Gene Incremental Learning for Single-Cell Transcriptomics

Jiaxin Qi[1], **Yan Cui**[2], **Jianqiang Huang**[1,2,3*], **Gaogang Xie**[1,3]

[1]Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
[2]Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou, China
[3]University of Chinese Academy of Sciences, Beijing, China
jxqi@cnic.cn, cuiyan.ch@gmail.com, jqhuang@cnic.cn, xie@cnic.cn

Abstract

Classes, as fundamental elements of Computer Vision, have been extensively studied within incremental learning frameworks. In contrast, tokens, which play essential roles in many research fields, exhibit similar characteristics of growth, yet investigations into their incremental learning remain significantly scarce. This research gap primarily stems from the holistic nature of tokens in language, which imposes significant challenges on the design of incremental learning frameworks for them. To overcome this obstacle, in this work, we turn to a type of token, gene, for a large-scale biological dataset—single-cell transcriptomics—to formulate a pipeline for gene incremental learning and establish corresponding evaluations. We found that the forgetting problem also exists in gene incremental learning, thus we adapted existing class incremental learning methods to mitigate the forgetting of genes. Through extensive experiments, we demonstrated the soundness of our framework design and evaluations, as well as the effectiveness of our method adaptations. Finally, we provide a complete benchmark for gene incremental learning in single-cell transcriptomics.

Introduction

The class of an object serves as a foundational concept in Computer Vision. It is observed that the number of classes often increases due to factors such as the discovery of new species in the natural world and the assignment of novel class labels to recently developed objects. In response, the Class Incremental Learning (CIL) framework has been introduced to evaluate a model's ability to continuously learn new classes (Wu et al. 2019; Zhang et al. 2020; Masana et al. 2022). The pipeline for this framework is illustrated in Figure 1(a). Initially, the model trains on images from specific classes, i.e., "cat" and "dog". Subsequent stages of training focus exclusively on datasets containing new classes, such as "deer" and "bird", ensuring that samples from previous classes remain inaccessible during these stages. Evaluations will be performed across all seen classes until the current stage. Considering the specified framework, extensive studies highlight catastrophic forgetting of previous classes as the crucial challenge in Class Incremental Learning. To mitigate such forgetting, methods designed on data replay (Zhu et al. 2021; Hu et al. 2021) and knowledge distillation (Dong et al. 2021; Kang, Park, and Han 2022) have been extensively proposed and explored.

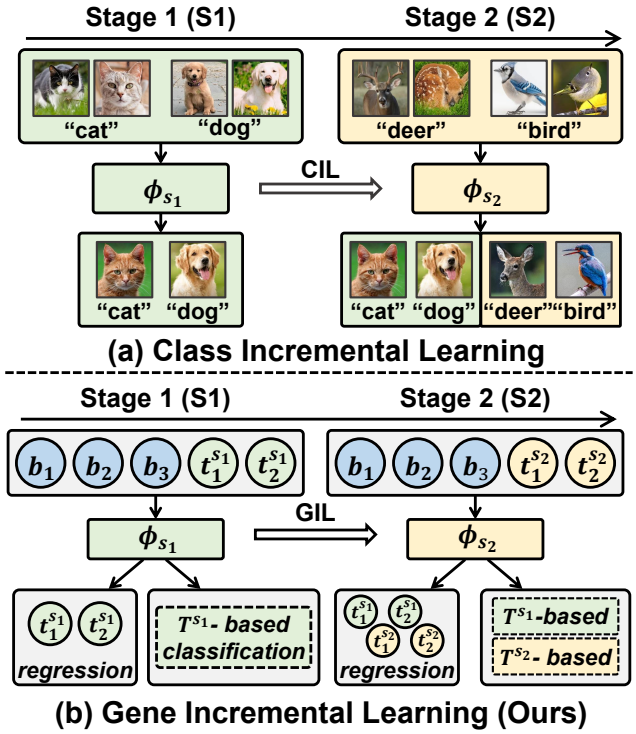

Figure 1: Illustrations of (a) Class Incremental Learning (CIL) framework and (b) our proposed Gene Incremental Learning (GIL) framework. In CIL, the given classes are exclusive at each stage, and classification accuracy is tested across all previously seen classes. In GIL, $b_i, i = 1, 2, 3, \ldots$ denote the base tokens given in every stage, while $T^{s_i} = \{t^{s_i}\}$ represents the set of specific tokens to be learned in stage i. For evaluation, regression refers to the token-wise regression loss, and T^{s_i}-based classification denotes performing the classification on the specific downstream dataset where the token set T^{s_i} is crucial.

Similarly, tokens, crucial elements for many fields (Pennington, Socher, and Manning 2014; Cui et al. 2024; Hao et al. 2024), also exhibit growth characteristics like classes.

*Corresponding author.

For example, in natural language processing, where tokens represent words, the continual invention of new words leads to an expansion of the vocabulary (Lehrer 2003). Likewise, in the biological field of single-cell transcriptomics, where tokens represent genes, new genes are continually discovered due to advancements in measurement technologies (Karaayvaz et al. 2018; AlJanahi, Danielsen, and Dunbar 2018), contributing to the expansion of the gene pool. Thus, incremental learning for tokens has practical significance. However, this framework has been consistently overlooked, primarily due to the challenges in defining it within the holistic nature of language data. For example, if we apply the settings of CIL to divide different words into different stages, e.g., the word "learning" no longer appears in one stage, it becomes impractical to either collect texts that do not contain the word "learning", which would significantly reduce the amount of data, or just remove the word "learning" from existing texts, which would change the original meaning. Moreover, these difficulties become significantly exacerbated when multiple works need to be excluded.

