# “Enhancing scBERT for Optimized Performance and Reduced Latency”

Aakash, Vishwa

# Why is cell annotation using scRNA important

- Single-cell RNA sequencing (scRNA) allows for the identification of distinct cell types and subtypes within a sample. This is crucial for understanding the complex cellular heterogeneity.
- Cell annotation using scRNA can reveal novel therapeutic targets by identifying specific cell types involved in disease mechanisms
- scRNA can uncover new insights into cellular development, differentiation, and function
- scRNA can provide context for the interpretation of other omics data, such as genomic and proteomic data
- Accurate cell annotation is essential for integrating these datasets and gaining a comprehensive understanding of biological systems

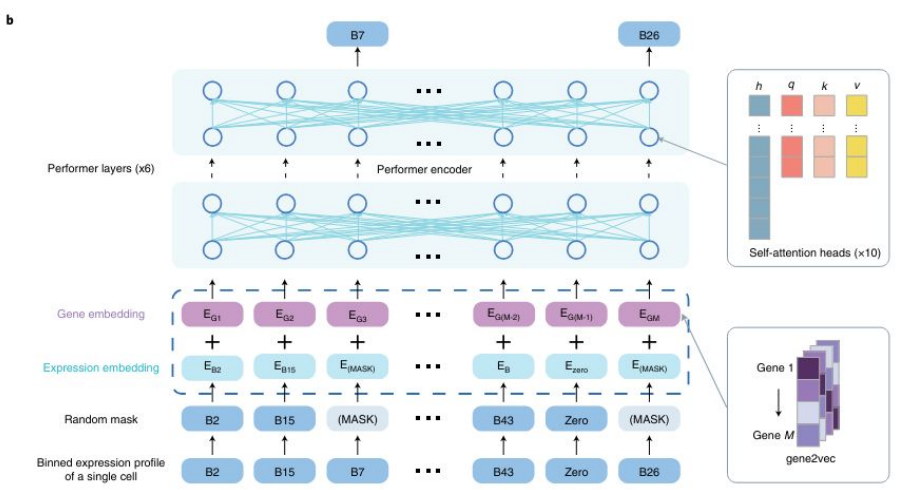
# Project Idea and Motivation

Why do we need a smaller model that can compete with the larger SOTA models?

- Inference efficiency
- Faster iteration and experimentation
- Democratization of AI

# Introduction to scBERT

- scBERT is a large-scale pretrained deep language model designed for cell type annotation in single-cell RNA-seq data.
- It utilizes Transformer(encoder) architectures with innovatively designed embeddings for genes, pioneering its application in scRNA-seq data analysis.
- scBERT employs Performer, maintaining full gene-level interpretation without relying on dimensionality reduction.


b
B7
B26
Performer layers (x6)
Performer encoder
h
q
k
v
Self-attention heads (×10)
Gene embedding
$E_{G1}$ $E_{G2}$ $E_{G3}$ ... $E_{G(M-2)}$ $E_{G(M-1)}$ $E_{GM}$
Expression embedding
$E_{B2}$ $E_{B15}$ $E_{(MASK)}$ ... $E_{B}$ $E_{zero}$ $E_{(MASK)}$
Random mask
B2 B15 (MASK) ... B43 Zero (MASK)
Binned expression profile of a single cell
B2 B15 B7 ... B43 Zero B26
Gene 1
Gene M
gene2vec

Source: https://www.nature.com/articles/s42256-022-00534-z

# Problems with scBERT

- High Computational requirement due to high sparsity of data -> 90% of values are 0 -> leads to unnecessary computations
    - 2.65 × 10^19 FLOPs to train 5 million samples over 5 epochs
- Limited or Loss of resolution for expression values -> scBERT rounds gene expression values into integers (1.99 and 2.01 are far, 1.99 and 1.01 are closer)
- During leave-one-out experiments, scBERT failed to identify novel cell types.
- The current masking strategy in scBERT, which involves non-zero masking, may need further optimization for efficiency.

