scWEAVE: An Efficient Multi-Species Single-Cell Foundation Model via Meta-Gene Compression and Contrastive Alignment

Anonymous authors
Paper under double-blind review

Abstract

Single-cell foundation models (scFMs) show great promise across various biological tasks. However, most existing models are computationally expensive and trained mainly on human data, limiting their applicability to multi-species analysis. Extending scFMs across species is non-trivial: gene sets differ substantially across species, and biological divergence together with batch effects induce significant cross-species distribution shift. We present scWEAVE, an efficient scFM that jointly reduces computational cost and generalizes across species. We introduce a gene compression module that uses knowledge from protein language models and gene ontology to softly compress genes from any species into a shared, compact set of meta-genes, making cross-species data directly comparable while shortening the token sequence the transformer must process. To address distribution shift, scWEAVE decomposes each cell's total variation into a shared cell factor encoding shared cell-state variation and a remnant factor that supplies species-specific information needed for accurate masked expression prediction – the main objective of scFMs. We further align the shared cell factor across species using a supervised contrastive regularizer based on cell-ontology identity, which accounts for inconsistent label granularity between species-specific annotations and corrects for residual batch-effect distortion during pretraining. We validate scWEAVE through extensive experiments on datasets spanning multiple species. scWEAVE matches or exceeds baseline performance on ortholog retrieval, gene expression imputation, batch effect removal, and novel cell type discovery, while cutting inference time up to 35x and memory up to 6x relative to TranscriptFormer.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) measures gene expression at single-cell resolution, enabling modern experiments to profile thousands to millions of cells. The rapid accumulation of scRNA-seq data has led to large-scale resources such as the Human Cell Atlas (Regev et al., 2017), the Human Ensemble Cell Atlas (Chen et al., 2022), and CELLxGENE (Program et al., 2023). By aggregating cells from diverse tissues, organisms, disease states, and experimental conditions, these atlases provide an unprecedented resource for learning generalizable representations of cellular states.

Single-cell foundation models (scFMs) aim to exploit these atlases through large-scale self-supervised pretraining. Many scFMs adopt Transformer-based architectures and represent each cell as a "cell sentence," in which genes constitute tokens and their expression measurements are encoded as token attributes (Cui et al., 2024; Hao et al., 2024; Bian et al., 2024a). These models are commonly pretrained using masked gene or expression prediction, and used to generate cell representations for tasks such as batch-effect removal, cell type identification, and gene expression imputation.

Models such as scGPT (Cui et al., 2024), scFoundation (Hao et al., 2024), and scMulan (Bian et al., 2024a) have demonstrated promising performance in various downstream tasks. However, these models are primarily focusing on human-centered tasks. Recently, single-cell foundation models are starting to expand to non-human species. UCE (Rosen et al., 2026) was pretrained on more than 36 million cells from eight species. TranscriptFormer (Pearce et al., 2026) further scales pretraining to 112 million cells from 12 species and reports strong cross-species transfer. Nevertheless, the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

increasing scale of these models introduces substantial training, adaptation, and inference costs, limiting their practical application in biological research.

On the other hand, cross-species analysis also amplifies the distribution distortion when integrating different sources of scRNA-seq data. Measurements obtained from different experiments are affected by batch effects arising from differences in sample handling, sequencing technologies, and processing protocols (Luecken & Theis, 2019; Zhang et al., 2024). Consequently, cells of the same cell state may exhibit markedly different expression distributions across datasets. This problem is more pronounced across species. Firstly, the gene sets differ across species, making it hard to compare data across species. Moreover, cross-species biological differences coexist with batch effect, making it difficult to align conserved cell states without eliminating meaningful species-specific biology (Rosen et al., 2024).

In this work, we address computational efficiency and cross-species generalization jointly by introducing **scWEAVE**, an efficient multi-species single-cell foundation model. scWEAVE introduces a gene compression module that softly maps genes from different species onto a compact, shared set of meta-genes, guided by multi-source gene function knowledge including protein-language-model representations and Gene Ontology annotations. Using these meta-gene tokens, scWEAVE constructs a "cell sentence" for each cell and processes it with a transformer architecture to extract cell-level representations. Reducing the tokens in a cell sentence to a small set of meta-genes significantly lowers the computational cost of self-attention, while the shared meta-gene vocabulary enables direct feature comparison across species. The primary pretraining objective, masked expression prediction, is not enough to resolve the distribution distortion across species caused by batch effect. To address this challenge, scWEAVE decomposes the total variation of each cell's expression into a shared cell factor, which encodes cell-state variation common across batches and species, and a remnant factor, which supplies the species- and batch-specific information needed for accurate masked expression prediction. We further introduce a supervised contrastive regularizer that aligns common cell states across species and removes batch effect in the shared cell factor. To handle mismatched label naming and granularity across species, we project labels onto a common cell-ontology vocabulary and incorporate the ontology hierarchy directly into the contrastive loss.

We evaluate scWEAVE on multi-species single-cell datasets across four tasks: gene expression imputation, ortholog retrieval, batch effect removal, and novel cell type discovery. scWEAVE achieves comparable or superior performance against state-of-the-art (SOTA) scFMs, while substantially reducing inference time and memory usage. These findings highlight substantial room for improvement in scFM architecture and motivate rethinking model design rather than pursuing increased model scale. By combining computational efficiency with strong performance across heterogeneous datasets, scWEAVE makes foundation models more accessible for biological research.

We summarize our contributions of this work as follows:

- We propose scWEAVE, a single-cell foundation model that is computationally efficient and generalizes well to multi-species applications.
- We introduce a gene compression module that reduces the number of gene tokens per cell sentence while preserving key biological information, lowering computational cost and enabling direct feature comparison across species.
- We decompose each cell's total variation into a shared cell factor encoding batch- and species-invariant cell-state variation and a remnant factor supplying complementary information for masked expression prediction, supporting both accurate prediction and batch effect removal.
- We introduce a supervised contrastive regularizer that aligns common cell states across species while accounting for mismatched label naming across species.
- We demonstrate across four evaluation tasks that a compact scFM can achieve competitive performance with much lower inference time and memory usage, highlighting architectural design as an important avenue for improving scFMs beyond increasing model scale.

2 Related work

single-cell Foundation Model. Most scFMs use Transformer architectures to represent cells as sequences of gene tokens, with expression levels encoded through token ordering or associated values

(Cui et al., 2024; Bian et al., 2024b; Theodoris et al., 2023). Common pretraining objectives include masked prediction and auto-regressive prediction. Geneformer (Theodoris et al., 2023) ranks genes by relative expression and is trained to predict masked gene identities. scGPT (Cui et al., 2024) separately encodes gene identities and expression values, and is trained by iteratively reconstructing masked expression. scMulan (Bian et al., 2024a) formulates single-cell analysis as conditional generation, and is trained using an auto-regressive next-token prediction.

scFM for multi-species application. GeneCompass (Yang et al., 2024) was among the first explicitly cross-species scFMs, but is limited to human and mouse. Following the protein-embedding approach introduced by SATURN (Rosen et al., 2024), UCE (Rosen et al., 2026) and scPRINT (Kalfon et al., 2025) represent genes using protein-language-model embeddings, supporting transfer to unseen species with available protein sequences. More recent models, such as TranscriptFormer (Pearce et al., 2026), have expanded both pretraining scale and species coverage. However, this expansion comes with substantial training and inference costs.

Efficient design of single-cell foundation models. Computational efficiency remains a major challenge for scFMs because self-attention scales quadratically with the number of gene tokens. Human transcriptomes contain approximately 20,000 protein-coding genes, making full-transcriptome modeling computationally expensive. To shorten input sequences, models such as Geneformer, GeneCompass, and scPRINT keep only highly expressed genes for modeling (Theodoris et al., 2023; Yang et al., 2024; Rosen et al., 2026; Kalfon et al., 2025). However, gene selection discards biological signals from weakly expressed genes, and attention costs remain substantial even for 1000–2000 highly expressed genes. Other models improve the underlying architecture: CellFM (Zeng et al., 2025) adopts the linear-complexity ERetNet architecture, while scLong (Bai et al., 2026) uses Performer layers and separate processing paths for highly and weakly expressed genes. Despite these architectural improvements, full-transcriptome modeling remains computationally demanding.

3 scWEAVE

3.1 Overview of scWEAVE

scWEAVE uses an encoder-only transformer architecture and is pretrained with masked expression prediction (MEP) as its main objective: scWEAVE masks the expression of a small portion of genes when feeding the data into the transformer, then predicts the masked-out expression from the transformer's output embeddings. scWEAVE differs from existing scFMs in three key design choices: (1) a gene compression module that compresses genes into a compact set of meta-genes based on multi-source gene functional features; (2) a decomposition of the encoder output used for masked prediction into a shared cell factor and a remnant factor; and (3) a cell-ontology-informed contrastive loss that aligns the distribution of shared cell factor of common cell states across batches and species.

The gene compression module takes gene functional embeddings derived from protein-language-model representations and Gene Ontology annotations as input, and softly decomposes each gene into a weighted combination of a compact set of 512 meta-genes. The module is trained to group functionally similar genes into a common meta-gene (Figs. 1a,b, detailed in Sec. 3.2) and retain biological variation between different meta-genes. This compression shortens the token sequence processed by the transformer, reducing the computational cost of self-attention.

The expression data of each cell is then transformed into meta-gene-level sentence before feeding into the transformer . scWEAVE derives two complementary signals from the transformer output: a shared cell factor from the output token at the CLS position, and a remnant factor for each masked gene (Fig. 1c). The predictor then fuses these two signals together to represent the cell identity, and predicts the expression of the query masked gene conditioning on the cell (Fig. 1c, details in Sec. 3.3). To remove the batch effect from the shared cell factor, scWEAVE introduces a supervised contrastive loss as a regularizer during training. The contrastive loss uses the widely available cell type labels in the training data as supervision, aligning cells of the same cell type label across batches and species. A challenge, however, is that cell type labels from different batches (usually from different sources) are annotated at different granularities. For example, cells of a certain type (e.g., T cell) may carry a detailed subtype annotation in one experiment (e.g., CD4+ T cell) and only a coarse annotation in another (e.g., immune cell); labels from different species may also follow entirely different naming conventions. We map labels onto standardized cell-ontology terms to make them

directly comparable across datasets, and design the contrastive loss over these mapped labels to account for the hierarchical granularity relationships between cell-ontology terms.