Fortunately, the challenge mentioned earlier does not exist in single-cell transcriptomics (simplified as transcriptomics) (Tang et al. 2009), allowing us to design the gene incremental learning framework to address the increase of new genes. In transcriptomics, genes are viewed as tokens, similar to words, and each sample consists of a sequence of gene expression values, analogous to a sentence, and the mainstream models in this field are based on Transformers (Cui et al. 2024). Unlike the holistic nature of language, transcriptomic data lacks relative orders among genes, allowing for straightforward division and rearrangement of genes in different incremental stages to establish an incremental framework. Therefore, we design the Gene Incremental Learning (GIL) framework for transcriptomics with the following details: As shown in Figure 1(b), we maintain some genes as base genes, which is essential for rendering samples meaningful under transcriptomic contexts. Then, we divide the remaining genes into various stages, ensuring they are mutually exclusive across stages. For example, in stage one, the samples contain base genes and genes t^{s_1}, while in stage two, models can only see base genes and genes t^{s_2}. Evaluations will be performed across all seen genes until the current stage. This framework effectively constructs a pipeline for Gene Incremental Learning and enables the assessment of the model's ability to continually learn new genes.

In addition, we propose comprehensive evaluations for Gene Incremental Learning. First, we introduce a gene-wise regression metric that directly assesses model forgetting for previous genes. Second, as shown in Figure 1(b), we propose a gene-based classification evaluation, where specific genes are selected for each stage, whose learning is crucial for the corresponding downstream classification datasets. This means learning for such genes in a stage will make the model perform better in the downstream classifications associated with that stage. Utilizing these datasets allows us to use classification accuracy to demonstrate whether genes have been memorized or forgotten. Furthermore, to mitigate gene forgetting, we have adopted several fundamental CIL methods to establish the baseline methods for GIL. Through extensive experiments, we validate the rationality of our Gene Incremental Learning framework, the consistency of our evaluation methods, and the effectiveness of our adapted methods. Ultimately, we present a straightforward yet comprehensive benchmark for Gene Incremental Learning for single-cell transcriptomics.

We summarize our main contributions as the following three aspects:

1. We thoroughly define the Gene Incremental Learning framework, using single-cell transcriptomic datasets, which addresses the research gap in incremental learning in the context of the continuous growth of genes.
2. We propose the evaluations for Gene Incremental Learning by introducing a gene-wise regression and gene-based classification to facilitate a thorough assessment of gene learning and forgetting within the GIL framework.
3. We adapt existing Class Incremental Learning methods to the GIL and validate the effectiveness of the adaptations through extensive experiments. Finally, we introduce a comprehensive Gene Incremental Learning benchmark for single-cell transcriptomics.

Related Works

Class Incremental Learning

Class Incremental Learning (CIL) (Chen and Liu 2018; Pentina 2016) , also known as lifelong learning (Silver and Mercer 2002; Silver, Yang, and Li 2013) or continual learning (Shi et al. 2024; De Lange et al. 2021), is inspired by the continual learning pattern observed in human brains (Constantinescu, O'Reilly, and Behrens 2016; McCaffary 2021). It involves training models sequentially on a series of classes while maintaining overall performance on all seen classes. Researchers have identified catastrophic forgetting as the major challenge for CIL (Goodfellow et al. 2013; McCloskey and Cohen 1989). To address this issue, two main camps of methods have been proposed: 1). Data replay (Castro et al. 2018; Hou et al. 2019; Zhao et al. 2020) demonstrates strong resistance to forgetting by storing exemplars of old classes. 2). Knowledge distillation (Hinton 2015; Rebuffi et al. 2017) retains model behavior by learning the outputs or features of the old models.

In addition to classic CIL, incremental learning also encompasses Task Incremental Learning (Qin and Joty 2022; Ke and Liu 2022) and Domain Incremental Learning (Lu et al. 2018), which divide tasks or different distributions of data into different incremental stages, respectively. However, these incremental learning frameworks are mainly focused on the increase of class or data, but do not discuss tokens. In this work, we follow the CIL settings, leveraging single-cell transcriptomic data, to define the Gene Incremental Learning, which is fundamentally different from the traditional incremental frameworks.

Single-Cell Transcriptomics

Single-cell transcriptomics, also known as single-cell RNA sequencing (*i.e.*, scRNA-seq), was initially developed by the Surani Lab (Tang et al. 2009). Additionally, the landscape of

computational tools and public data repositories for scRNA-seq has rapidly expanded (Voigt et al. 2021; Kharchenko 2021). Today, scRNA-seq is extensively utilized in human health research, primarily to characterize cell types across various organs (Ramachandran et al. 2020; Gustafsson and Johansson 2022) or clarify temporal processes such as human tissue development (Olaniru et al. 2023; Collin et al. 2021).

As genes can be viewed as tokens, researchers have applied NLP methods to transcriptomics, particularly using Transformers as feature extractors. scBERT (Yang et al. 2022) was a pioneer in proposing a single-cell pre-training framework utilizing Transformers. Subsequent studies have focused on increasing the data volume (Cui et al. 2024), expanding dataset diversity (Yang et al. 2023), and modifying Transformer architectures (Hao et al. 2024; Theodoris et al. 2023). However, researchers have overlooked the potential of transcriptomics to pioneer token learning. Leveraging the properties of the transcriptomic dataset, we have successfully divided tokens and defined the Token Incremental Learning framework.

Method

Gene Learning in Transcriptomics

In single-cell transcriptomics, given the sample $(\boldsymbol{x}, \boldsymbol{v}) = (t_1, t_2, \ldots, t_l; v_1, v_2, \ldots, v_l)$, where $\boldsymbol{x}$ denotes genes and $\boldsymbol{v}$ denotes corresponding expression values, the gene learning strategy can be realized by masked value prediction (Yang et al. 2022, 2023) under a self-supervised framework. Define the training set as $\mathcal{D}=\{\boldsymbol{x}_i, \boldsymbol{v}_i\}_{i=1}^{N}$, and the output is the predictions for the masked values. Then, the loss function can be written as:

$$\mathcal{L}_{\text{tran}}(\mathcal{D}, \phi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j} \left\| v_{ij} - \hat{v}_{ij} \right\|^2, \tag{1}$$

where $\hat{v}_{ij}$ represents the corresponding predicted values, v_{ij} denotes the ground-truth value for j-th masked value in $\tilde{\boldsymbol{v}}_i$ and $\tilde{\boldsymbol{v}}_i$ denotes the masked input values. We will further elaborate on the value prediction process by Eq. (4) to Eq. (6), to demonstrate that learning the values associated with genes is equivalent to learning the genes themselves.

