CS690: Computational Genomics

Instructor: Hamim Zafar
Team Members: Vineet Kumar, Tejas Chikoti, Prasoon Patel
Team Number: 3
Nov 27, 2023

1 INTRODUCTION

The integration of mosaic single-cell data has become a pivotal challenge in leveraging the wealth of information contained within diverse single-cell datasets. This process, often referred to as 'mosaic data integration,' involves consolidating and extracting insights from various molecular assays. While current methods for mosaic data integration primarily rely on overlapping features between modalities, the increasing number and complexity of single-cell datasets necessitate the development of techniques specifically tailored for this purpose.

Several existing approaches address the mosaic data integration challenge. Notable examples include UINMF, which introduces a latent metagene matrix in the factorization problem, MultiMAP, a graph-based method assuming a uniform distribution of cells across a latent manifold structure, and StabMap[3], which projects cells onto reference coordinates, utilizing all available features regardless of dataset overlap. However, as the field advances, there is a need for more sophisticated methods.

This research project aims to develop a deep generative model for mosaic data integration that outperforms existing methods. Unlike some current approaches that separate integration and batch correction tasks, our proposed model, based on a VAE-based architecture, unifies these tasks into a cohesive framework.

We present two approaches for mosaic data integration using deep generative models. Firstly, we utilize ScVI[4] to reconstruct shared features from highly variable RNA features and simultaneously train TotalVI[2] to reconstruct shared protein features from all protein features. By modifying the loss function, we aim to preserve the distance relationships between cells in the embedding space of RNA and protein proportionally to their distances in the shared feature space.

Secondly, employing three encoder-decoder networks, we seek to learn a joint embedding of both protein and RNA shared features. The first encoder learns the embedding of protein features by reconstructing shared protein features from unshared features, the second encoder learns the embedding of RNA shared features by reconstructing shared RNA features from unshared RNA features, and the third encoder is trained to simultaneously reconstruct shared RNA and protein features from their corresponding shared features. The resulting embeddings are aligned based on the modality from which the cells originate. Detailed descriptions of these methods are provided in Section 3, while Sections 4 and 5 present the results and ablation studies, respectively.

2 RELATED WORKS

MaxFuse[1] is a data integration method that works on weakly related-modal data and performs integration through iterative coembedding, data smoothing and cell matching from each modality. It has shown a high-quality integration on spatial proteomic data with single-cell sequencing data (CITE-seq PBMC[Human peripheral blood mononuclear cells] data with 228 antibodies **??**). MaxFuse uses a cell-cell nearest-neighbour graph using features from one modality represents cell-cell similarities for that modality. It boosts the signal to noise ratio in linked features by shrinking the value of linked features, for each cell, towards the cells's graph neighbour average. After obtaining the initial matching of cells across modalities by applying linear assignment to the distance between cross-modal cell-pairs, we create a joint embedding based on all features of the cross-modal matched cell pairs. Fuzzy smoothing is applied to these joint embedding coordinates based on the previously calculated nearest-neighbor graph. Subsequently, linear assignment is applied to update the cell-matching across modalities based on pairwise distances of the fuzzy-smoothed joint embedding coordinates. The screened matched pair from the last iteration is used as pivots to compute the final joint-embedding and calculate the best match for previously un-matched cells.

3.1 Data

3.1.1 Dataset Description. In our experiments, we utilize the **CITE-seq PBMC** dataset. This dataset is a comprehensive and well-annotated collection of human peripheral blood mononuclear cells (PBMCs), widely employed in various data integration studies. It encompasses transcriptome-wide expression data for over 10,000 cells and surface protein expression data for 224 proteins. For our specific analysis, we focus on a subset of 10,000 cells. These cells were derived from healthy donors and underwent profiling using the 10x Genomics CITE-seq platform. Within our chosen subset, we encounter two levels of annotation for cell types in both modalities. The broader annotation level comprises 8 labels, while the finer level offers a more detailed annotation with 31 distinct labels.

3.1.2 Dataset Preprocessing. Given the extensive gene expression data in the CITE-seq dataset, comprising over 19,000 genes, attempting to model the entire set using ScVI is practically unfeasible. To address this, we opted to extract highly variable genes for training with ScVI. Additionally, we augmented these highly variable genes with shared genes, ensuring no duplication. For the majority of our experiments, we selected the top 2000 highly variable genes, along with 177 shared genes. After eliminating duplicates, we retained approximately 2080 gene features within the RNA modality.

No specific preprocessing steps were applied to the Protein modality.