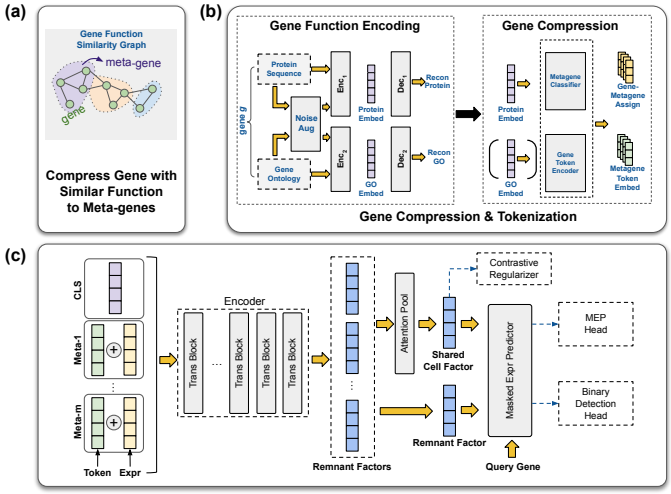

Figure 1: Overview of scWEAVE. (a) scWEAVE compresses functionally similar genes into meta-genes. (b) Architecture of gene compression and tokenization. (c) Transformer architecture of scWEAVE, details are provided in Appendix Sec. A.5.

3.2 Gene Compression with Protein Language Model and Gene Ontology Annotations

The gene compression module compresses functionally similar genes into shared meta-genes (Fig. 1a). To characterize gene function, we integrate two sources of functional knowledge: (1) gene-specific protein representations generated by the ESM-2 protein language model (Lin et al., 2022), which capture functional properties of the proteins encoded by individual genes; and (2) Gene Ontology (GO) annotations, which describe genes using a standardized set of functional terms (Ashburner et al., 2000). The two sources provide a complementary view of gene function from protein-product- and ontology-based aspects. At the same time, both sources are uniformly defined across species, allowing the gene compression module to generalize to a multi-species setting.

Given a gene g ($g = 1, \ldots, n$), we separately encode its protein representation and GO annotation into 256-dimensional latent embeddings, denoted by $\mathbf{p}_g$ and $\mathbf{o}_g$ respectively, using two denoising autoencoders (Fig. 1b, details in Appendix Sec. A.1). Using the functional information captured by the protein and GO latent embeddings, the gene compression module assigns the n genes to a fixed set of m meta-genes ($m = 512$). For each gene g, the module fuses $\mathbf{p}_g$ and $\mathbf{o}_g$ into a joint functional embedding. An assignment network then maps this embedding to m logits, which are transformed using a softmax function to obtain a soft assignment of the gene over the meta-genes.

We then construct two k-nearest neighbor (kNN) graphs that connect genes with similar protein representation and gene ontology annotations, using the protein and GO latent embedding, respectively. Denoting the adjacency matrices of kNN graphs built based on protein and GO latent embedding as $\mathbf{G}_p \in \mathbb{R}^{n\times n}$ and $\mathbf{G}_f \in \mathbb{R}^{n\times n}$, we calculate the adjacency matrix of fused graph $\mathbf{G} = 0.5\mathbf{G}_p + 0.5\mathbf{G}_f$. Functionally similar genes agreed by two sources are connected with edges of higher weight in $\mathbf{G}$. As a result, assigning genes into meta-genes is equivalent to partitioning the graph $\mathbf{G}$. We applied a relaxed min-cut loss (Bianchi et al., 2020) to partition the graph, which maximizes the total edge weights within each partition. To prevent the collapsing of partition assignment to trivial result, we further introduce two complementary regularization terms: a load-balancing regularization preventing a small number of meta-genes from absorbing most genes, and a rank-preserving regularization encouraging the mutual exclusiveness of gene composition for different meta-genes.

Assuming the gene-to-meta-gene soft assignment matrix is $\mathbf{S} \in \mathbb{R}^{n\times m}$, the full objective follows:

$$\mathcal{L}_G(\mathbf{S}) = \underbrace{-\frac{\mathrm{Tr}(\mathbf{S}^\top \mathbf{G}\mathbf{S})}{\mathrm{Tr}(\mathbf{S}^\top \mathbf{D}\mathbf{S})}}_{\text{mincut}} + \lambda_{bal} \underbrace{\sum_{m_g} f_{m_g} \cdot p_{m_g}}_{\text{load balance}} + \mathcal{L}_{rank} \tag{1}$$

$\mathbf{D}$ is the degree matrix of $\mathbf{G}$. The first term is the relaxed min-cut loss, and the second term is the load-balancing regularization, and the third term is the rank-preserving regularization.

For each meta-gene m_g, the load balancing regularization calculates the average assignment score $p_{m_g} = \frac{1}{n}\sum_g \mathbf{S}_{g,m_g}$, and weights it by f_{m_g}, the fraction of genes that are hard-assigned to m_g (with "argmax"). By minimizing the load balancing loss, the module learns to evenly distribute gene assignment across all meta-genes.

The rank-preserving regularization is defined as:

$$\mathcal{L}_{rank} = \underbrace{\frac{1}{m(m-1)} \sum_{m_g \neq m'_g} \left[\lambda_{ort}\psi^2_{m_g,m'_g} + \lambda_{cov}\rho^2_{m_g,m'_g}\right]}_{\text{covariance}} + \lambda_{var} \underbrace{\frac{1}{m}\sum_{m_g} \text{ReLU}(\frac{1}{m} - \sigma_{m_g})}_{\text{variance}} \quad (2)$$

The first term minimizes the correlation between different meta-gene assignments, encouraging different meta-genes to capture distinct sets of genes. $\psi_{m_g,m'_g} = \frac{(\mathbf{S}^\top\mathbf{S})_{m_g,m'_g}}{(\mathbf{S}^\top\mathbf{S})_{m_g,m_g}}$ measures the orthogonality between different meta-genes, while ρ_{m_g,m'_g} measures the Pearson correlation between different zero-centered meta-genes residuals. The second term encourages the total variance of each meta-genes σ_{m_g}, preventing meta-gene assignments from flattening into a near-constant value across genes. Further training details of gene compression module, including GO-masking augmentation and hyperparameter settings, are provided in Appendix Sec. A.2.

For each cell i, the compression model generates meta-gene expression x_{i,m_g} and token embedding $\mathbf{e}_{m_g}$ using $\mathbf{S}$ (Fig. 1a): $x_{i,m_g} = \sum_{g=1}^n x_{i,g}\mathbf{S}_{g,m_g}$, $\mathbf{e}_{m_g} = \sum_{g=1}^n \mathbf{e}_g\mathbf{S}_{g,m_g}$. The gene token embedding $\mathbf{e}_g$ is calculated from its protein and GO latent embeddings $(\mathbf{p}_g, \mathbf{o}_g)$ using an MLP.

3.3 Tokenization and pretraining objective of scWEAVE

After obtaining the meta-genes' expression x_{i,m_g} and token embedding $\mathbf{e}_{m_g}$, we construct meta-gene sentence c_i for cell i with sequence:

$$c_i = [\text{cls}, [x_{i,m_1}, \mathbf{e}_{m_1}], [x_{i,m_2}, \mathbf{e}_{m_2}], \cdots, [x_{i,m}, \mathbf{e}_m]] \quad (3)$$

the "cls" token embedding is used as the position for the shared cell factor generation of scWEAVE. The meta-gene expression x_{i,m_g} is transformed into 32-dimensional value embedding $\mathbf{x}_{i,m_g}$ using Fourier encoding function, which better preserves the magnitudes of the continuous expression values (Appendix Sec. A.3).

scWEAVE then preprocesses the token and value embeddings ($\mathbf{e}_{m_g}$ and $\mathbf{x}_{i,m_g}$) using MLP, and sums up the two embeddings element-wise to obtain the final token embedding for the construction of c_i (Equ. 3). For cell i, the tokenized c_i is then fed into the transformer to generate the shared cell factor $\mathbf{h}_i$ and gene-specific remnant factors $\{\mathbf{r}_{ig}\}_{g=1}^n$, which are then used for the masked expression prediction (MEP) pretraining (Fig. 1c).

In MEP pretraining, we mask the expression of a portion of genes in each cell, and train scWEAVE to predict the masked genes' expression. Since scRNA-seq data are highly sparse, we apply different masking rates for zero- and non-zero-expression genes (0.5 for non-zero genes, 0.18 for zero genes) so their composition is balanced in the masked gene pool. Given a cell i and a masked gene g, scWEAVE predicts the masked expression $\hat{x}_{ig}$ in three steps: (1) it predicts the baseline expression of gene g through cell-ablated prediction network $\hat{x}^{\text{bl}}_g = f_{\text{bl}}(\mathbf{e}_g)$ using its token embedding $\mathbf{e}_g$; (2) it predicts a cell-specific expression correction term using cell encoding (shared cell factor $\mathbf{h}_i$ and remnant factor $\mathbf{r}_{ig}$) through a cell-context prediction network $\hat{x}^{\text{context}}_{ig} = f_{\text{context}}(\mathbf{h}_i, \mathbf{r}_{ig}; \mathbf{e}_g)$, (3) it combines the two into the final expression, $\hat{x}_{ig} = \hat{x}^{\text{bl}}_g \cdot \hat{x}^{\text{context}}_{ig}$. scWEAVE models the masked expression using negative binomial distribution, which better accounts for the long-tailed nature of data distribution (Lopez et al., 2018). The negative binomial requires both a mean and a dispersion parameter; scWEAVE uses $\hat{x}_{ig}$ as the estimated mean, and estimates the dispersion of gene g through a cell-ablated prediction network, $\hat{\theta}_g = f_{\text{theta}}(\mathbf{e}_g)$. Given the ground-truth expression x_{ig} and scWEAVE's estimated $(\hat{x}_{ig}, \hat{\theta}_g)$ parameters, the negative binomial distribution follows $\text{NB}(x_{ig}; \hat{x}_{ig}, \hat{\theta}_g) = \frac{\Gamma(x_{ig}+\hat{\theta}_g)}{\Gamma(\hat{\theta}_g)\Gamma(x_{ig}+1)} \left(\frac{\hat{\theta}_g}{\hat{\theta}_g+\hat{x}_{ig}}\right)^{\hat{\theta}_g} \left(\frac{\hat{x}_{ig}}{\hat{\theta}_g+\hat{x}_{ig}}\right)^{x_{ig}}$.

scWEAVE is trained to minimize the negative log-likelihood of the negative binomial distribution, $\mathcal{L}_{\text{NB}}(x_{ig}; \hat{x}_{ig}, \hat{\theta}_g) = -\log NB(x_{ig}; \hat{x}_{ig}, \hat{\theta}_g)$.

We improve MEP accuracy on highly sparse scRNA-seq data through multi-task learning, and introduce an auxiliary expression detection head that predicts whether the masked gene is expressed, $\hat{y}_{ig}^{\text{detect}} = f_{\text{detect}}(\mathbf{h}_i, \mathbf{r}_{ig}; \mathbf{e}_g)$. Given the ground-truth expression flag y_{ig}^{detect} for cell i and gene g, we train scWEAVE to predict it using a binary cross-entropy loss, $\mathcal{L}_{\text{BCE}}(y_{ig}^{\text{detect}}, \hat{y}_{ig}^{\text{detect}})$.

