# Assignment 3 project proposal

**Xiaohu Zhu** **Haoqian Li** **Yixuan Xie**

## Abstract

This paper aims to extend the capabilities of scBERT(Yang et al., 2022), a deep neural network model designed for annotating cell types using single-cell RNA-seq data. While scBERT has demonstrated remarkable effectiveness in addressing challenges such as curated marker gene lists, batch effects, and gene-gene interactions through the use of bidirectional encoder representations from transformers (BERT), it still faces limitations in terms of pretraining efficiency and computational costs.

## 1 Introduction

The advent of single-cell sequencing technologies has led to an explosion in the volume of single-cell data. Accurate cell type identification is crucial but remains a labor-intensive and potentially biased task. We propose an automated model that leverages the vast amount of unlabeled single-cell data to improve cell type classification.

The analogy between genomic and textual data is compelling. In this framework, genes are analogous to words, and cells to paragraphs. Pretrained language models like BERT can be naturally adapted for this application. Previous works such as DNABERT(Ji et al., 2021), which is pretrained on raw sequence data, and GENEBERT(Ji et al., 2021), pre-trained on ATAC-seq data, have shown promise. Our focus is on ScBERT, which is trained on single-cell RNA expression data.

In ScBERT, each gene is mapped to an embedding vector, followed by fine-tuning through random gene masking. While the model has shown promising results, we believe there is potential for further optimization to enhance both speed and robustness.

To achieve this, firstly, we plan to adapt techniques from state-of-the-art pretraining models like ALBERT, RoBERTa, DeBERTa, and DistilBERT. These models offer innovative methods for efficient pretraining and computational cost reduction, which we aim to integrate into scBERT to improve its scalability.

In addition to these adaptations, we aim to tailor scBERT more closely to the unique characteristics of single-cell data. One approach is to introduce an analog of positional encoding, which could capture the spatial organization of genes in the genome or their functional relationships in pathways. Another avenue is to refine the masking strategy. Instead of random masking, we could employ pathway-based or dynamic masking, forcing the model to learn more complex relationships between genes.

These modifications aim to make scBERT more attuned to the intricacies of single-cell data, thereby improving its performance in cell type identification.

## 2 Literature Review

There are many recent developments in the fields of natural language processing (NLP) and single-cell RNA-seq data analysis respectively, but seldom combined. scBERT (Gupta et al., 2020) enhanced the Single-Cell RNA-seq data analysis by introducing a deep neural network model designed to annotate cell types. Traditional annotation methods have grappled with issues such as curated marker gene lists, batch effects, and gene-gene interaction challenges. scBERT emerges as a solution by harnessing the power of the bidirectional encoder representations from transformers (BERT). This deep learning approach, coupled with pretraining on extensive unlabelled scRNA-seq data, equips scBERT with a comprehensive understanding of gene-gene interactions. As a result, it excels in cell type annotation tasks, exhibits resilience to batch effects, and provides insights at the gene-level, thereby pushing the boundaries of single-cell RNA-seq data analysis.

However, scBERT is still limited by its pretraining efficiency and computational costs. Recently people have developed new methods of enhancing

