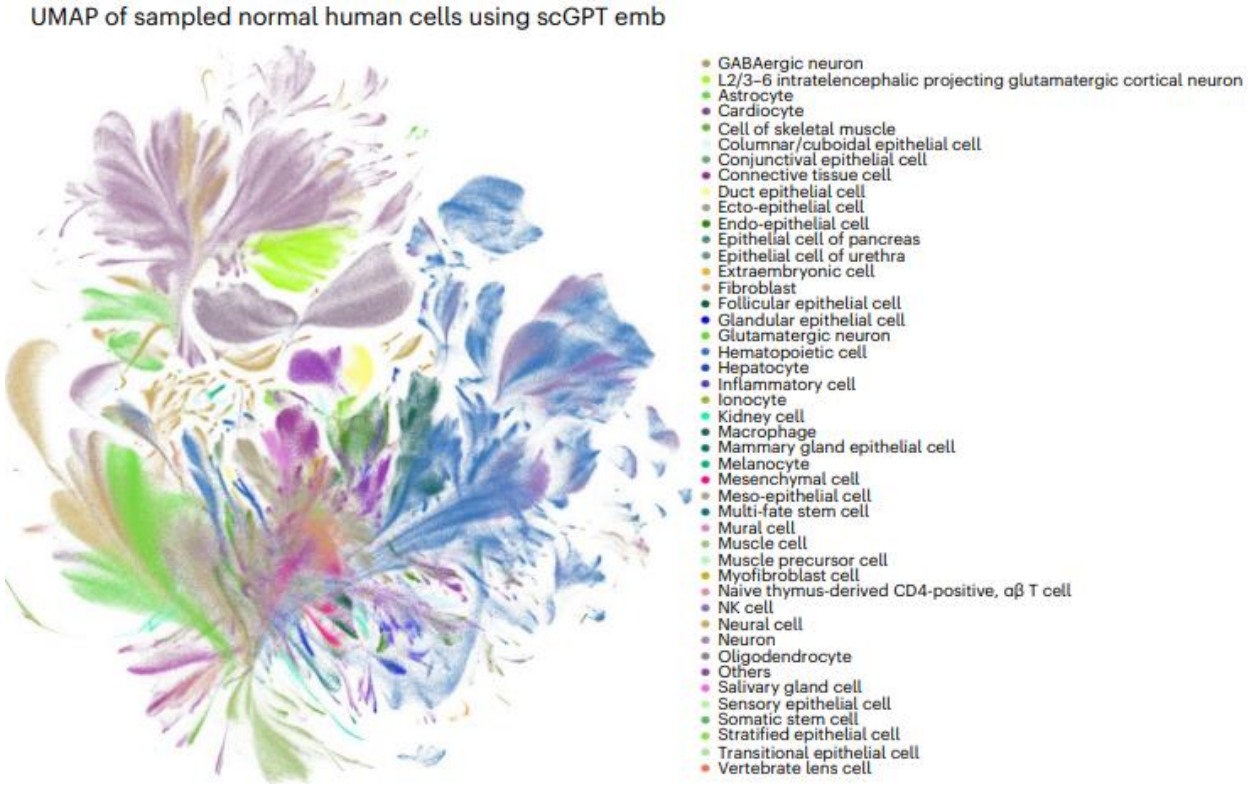


Data

- To collect diverse and extensive sequencing data for self-supervised pretraining of scGPT, we assembled scRNA-seq data from 33 million human cells under normal (non-disease) conditions, obtained from the CELLxGENE collection
- From 51 organs or tissues and 441 studies, providing a rich representation of cellular heterogeneity across the human body



- scGPT cell embeddings on 10% of the human cells of the 33 million cells
- The resulting UMAP plot exhibits intriguing clarity, with cell types accurately represented by distinct colors at localized regions and clusters