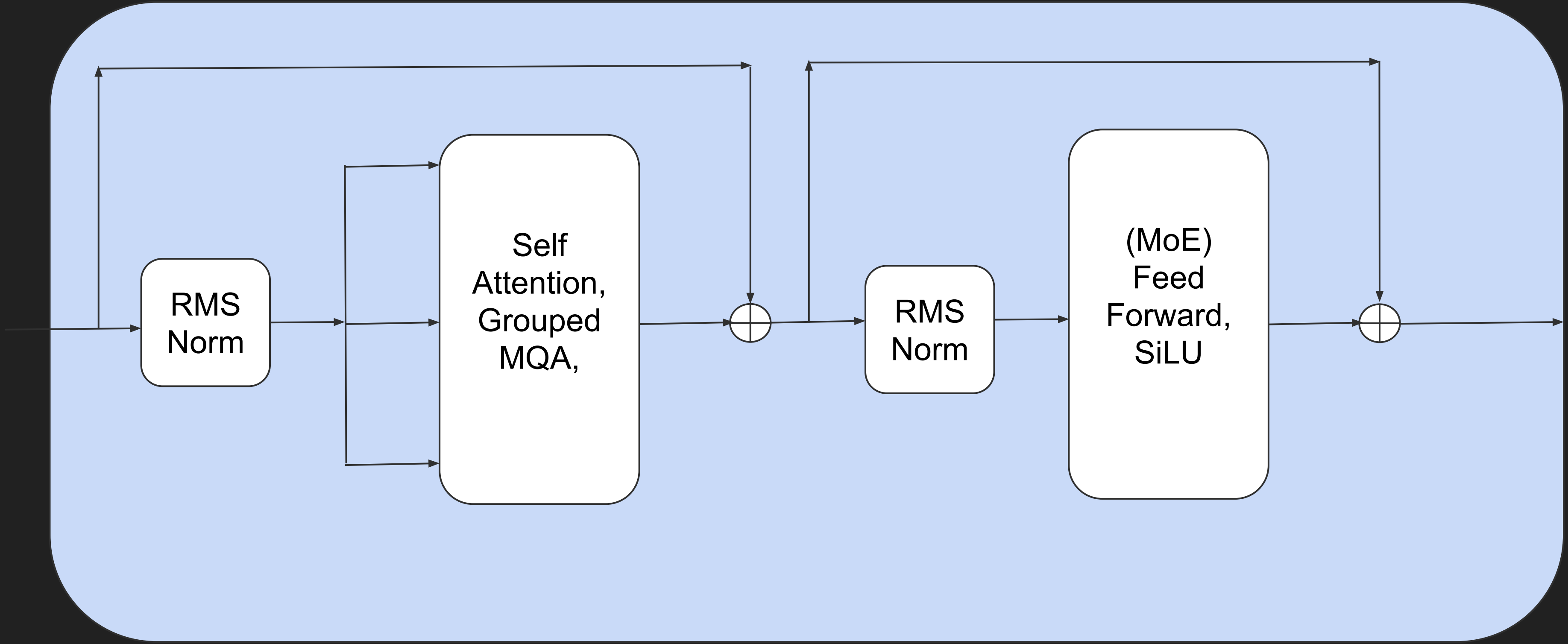
# Smaller & Faster scBERT
# (Not necessarily better)

| Model | Params | dtype |
|---|---|---|
| scBERT | 8.3M | Float32 |
| Q-scBERT (ours) | 8.3M | Int8 |
| Distilled scBERT (ours) | 5.7M | Float32 |
| Q-Distilled scBERT (ours) | 5.7M | Int8 |
| scBERT1.58 (ours) | 8.3M | Int8 |

# Better scBERT

| Model | Params | Pre Training strategy |
|---|---|---|
| scBERT(Encoder) | 8.3M | Masked Language Modelling |
| scGPT(Decoder) | 53M | Next token prediction |
| xTrimoGene(Encoder + Decoder) | 100M | Masked Regression task |
| scBERT-2.0 (Encoder) | 8x10M | MLM + Masked Regression task |

# scBERT2 Architecture



# scBERT 2.0 Improvements

### Architectural Improvements for faster training and inference

1. Grouped Multi Query Attention (5x faster than Vanilla)
2. RMS Norm in place of LayerNorm (Faster training convergence)
3. Flash attention 2.0 (2.3x faster than Performer-scBERT)
4. SiLU in place of ReLU/GLU (Objectively better than ReLU)

### Improvements to improve parameter count(model complexity) while keeping the computational cost to a minimum

1. Sparse Mixture of Experts (8x improvement in model complexity, slight increase in computational cost)

### Improvements for faster inference

1. Torch.compile (30% faster inference time)

### Improvements to pre-training strategy

1. Improved Token Embeddings (
2. Improved masking (MLM + Masked Regression task)

### For Faster training

1. Mixed precision training (2x improvement in throughput)
2. Distributed Data Parallel Training (Nx improvement in training time)
3. Faster Data Loading using MultDL (reduces cpu-gpu transfer latency)
4. Adafactor (More computationally efficient and memory efficient than Adam)
5. Data preloading (reduces data loading latency)

# What downstream tasks are we planning to cover?

- Cell type annotation
- Perturb-seq effect prediction
- Drug combination prediction

# Why these tasks?

The data is already available for scGPT and xTrimoGene. Making it easier for us to compare them.

# Current Progress

| Model | Code | Pre-training/Conversion | Benchmarking on downstream tasks |
|---|---|---|---|
| Q-scBERT | Done | Done (~1 hr) | Most of the data and code required to benchmark on downstream tasks is done. We are waiting on the models to complete pre-training stage |
| Distilled scBERT (ours) | Done | In Progress(~4 hrs) | |
| Q-Distilled scBERT (ours) | Done | In progress(~30 minutes) | |
| scBERT1.58 (ours) | Done | Done | |
| scBERT 2.0 | Done* | To do (~30 hrs) | |