Gene Incremental Learning Formulation

Revisit Class Incremental Learning (CIL). CIL is designed to enable the models to progressively learn new classes, naturally introducing the concept of stages $S = (s_1, s_2, \ldots, s_n)$. In each stage, the model should learn different classes and thus CIL separates all classes Y into different stages $Y = (Y^{s_1}, Y^{s_2}, \ldots, Y^{s_n})$, where $Y = Y^{s_1} \cup Y^{s_2} \cup \cdots \cup Y^{s_n}$ and the classes designated to different stages are disjoint, i.e., $Y^{s_i} \cap Y^{s_j} = \varnothing, i \neq j, i, j = 1, 2, \ldots, n$. Since each class corresponds to specific samples in the classification dataset $\mathcal{D}$, distributing classes across various stages effectively partitions the dataset $\mathcal{D} = (\mathcal{D}^{s_1}, \mathcal{D}^{s_2}, \ldots, \mathcal{D}^{s_n})$, where $\mathcal{D}^{s_k} = \{(x, y), y \in Y^{s_k}\}, k = 1, 2, \ldots, n$.

The ultimate goal of CIL is to continually build a classification model for all seen classes. In other words, the model should not only learn classes from the current datasets but also preserve the classification ability learned from former datasets. Formally the objective function for the model ϕ in stage k is usually written as:

$$\mathcal{L}_{\text{CIL},s_k} = \mathcal{L}(\mathcal{D}^{s_k}, \phi) + \mathcal{L}(\bigcup_{i=1}^{k-1} \mathcal{D}^{s_i}, \phi), \tag{2}$$

where $\mathcal{L}(\mathcal{D}^{s_k}, \phi)$ represents the loss for the current dataset, designed to evaluate the classification ability of classes Y^{s_k}. The term $\mathcal{L}(\bigcup_{i=1}^{k-1} \mathcal{D}^{s_i}, \phi)$ quantifies the risk of model ϕ when performing previous datasets. Due to the invisibility of previous data, this term cannot be directly computed. Thus, it is typically implemented as an approximate constraint, such as mimicking the optimal model in the last stage $\phi^*_{s_{k-1}}$, which approximately reflects the former data distributions.

A good CIL model that minimizes Eq. (2) for every stage finally demonstrates discriminability across all classes, thereby fulfilling the initial goal for CIL.

Gene Incremental Learning (GIL). Inspired by the formulation of CIL, we define the Gene Incremental Learning framework, where our motivation is to enable the model to continuously learn genes in the context of single-cell transcriptomics. Although the transcriptomic data avoids the holistic nature of language when designing the incremental framework, there remain two significant challenges compared to CIL. First, unlike the direct correspondence between classes and samples, genes and samples do not align; each sample consists of all genes and their expression values, and thus dividing genes into different stages does not automatically partition datasets. Second, there is a significant difference in the roles of genes and classes; a class can represent the meaning of a sample as a standalone unit, while a gene and its value, being a single component of the sample, lacks the ability to express the sample's overall meaning.

To address the above challenges, we propose the separate partitioning of datasets and genes, along with a base gene mechanism. To be specific, given a dataset $\mathcal{D}$ and a gene set T, we partition both of them into different stages: $\mathcal{D} = (\mathcal{D}^{s_1}, \mathcal{D}^{s_2}, \ldots, \mathcal{D}^{s_n})$ and $T = ((B, T^{s_1}), (B, T^{s_2}), \ldots, (B, T^{s_n}))$. Here, some genes are designated as base genes B, which consistently appear in each stage, while the remaining genes are specifically assigned to different stages such that $T^{s_i} \cap T^{s_j} = \varnothing, i \neq j$ and $T = B \cup T^{s_1} \cup T^{s_2} \cup \cdots \cup T^{s_n}$. The base gene mechanism enables valid gene learning in each stage, reducing the risk of constructing meaningless samples with insufficient genes. For stage k, as shown in Figure 1, the dataset is represented as $\mathcal{D}^{s_k} = \{(b_1, b_2, \ldots, t_1, t_2, \ldots)_j\}, b_i \in B, t_i \in T^{s_k}$. In transcriptomics, values and genes correspond one-to-one. Thus, when genes are determined in each stage, the associated values are correspondingly divided, thereby we omit the notation of values here. Following Eq. (2), the GIL objective function for model ϕ in stage k is defined as:

$$\mathcal{L}_{\text{GIL},s_k} = \mathcal{L}(\mathcal{D}^{s_k}, T^{s_k}, \phi) + \sum_{i=1}^{k-1} \mathcal{L}(\mathcal{D}^{s_i}, T^{s_i}, \phi), \tag{3}$$

where $\mathcal{L}(\mathcal{D}^{s_k}, T^{s_k}, \phi)$ represents the loss associated with the current dataset, which can be implemented by Eq. (1). The