3.4 Contrastive regularization with multi-granular labels

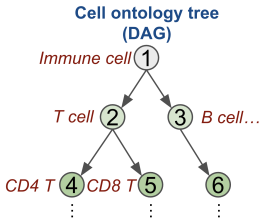

Figure 2: Project label to CL-Tree

We adapt supervised contrastive learning (Khosla et al., 2020) to align shared cell factors across batches and species while accommodating differences in annotation granularity across data sources (Sec. 3.1). We first map annotations of different granularities onto a common cell ontology tree (CL-tree) (Diehl et al., 2016), making their hierarchical relationships explicit (Fig. 2). Prior work scCello (Yuan et al., 2024) uses ontology-derived label similarities to preserve relative similarity between cell types in the embedding space. Our objective focuses on annotation compatibility: identical and ancestor–descendant labels receive positive weights that decay with hop distance. This allows cells with potentially matching identities but differently resolved annotations to contribute to alignment within the shared cell factor space.

Given cell i and j, we denote their shared cell factors as $\mathbf{h}_i$ and $\mathbf{h}_j$, and their labels as ℓ_i and ℓ_j respectively. Their label similarity w_{ij} can then be calculated by applying an exponential decay function onto the hop-distance between ℓ_i and ℓ_j in the CL-tree

$$w_{ij} = \exp(-\operatorname{hop}(\ell_i, \ell_j)/\tau_1)\, \mathbf{1}[\ell_i \sim \ell_j] \tag{4}$$

where $\tau_1 = 1.0$ and $\operatorname{hop}(\ell_i, \ell_j)$ denotes the number of hops between ℓ_i and ℓ_j in the CL-tree. The indicator $\mathbf{1}[\ell_i \sim \ell_j] = 1$ if the labels are identical or have an ancestor–descendant relationship, allowing matches across annotation granularities, and 0 otherwise.

We then calculate the similarity between the shared cell factors of cells i and j (p_{ij}):

$$p_{ij} = \frac{\exp(\mathbf{h}_i^T \cdot \mathbf{h}_j/\tau_2)}{\sum_{k \neq i} \exp(\mathbf{h}_i^T \cdot \mathbf{h}_k/\tau_2)} \tag{5}$$

We set $\tau_2 = 0.07$ in the calculation. For cell i, the contrastive loss follows

$$\mathcal{L}_{\text{contr}}(i) = -\frac{\sum_{j \neq i} w_{ij} \log p_{ij}}{\sum_{j \neq i} w_{ij}} \tag{6}$$

This objective extends supervised contrastive learning with ontology-dependent positive weights (Appendix Sec. A.4).

We train scWEAVE using MEP and contrastive regularization through a three-stage procedure designed to stabilize training (Appendix Sec. A.6).

4 Experimental Results

We pretrained scWEAVE on 30 million healthy human cells and 26 million healthy mouse cells from CELLxGENE (Program et al., 2023). We evaluate the model on 11 scRNA-seq datasets spanning single- and multi-species settings: three multi-species, two mouse, and six human datasets (Luecken et al., 2022; Liu et al., 2026; Rosen et al., 2024; Zhang et al., 2024). Dataset statistics are summarized in Table 1, with sources and preprocessing details provided in Appendix Sec. A.7.

We selected SOTA scFMs as baseline methods for evaluation, including TranscriptFormer (Pearce et al., 2026), scGPT (Cui et al., 2024), scMulan (Bian et al., 2024a), UCE (Rosen et al., 2026), and scFoundation (Hao et al., 2024). We first compare the computational efficiency of scWEAVE with baseline scFMs, and then evaluate scWEAVE's performance in computational tasks including ortholog retrieval, gene expression imputation, batch effect removal, and novel cell type discovery. The experiment details and results are shown in the following sections.

Table 1: Overview of datasets. Species: [a]Human, pig, and *M. mulatta*; [b]Human, mouse, *M. mulatta*, and *C. jacchus*; [c]Human, mouse, pig, *M. mulatta*, and *M. fascicularis*.

Dataset	Hippo	Motor	AH-Outflow	Cortex-3B	Retina	Immune	Pancreas	Lung Atlas	Covid-19	Dendritic	PBMC
# cells	50,000	50,000	50,112	50,000	50,000	33,506	16,382	32,472	11,956	569	4,638
# batches	3	4	5	3	61	10	9	16	27	2	2
Species	3[a]	4[b]	5[c]	Mouse	Mouse	Human	Human	Human	Human	Human	Human

4.1 Inference running time and memory usage comparison

We compared the inference running time and peak GPU memory of all methods on one A40 GPU with 48 GB of GDDR6 memory. We sample 500,000 cells from our training data and infer the cell embedding of 32 cells per mini-batch. The cumulative running time comparison on varying sizes of inferred data is shown in Fig. 3. Table 2 shows the average running time of 32 cells for different methods over 200 mini-batches, along with the peak memory usage of different methods during inference. scWEAVE has significant runtime and GPU memory consumption advantage compared to other baseline methods. scGPT achieves the second-lowest runtime and peak GPU memory usage. UCE is the slowest method while scFoundation has the largest memory consumption.

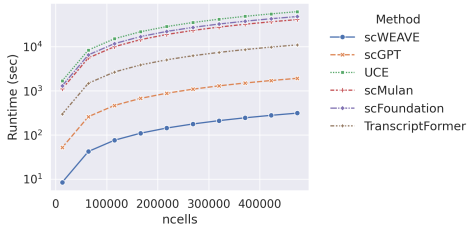

Figure 3: Running time comparison of scWEAVE and baseline methods.

Table 2: Running time and memory usage comparison between scWEAVE and baseline methods.

Methods	ncells	runtime (s)	peak memory (GB)
scWEAVE	32	**0.021**	**0.86**
scGPT	32	0.129	1.37
UCE	32	4.196	9.81
scMulan	32	2.776	4.20
scFoundation	32	3.255	18.88
TranscriptFormer	32	0.741	5.18

4.2 Test on imputation of gene expression dropouts

Experiment setup and evaluation metrics Gene expression dropout is a long-standing challenge in scRNA-seq analysis. We simulate gene expression dropout by sampling 2,000 cells from each test dataset and randomly masking expression entries with 0s (except "Dendritic" with less than 2000 total cells). To reflect practical imputation settings, we evaluate the scFMs' ability to impute expression dropouts without informing the models which entries were masked. We compared scWEAVE with scFoundation, scGPT, and scMulan, the baseline scFMs that provide gene expression prediction heads. We evaluate imputation accuracy at the masked entries using Pearson and Spearman correlations with the ground-truth values (Appendix Sec. A.10).

Results As shown in Table 3, scWEAVE achieved the highest PCC on 10 of 11 datasets and the highest SCC on 8 of 11 datasets. Its advantages were particularly pronounced on Cortex-3B, Retina, Hippo, and Motor, where it outperformed all baselines on both metrics. These results demonstrate that the compact design of scWEAVE still outperforms larger scFMs on expression imputation, and challenge the need for increased model scale for this task.

Table 3: Imputation accuracy of scFMs, measured using Pearson correlation (PCC) and Spearman correlation (SCC). Bold and underlined values indicate the best and second-best scores, respectively.

	scWEAVE		**scFoundation**		**scGPT**		**scMulan**	
Dataset	PCC ↑	SCC ↑	PCC ↑	SCC ↑	PCC ↑	SCC ↑	PCC ↑	SCC ↑
Immune	**0.707**	0.503	0.421	0.416	0.697	**0.548**	0.493	0.327
Pancreas	0.488	**0.463**	0.417	0.411	**0.499**	0.419	0.281	0.154
Lung Atlas	**0.541**	0.458	0.456	**0.514**	0.476	0.475	0.536	0.398
Covid-19	**0.624**	**0.500**	0.381	0.437	0.207	0.279	0.597	0.375
PBMC	**0.658**	0.403	0.493	0.452	0.657	**0.576**	0.555	0.352
Dendritic	**0.604**	**0.520**	0.462	0.434	0.199	0.252	0.591	0.337
Cortex-3B	**0.437**	**0.425**	0.227	0.231	0.155	0.194	0.206	0.116
Retina	**0.534**	**0.510**	0.286	0.276	0.357	0.309	0.284	0.054
Hippo	**0.491**	**0.432**	0.319	0.299	0.263	0.312	0.242	0.140
Motor	**0.647**	**0.606**	0.356	0.378	0.502	0.434	0.362	0.205
AH-Outflow	**0.529**	**0.449**	0.352	0.318	0.479	0.393	0.511	0.292

4.3 EVALUATION OF MULTISPECIES APPLICABILITY THROUGH ORTHOLOG RETRIEVAL

Experiment setup Orthologous gene pairs are commonly used as a gold standard for evaluating functional similarity of genes across species. The gene compression module of scWEAVE assigns genes from different species to shared meta-gene space, allowing candidate orthologs to be retrieved by representation similarity. We evaluate ortholog retrieval on 3 multi-species datasets: "Hippo", "Motor", and "AH-Outflow" (Table 1). For each dataset, we evaluate scWEAVE on species that are held out from pretraining, and measure the proportion of orthologs it can retrieve between the query species and human reference. Since none of the baseline scFMs considered in our evaluation provides an ortholog retrieval procedure, we compare scWEAVE with "PCA-ESM" and "SATURN" (Rosen et al., 2024) (Details in Appendix Sec. A.8). "PCA-ESM" applies principal component analysis to ESM-2-derived protein embeddings to obtain gene representations. "SATURN" is a multi-species integration method that learns gene and cell representations. We evaluate its initial and refined gene-to-macrogene assignment scores, denoted as "SATURN (init)" and "SATURN (trained)".

Evaluation metrics For each query gene g, we rank candidate human genes by cosine similarity to its representation (Appendix Sec. A.11.1). For scWEAVE, this representation is the meta-gene assignment vector $\mathbf{S}_g$. We report top-k ortholog retrieval accuracy ($k \in \{1, 5\}$), defined as the percentage of query genes for which at least one known human ortholog was identified among the top-k nearest human genes.

Table 4: Top-k human ortholog retrieval accuracy (%) of scWEAVE and baseline methods.

		scWEAVE		PCA-ESM		SATURN (init)		SATURN (trained)	
Dataset	Species	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
AH-Outflow	Pig	**81.1**	90.4	79.3	**90.7**	38.3	56.6	23.5	44.1
	M. fascicularis	**81.6**	89.5	80.4	**90.5**	40.8	56.9	26.8	46.6
	M. mulatta	**79.6**	88.6	77.9	**89.4**	38.7	56.3	25.0	44.8
Hippo	Pig	**81.7**	**90.9**	80.6	90.8	39.6	56.9	11.3	24.9
	M. mulatta	**82.6**	**91.1**	81.2	90.7	44.3	60.7	13.2	27.0
Motor	*M. mulatta*	**83.3**	**91.2**	81.0	90.9	42.6	58.5	19.0	40.9
	C. jacchus	**83.8**	**91.9**	81.7	91.0	41.4	58.5	31.3	48.9

Results As shown in Table 4, scWEAVE achieves the highest top-1 accuracy in all seven comparisons and the highest top-5 accuracy in four. PCA-ESM achieves slightly higher top-5 accuracy on the three AH-Outflow comparisons. Both methods substantially outperform the SATURN variants. SATURN (trained) consistently underperforms SATURN (init), indicating that its training refinement reduces ortholog retrieval accuracy on this benchmark.