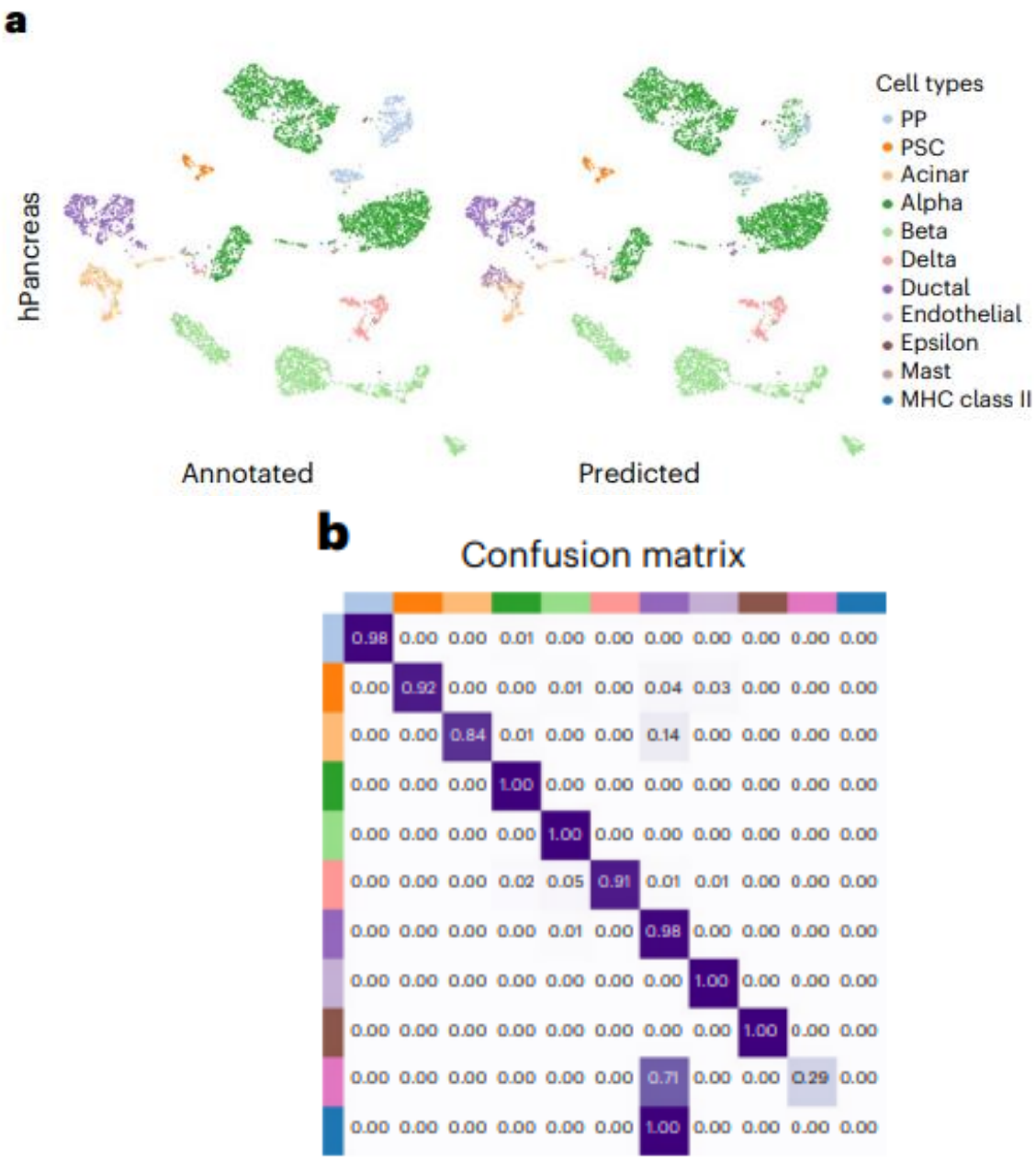
Application

1., scGPT improves the precision of cell type annotation

- To fine-tune the pretrained scGPT for cell type annotation, a neural network classifier takes the scGPT transformer output cell embedding as input and outputs categorical predictions for cell types
- The whole model was trained with cross-entropy on a reference dataset with expert annotations and then used to predict cell types on a held-out query data partition