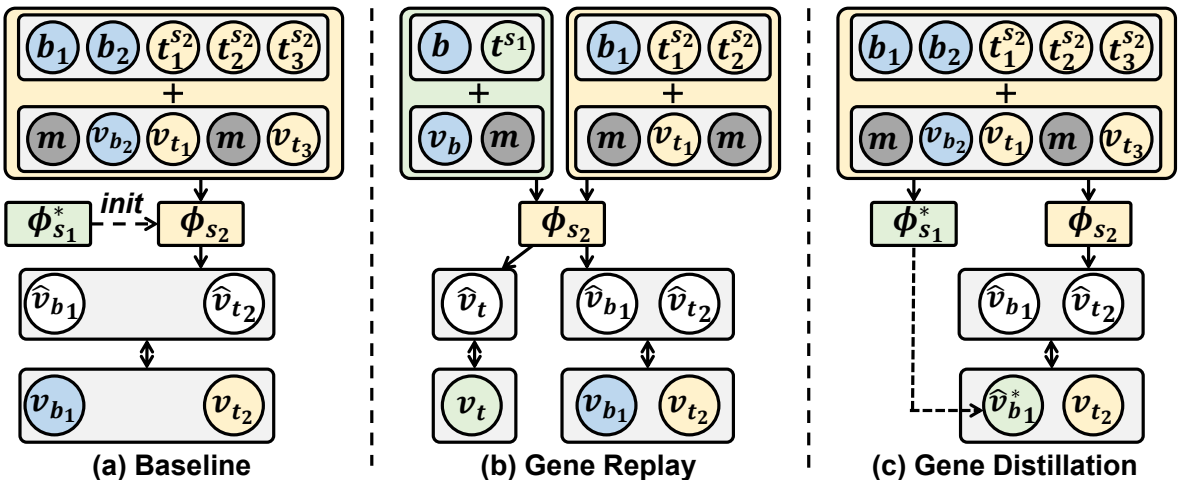

Figure 2: Illustrations of baseline methods for GIL in stage k, where we use $k = 2$ as an example, and the samples from stage 2 are in yellow background. (a) The baseline shows the masked token prediction loss formulated in Eq. (1). *init* denotes that the current model ϕ_{s_2} is initialized by the previous optimal model $\phi^*_{s_1}$ (b) Data Replay shows that some previous samples (with a green background) are maintained for training in the current stage. (c) Token Distillation shows how the previous optimal model distills knowledge through base token regression, which is formulated in Eq. (10).

second term also cannot be explicitly calculated and it is estimated to ensure performance across all seen genes. Note that both gene and dataset partitions could be randomized; alternatively, the partitioning of genes can follow a specific order. For example, we deliberately selected genes, which are crucial for downstream datasets, and assigned them to different stages to align the downstream datasets with stages for our proposed evaluation.

Incremental Learning Method Adaptations

To establish a comprehensive benchmark for our proposed Gene Incremental Learning, we draw inspiration from the methodologies of CIL, adapting several baseline methods to provide foundational directions for this field. For better formulation, we first provide a detailed explanation of feature extraction for genes in transcriptomics. Assume that the current stage is k, for a transcriptomic dataset $\mathcal{D}^{s_k}$, the input is $(\boldsymbol{x}, \boldsymbol{v})$ and the gene feature extraction process could be formulated as:

$$\boldsymbol{e} = \mathbf{E}_\phi(\boldsymbol{x}) + \tilde{\boldsymbol{v}}\mathbf{L}_{1,\phi}, \tag{4}$$

$$\boldsymbol{e}' = \mathbf{M}_\phi(\boldsymbol{e}), \tag{5}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{e}'\mathbf{L}_{2,\phi}, \tag{6}$$

where $\tilde{\boldsymbol{v}}$ is masked values $\boldsymbol{v}$, $\hat{\boldsymbol{v}}$ is the predicted values, $\mathbf{E}_\phi$ is gene embeddings layer, $\mathbf{L}_{1,\phi} \in \mathbb{R}^{1\times d}$ is a linear layer that encodes the values into embeddings, $\mathbf{L}_{2,\phi} \in \mathbb{R}^{d\times 1}$ is a linear layer to predict masked values from encoded features, d is the hidden dimension, $\mathbf{M}_\phi$ is the backbone, which is usually implemented as Transformers. According to Eq. (4), genes are bound to corresponding values, demonstrating learning values are indeed learning genes as we mentioned.

Baseline and Oracle. As shown in Figure 2(a), our baseline is defined as optimizing the model only using the current dataset $\mathcal{D}^{s_k}$ for learning T^{s_k} at stage k, while ignoring the second term in Eq. (3). The loss can be written as:

$$\mathcal{L}_{\text{base},s_k} = \mathcal{L}_{\text{tran}}(\mathcal{D}^{s_k}, \phi). \tag{7}$$

Here we omit the input T^{s_k} as formulated in Eq. (3) because T^{s_k} is already bound to $\mathcal{D}^{s_k}$ based on our GIL design. Note that the parameters ϕ at each stage are initialized with the optimal parameters trained from the previous stage. Unless otherwise specified, we assume $k > 1$ because stage one can only take the baseline training.

To establish an upper bound for reference, we train the model on all datasets $\{\mathcal{D}^{s_i}\}_{i=1}^n$ derived from GIL splits, as the oracle method:

$$\mathcal{L}_{\text{oracle}} = \sum_{i=1}^{n} \mathcal{L}_{\text{tran}}(\mathcal{D}^{s_i}, \phi). \tag{8}$$

The oracle is expected to provide the best global performance across all genes, while for specific genes at a given stage, the performance may not exceed that of the baseline.

Gene Replay. A mainstream method in CIL leverages the incremental settings by retaining a subset of samples from previous datasets for current training, referred as data replay, and in our GIL framework, the gene replay strategy could also be implemented as data replay. Some advanced methods are proposed such as dataset condensation (Mitra, Murthy, and Pal 2000) to further improve the performance. To provide a basic reference for our GIL benchmark, we evaluate the native implementation:

$$\mathcal{L}_{\text{dr},s_k} = \mathcal{L}_{\text{tran}}(\mathcal{D}^{s_k}, \phi) + \sum_{i=1}^{k-1} \mathcal{L}_{\text{tran}}(\mathcal{D}^{s_i}_{\text{dr}}, \phi), \tag{9}$$

where $\mathcal{D}^{s_i}_{\text{dr}} \subset \mathcal{D}^{s_i}$ is a subset for previous dataset $\mathcal{D}^{s_i}$, and the training pipeline in stage k is shown in Figure 2(b).