4.4 TEST ON BATCH EFFECT REMOVAL AND CELL TYPE IDENTIFICATION

Experiment setup Cell type identification is a key task in single-cell analysis, and its major challenge lies within cell clustering, which is confounded by batch-effect-induced distortion. scWEAVE generates a shared cell factor designed to reduce batch effect and improve cell clustering accuracy. We evaluate this representation on test datasets in Table 1. For each test dataset, we extract cell embeddings from scWEAVE (the shared cell factor) and baseline scFMs, and assess batch mixing and agreement between the identified cell clusters and ground-truth cell type labels.

Evaluation metrics We evaluate the batch mixing using ASW-B (Average Silhouette Width on Batch), and GC (Graph Connectivity) scores (Luecken et al., 2022). To evaluate cell type identification, we apply Leiden clustering algorithm on the cell embedding, and measure its agreement with ground truth cell type label using ARI (Adjusted Rand Index). We also use ASW (Average Silhouette Width) (Luecken et al., 2022) to measure cell type separation in the embedding space (Details in Appendix Sec. A.11.2).

Results The evaluation scores in zero-shot setting are shown in Table 5. scWEAVE achieves the best or second-best GC on 8 of 11 datasets, showing its superior performance in batch mixing. For cell clustering accuracy, scWEAVE achieves the best or second-best ARI on 9 of 11 datasets, indicating its embedding supports a better cell type identification. The UMAP visualization also confirms that scWEAVE's embedding improves batch mixing and cell type separation compared to the baselines (Appendix Sec. A.13).

Table 5: Performance of scWEAVE and baseline scFMs on batch effect removal and cell type identification. Bold and underlined values indicate the best and second-best distinct scores, respectively

Methods	scWEAVE				scGPT				scMulan				scFoundation				TranscriptFormer				UCE			
	ARI	ASW	ASW-B	GC	ARI	ASW	ASW-B	GC	ARI	ASW	ASW-B	GC	ARI	ASW	ASW-B	GC	ARI	ASW	ASW-B	GC	ARI	ASW	ASW-B	GC
Immune	**0.70 ± 0.00**	**0.58**	0.85	**0.95**	0.15 ± 0.02	0.48	**0.93**	0.83	0.48 ± 0.00	0.56	0.89	0.81	0.34 ± 0.04	0.57	0.82	0.88	0.50 ± 0.06	0.53	0.82	0.93	0.62 ± 0.01	0.56	0.89	0.91
Pancreas	**0.51 ± 0.01**	**0.55**	0.86	0.90	0.05 ± 0.00	0.43	**0.93**	0.55	0.34 ± 0.00	0.51	0.88	0.76	0.26 ± 0.06	0.53	0.83	0.89	0.27 ± 0.06	0.48	0.84	0.84	0.40 ± 0.01	**0.55**	0.88	**0.91**
Lung Atlas	**0.50 ± 0.02**	0.52	0.88	0.96	0.23 ± 0.02	0.46	0.92	0.80	0.44 ± 0.02	0.53	0.91	0.86	0.41 ± 0.05	0.55	0.89	0.90	0.45 ± 0.02	0.49	0.87	0.97	0.45 ± 0.03	**0.56**	**0.93**	**0.98**
Covid-19	**0.62 ± 0.01**	0.59	0.82	0.95	0.20 ± 0.02	0.48	**0.90**	0.84	0.51 ± 0.01	**0.60**	0.84	0.74	0.50 ± 0.05	0.57	0.87	0.96	0.59 ± 0.01	0.57	0.83	**0.97**	0.51 ± 0.05	0.58	0.88	0.96
PBMC	0.76 ± 0.02	**0.63**	0.82	0.96	0.04 ± 0.02	0.50	**0.92**	0.86	0.37 ± 0.00	0.58	0.90	0.93	0.54 ± 0.01	0.59	0.81	**0.98**	0.57 ± 0.13	0.56	0.74	**0.98**	**0.81 ± 0.00**	0.53	0.89	0.96
Dendritic	0.80 ± 0.00	0.63	0.77	0.99	0.34 ± 0.01	0.52	0.91	0.99	0.69 ± 0.00	0.69	**0.93**	0.83	0.58 ± 0.00	**0.76**	0.85	0.98	**0.86 ± 0.00**	0.62	0.91	**1.00**	0.72 ± 0.01	0.66	0.86	0.99
Hippo	0.67 ± 0.09	0.60	0.80	0.82	0.10 ± 0.01	0.48	**0.91**	0.62	0.70 ± 0.00	0.59	0.83	0.79	0.70 ± 0.02	0.61	0.76	**0.90**	**0.72 ± 0.00**	0.59	0.68	0.79	**0.72 ± 0.01**	**0.62**	0.76	0.83
Motor	0.26 ± 0.00	0.51	0.72	**0.89**	0.06 ± 0.00	0.48	**0.93**	0.72	0.28 ± 0.01	0.51	0.82	0.78	0.32 ± 0.00	0.53	0.69	0.88	0.32 ± 0.01	**0.55**	0.59	0.54	**0.34 ± 0.02**	0.54	0.46	**0.89**
Cortex-3B	**0.78 ± 0.11**	0.64	0.82	0.99	0.30 ± 0.10	0.50	**0.93**	0.89	0.64 ± 0.07	0.67	0.90	0.77	0.46 ± 0.02	**0.70**	0.74	**1.00**	0.30 ± 0.00	0.60	0.59	0.77	0.32 ± 0.02	0.66	0.82	**1.00**
Retina	0.58 ± 0.06	0.54	0.87	0.86	0.18 ± 0.03	0.46	0.88	0.51	0.41 ± 0.01	0.51	0.86	0.64	**0.66 ± 0.00**	**0.55**	0.86	**0.87**	0.48 ± 0.02	0.51	0.81	0.73	0.50 ± 0.04	**0.55**	**0.90**	**0.87**
AH-Outflow	**0.37 ± 0.02**	0.56	0.85	0.98	0.17 ± 0.03	0.48	**0.90**	0.86	0.18 ± 0.00	0.54	0.84	0.82	0.22 ± 0.00	0.58	0.77	0.86	0.33 ± 0.00	0.57	0.75	0.94	0.27 ± 0.02	**0.59**	0.82	**0.99**

4.5 TEST ON NOVEL CELL TYPE DETECTION

Experiment setup A major challenge in analyzing multi-species single-cell data is identifying novel cell types that only exist in a query species. In the cell embedding space, a novel cell type is expected to lie farther from known cell types in the reference species. For each query cell type, we define a novelty score based on its cells' average distance to their nearest reference cells. A higher score indicates greater novelty of the query cell type. We evaluate scFMs on novel cell type detection on multi-species datasets "Hippo", "Motor", and "AH-Outflow" (Table 1), and use human cells as the reference for novelty classification (details in Appendix Sec. A.9).

Evaluation metrics For each cell in the query species, we define its *novelty score* as its mean cosine distance to its k nearest cells in the reference species. We then average these scores within each ground-truth query cell type to obtain a cell-type-level novelty score. We generate the ground truth novelty label by checking if each query cell type is absent from the corresponding human reference. We then measure how well the cell-type-level novelty scores predict these novelty labels using the area under the precision–recall curve (AUPRC) (Appendix Sec. A.11.3).

Results Table 6 summarizes novel cell type detection performance across species from "Hippo", "Motor", and "AH-Outflow" datasets. scWEAVE achieves the highest AUPRC in four tests, followed by scFoundation and scGPT in three each. The result shows that the cell embedding generated by scWEAVE is capable of capturing the novel cell type with high accuracy.

Table 6: AUPRC scores for novel cell type detection. Bold and underlined values indicate the best and second-best scores, respectively.

Dataset	Species	scWEAVE	scFoundation	scMulan	UCE	scGPT	TranscriptFormer
Hippo	Pig	**1.00**	**1.00**	**1.00**	0.33	0.11	**1.00**
	Rhesus	**0.83**	0.75	0.75	0.70	0.14	0.50
Motor	Marmoset	0.39	**0.64**	0.62	0.61	0.09	0.29
	Mouse	**1.00**	**1.00**	0.92	0.23	0.09	**1.00**
AH-Outflow	MacaM	0.32	0.48	0.33	0.51	**0.53**	0.29
	MacaF	**0.43**	0.32	0.27	0.39	0.29	0.36
	Pig	0.25	0.25	0.19	0.27	**0.72**	0.18
	Mouse	0.59	0.46	0.47	0.46	**0.77**	0.65

5 CONCLUSION AND FUTURE WORK

The primary strengths of scWEAVE lie in its multi-species applicability and computational efficiency. These strengths make scWEAVE more accessible for biological research, where rapid single-cell analysis with limited GPU resources is often important. Notably, this computational efficiency does not come at the expense of performance relative to larger scFMs. In particular, scWEAVE outperforms larger baseline scFMs on dropout imputation, a task closely aligned with the pretraining objective of many scFMs. This result challenges the assumption that increasing model scale is necessary for better imputation performance. It also motivates greater attention to biologically informed model design as a way to improve scFMs without increasing their size.

Despite these strengths, scWEAVE is currently pretrained only on healthy human and mouse scRNA-seq data because of limited training data availability. In future work, we plan to improve data collection and curation to train scWEAVE on more diverse, high-quality single-cell data and improve its ability to generalize across species and biological conditions. We also aim to extend scWEAVE into a multi-condition, multi-modal scFM while maintaining its emphasis on computational efficiency.

6 AI USE STATEMENT

In this work, we used OpenAI Codex to assist with language editing, manuscript organization, literature and citation lookup. We also used Claude to assist with setting up baseline methods and

debugging evaluation scripts. The authors reviewed and revised the AI-assisted content and take full responsibility for the manuscript.

7 Reproducibility statement

To ensure the reproducibility of the model, the link to the complete source code of scWEAVE is provided in Appendix Sec. A.12. The sources and download links of the training and evaluation datasets are also provided in Appendix Sec. A.12.

References

Prabhu S Arunachalam, Florian Wimmers, Chris Ka Pun Mok, Ranawaka APM Perera, Madeleine Scott, Thomas Hagan, Natalia Sigal, Yupeng Feng, Laurel Bristow, Owen Tak-Yin Tsang, et al. Systems biological assessment of immunity to mild versus severe COVID-19 infection in humans. Science, 369(6508):1210–1220, 2020.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. Nature genetics, 25(1):25–29, 2000.

Ding Bai, Shentong Mo, Ruiyi Zhang, Yingtao Luo, Jiahao Gao, Jeremy Parker Yang, Qiuyang Wu, Hamidreza Rahmani, Tiffany Amariuta, Danielle Grotjahn, et al. sclong: A billion-parameter foundation model for capturing long-range gene context in single-cell transcriptomics. Nature Communications, 2026.