3.2 ProtSc-I

As you can see in Figure 1, the entire model consists of two encoder-decoder networks, hence the name. The VAE used for modeling gene expression data is standard ScVI model, we made some minor changes to have more control, so that we can run experiments like using ushared features to reconstruct shared features, reconstructing shared features using all features, or only shared features. The input to the ScVI is our 10000×2081 dimension gene expression matrix. We used 20 as latent dimension in both the VAEs. We train the scvi model (rodel-name of model) to reconstruct 2081 gene features and modified TotalVI (podel-name of the model) to reconstruct 224 protein features. For loss functions, we use a reconstruction loss for both podel and rodel. The reconstruction losses help the models to learn better lower dimensional embeddings. On top of that we have to make the latent space of rodel and podel to overlap. For this task we define a cross-loss, which forces the embedding spaces to overlap. Suppose, x_i and y_j are two cells from rna and protein modality respectively. Let $\hat{x}_i \in \mathbb{R}^{177}$ and $\hat{y}_j \mathbb{R}^{177}$ represent shared features of x_i and y_j with one-one correspondence. The the cross-loss is defined as:

$$\text{Cross Loss}(x_i, y_j) = (Distance1(f(x_i), g(y_i)) - Distance2(\hat{x}_i, \hat{y}_j))^2$$

where $f(x)$ and $g(y)$ represents encoders of rodel and podel respectively. $Distance1(x, y)$ is some metric to quantify distance between embeddings of x and y, while $Distance2$ is another metric to quantify the distance (similarity) between shared linked features of x and y. $Distance1(x, y)$ and $Distance2(x, y)$ can also be same function like Euclidean.

We cannot directly apply distance metric on linked shared features as both $\hat{x}_i$ and $\hat{y}_j$ have different scales and we also do not know what is the idea of distance inside shared feature space. So we took inspiration from [1], and projected to respective leading singular sub-spaces $(\hat{x}_i^p, \hat{y}_j^p) \in \mathbb{R}^{20 \times 20}$ (we choose to use only 20 components from decomposition) and then calculated the distances. In our experiments, we have experimented with 2 different distance metrics with using same function for both $Distance1()$ and $Distance2()$:

(1) **Pearson Correlation Distance:** The Pearson correlation coefficient (r) measures the strength and direction of a linear relationship between two variables. We define Pearson Correlation Distance as:

$$d_{corr}(x, y) = 1 - r = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

This distance ranges from $[0, 2]$ as the correlation coefficient can range from -1 to +1:

- -1: Perfect negative correlation
- 0: No correlation
- +1: Perfect positive correlation

(2) **Euclidean Distance:** The euclidean distance as simply specified as MSE Los.

$$\mathrm{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$$

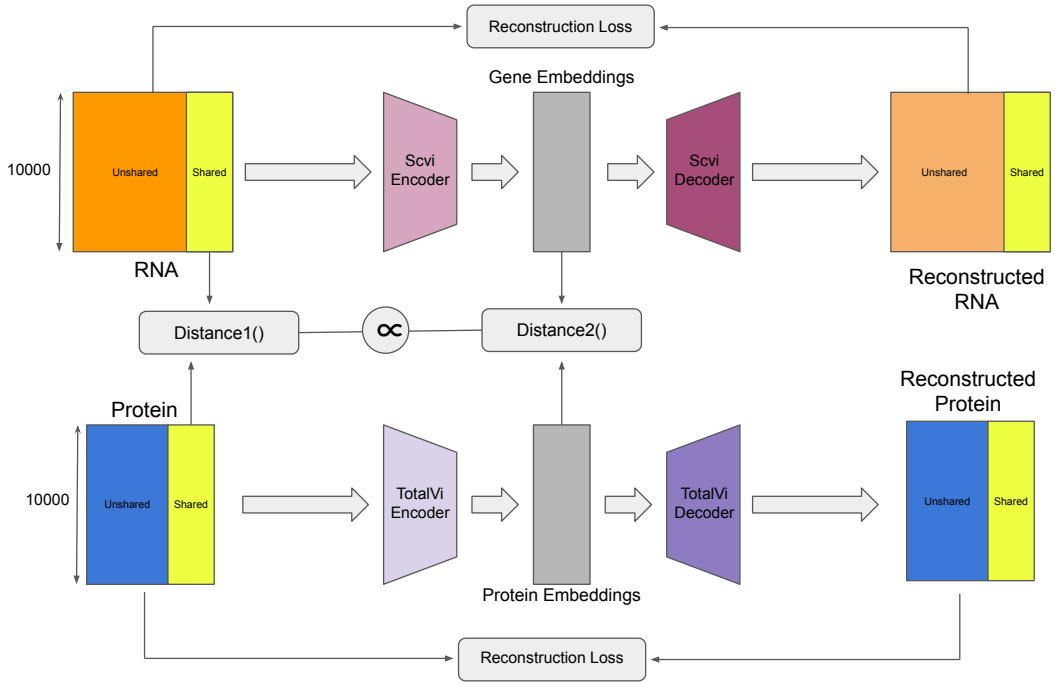


Fig. 1. Overview of ProtSc model architecture

3.3 ProtSci-II

In this method we use ProteinVI for unshared protein feature matrix of 10, 000 cells, with 54 proteins; ScVI for unshared RNA expression matrix of 10, 000 cells and 1911 genes and TotalVI[2] for shared protein and RNA features of 10, 000 cells and 170 common features.