Human pancreas dataset

- scGPT achieved high precision (>0.8) for most cell types shown in the confusion matrix except only for rare cell types with extremely low cell numbers in the reference partition



- They further explored the ability of scGPT to project unseen query cells to reference datasets through reference mapping
- They discovered that scGPT, with only pretrained weights, achieved competitive performance compared with existing methods

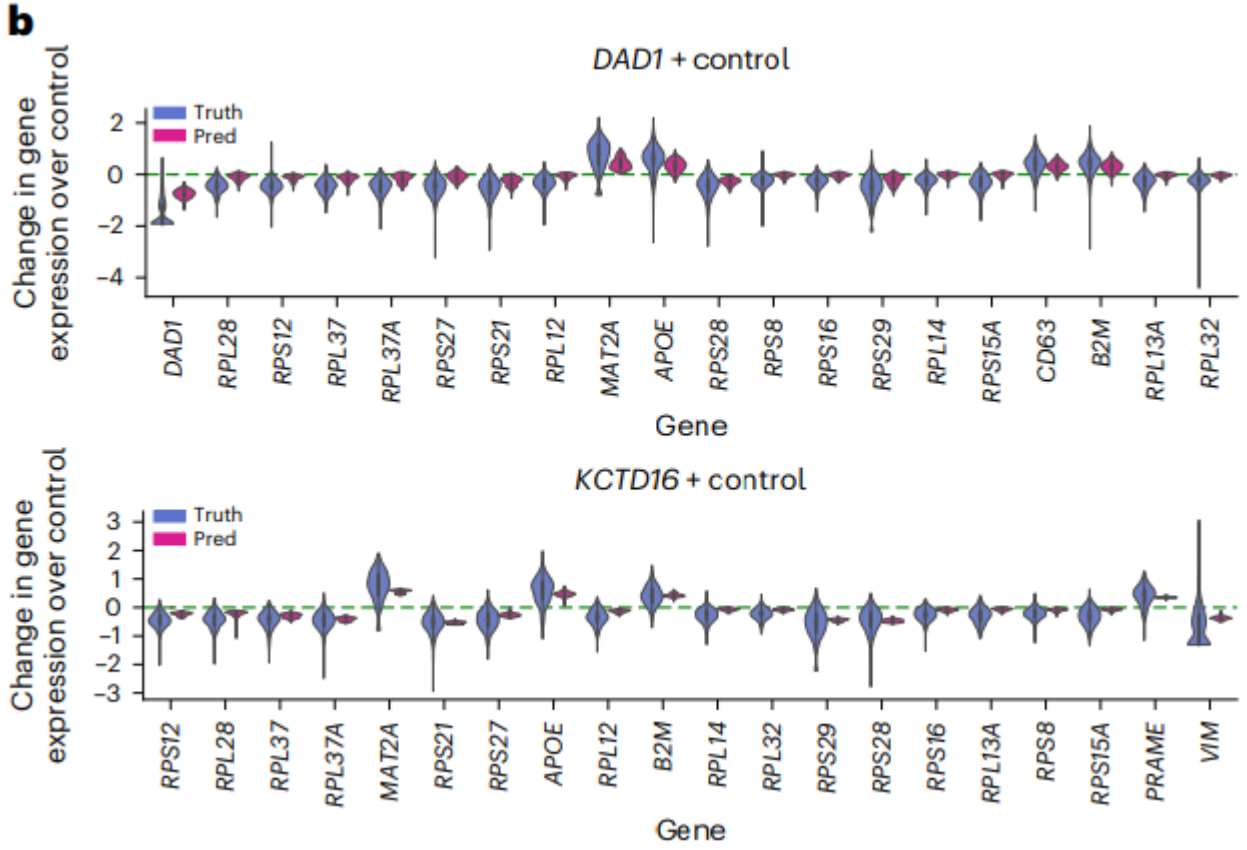
scGPT predicts unseen genetic perturbation responses