Gene Distillation. Another method in CIL involves distilling knowledge from the previous model, which assumes old models can represent the current sample by old classes. In GIL, we adapt the class distillation to gene distillation, as shown in Figure 2(c). According to Eq. (1), we have:

$$\mathcal{L}_{\text{fd},s_k} = \frac{1}{N_k}\sum_{i=1}^{N_k}(\sum_j \|v_{ij}-\hat{v}_{ij}\|^2+\lambda\|\hat{\boldsymbol{v}}_i-\hat{\boldsymbol{v}}^*_{i,s_{k-1}}\|^2), \quad (10)$$

where N_k denotes the number of samples in this stage, $\hat{v}_{ij}$ is the prediction for the masked values, $\hat{\boldsymbol{v}}^*_{i,s_{k-1}}$ denotes the output derived from the optimal model $\phi^*_{s_{k-1}}$ in the last stage, λ is the coefficient for distillation. Note that the specific genes for the current stage are removed from the second term in the implementation due to $\phi^*_{s_{k-1}}$ do not have the ability to predict the unseen genes.

Evaluations

Gene-wise Regression. The most straightforward method to evaluate gene learning is using masked gene prediction loss in Eq. (1). As multiple genes are learned in a single stage, we average the performance across all genes learned specifically in that stage. For stage k, we have:

$$\mathcal{L}_{\text{regress},s_k} = \mathbb{E}[\sum_k \|v_{ik} - \hat{v}^*_{ik}\|^2], t_{ik} \in T^{s_k}, \quad (11)$$

where $\hat{v}^*_{ik}$ is the predicted masked values by the optimal model trained in the current stage $\phi^*_{s_k}$, v_{ik} is the expression value of its corresponding gene t_{ik}, and T^{s_k} denotes the set of learned specific genes in stage k.

Gene-based Classification. The above evaluation might not provide a universally comparable measure due to variations in different datasets and value scales. Therefore we design gene-based classification as another evaluation for GIL. Specifically, we identified some crucial genes for different downstream transcriptomics classification tasks and divided these genes into different stages. Then, at each stage, we can assess the gene performance through the corresponding downstream task. For example, for a downstream dataset $\mathcal{D}_{d_1}$, where T^{s_1} is crucial for its classification, we learn T^{s_1} at stage one and then measure how well the model retains T^{s_1} through tests on $\mathcal{D}_{d_1}$ in the following stages. The downstream classification loss is written as:

$$\mathcal{L}_{\text{class},s_k} = \frac{1}{N_{d_k}}\sum_{i=1}^{N_{d_k}} -\boldsymbol{y}_i \cdot \log p(\boldsymbol{e}'^*_{s_k}\mathbf{L}), \quad (12)$$

where N_{d_k} is the number of samples in downstream dataset $\mathcal{D}_{d_k}$, $\boldsymbol{y}_i$ is the one-hot class label, $\mathbf{L}$ is a trainable linear layer to project the feature into class space and $\boldsymbol{e}'^*_{s_k}$ is the extracted feature by the optimal model $\phi^*_{s_k}$, formulated in Eq. (4) and Eq. (5), which is frozen to only extract gene features.

Experiment

Dataset

In this paper, we leveraged the data collection method outlined by scGPT (Cui et al. 2024), drawing from the CELLxGENE collection (Megill et al. 2021; Biology et al. 2023), which consists of human cell data characterized by gene-expression pairs. This extensive dataset covers over 50 organs and tissues such as blood and heart, derived from more than 400 studies, providing a comprehensive view of cellular diversity within the human body. We randomly selected 906,890 samples for training and 204,871 samples for gene-wise evaluation. We also followed scGPT (Cui et al. 2024) to construct the gene vocabulary consisting of 60,697 genes.

For gene-based downstream classification, we collected six transcriptomic datasets for comprehensive evaluations: Norman (Norman et al. 2019) explores the relationship between the set of genes expressed by a cell and its phenotype; Lupus (Perez et al. 2022) shows an increase in type 1 interferon-stimulated genes; Inhibitor Colitis (ICol) (Thomas et al. 2024) reveals the interactions between circulating T cells and epithelial cells; Adamson (Adamson et al. 2016) applies Perturb-seq to dissect the mammalian unfolded protein response; Pancreas (Panc) (Chen et al. 2023) consolidates data from five human pancreas studies; and Myeloid (Myel) (Cheng et al. 2021) provides a comprehensive pan-cancer analysis of myeloid cells.

Implementation Details

To ensure consistent and fair comparisons, we configured the same model and training parameters for all experiments. We followed the scGPT (Cui et al. 2024) and employed a Transformer as the feature extractor with 6 layers, 8 heads for multi-head attention, and hidden dimensions of 256. The experiments were conducted on an 8-NVIDIA A100 GPU server. In the GIL training, we applied a batch size of 128, and the Adam (Kingma 2014) optimizer with a learning rate of 0.0005 across 5 epochs for each stage, and here a warm-up strategy is applied in the first 5,000 iterations for all methods. We selected important genes for downstream datasets based on their cumulative gene values calculated across all samples in the corresponding datasets and removed duplicate genes across datasets to prevent confusion.

For gene-wise evaluation, we only considered the seen stage-specific genes in each stage and calculated the average loss across these genes. Following scGPT, the length of the input is limited to 512, and genes are randomly selected. Therefore, the evaluations for each trial may contain different genes for each sample. To mitigate randomness, we construct a large-scale evaluation set, and the experimental results demonstrate that gene-wise regression evaluation is stable. For gene-based classification, the GIL model was frozen to extract features, and only a single linear layer was optimized. All experiments were independently conducted three times, and the average performance was reported. More details can be found in the Appendix.