Trygve E. Bakken, Nikolas L. Jorstad, Qiwen Hu, Blue B. Lake, Wei Tian, Brian E. Kalmbach, Megan Crow, Rebecca D. Hodge, Fenna M. Krienen, Staci A. Sorensen, et al. Comparative cellular analysis of motor cortex in human, marmoset and mouse. Nature, 598(7879):111–119, 2021. doi: 10.1038/s41586-021-03465-8. URL https://doi.org/10.1038/s41586-021-03465-8.

Haiyang Bian, Yixin Chen, Xiaomin Dong, Chen Li, Minsheng Hao, Sijie Chen, Jinyi Hu, Maosong Sun, Lei Wei, and Xuegong Zhang. scmulan: a multitask generative pre-trained language model for single-cell analysis. In International Conference on Research in Computational Molecular Biology, pp. 479–482. Springer, 2024a.

Haiyang Bian, Yixin Chen, Erpai Luo, Xinze Wu, Minsheng Hao, Lei Wei, and Xuegong Zhang. General-purpose pre-trained large cellular models for single-cell transcriptomics. National Science Review, 11(11):nwae340, 2024b.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In International conference on machine learning, pp. 874–883. PMLR, 2020.

BRAIN Initiative Cell Census Network (BICCN). A multimodal cell census and atlas of the mammalian primary motor cortex. Nature, 598(7879):86–102, 2021. doi: 10.1038/s41586-021-03950-0. URL https://doi.org/10.1038/s41586-021-03950-0.

Sijie Chen, Yanting Luo, Haoxiang Gao, Fanhong Li, Yixin Chen, Jiaqi Li, Renke You, Minsheng Hao, Haiyang Bian, Xi Xi, et al. heca: the cell-centric assembly of a cell atlas. Iscience, 25(5), 2022.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nat. Methods, 21(8):1470–1480, August 2024.

Alexander D Diehl, Terrence F Meehan, Yvonne M Bradford, Matthew H Brush, Wasila M Dahdul, David S Dougall, Yongqun He, David Osumi-Sutherland, Alan Ruttenberg, Sirarat Sarntivijai, et al. The cell ontology 2016: enhanced content, modularization, and ontology interoperability. Journal of biomedical semantics, 7:1–10, 2016.

Daniel Franjic, Mario Skarica, Shaojie Ma, Jon I. Arellano, Andrew T. N. Tebbenkamp, Jinmyung Choi, Chuan Xu, Qian Li, Yury M. Morozov, David Andrijevic, Zvonimir Vrselja, Ana Spajic, Gabriel Santpere, Mingfeng Li, Shupei Zhang, Yang Liu, Joshua Spurrier, Le Zhang, Ivan Gudelj, Lucija Rapan, Hideyuki Takahashi, Anita Huttner, Rong Fan, Stephen M. Strittmatter, Andre M. M. Sousa, Pasko Rakic, and Nenad Sestan. Transcriptomic taxonomy and neurogenic trajectories of adult human, macaque, and pig hippocampal and entorhinal cells. Neuron, 110(3):452–469.e14, 2022. doi: 10.1016/j.neuron.2021.10.036. URL https://doi.org/10.1016/j.neuron.2021.10.036.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. Nat. Methods, 21(8):1481–1491, August 2024.

Sinisa Hrvatin, Daniel R. Hochbaum, M. Aurel Nagy, Marcelo Cicconet, Keiramarie Robertson, Lucas Cheadle, Rapolas Zilionis, Alex Ratner, Rebeca Borges-Monroy, Allon M. Klein, Bernardo L. Sabatini, and Michael E. Greenberg. Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. Nature Neuroscience, 21(1):120–129, 2018. doi: 10.1038/s41593-017-0029-5. URL https://doi.org/10.1038/s41593-017-0029-5.

Jérémie Kalfon, Jules Samaran, Gabriel Peyré, and Laura Cantini. scprint: pre-training on 50 million cells allows robust gene network predictions. Nature Communications, 16(1):3607, 2025.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. Advances in neural information processing systems, 33:18661–18673, 2020.

Jeong Seok Lee, Seongwan Park, Hye Won Jeong, Jin Young Ahn, Seong Jin Choi, Hoyoung Lee, Baekgyu Choi, Su Kyung Nam, Moa Sa, Ji-Soo Kwon, et al. Immunophenotyping of COVID-19 and influenza highlights the role of type I interferons in development of severe COVID-19. Science immunology, 5(49):eabd1554, 2020.

Jin Li, Jongsu Choi, Xuesen Cheng, Justin Ma, Shahil Pema, Joshua R. Sanes, Graeme Mardon, Benjamin J. Frankfort, Nicholas M. Tran, Yumei Li, and Rui Chen. Comprehensive single-cell atlas of the mouse retina. iScience, 27(6):109916, 2024. doi: 10.1016/j.isci.2024.109916. URL https://doi.org/10.1016/j.isci.2024.109916.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. bioRxiv, 2022.

Tianyu Liu, Kexing Li, Yuge Wang, Hongyu Li, and Hongyu Zhao. Evaluating the utilities of foundation models in single-cell data analysis. Advanced Science, 13(27):e14490, 2026.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. Nat. Methods, 15(12):1053–1058, December 2018.

Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. Molecular systems biology, 15(6):e8746, 2019.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. Nature methods, 19(1):41–50, 2022.

James D Pearce, Sara E Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, et al. Transcriptformer: A generative cell atlas across 1.5 billion years of evolution. Science, 393(6806): aec8514, 2026.

CZI Single-Cell Biology Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M. Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, J. Michael Cherry, Tiffany Chi, Jennifer Chien, Leah Dorman, Pablo Garcia-Nieto, Nayib Gloria, Mim Hastie, Daniel Hegeman, Jason Hilton, Timmy Huang, Amanda Infeld, Ana-Maria Istrate, Ivana Jelic, Kuni

Katsuya, Yang Joon Kim, Karen Liang, Mike Lin, Maximilian Lombardo, Bailey Marshall, Bruce Martin, Fran McDade, Colin Megill, Nikhil Patel, Alexander Predeus, Brian Raymor, Behnam Robatmili, Dave Rogers, Erica Rutherford, Dana Sadgat, Andrew Shin, Corinn Small, Trent Smith, Prathap Sridharan, Alexander Tarashansky, Norbert Tavares, Harley Thomas, Andrew Tolopko, Meghan Urisko, Joyce Yan, Garabet Yeretssian, Jennifer Zamanian, Arathi Mani, Jonah Cool, and Ambrose Carr. Cz cell×gene discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. bioRxiv, 2023. doi: 10.1101/2023.10.30.563174. URL https://www.biorxiv.org/content/early/2023/11/02/2023.10.30.563174.

Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. The human cell atlas. elife, 6:e27041, 2017.

Yanay Rosen, Maria Brbić, Yusuf Roohani, Kyle Swanson, Ziang Li, and Jure Leskovec. Toward universal cell embeddings: integrating single-cell rna-seq datasets across species with saturn. Nature Methods, 21(8):1492–1500, 2024.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embedding provides a foundation model for cell biology. Nature, pp. 1–9, 2026.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65, 1987.

Arpiar Saunders, Evan Z. Macosko, Alec Wysoker, Melissa Goldman, Fenna M. Krienen, Heather de Rivera, Elizabeth Bien, Matthew Baum, Laura Bortolin, Shuyu Wang, Aleksandrina Goeva, James Nemesh, Nolan Kamitaki, Sara Brumbaugh, David Kulp, and Steven A. McCarroll. Molecular diversity and specializations among the cells of the adult mouse brain. Cell, 174(4): 1015–1030.e16, 2018. doi: 10.1016/j.cell.2018.07.028. URL https://doi.org/10.1016/j.cell.2018.07.028.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. Nature, 618(7965):616–624, 2023.

V A Traag, L Waltman, and N J van Eck. From louvain to leiden: guaranteeing well-connected communities. Sci. Rep., 9(1):5233, March 2019.

Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biology, 21(1):12, 2020. doi: 10.1186/s13059-019-1850-9.

Tavé van Zyl, Wenjun Yan, Alexi McAdams, Yi-Rong Peng, Karthik Shekhar, Aviv Regev, Dejan Juric, and Joshua R. Sanes. Cell atlas of aqueous humor outflow pathways in eyes of humans and four model species provides insight into glaucoma pathogenesis. Proceedings of the National Academy of Sciences, 117(19):10339–10349, 2020. doi: 10.1073/pnas.2001250117. URL https://doi.org/10.1073/pnas.2001250117.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

Alexandra-Chloé Villani, Rahul Satija, Gary Reynolds, Siranush Sarkizova, Karthik Shekhar, James Fletcher, Morgane Griesbeck, Andrew Butler, Shiwei Zheng, Suzan Lazo, Laura Jardine, David Dixon, Emily Stephenson, Emil Nilsson, Ida Grundberg, David McDonald, Andrew Filby, Weibo Li, Philip L. De Jager, Orit Rozenblatt-Rosen, Andrew A. Lane, Muzlifah Haniffa, Aviv Regev, and Nir Hacohen. Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. Science, 356(6335):eaah4573, 2017. doi: 10.1126/science.aah4573. URL https://doi.org/10.1126/science.aah4573.

Xiaodong Yang, Guole Liu, Guihai Feng, Dechao Bu, Pengfei Wang, Jie Jiang, Shubai Chen, Qinmeng Yang, Hefan Miao, Yiyang Zhang, et al. Genecompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. Cell Research, 34(12):830–845, 2024.

Xinyu Yuan, Zhihao Zhan, Zuobai Zhang, Manqi Zhou, Jianan Zhao, Boyu Han, Yue Li, and Jian Tang. Cell ontology guided transcriptome foundation model. Advances in Neural Information Processing Systems, 37:6323–6366, 2024.

Amit Zeisel, Ana B. Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science, 347(6226):1138–1142, 2015. doi: 10.1126/science.aaa1934. URL https://doi.org/10.1126/science.aaa1934.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, Shuangyu Yang, Chengyang Zhang, Jinbo Zhang, Nan Fang, et al. Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. Nature Communications, 16(1):4679, 2025.

Ziqi Zhang, Xinye Zhao, Mehak Bindra, Peng Qiu, and Xiuwei Zhang. scdisinfact: disentangled learning for integration and prediction of multi-batch multi-condition single-cell rna-sequencing data. Nature Communications, 15(1):912, 2024.

Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. Nature Communications, 8:14049, 2017. doi: 10.1038/ncomms14049. URL https://doi.org/10.1038/ncomms14049.

A Appendix

A.1 Details on Protein and GO encoding

We encode the protein representation and GO annotation into continuous latent embeddings using two denoising autoencoders respectively. Each autoencoder is trained to reconstruct its input from the noise-corrupted version of that input (Fig. 1b).

Because the protein representation is continuous-valued, we corrupt it by adding Gaussian noise with $\mu = 0$ and $\sigma = 0.1$, and optimize its reconstruction using a mean squared error (MSE) loss. Because the GO annotation is represented as a binary vector, we corrupt it by flipping annotated terms to unannotated with probability 0.05, and optimize its reconstruction with a binary cross-entropy (BCE) loss.