- This approach holds immense promise for uncovering new gene interactions and advancing regenerative medicine
- scGPT can be used to leverage the knowledge gained from cellular responses in known experiments and extrapolate them to predict unknown responses
- The utilization of self-attention mechanisms over the gene dimension enables encoding of intricate interactions between perturbed genes and the responses of other genes

Prediction of unseen gene perturbations

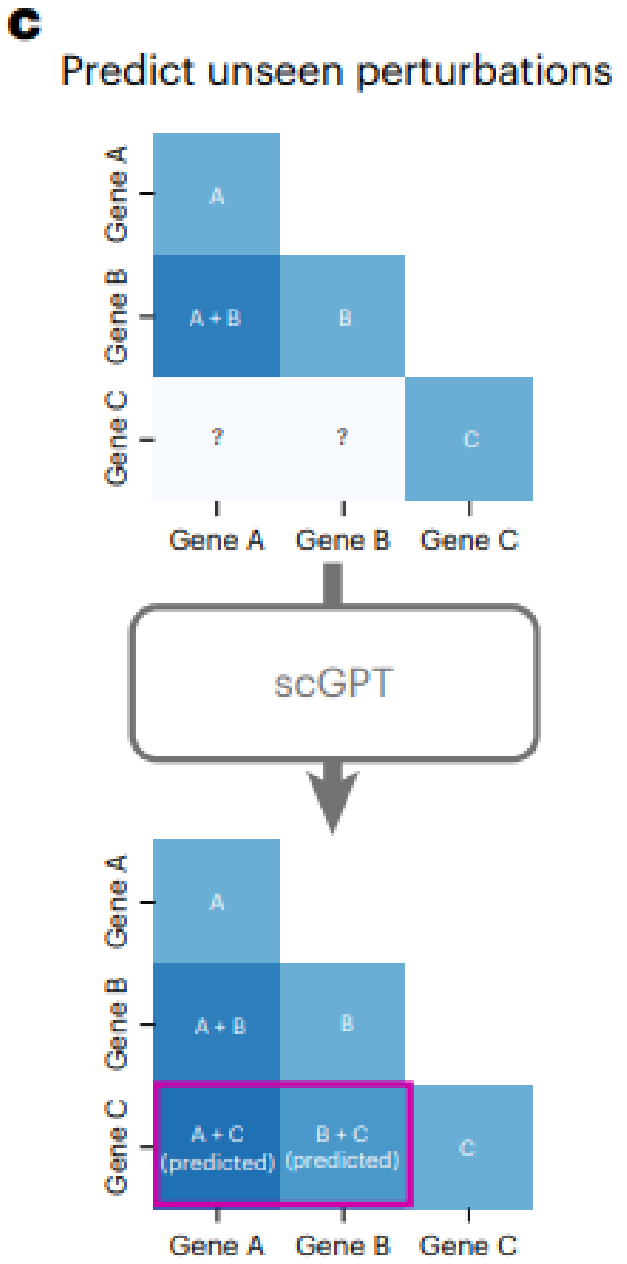
- Perturb-seq datasets
- fLeukemia cell lines: the Adamson dataset consisting of 87 one-gene perturbations, the curated Replogle dataset34 consisting of 1,823 one-gene perturbations and the Norman dataset consisting of 131 two-gene perturbations and 105 one-gene perturbations
- They fine-tuned the model on a subset of perturbations to predict the perturbed expression profile given an input control cell state and the genes of intervention
- Next, the model was tested on perturbations involving unseen genes

- We calculated the Pearsondelta metric, which measures the correlation between predicted and observed post-perturbation expression changes
- Reported this metric on the top 20 most significantly changed genes for each perturbation, denoted as Pearsondelta on differentially expressed genes