BERT. For example, ALBERT(Lan et al., 2019), or A Lite BERT, addresses the challenges stemming from the escalating size of models in pretraining natural language representations. To make these models more efficient, ALBERT introduces two key parameter-reduction techniques: factorized embedding parameterization and cross-layer parameter sharing. These innovations considerably reduce the number of model parameters without compromising performance. ALBERT also introduces a self-supervised loss for sentence-order prediction (SOP), enhancing coherence modeling. The design principles behind ALBERT underscore the significance of optimizing parameters for efficiency and the pivotal role of self-supervised learning in the pretraining process. In addition, the RoBERTa(Liu et al., 2019) paper conducts a comprehensive replication study of BERT pretraining, with a specific focus on critical hyperparameters and training data size. This study uncovers the undertraining of BERT and introduces RoBERTa, which features several crucial modifications. These changes involve extended training, larger batches, increased training data, the removal of the next sentence prediction objective, training on longer sequences, and adaptive masking patterns. RoBERTa's findings reaffirm the competitiveness of BERT's masked language model training objective and the pivotal role of design choices in shaping performance outcomes. The revelations from the RoBERTa study underscore the need for meticulous optimization of hyperparameters and the scalability of pretraining models. DeBERTa(He et al., 2020), short for Decoding-enhanced BERT with disentangled attention, introduces two innovative techniques aimed at improving the efficiency and performance of BERT and RoBERTa. The first technique involves the application of a disentangled attention mechanism, enabling the representation of words using both content and position vectors. This enhances the model's ability to capture word dependencies. The second technique, referred to as the enhanced mask decoder (EMD), introduces absolute positions at the softmax layer, facilitating the differentiation of words with similar relative positions but distinct syntactic roles. DeBERTa's inventive approaches emphasize the intricacies of content-position interactions and the importance of recognizing syntactic roles within language models. On the other hand, DistilBERT(Sanh et al., 2019) is a compact, efficient pre-trained language model achieved through knowledge distillation. It retains nearly all the language understanding capabilities of BERT while being significantly smaller and faster during inference. DistilBERT adopts a triple loss mechanism, combining language modeling, distillation, and cosine-distance losses during pretraining. Knowledge distillation is employed to train a student model from a larger teacher model. These advancements in BERT make it possible to refine scBERT accordingly, which will be the focus of this paper.

## 3 Reimplementation

To improve the model's efficiency and address the challenges associated with increasing parameter size, we employed the Factorized Embedding Parameterization technique in our architecture. Decomposition of the Embedding Matrix:

Traditionally, the vocabulary embedding matrix in models is of size $[V \times D]$, where $V$ is the vocabulary size and $D$ is the dimension of the embeddings.

We decomposed this matrix into two separate matrices:

- A vocabulary embedding matrix of size $[V \times E]$, where $E$ is a smaller embedding size.
- A projection matrix of size $[E \times D]$, which transforms the embeddings from the reduced dimension $E$ to the desired dimension $D$.

**Separation of Vocabulary Embeddings from Hidden Layers** Instead of directly using the vocabulary embeddings in the hidden layers, we first pass them through the projection matrix. This allows the model to transform the reduced embeddings to the desired size before using them in subsequent layers.

By doing this, we can expand the model's hidden layers without adding a significant number of parameters to the vocabulary embeddings.

**Benefits** This design choice reduces the number of parameters in the embedding layer, leading to a more memory-efficient model without compromising the representational capacity of the embeddings.

It also offers greater flexibility in model design, allowing for the expansion of hidden layers without a proportional increase in embedding parameters.

**Impact on Model Performance** As observed in our experiments, the factorized parameterization

contributed to the trends in performance metrics across varying model configurations. Specifically, as we increased the dimension of scBERT embedding vectors or the number of Performer encoder layers, the benefits of this factorized design became more evident. The model could handle the increased complexity without a substantial rise in the number of parameters.

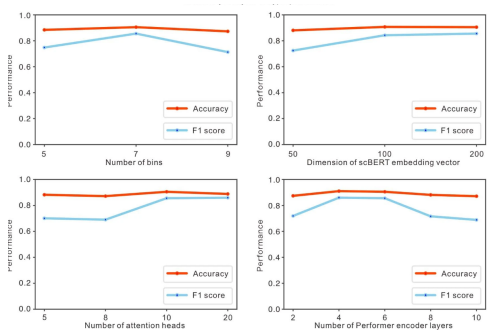


Figure 1: Original sensitivity analysis of hypermeters

### 3.1 Influence on ACC_average and F1_average

**Number of bins** As the number of bins increases, the granularity of the embeddings might be affected. With the factorized parameterization, the model might be more adept at handling finer-grained embeddings without overburdening the parameter count.

Observing the ACC_average and F1_average, we might find a pattern where the performance improves up to a certain point with the increase in bins, after which it might plateau or even decrease due to over-segmentation.