A.2 Training Details of Gene Compression Module

One input source, GO annotations, is not broadly available across species. We adopted a training scheme with missing GO augmentation to improve the robustness of the gene compression module when GO information is not available.

To keep the module reliable when GO annotation is unavailable, every training step computes the assignment matrix twice: once from the fused protein & GO representation (assignment matrix denoted as $\mathbf{S}_f$) and once from the protein embedding alone, with GO masked out (assignment matrix denoted as $\mathbf{S}_p$). Both variants are supervised by the same clustering objective ($\mathcal{L}_G(\mathbf{S}_f)$ and $\mathcal{L}_G(\mathbf{S}_p)$ following Equ. 1), the full training objective for the gene compression module is:

$$\mathcal{L}_{G_{complete}} = \frac{1}{2} \sum_{v \in f,p} \mathcal{L}_G(\mathbf{S}_v) + \lambda_{align} \mathcal{L}_{align} \tag{7}$$

$\mathcal{L}_{align}$ keeps $\mathbf{S}_f$ and $\mathbf{S}_p$ consistent with each other by minimizing the squared distance between their embeddings before classification output ($\mathbf{z}_f$ and $\mathbf{z}_p$, respectively):

$$\mathcal{L}_{align} = \|\mathbf{z}_p - \mathbf{z}_f\|_2^2 \tag{8}$$

The hyper-parameter and optimization settings that are used to train the gene compression module are summarized in Appendix Table 7. To prevent the training from collapsing to a small number of dominant meta-genes, we additionally anneal the softmax temperature used to compute the soft assignment of meta-genes from 2.0 to 0.2 linearly over the first 500 training epochs, then hold it fixed for the remaining epochs.

Hyperparameter	Value
λ_{ort}	4.87×10^{-5}
λ_{bal}	0.0368
λ_{var}	336.9
λ_{cov}	74.4
λ_{align}	1.0
Number of meta-genes m	512
Encoder/embedding dim (n_{latent}, n_{embed})	256
kNN graph neighbors k	30
Dropout	0
Optimizer	Adam, lr $= 5 \times 10^{-3}$
Training epochs	2000
Temperature schedule	$2.0 \rightarrow 0.2$ over first 500 epochs, then fixed

Table 7: Hyperparameters for gene compression module training. Loss weights $\lambda_{ort}, \lambda_{bal}, \lambda_{var}, \lambda_{cov}$ are calibrated so that $\lambda \cdot \mathcal{L}(\text{init}) \approx 0.75 \approx |\mathcal{L}_{mincut}(\text{init})|$ for each term, giving each regularizer comparable influence at the start of training.

A.3 Encoding Continuous Expression Value

There are three major ways to transform a continuous value x_{i,m_g} into embedding: (1) Digitizing x_{i,m_g} and learns embedding for each digit; (2) Feeding x_{i,m_g} into Multilayer Perceptron (MLP); and (3) Feeding x_{i,m_g} into Fourier encoding function (Vaswani et al., 2017). Method (1) prevents the gradient from back-propagating to the gene compression model, and method (2) does not work well with the long-tail gene expression level distribution (Lopez et al., 2018). Therefore, scWEAVE adopts Fourier encoding function to transform x_{i,m_g} into embedding.

A.4 Multi-granular Contrastive Loss

To improve the influence of the exact label match relative to its hierarchically-related (ancestor/descendant) matches, we cap the latter's aggregate contribution. For cell i, let $W_{s_i} = \sum_{j:\text{hop}(\ell_i,\ell_j)=0, j\neq i} w_{ij}$ denote the total weight contributed by exact-match partners, and $W_{r_i} = \sum_{j:\text{hop}(\ell_i,\ell_j)\neq 0, j\neq i} w_{ij}$ the total weight contributed by the remaining hierarchically-related partners. Whenever $W_{s_i} > 0$, we rescale every related weight by a common factor so their aggregate never exceeds an α fraction of the exact weight:

$$w_{ij} \leftarrow w_{ij} \cdot \min\left(1,\ \frac{\alpha W_{s_i}}{W_{r_i}}\right), \quad \text{for } j \text{ with } \ell_i, \ell_j \text{ related but not exact} \tag{9}$$

This is necessary because hop-distance decay alone does not bound the exact match's relative contribution: an anchor with many weakly-related labels co-occurring in the batch/queue can still have its exact-match signal outweighed in aggregate, even though each individual related weight is small. We set $\alpha = 0.3$. Anchors with no exact match present ($W_{s_i} = 0$) are unaffected and use their related weights unscaled.

A.5 scWEAVE Transformer Architecture

The overall transformer architecture of scWEAVE follows Fig. 1c. scWEAVE uses a 6-layer transformer with 8 attention heads, each head having 64 dimensions, to encode the input. For predicting masks, it uses a separate 6-layer MLP. The shared cell factor has 128 dimension, while the remnant factor has 512 dimension.

A.6 Pretraining process

To ensure the stability of the training process, we use a three-step training strategy.

In **Step 1**, we warm-up the model through meta-gene-level MEP. We freeze the parameters of the gene compression module, and apply masking directly on the expression of meta-genes. The model is then trained to predict the masked meta-gene expression by minimizing the negative log-likelihood of negative binomial distribution (Sec. 3.3). For each cell i and masked meta-gene m_g, the prediction loss follows

$$\mathcal{L}_{s1}(i) = \sum_{m_g} \mathcal{L}_{\text{NB}}(x_{im_g}; \hat{x}_{im_g}, \hat{\theta}_{m_g}) \tag{10}$$

In **Step 2**, we unfreeze gene compression module, and train scWEAVE through MEP task on the original gene sets (Sec. 3.3). For each cell i and masked gene g, the prediction loss is calculated by combining the negative log-likelihood loss $\mathcal{L}_{\text{NB}}(x_{ig}; \hat{x}_{ig}, \hat{\theta}_g)$ and detection loss $\mathcal{L}_{\text{BCE}}(y_{ig}^{\text{detect}}, \hat{y}_{ig}^{\text{detect}})$ (Sec. 3.3):

$$\mathcal{L}_{s2}(i) = \sum_{g} \left(\mathcal{L}_{\text{NB}}(x_{ig}; \hat{x}_{ig}, \hat{\theta}_g) + \lambda \mathcal{L}_{\text{BCE}}(y_{ig}^{\text{detect}}, \hat{y}_{ig}^{\text{detect}})\right) \tag{11}$$

The weight λ is set to 0.5.

In **Step 3**, we incorporate the multi-granular contrastive regularization into the train objective to refine the shared cell factor space, and train the model with both $\mathcal{L}_{s2}(i)$ and the contrastive regularization $\mathcal{L}_{\text{contr}}(i)$ (Sec. 3.4). $\mathcal{L}_{s2}(i)$ requires a higher masking rate to maintain the MEP performance, while the contrastive regularization requires the shared cell factor to be less corrupted by the masking. To resolve the contradiction between the two terms, we adopt an alternating training strategy, by

iteratively training the model with $\mathcal{L}_{s2}(i)$ and $\mathcal{L}_{\text{contr}}(i)$. When minimizing $\mathcal{L}_{s2}(i)$, we set the mask rate to 0.5 for non-zero genes and 0.18 for zero genes, following Sec. 3.3. When minimizing $\mathcal{L}_{\text{contr}}(i)$, we set the mask rate to be 0 to maintain the integrity of shared cell factor.

scWEAVE is trained with 4 Nvidia A40 GPUs on CELLxGENE healthy human and mouse dataset for 3 epochs, where training **stage 1**, **stage 2**, **stage 3** each takes up 1 epoch. We used AdamW as the training optimizer, and schedule the learning rate with a linear warmup followed by cosine decay: the learning rate increases linearly over the first 3% of training steps (at least 500 steps), then decays following a cosine curve to 10% of its maximum value. The global training batch size is set to 2048 in stage 1 and 1024 in stages 2 and 3, with the maximum learning rate 1.2×10^{-5} in stage 1 and 6.0×10^{-6} in stages 2 and 3. Gradients are clipped to a maximum norm of 1.0.

A.7 Details of evaluation datasets

We evaluate scFMs on eleven datasets covering three multi-species, two mouse, and six human settings. These datasets span different tissues, biological conditions, and sequencing technologies. Dataset statistics are summarized in Table 1, and download sources are provided in Appendix Sec. A.12.

"Hippo" dataset This dataset contains single-nucleus RNA-seq profiles from five hippocampal and entorhinal subregions of adult humans, rhesus macaques (*M. mulatta*), and pigs (Franjic et al., 2022). It characterizes cellular diversity across these brain regions and species.

"Motor" dataset This dataset profiles the primary motor cortex using single-nucleus RNA-seq from humans, mice, rhesus macaques (*M. mulatta*), and marmosets (*C. jacchus*) (Bakken et al., 2021; BRAIN Initiative Cell Census Network (BICCN), 2021). It provides cross-species annotations for comparing cortical cell populations.

"AH-Outflow" dataset This dataset profiles the aqueous humor outflow tissues of the eye in humans, mice, pigs, rhesus macaques (*M. mulatta*), and cynomolgus macaques (*M. fascicularis*) (van Zyl et al., 2020). It was also used as a multi-species integration benchmark in SATURN (Rosen et al., 2024).

"Cortex-3B" dataset This dataset combines three mouse studies profiling the somatosensory cortex and hippocampal CA1, visual cortex, and frontal and posterior cortex (Zeisel et al., 2015; Hrvatin et al., 2018; Saunders et al., 2018). The studies used different sequencing protocols, providing a benchmark for integrating neuronal and non-neuronal populations across experiments.

"Retina" dataset The Mouse Retina Cell Atlas integrates single-cell RNA-seq profiles from nine studies using multiple sequencing technologies (Li et al., 2024). It provides a broad collection of retinal cell populations for evaluating integration across samples.

"Immune" dataset This human immune-cell dataset combines bone marrow and peripheral blood profiles from five studies using four sequencing chemistries. It is part of the scIB benchmark for atlas-level data integration (Luecken et al., 2022).

"Pancreas" dataset This human pancreas dataset combines single-cell expression profiles generated using multiple sequencing technologies. It is part of the scIB benchmark and contains annotations for 14 pancreatic cell types (Luecken et al., 2022).

"Lung Atlas" dataset This human lung dataset is included in the scIB benchmark (Luecken et al., 2022). It comprises 17 annotated cell types across multiple batches, supporting evaluation of integration within a heterogeneous tissue.

"Covid-19" dataset This dataset combines human peripheral blood immune-cell profiles from studies of COVID-19 (Arunachalam et al., 2020; Lee et al., 2020). It supports evaluation of cell representations across studies and biological conditions.

"Dendritic" dataset This dataset profiles human blood dendritic cell populations using Smart-seq2 (Villani et al., 2017). It contains four annotated cell types and was included in the batch-correction benchmark of Tran et al. (2020).

"PBMC" dataset This dataset combines human peripheral blood mononuclear cells from the 10x Genomics PBMC 3k dataset and the 68k PBMC dataset of Zheng et al. (2017). The evaluation dataset contains eight annotated cell types.