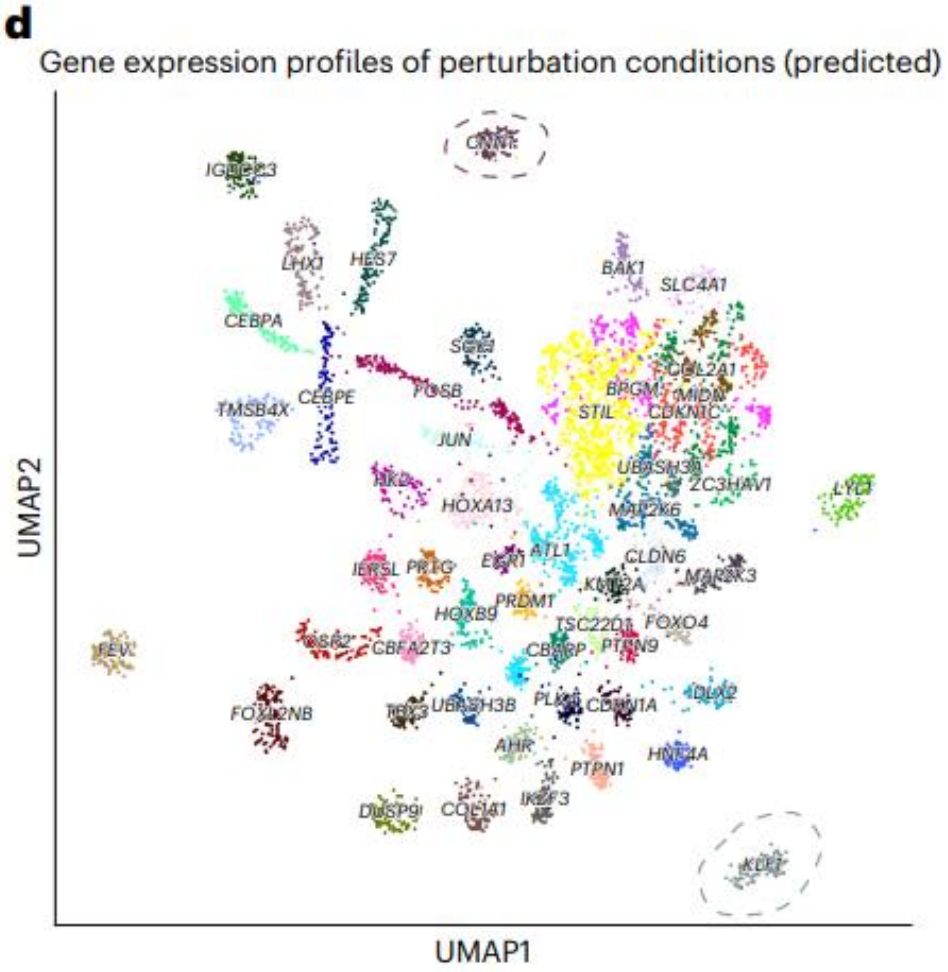


Two example perturbations in the Adamson test dataset, distribution of predicted

- The ability to predict unseen perturbation responses could expand the scope of perturbation experiments
- To explore the expanded space of predicted perturbation responses, we conducted clustering analysis using the Norman dataset to validate biologically relevant functional signals
- The original Perturb-seq study covered 236 perturbations targeting 105 genes - a total of 5,565 potential perturbations