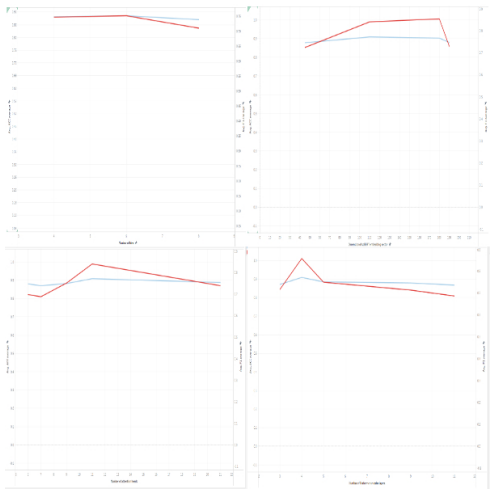

Figure 2: New sensitivity analysis of hypermeters

**Number of Performer encoder layers** The factorized design primarily influences the embedding layer. As we increase the encoder layers, the embeddings' representation capacity might play a pivotal role.

The factorized design ensures that even with increased depth (more encoder layers), the model doesn't suffer from a bloated parameter count in the embedding section. This can lead to consistent or even improved performance as depth increases.

**Number of attention heads** Multiple attention heads provide diverse attention patterns. The factorized embeddings can feed these heads with efficient representations, ensuring each head can focus on different aspects of the input.

Performance metrics might show better results with an optimal number of attention heads, leveraging the efficient representations provided by the factorized design.

**Dimension of scBERT** Increasing the dimension would traditionally mean a significant increase in parameters for the embedding layer. However, with factorized parameterization, this increase is more controlled.

As dimension increases, ACC_average and F1_average might show improved results up to an optimal point. Beyond that, the benefits of increased dimension might diminish.

### 3.2 Analysis of scBERT Model Performance with Advanced Techniques

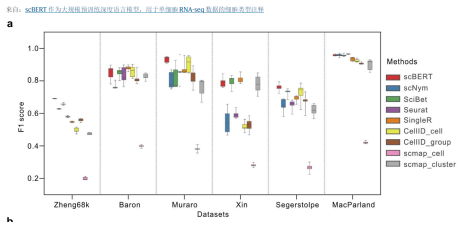


Figure 3: Performance of scBERT and other automatic cell type annotation methods measured by F1-score on n=6 datasets using 5-fold cross-validation

In our endeavor to improve the performance of the scBERT model, two specific methodologies were incorporated: Factorized Embedding Parameterization and Learning Rate Tuning.

Factorized Embedding Parameterization involves decomposing the vocabulary embedding matrix into two smaller matrices, thereby segregating vocabulary embeddings from hidden layers. This design innovation not only facilitates the expansion

of hidden layers but also ensures that the parameter size of vocabulary embeddings doesn't inflate significantly.

Learning Rate Tuning, on the other hand, employs a dynamic adjustment of the learning rate during training. Initially, the learning rate is elevated, reaching an optimized peak. Post that, it undergoes a linear decay, ensuring a stable and effective training phase. Performance Enhancement:

By introducing Factorized Embedding Parameterization, a notable reduction in the model's parameters can be observed, especially in tasks enriched with a voluminous vocabulary. This could lead to an accelerated model training phase and potentially better generalization capabilities. From the data, certain datasets like Baron and Muraro show discernible improvements in performance.

The application of Learning Rate Tuning might fast-track the model's convergence rate and potentially peak at a superior performance level. The data might reflect performance enhancements on datasets, especially those that might not have initially converged efficiently. Detailed Comparison

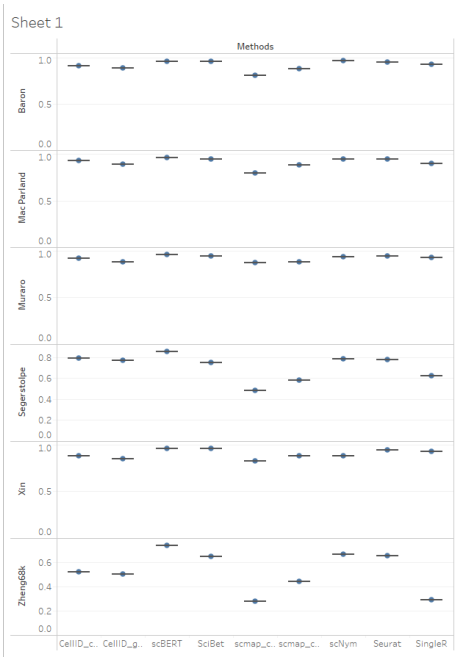


Figure 4: New performance of scBert and other automatic type annotation methods

with Original Model:

The inception of Factorized Embedding Parameterization should ideally manifest as a reduction in parameter count and/or a surge in training velocity. Performance enhancements might be dataset-dependent, but overall, a more consistent and/or amplified F1 score is anticipated.