A.8 Details of ortholog retrieval

SATURN. SATURN produces gene-to-macrogene assignment scores that can be used to retrieve ortholog genes across species. It generates two sets of assignment scores: (1) initial assignments obtained by applying k-means clustering to the protein embedding space (ESM-2) and (2) refined assignments obtained after training SATURN on the evaluation datasets. We evaluate ortholog retrieval using each set of assignments separately. For every query gene from a held-out species, we computed the cosine similarity between its assignment vector and those of all human genes. We then treated the top-k nearest human genes ($k \in \{1, 5\}$) as candidate ortholog genes, and evaluated retrieval accuracy using the same procedure as for scWEAVE. We denote the results obtained from the initial and refined assignments by "SATURN (init)" and "SATURN (trained)", respectively.

PCA-ESM. We projected the ESM-2 protein embedding into a 512-dimensional space using principal component analysis (PCA). For every query gene from a held-out species, we ranked all human genes according to their cosine similarity in the PCA space and treated the top-k nearest human genes ($k \in \{1, 5\}$), as candidate ortholog genes. We evaluate retrieval accuracy using the same procedure as for scWEAVE.

A.9 Details of novel cell type detection

For each method, let H_{ref} and H_{qry} denote the cell embeddings from the reference species (human) and the query species, respectively. For a query cell i with embedding $\mathbf{h}_i \in H_{\mathrm{qry}}$ and a reference cell j with embedding $\mathbf{h}_j \in H_{\mathrm{ref}}$, we define the cosine distance as:

$$d_{\cos}(\mathbf{h}_i, \mathbf{h}_j) \;=\; 1 - \frac{\mathbf{h}_i^\top \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \cdot \|\mathbf{h}_j\|}, \tag{12}$$

Let $\mathcal{N}_k(\mathbf{h}_i) \subseteq H_{\mathrm{ref}}$ denote the k reference embeddings with the smallest cosine distances to $\mathbf{h}_i$ ($k = 15$). The novelty score of query cell i is the mean cosine distance to these neighbors:

$$s_i \;=\; \frac{1}{k} \sum_{\mathbf{h}_j \in \mathcal{N}_k(\mathbf{h}_i)} d_{\cos}(\mathbf{h}_i, \mathbf{h}_j) \tag{13}$$

A higher score indicates that the query cell is more distant from its nearest reference cells.

Let $\mathcal{I}_c$ denote the set of query cells with ground-truth cell type label c. We obtain the cell-type-level novelty score by averaging the scores of these cells:

$$s_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} s_i. \tag{14}$$

Let $\mathcal{L}_{\mathrm{ref}}$ denote the set of ground-truth cell type labels represented in the human reference. We assign each query cell type a binary novelty label,

$$y_c = \mathbf{1}[c \notin \mathcal{L}_{\mathrm{ref}}], \tag{15}$$

where $y_c = 1$ indicates a novel cell type and $y_c = 0$ indicates a cell type represented in the reference. Novelty is therefore defined relative to the cell types represented in the reference dataset. We evaluate the correspondence between s_c and y_c using cell-type-level AUPRC, as described in Appendix Sec. A.11.3.

A.10 Details on imputation of missing gene expression

Since the gene expression data is extremely sparse, applying random mask uniformly across all genes will bias heavily towards zero-expression genes. We instead select masking percentage separately for non-zero and zero expression genes (masking percentage 50% for non-zero values, and percentage 18% for zero values). For each masked gene, we measure the imputation accuracy using Pearson and Spearman Correlation, because these correlation-based metrics are invariant to the scaling of the data and can better capture the variance of expression that encodes true biological information. The details of the evaluation metrics are described in Appendix Sec. A.11.4.

A.11 Details on evaluation metrics

A.11.1 Cosine similarity score

The cosine similarity score is used to calculate the similarity of meta-gene assignment score between two genes across two species. Denoting gene g and g', with meta-gene assignment score denoted as $\mathbf{S}_g$ and $\mathbf{S}_{g'}$ respectively, the cosine similarity score is calculated following:

$$\cos(\mathbf{S}_g, \mathbf{S}_{g'}) = \frac{\mathbf{S}_g^T \cdot \mathbf{S}_{g'}}{\|\mathbf{S}_g\| \cdot \|\mathbf{S}_{g'}\|} \tag{16}$$

A.11.2 ARI, ASW, ASW-B, and GC scores

Given the clustering output and ground truth cell type label, ARI score measures how similar the cluster label is to the ground truth label. A higher ARI indicates better agreement between the inferred clusters and ground-truth cell type labels.

On the other hand, ASW score directly measures the separation of cell types in the embedding space by comparing the distance between cells of different cell types with the distance between cells of the same cell type. The ASW score ranges between -1 and 1. A higher score means a better separation of cells of different cell types and a better mixing of cells of the same cell type.

ARI Given the ground truth cell type label and the cluster label, we first construct the contingency table $\mathbf{N}$, where element n_{ij} encodes the number of cells with both ground truth label i and cluster label j. ARI score then follows:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \Big/ \binom{n}{2}\right]}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \Big/ \binom{n}{2}\right]} \tag{17}$$

a_i is the row sum of the contingency table: $a_i = \sum_j n_{ij}$. b_j is the column sum of the contingency table: $b_j = \sum_i n_{ij}$. n is the total number of cells.

Calculate ARI with optimal clustering We calculate ARI scores following the scIB benchmarking pipeline (Luecken et al., 2022). We cluster the cells by applying Leiden clustering algorithm (Traag et al., 2019) on cell embeddings. To decide the optimal cluster "resolution" for the Leiden clustering algorithm, we conduct grid search on the "resolution" starting from 0 to 2 with step-size 0.2 and select the optimal clustering result that has the highest score against the ground truth label. For the cell embedding generated by each method, we conduct the grid search and optimal clustering separately, and then calculate the ARI score between the optimal clustering label and the ground truth cell type label. We run optimal clustering under 5 different random seeds, and report the mean and standard deviation of the ARI scores in Table 5.

ASW ASW is calculated directly on cell embedding using the ground truth cell type label. For each cell i, the silhouette width (Rousseeuw, 1987) measures the average distance of the cell to other cells of the same cell type $d_1(i)$, and the average distance of the cell to the other cells of the closest cell type $d_2(i)$, and then calculates score following:

$$s(i) = \frac{d_2(i) - d_1(i)}{\max(d_1(i), d_2(i))} \tag{18}$$

ASW is then obtained by averaging silhouette scores across all cells.

ASW-B ASW-Batch measures the mixing of cell batches in the embedding space. Similar to ASW score, we calculate $s(i)$ for cell i, but use the batch label instead of cell type label. For each cell type label j, we then calculate

$$\text{ASW-B}_j = \frac{1}{|C_j|} \sum_{i \in C_j} 1 - |s(i)| \tag{19}$$

The final ASW-B score is ASW-B$_j$ averaged across cell type js.

GC Graph connectivity (GC) score measures how well the cells of the same cell type between batches are mixed in the latent space. To calculate GC, we first construct a kNN graph connecting cells from all batches. Then for each cell type j, we select the cells that belong to the cell type and denote the corresponding subgraph by $G_j(N_j, E_j)$ where j denotes the cell type. The GC score of cell type j follows

$$\text{GC}_j = \frac{|\text{LCC}(G_j)|}{|N_j|} \tag{20}$$

$|\text{LCC}(G_j)|$ denotes the largest number of connected cells within the subgraph G_j, and $|N_j|$ denotes the total number of cells in the subgraph. The GC score of the whole dataset is the average GC score of all cell types.

A.11.3 AUPRC SCORE

We use AUPRC to assess how well the cell-type-level novelty scores rank novel cell types above those represented in the reference. For each of the C evaluated query cell types, let s_c denote its novelty score and $y_c \in \{0, 1\}$ its novelty label, as defined in Appendix Sec. A.9.

At threshold t, a cell type is predicted to be novel if $s_c \geq t$. Precision and recall are defined as

$$P(t) = \frac{\sum_{c=1}^{C} \mathbf{1}[s_c \geq t]\, y_c}{\sum_{c=1}^{C} \mathbf{1}[s_c \geq t]}, \qquad R(t) = \frac{\sum_{c=1}^{C} \mathbf{1}[s_c \geq t]\, y_c}{\sum_{c=1}^{C} y_c}. \tag{21}$$

Let $t_1 > t_2 > \cdots > t_M$ denote the $M \leq C$ distinct novelty scores, and let $P_n = P(t_n)$ and $R_n = R(t_n)$. We calculate AUPRC using the non-interpolated average precision formulation:

$$\text{AUPRC} = \sum_{n=1}^{M} (R_n - R_{n-1}) P_n, \qquad R_0 = 0. \tag{22}$$

Cell types with identical scores are included at the same threshold. Each evaluated cell type contributes equally, regardless of its number of cells.

A.11.4 PEARSON AND SPEARMAN CORRELATION

We evaluate imputation accuracy using Pearson and Spearman correlations between ground-truth and imputed gene expression. Before calculating either metric, we log-normalized both expression data using the ground-truth library size:

$$\bar{x}_{ig}^{a} = \log_2 \left(1 + \frac{x_{ig}^{a}}{\ell_i} \right), \qquad a \in \{\text{gt}, \text{pred}\}, \tag{23}$$

where x_{ig}^{gt} and x_{ig}^{pred} denote the ground-truth and imputed expression of gene g in cell i, respectively, and ℓ_i is the ground-truth library size of cell i.

For each cell, we calculated correlations across the G masked genes. Pearson correlation follows:

$$r_i = \frac{\sum_{g=1}^{G} \left(\bar{x}_{ig}^{\text{gt}} - \mu_i^{\text{gt}} \right) \left(\bar{x}_{ig}^{\text{pred}} - \mu_i^{\text{pred}} \right)}{\sqrt{\sum_{g=1}^{G} \left(\bar{x}_{ig}^{\text{gt}} - \mu_i^{\text{gt}} \right)^2} \sqrt{\sum_{g=1}^{G} \left(\bar{x}_{ig}^{\text{pred}} - \mu_i^{\text{pred}} \right)^2}}, \tag{24}$$

where $\mu_i^a = G^{-1} \sum_{g=1}^{G} \bar{x}_{ig}^{a}$ is the mean normalized expression across evaluated genes.

Spearman correlation is calculated as the Pearson correlation between the ranks of the two expression profiles. Let $q_{ig}^a = \mathrm{rank}_g(\bar{x}_{ig}^a)$ denote the rank of gene g within cell i, Spearman correlation follows:

$$\rho_i = \frac{\sum_{g=1}^{G} \left(q_{ig}^{\mathrm{gt}} - \bar{q}_i^{\mathrm{gt}}\right)\left(q_{ig}^{\mathrm{pred}} - \bar{q}_i^{\mathrm{pred}}\right)}{\sqrt{\sum_{g=1}^{G} \left(q_{ig}^{\mathrm{gt}} - \bar{q}_i^{\mathrm{gt}}\right)^2}\sqrt{\sum_{g=1}^{G} \left(q_{ig}^{\mathrm{pred}} - \bar{q}_i^{\mathrm{pred}}\right)^2}}, \tag{25}$$

where $\bar{q}_i^a = G^{-1}\sum_{g=1}^{G} q_{ig}^a$. Both metrics range from -1 to 1, with higher values indicating stronger positive agreement between ground-truth and imputed expression.