Result Analysis

Q1. *Is our GIL framework and evaluation reasonable?*

A1. The performance of the baseline in Table 1, Table 2, Table 3, and Figure 3 demonstrate that, in the absence of any knowledge-preserving methods, the model progressively forgets previously learned genes. In the 2-stage GIL settings, the performance drops by 0.279 in regression and 1.816% in downstream classification, on average, and drops

Method	Stage	Norman	Lupus	Δ	ICol	Adamson	Δ	Lupus	Panc	Δ	Avg
Baseline	1	0.172	-	-	0.164	-	-	0.145	-	-	-
	2	0.424	0.134	0.253	0.496	0.137	0.333	0.397	0.204	0.252	0.279
Oracle	-	0.173	0.136	-	0.164	0.115	-	0.145	0.206	-	-
Replay	1	0.172	-	-	0.164	-	-	0.146	-	-	-
	2	0.215	0.134	0.043	0.200	0.124	0.036	0.177	0.213	0.031	0.037
Distill	1	0.172	-	-	0.163	-	-	0.147	-	-	-
	2	0.365	0.139	0.193	0.420	0.145	0.257	0.332	0.220	0.185	0.212

Table 1: Averaged regression loss for specific genes in each stage for three 2-stage GIL settings on evaluation set (Gene-wise Regression). The three settings are Norman-Lupus, ICol-Adamson, and Lupus-Panc. The model learns crucial genes (T^{s_k}) for the associated dataset at each stage, thus using the name of the dataset to represent corresponding genes. Δ represents the forgetting of genes learned in the previous stage (here is stage 1), as reflected by the difference in regression loss. Avg denotes the averaged Δ across three GIL settings. The smaller the absolute value of Δ, the better, and lower regression losses for others are preferred. The default replay number of samples is 1,000 and the default λ of distillation is 5.0. "-" denotes the result is not applicable. Results are the mean of three independent trials.

Method	Stage	ICol	Myel	Panc	Δ
Baseline	1	0.163	-	-	-
	2	0.452	0.263	-	0.289
	3	0.498	0.290	0.192	0.181
Oracle	-	0.163	0.209	0.175	-
Replay	1	0.163	-	-	-
	2	0.205	0.263	-	0.042
	3	0.209	0.272	0.190	0.028
Distill	1	0.164	-	-	-
	2	0.437	0.248	-	0.273
	3	0.444	0.323	0.214	0.177

Table 2: Averaged regression loss for specific genes at each stage in a 3-stage GIL setting (ICol-Myel-Panc) on the evaluation set. Selected genes for each dataset correspond to the specific genes at each respective stage. Δ denotes the averaged forgetting for the previous genes associated with their datasets, e.g., Δ in Stage 3 is calculated by $((\text{ICol}_{s_3} - \text{ICol}_{s_1}) + (\text{Myel}_{s_3} - \text{Myel}_{s_2}))/2$, where the subscript denotes the performance of the dataset at that stage. Other settings are the same as those in Table 1.

by 0.181 in regression in the 3-stage GIL setting. This confirms the gene forgetting problem in GIL, justifying the effectiveness of our proposed GIL framework. Furthermore, by comparing Table 1 and Table 3, we observe that both of the evaluations, gene-wise regression, and gene-based classification, show declines with the increase of stages when evaluating the baseline, indicating the consistency of our proposed evaluations.

Q2. *Are the adapted methods we used effective?*

A2. In Table 1 and Table 2, we observe that both methods achieve improvements in gene-wise regression evaluation across different settings, indicating that they are effective and can prevent gene forgetting. As shown in Table 4, we also find that both methods consistently reduce gene forgetting, and as the hyperparameters increase, better performance is achieved, where the best performance is 0.018 and 0.154 for gene replay and gene distillation, respectively, under the selected scope of hyperparameters.

Note that, for gene replay, as the number of replayed samples increases, it degenerates into the oracle method. For gene distillation, as λ increases, the model may retain more previous knowledge, but it could impair the learning of new genes. For example, in Table 4, with λ increasing from 0.5 to 10.0, the loss for Lupus increases from 0.134 to 0.143, demonstrating that learning for stage 2 is gradually failing.

Q3. *Why does gene distillation perform well in gene-wise regression but poorly in gene-based classification?*

A3. In Table 1, we find that the method gene distillation achieves an average Δ of 0.212 outperforming the baseline of 0.279. However, in Table 3, its average performance drop is 2.473%, which is worse than the baseline of 1.816%. This discrepancy indicates that gene distillation has limitations. A possible reason is that gene distillation indeed prevents gene forgetting on average, but degrades the features of the genes learned by the model. This conflict highlights the effectiveness and comprehensiveness of the two evaluations we proposed. Only methods that show consistent improvements across all evaluations could be considered robust.

Q4. *Why are the oracle accuracies different for the same dataset across different settings?*

A4. The reason is we removed crucial genes shared across downstream datasets when constructing different settings, to prevent cross-contamination of downstream classifications between datasets. For example, in Table 1, although both the Norman-Lupus and Lupus-Panc settings include the Lupus, the selected crucial genes for Lupus are different and thus induce the mentioned phenomenon.

Q5. *Why have we only provided a few settings, and why is the number of stages limited?*

A5. According to our proposed gene-based classification, we should identify genes that are crucial for each downstream dataset. However, all transcriptomic downstream datasets have a large amount of overlapping crucial genes. For example, in Table 3, the baseline achieves 67.313% on Lupus at the first stage because some base genes and specific genes for Norman could help its classification. If a gene is crucial for many datasets but is assigned to a specific stage,

Method	Stage	Norman	Lupus	Δ	ICol	Adamson	Δ	Lupus	Panc	Δ	Avg
Baseline	1	37.734	67.313	-	59.370	42.209	-	74.451	93.542	-	-
	2	35.590	75.386	-2.144	56.771	43.514	-2.599	73.747	97.826	-0.704	-1.816
Oracle	-	38.112	75.422	-	59.201	43.782	-	74.893	97.913	-	-
Replay	1	37.734	67.315	-	59.366	42.209	-	74.456	93.550	-	-
	2	36.449	75.000	-1.285	57.738	43.618	-1.628	74.143	97.948	-0.313	-1.075
Distill	1	37.902	67.577	-	59.367	42.209	-	74.659	93.565	-	-
	2	34.162	72.939	-3.740	56.347	42.480	-3.020	74.000	97.441	-0.659	-2.473

Table 3: Test accuracy (%) for three 2-stage GIL settings on downstream classification datasets (Gene-based Classification). The specific genes T^{s_k} for each stage are crucial for the associated dataset classification (e.g., in the first settings, the specific genes in stage 1 are important for Norman classification). Δ represents the forgetting of genes, as reflected by the difference in classification accuracy. The smaller the absolute value of Δ, the better, while higher accuracies for others are preferred. Other settings are the same as those in Table 1.