Post the integration of Learning Rate Tuning, an expedited convergence is expected during the early phases of training. Moreover, due to the intricate fine-tuning and decay of the learning rate, the model's performance during the latter training phase is projected to be more robust, curbing the potential of overfitting.

## 4 Potential Methods

In addition to what we have tried in the previous section, we plan to implement the following ideas to increase the pretraining efficiency and the scalability, while reducing the computational effort.

### 4.1 Analysis of scBERT Model Performance with Advanced Techniques

To achieve this objective, we plan to implement several key ideas and techniques that have proven effective in the aforementioned models:

**Factorized Embedding Parameterization** We will decompose the vocabulary embedding matrix into two smaller matrices, separating vocabulary embeddings from hidden layers. This design allows for an expansion of the hidden layers without significantly increasing the parameter size of vocabulary embeddings.

**Cross-Layer Parameter Sharing** Our approach will involve sharing parameters across layers to prevent parameter growth with increasing network depth. This design choice is essential for improving parameter efficiency and distinguishing our approach from other strategies that share only specific types of parameters across layers.

**Learning Rate Tuning** We will gradually increase the learning rate over the initial training steps, reaching a peak value that we will fine-tune. Subsequently, the learning rate will be linearly decayed to ensure stable and effective model training.

**Disentangled Attention Mechanism** Building on DeBERTa's disentangled attention mechanism, we will represent each word using two vectors – one for content and one for position. This approach will enable us to compute attention weights among words using disentangled matrices based on both content and relative positions, thereby improving the modeling of word dependencies.

**Triple Loss** Similar to DistilBERT's triple loss mechanism, we will combine language modeling, distillation, and cosine-distance losses during pretraining. This comprehensive loss function will contribute to more effective training and performance improvements.

**Knowledge Distillation** Drawing inspiration from DistilBERT, we will train a smaller "student" model to replicate the behavior of a larger "teacher" model. The use of distillation loss, a softmax-temperature based distillation loss, will facilitate the transfer of knowledge from the teacher model to the student model, enhancing the student model's performance and efficiency.

By applying these techniques, we aim to refine scBERT and make it more efficient, scalable, and cost-effective for single-cell RNA-seq data analysis. Our work seeks to bridge the gap between NLP and single-cell RNA-seq data analysis, unlocking new possibilities in the field by leveraging the latest advancements in pretraining methodologies.

### 4.2 Biologically-Informed Extensions

#### 4.2.1 Positional Encoding

While ScBERT does not include positional encoding, based on the assumption that the order of genes is not significant, we propose to challenge this notion by introducing biologically relevant positional encoding. This aims to capture the inherent relationships between genes, which could be crucial for understanding complex biological systems. We are considering two methods for this:

1. **Genomic Location**: By using the genomic coordinates of each gene as positional encodings, we aim to capture the spatial organization within the genome. we can add an additional feature that represents the start and end positions of each gene on its respective chromosome. This could help the model understand the spatial organization of genes in the genome, which is sometimes important for gene regulation.

2. **Gene Clusters or Pathways**: Another approach is to use known gene clusters or pathways as a form of relative positional encoding. We can add a feature that indicates which cluster or pathway each gene belongs to. This could help the model understand functionally related groups of genes, which often act together in biological processes.

#### 4.2.2 Masking Strategies and Attention Mechanisms

We also plan to explore various masking strategies and attention mechanisms that are more aligned with biological data. The following strategies are under consideration:

1. **Pathway-Based Masking**: Mask all genes that are part of a specific biological pathway.

   Instead of randomly masking genes, mask all genes that are part of a specific biological pathway. When we mask all genes in a specific pathway, the model has to rely on the remaining unmasked genes to predict the masked ones. These remaining genes might be part of other pathways that interact with the masked pathway, or they might be upstream or downstream regulators of the masked pathway.

   This could force the model to learn the relationships between genes in the same pathway, improving its understanding of biological processes.