A.12 Data and code availability

Pretraining data The healthy human and mouse single-cell data used for pretraining were obtained from CZ CELLxGENE Discover (`https://cellxgene.cziscience.com/collections`) and accessed using the `cellxgene-census` Python package (`https://chanzuckerberg.github.io/cellxgene-census/`).

Multi-species evaluation datasets The "Hippo" dataset is available from the Gene Expression Omnibus (GEO) under accession GSE186538. The "Motor" dataset is available from CZ CELLxGENE Discover under collection identifier `367d95c0-0eb0-4dae-8276-9407239421ee`. The original "AH-Outflow" dataset is deposited in GEO under accession GSE146188. The processed five-species dataset released with SATURN (Rosen et al., 2024) is also available at `http://snap.stanford.edu/saturn/data/ah_atlas_export.tar.gz`.

Mouse evaluation datasets The three studies comprising "Cortex-3B" are available from GEO under accessions GSE60361, GSE102827, and GSE116470. The "Retina" dataset is available from CZ CELLxGENE Discover under dataset identifier `23f77ae6-10af-4307-b136-76b26654ae7d`.

Human evaluation datasets The "Immune", "Pancreas", and "Lung Atlas" datasets are available from the scIB benchmark (Luecken et al., 2022) through its Figshare deposit: `https://doi.org/10.6084/m9.figshare.12420968`. The source datasets for "Covid-19" are available from GEO under accessions GSE155673, and GSE149689. The processed "Dendritic" and "PBMC" datasets were obtained from the scEval collection (Liu et al., 2026). Dataset access information is provided in the scEval repository: `https://github.com/HelloWorldLTY/scEval`.

Code availability The source code of scWEAVE is available at `https://anonymous.4open.science/r/scWEAVE-EA55`.

A.13 Visualizations of cell representation

We visualized the cell embedding of scWEAVE and baseline methods using UMAP, and annotated the cells by the ground truth cell type label and cells' batch id (Figs. 4, 5, 6). Ideally, cell embedding that is removed of batch effect should have (1) cells of same cell type aligned across batches in the embedding space, and (2) cells of different cell types well separated in the embedding space.

Fig. 4 shows the cell embedding visualization of scWEAVE and baseline scFMs on multi-species "Hippo" dataset. Among all methods, scWEAVE is the only method that align common cell type across species in the UMAP visualization.

Fig. 5 shows the cell embedding visualization of mouse "Cortex-3B" dataset. Among all methods, scWEAVE and scMulan have the best batch alignment and cell type separation in the cell embedding space. scGPT performs the worst by mixing all cell types in UMAP visualization.

Fig. 6 shows the cell embedding visualization of human "Immune" dataset. Among all methods, scWEAVE, TranscriptFormer, and UCE have the best batch alignment and cell type separation in the cell embedding space.

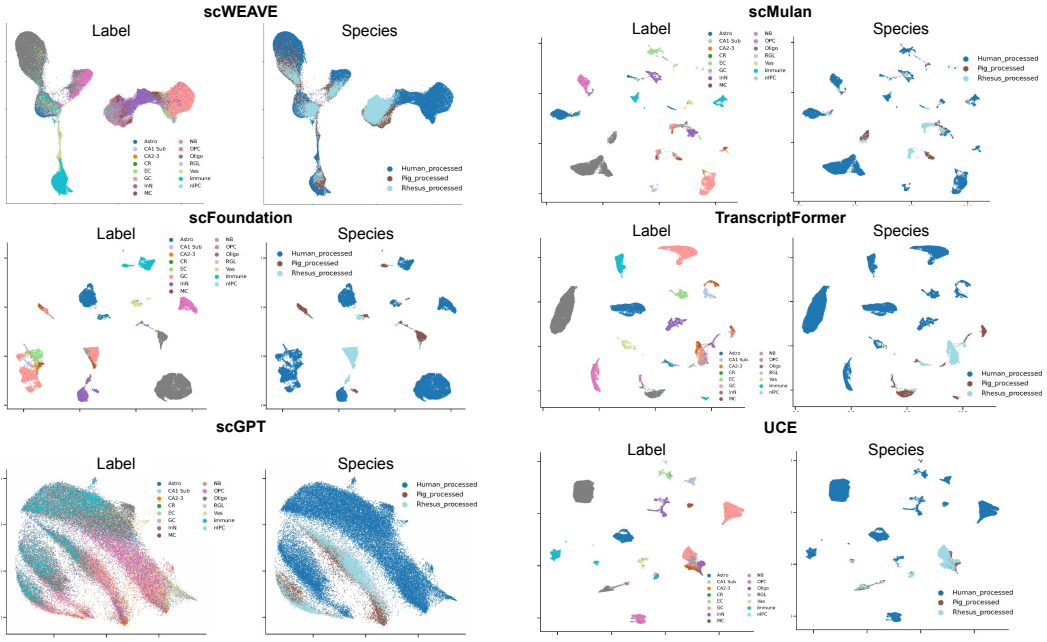

Figure 4: UMAP visualization of cell embedding of "Hippo" 3-species dataset from scWEAVE and baseline scFMs. Cells are colored by the ground truth cell type label in the left column and the batch id in the right column.

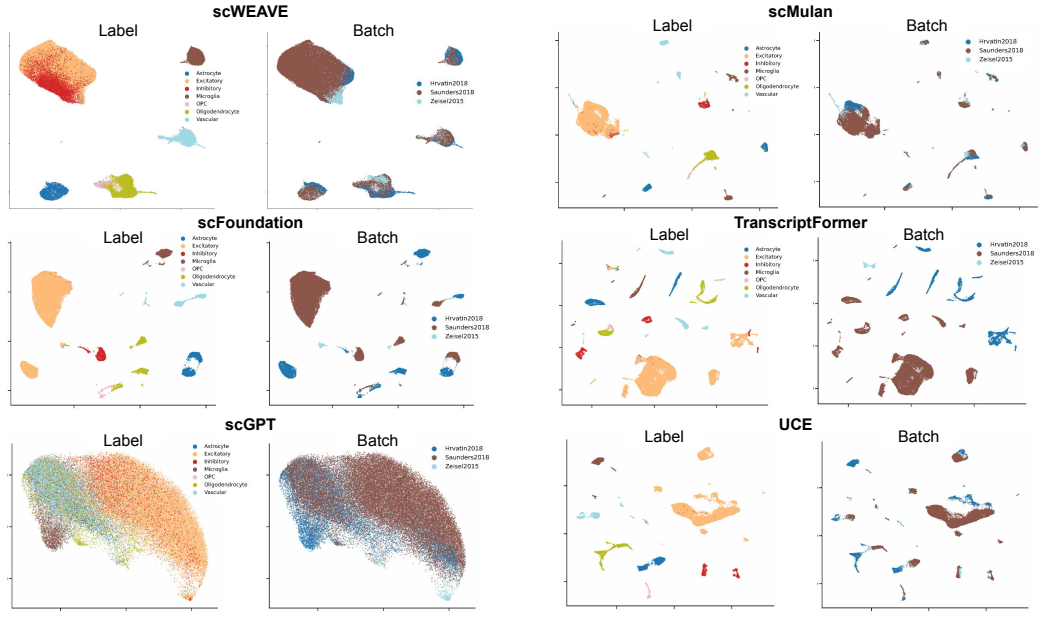

Figure 5: UMAP visualization of cell embedding of "Cortex-3B" mouse dataset from scWEAVE and baseline scFMs. Cells are colored by the ground truth cell type label in the left column and the batch id in the right column.

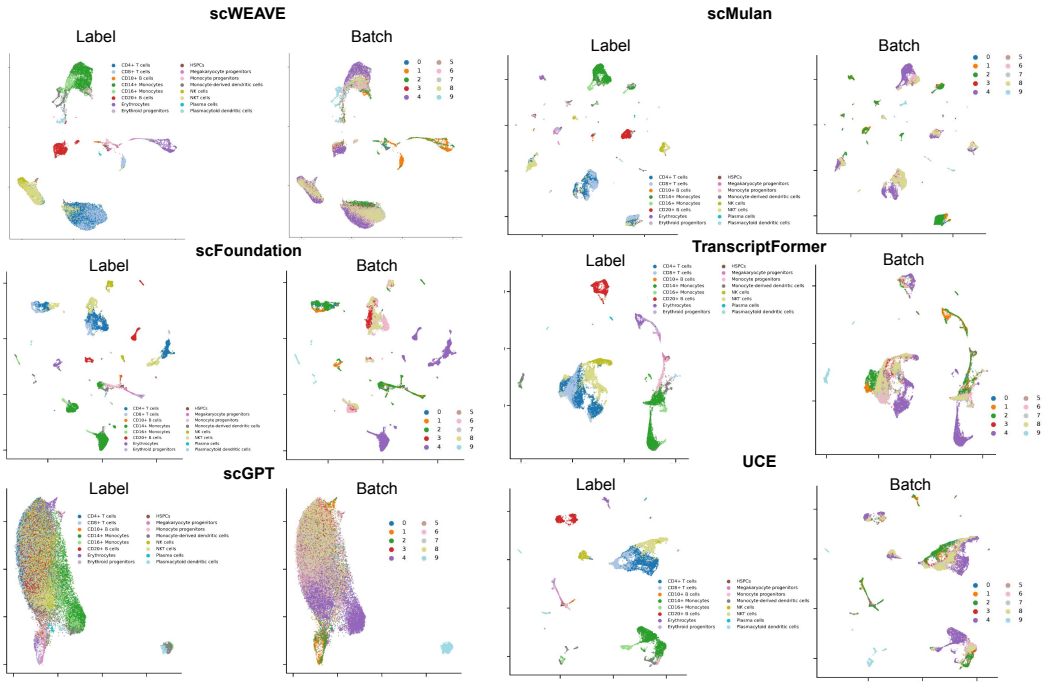

Figure 6: UMAP visualization of "Immune" dataset cell embedding from scWEAVE and baseline scFMs. Cells are colored by the ground truth cell type label in the left column and the batch id in the right column.

A.14 Experiments compute resources

Pretraining of scWEAVE was done using four Nvidia A40 GPUs with 48 GB of GDDR6 memory. Inference of each test dataset was performed using one Nvidia A40 GPU with 48 GB of GDDR6 memory.

A.15 Potential Societal Impacts

scWEAVE's efficient design democratizes access to single-cell analysis, and indirectly contributes to healthcare and medicine by enabling more labs to contribute to precision medicine. However, we caution that biased training data could amplify health disparities, necessitating careful dataset curation. We emphasize the importance of using balanced datasets.