Method	Params	Norman	Lupus	Δ
Baseline	1	0.172	-	-
	2	0.424	0.134	0.253
Replay	50	0.293	0.136	0.121
	10^2	0.263	0.138	0.091
	10^3	0.215	0.134	0.043
	10^4	0.190	0.133	0.018
Distill	0.5	0.420	0.134	0.248
	1.0	0.402	0.135	0.230
	5.0	0.365	0.139	0.193
	10.0	0.326	0.143	0.154

Table 4: Averaged regression loss for ablation studies for gene replay and gene distillation under setting Norman-Lupus on the evaluation set. Except for the baseline method, all other methods only report the performance for stage 2. Params for baseline denote stages, for gene replay denotes the numbers of replayed samples, and for the distillation denote coefficients λ in Eq. (10). For the regression results, smaller values are better.

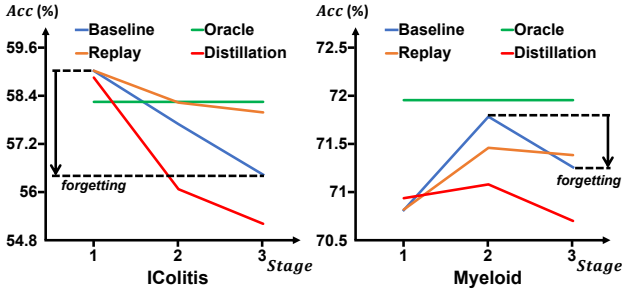

Figure 3: Test Accuracy (%) for the 3-stage GIL setting (ICol-Myel-Panc) on the corresponding downstream classification datasets. The crucial genes for the last dataset Panc only learned in the last stage, thus there is no forgetting problem and we omit the results for Panc here.

the gene-based classification evaluation will fail. Thus, it is not as straightforward as just splitting two datasets into two stages to create a setting. In our experiment, we selected the above effective settings through preliminary experiments to serve as the benchmark settings for GIL, which is also one of our contributions.

Q6. *What are the key differences between CIL and GIL according to the evaluations?*

A6. The class as the label has a significant impact on classification performance, and new classes are exclusive from old classes, leading to the forgetting problem in CIL being easily observed and intuitively reflected in the results. However, for GIL, evaluating forgetting is more challenging. A gene is only a small part of the input, so measuring its significance is not obvious. For example, even if a gene is not learned, it may not influence the downstream tasks. Even though we construct the GIL framework and consistent evaluations, it is still difficult to observe a significant performance drop, as shown in Table 3, unlike CIL. Therefore, we propose the GIL framework, evaluations, and benchmark to illustrate the potential for this task and draw more attention to establishing better GIL evaluations and methods in this field.

Conclusion

In this paper, we introduce Gene Incremental Learning (GIL), a novel framework for single-cell transcriptomics to address the problem of the gradual growth of genes. In this framework, we propose a series of novel designs to address the challenges in creating the GIL framework, including defining base genes to prevent the generation of semantically meaningless samples, and designing stage-specific gene subsets alongside corresponding datasets, thereby establishing a semantic evaluation protocol under the transcriptomics settings. Through extensive experiments, we demonstrate the rationale behind our GIL framework and the effectiveness of the proposed evaluation protocol and method adaptations, establishing a comprehensive benchmark for GIL in single-cell transcriptomics. For future work, we will try to design specific GIL algorithms, and we also aim to extend the framework into other token-learning fields.

Acknowledgments

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA0460205.

References

Adamson, B.; Norman, T. M.; Jost, M.; Cho, M. Y.; Nuñez, J. K.; Chen, Y.; Villalta, J. E.; Gilbert, L. A.; Horlbeck, M. A.; Hein, M. Y.; et al. 2016. A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7): 1867–1882.

AlJanahi, A. A.; Danielsen, M.; and Dunbar, C. E. 2018. An introduction to the analysis of single-cell RNA-sequencing data. *Molecular Therapy Methods & Clinical Development*, 10: 189–196.

Biology, C. S.-C.; Abdulla, S.; Aevermann, B.; Assis, P.; Badajoz, S.; Bell, S. M.; Bezzi, E.; Cakir, B.; Chaffer, J.; Chambers, S.; et al. 2023. CZ CELLxGENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *BioRxiv*, 2023–10.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, 233–248.

Chen, J.; Xu, H.; Tao, W.; Chen, Z.; Zhao, Y.; and Han, J.-D. J. 2023. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1): 223.

Chen, Z.; and Liu, B. 2018. *Lifelong machine learning*. Morgan & Claypool Publishers.

Cheng, S.; Li, Z.; Gao, R.; Xing, B.; Gao, Y.; Yang, Y.; Qin, S.; Zhang, L.; Ouyang, H.; Du, P.; et al. 2021. A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells. *Cell*, 184(3): 792–809.

Collin, J.; Queen, R.; Zerti, D.; Bojic, S.; Dorgau, B.; Moyse, N.; Molina, M. M.; Yang, C.; Dey, S.; Reynolds, G.; et al. 2021. A single cell atlas of human cornea that defines its development, limbal progenitor cells and their interactions with the immune cells. *The ocular surface*, 21: 279–298.

Constantinescu, A. O.; O'Reilly, J. X.; and Behrens, T. E. 2016. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292): 1464–1468.

Cui, H.; Wang, C.; Maan, H.; Pang, K.; Luo, F.; Duan, N.; and Wang, B. 2024. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 1–11.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385.