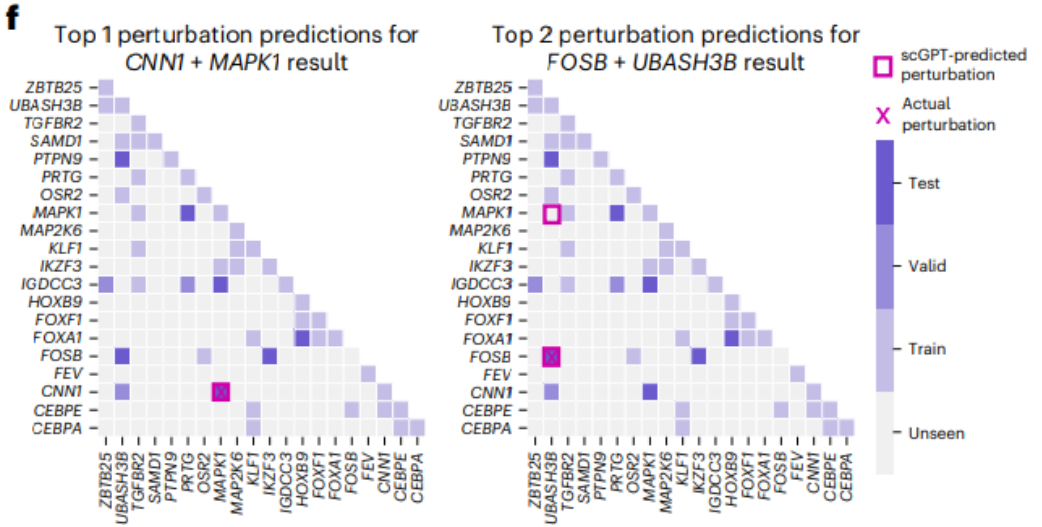


- Applied fine-tuned scGPT to expand the perturbation in silico and visualized the predicted mean response for each perturbation using UMAP
- Using the annotations from the original study, we found that perturbation conditions of the same functional groups clustered in neighboring regions
- Next, we clustered the predicted expression using Leiden and observed that the clusters exhibited high association with the ‘dominant gene’ within the perturbation combinations
- The dominant gene clusters demonstrate the capability of scGPT to uncover associations between perturbation combinations



In silico reverse perturbation prediction

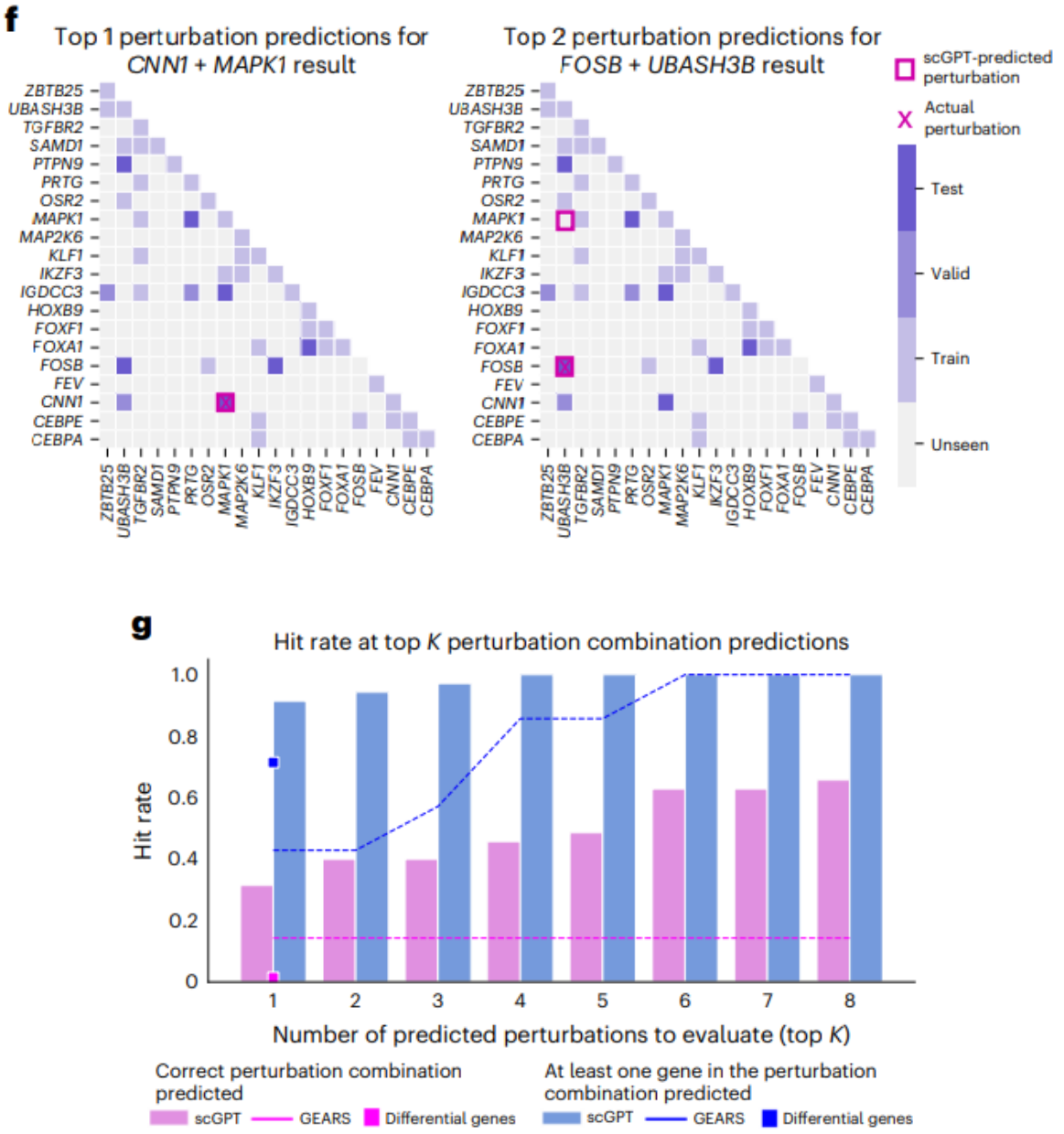
- scGPT is also capable of predicting the source of genetic perturbation for a given resulting cell state, which we refer to as in silico reverse perturbation prediction
- Reverse prediction can be used to infer important driving genes for lineage development or to facilitate the discovery of potential therapeutic gene targets
- A hypothetical example application of such capability could be to predict CRISPR target genes that influence cells to recover from a disease state
- To showcase the effectiveness of reverse perturbation
- prediction, we used a subset of the Norman dataset focusing on perturbations involving 20 genes