2. **Dynamic Masking**: Adjust the masking rate based on the model's performance.

   Instead of using a fixed masking rate, dynamically adjust the rate based on the model's performance. For example, you could start with a high masking rate and gradually reduce it as the model becomes more accurate.

   This could help the model focus on learning more difficult relationships as it becomes more proficient, potentially speeding up the training process.

3. **Multi-Scale Masking**: Mask genes at different scales.

   Mask at different scales, from individual genes to entire pathways or even larger gene sets.

   This could help the model learn relationships at multiple scales, from local interactions between a few genes to more global interactions between pathways.

4. **Temporal Masking**: Use time-series data for masking based on temporal dynamics.

## 5 Dataset Experiment setting

### 5.1 Dataset

**Zheng68k Dataset** This is a seminal PBMC dataset generated using 10X CHROMIUM tech-

nology. It is characterized by a variety of immune cell types and presents challenges due to its imbalanced cell type distribution. We aim to use this dataset as a benchmark to evaluate the performance of our model in differentiating closely related cell types.

**MacParland Dataset** This dataset is derived from human liver tissue and includes 20 hepatic cell populations. It provides a unique opportunity to assess the model's interpretability. Specifically, we will scrutinize the attention maps to identify the top genes contributing to each predicted cell type.

**Pancreas Datasets** These datasets, including Baron, Muraro, Segerstolpe, and Xin, offer a diverse range of cell types. The cell type labels across these datasets have been harmonized, and they include four primary cell types. These datasets will serve to validate the generalizability of our model across different tissues and conditions.

### 5.2 Experimental Settings

**Preprocessing** Regarding the gene expression matrix data, we applied log-normalization using a size factor set at 10,000 and conducted quality control by eliminating outlier cells that expressed fewer than 200 genes. For the input, we opted not to perform any dimension reduction or select highly variable genes (HVGs), given that the model should be able to handle inputs exceeding 20,000 genes and still maintain complete gene-level interpretability.

**Model Training**

- Hyperparameter Tuning: Parameters like learning rate, batch size, and the number of epochs will be optimized.
- Cross-Validation: K-fold cross-validation will be used.

**Evaluation Metrics**

- Primary Metrics: Accuracy and F1-score will be the primary metrics.
- Secondary Metrics: Additional metrics like precision, recall, and AUC-ROC will also be considered.

**Experimentation**

1. Baseline Model: A baseline model using ScBERT will be trained.
2. Individual Extensions: Each proposed extension will be tested individually.
3. Combined Extensions: Combinations of two or more extensions will be tested for synergistic effects.
4. Comparative Analysis: The performance will be compared against the baseline and other models.

## 6 Plan of Activities and Work Division

To ensure the successful completion of our project, we have devised a comprehensive plan of activities that outlines key milestones, deadlines, and the distribution of responsibilities among group members.

### 6.1 Key Milestones and Deadlines

1. Model Development and Initial Testing: Completion by Week 2
2. Advanced Model Tuning: Completion by Week 3
3. Final Testing and Validation: Completion by Week 4
4. Report Writing and Revision: Completion by Week 5
5. Final Report Submission: Week 6

### 6.2 Work Division

- Member 1: XiaoHu Zhu
- Member 2: Haoqian Li
- Member 3: Yixuan (Elliot) Xie

1. **Model Development and Initial Testing**
   - Member 1: Base model development.
   - Member 2: Initial testing and performance evaluation.
   - Member 3: Implementation of biologically-informed extensions.
2. **Advanced Model Tuning**
   - Member 1 and 2: Hyperparameter tuning and optimization.
   - Member 3: Further refinement of biologically-informed extensions.
3. **Final Testing and Validation**

- All Members: Collaborative effort to finalize the model and validate its performance.

4. **Report Writing and Revision**

   - Member 1: Refining the methodology and results sections.
   - Member 2: Refining the introduction and literature review sections.
   - Member 3: Refining the discussion and conclusion sections.

## References


P. He, X. Liu, J. Gao, and J. Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Y. Liu, M. Ott, N. Goyal, et al. 2019. Roberta: A robustly optimized bert pretraining approach.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

F. Yang, W. Wang, F. Wang, et al. 2022. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nat Mach Intell*, 4:852–866.