Dong, S.; Hong, X.; Tao, X.; Chang, X.; Wei, X.; and Gong, Y. 2021. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1255–1263.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Gustafsson, J. K.; and Johansson, M. E. 2022. The role of goblet cells and mucus in intestinal homeostasis. *Nature reviews Gastroenterology & hepatology*, 19(12): 785–803.

Hao, M.; Gong, J.; Zeng, X.; Liu, C.; Guo, Y.; Cheng, X.; Wang, T.; Ma, J.; Zhang, X.; and Song, L. 2024. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 1–11.

Hinton, G. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 831–839.

Hu, X.; Tang, K.; Miao, C.; Hua, X.-S.; and Zhang, H. 2021. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 3957–3966.

Kang, M.; Park, J.; and Han, B. 2022. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16071–16080.

Karaayvaz, M.; Cristea, S.; Gillespie, S. M.; Patel, A. P.; Mylvaganam, R.; Luo, C. C.; Specht, M. C.; Bernstein, B. E.; Michor, F.; and Ellisen, L. W. 2018. Unravelling subclonal heterogeneity and aggressive disease states in TNBC through single-cell RNA-seq. *Nature communications*, 9(1): 3588.

Ke, Z.; and Liu, B. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

Kharchenko, P. V. 2021. The triumphs and limitations of computational methods for scRNA-seq. *Nature methods*, 18(7): 723–732.

Kingma, D. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lehrer, A. 2003. Understanding trendy neologisms. *Italian Journal of Linguistics*, 15: 369–382.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2018. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12): 2346–2363.

Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A. D.; and Van De Weijer, J. 2022. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5513–5533.

McCaffary, D. 2021. Towards continual task learning in artificial neural networks: current approaches and insights from neuroscience. *arXiv preprint arXiv:2112.14146*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Megill, C.; Martin, B.; Weaver, C.; Bell, S.; Prins, L.; Badajoz, S.; McCandless, B.; Pisco, A. O.; Kinsella, M.; Griffin, F.; et al. 2021. Cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices. *bioRxiv*, 2021–04.

Mitra, P.; Murthy, C.; and Pal, S. K. 2000. Data condensation in large databases by incremental learning with support vector machines. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, 708–711. IEEE.

Norman, T. M.; Horlbeck, M. A.; Replogle, J. M.; Ge, A. Y.; Xu, A.; Jost, M.; Gilbert, L. A.; and Weissman, J. S. 2019. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455): 786–793.

Olaniru, O. E.; Kadolsky, U.; Kannambath, S.; Vaikkinen, H.; Fung, K.; Dhami, P.; and Persaud, S. J. 2023. Single-cell transcriptomic and spatial landscapes of the developing human pancreas. *Cell Metabolism*, 35(1): 184–199.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pentina, A. 2016. *Theoretical foundations of multi-task lifelong learning*. Ph.D. thesis.

Perez, R. K.; Gordon, M. G.; Subramaniam, M.; Kim, M. C.; Hartoularos, G. C.; Targ, S.; Sun, Y.; Ogorodnikov, A.; Bueno, R.; Lu, A.; et al. 2022. Single-cell RNA-seq reveals cell type–specific molecular and genetic associations to lupus. *Science*, 376(6589): eabf1970.

Qin, C.; and Joty, S. 2022. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *ICLR*.

Ramachandran, P.; Matchett, K. P.; Dobie, R.; Wilson-Kanamori, J. R.; and Henderson, N. C. 2020. Single-cell technologies in hepatology: new insights into liver biology and disease pathogenesis. *Nature reviews Gastroenterology & hepatology*, 17(8): 457–472.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Shi, H.; Xu, Z.; Wang, H.; Qin, W.; Wang, W.; Wang, Y.; Wang, Z.; Ebrahimi, S.; and Wang, H. 2024. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*.

Silver, D. L.; and Mercer, R. E. 2002. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2002 Calgary, Canada, May 27–29, 2002 Proceedings 15*, 90–101. Springer.

Silver, D. L.; Yang, Q.; and Li, L. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*.

Tang, F.; Barbacioru, C.; Wang, Y.; Nordman, E.; Lee, C.; Xu, N.; Wang, X.; Bodeau, J.; Tuch, B. B.; Siddiqui, A.; et al. 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5): 377–382.

Theodoris, C. V.; Xiao, L.; Chopra, A.; Chaffin, M. D.; Al Sayed, Z. R.; Hill, M. C.; Mantineo, H.; Brydon, E. M.; Zeng, Z.; Liu, X. S.; et al. 2023. Transfer learning enables predictions in network biology. *Nature*, 618(7965): 616–624.

Thomas, M. F.; Slowikowski, K.; Manakongtreecheep, K.; Sen, P.; Samanta, N.; Tantivit, J.; Nasrallah, M.; Zubiri, L.; Smith, N. P.; Tirard, A.; et al. 2024. Single-cell transcriptomic analyses reveal distinct immune cell contributions to epithelial barrier dysfunction in checkpoint inhibitor colitis. *Nature Medicine*, 1–14.

Voigt, A. P.; Mullin, N. K.; Stone, E. M.; Tucker, B. A.; Scheetz, T. E.; and Mullins, R. F. 2021. Single-cell RNA sequencing in vision research: Insights into human retinal health and disease. *Progress in retinal and eye research*, 83: 100934.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 374–382.

Yang, F.; Wang, W.; Wang, F.; Fang, Y.; Tang, D.; Huang, J.; Lu, H.; and Yao, J. 2022. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence*, 4(10): 852–866.

Yang, X.; Liu, G.; Feng, G.; Bu, D.; Wang, P.; Jiang, J.; Chen, S.; Yang, Q.; Zhang, Y.; Man, Z.; et al. 2023. Genecompass: Deciphering universal gene regulatory mechanisms with knowledge-informed cross-species foundation model. *bioRxiv*, 2023–09.

Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; and Kuo, C.-C. J. 2020. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1131–1140.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13208–13217.

Zhu, F.; Cheng, Z.; Zhang, X.-Y.; and Liu, C.-l. 2021. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34: 14306–14318.