- This combinatorial space consists of a total of 210 one-gene or two-gene perturbation combinations
- Fine-tuned scGPT using 39 (18%) known perturbations (the training group
- We then tested the model on queries of unseen perturbed cell states, and scGPT successfully predicted the source of perturbations (within top-ranked predictions)

Example

- scGPT ranked the correct perturbation of CNN1 + MAPK1 genes as the top prediction for one test example, and the correct perturbation of FOSB + UBASH3B genes was ranked as the second prediction for another case
- Overall, scGPT identified on average 91.4% relevant perturbations (6.4 of seven) within the top 1 predictions (blue bars in Fig. 3g) and 65.7% correct perturbations (4.6 of seven test cases) within the top 8 predictions



Others

- scGPT enables multi-batch and multi-omic integration
- scGPT uncovers gene networks for specific cell states

Methods

Input embeddings

- Single-cell sequencing data are processed into a cell-by-gene matrix
- The input to scGPT consists of three main components: (1) gene (or peak) tokens, (2) expression values and (3) condition tokens. For each modeling task, gene tokens and expression values are preprocessed from the raw count matrix

Gene tokens

- Each gene is considered the smallest unit of information, analogous to a word in NLG
- Use gene names as tokens, and assign each gene gja unique integer identifier id(gj)
- These identifiers form the vocabulary of tokens used in scGPT
- Additionally, we incorporate special tokens in the vocabulary, such as < cls > for aggregating all genes into a cell representation and < pad > for padding the input to a fixed length
- The input gene tokens of each cell i are hence represented by a vector t (i) g ∈ ℕM:

$$\boldsymbol{t}_g^{(i)} = \left[\mathrm{id}(g_1^{(i)}), \mathrm{id}(g_2^{(i)}), \ldots, \mathrm{id}(g_M^{(i)})\right],$$