The approach can be summarized as follows:

- X cells of RNA shared features concatenated with Y cells of Protein shared features are sent to reconstruct in TotalVI using the same encoder and different decoders.
- The same X cells of RNA unshared features are sent to reconstruct in ScVI.
- The same Y cells of Protein unshared features are sent to reconstruct in ProteinVI.
- There are two cross-loss(L1-loss) terms namely for the X cells of RNA-unshared features and Y cells of Protein-unshared features for ProteinVI-TotalVI and ScVI-TotalVI respectively.

We modify the TotalVI architecture for our requirements as follows:

- We concatenate the protein with [0.0,1.0] and RNA with [1.0,0.0] respectively will enable the model to implicitly learn a better representation
- In modeling the frequency counts of the Protein and RNA the same vector was used in the original implementation, but we have separated the two from each other.
- We have taken care in isolating the decoders and passing the corresponding embeddings though the KL-divergence loss is calculated as a whole.

We then train these three networks simultaneously in three experiments.

- The first experiment consists of using a warmup with prioritizing the reconstruction loss in the initial epochs and then linearly increasing the weight of KL-divergence loss, during the end of the training paradigm.
- The second experiment consists of using without warmup and fixing the coefficients of loss across epochs.
- The third experiment consists of changing the batch paradigm,sending the same randomly linked protein and RNA in the Shared encoder
- The fourth experiment consists of using SGD as an optimizer

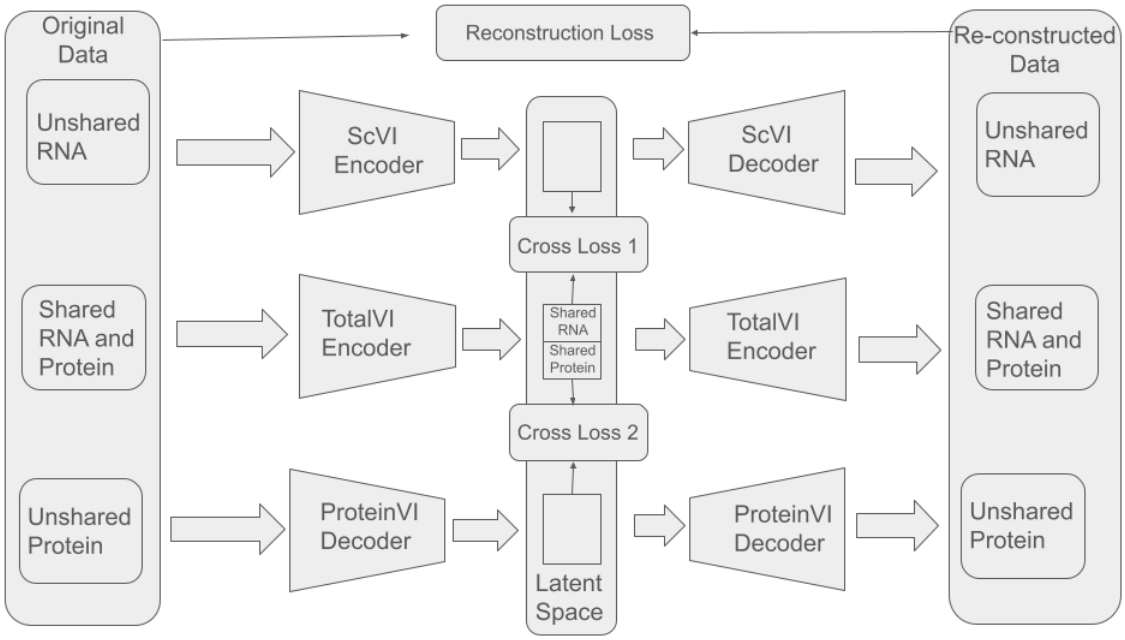


Fig. 2. Overview of ProtSci-2 architecture

4 RESULTS

4.1 Evaluation

We use the standard label transfer task, in which we transfer cell type labels from RNA to Protein and vice-versa with coarse cell-type annotations only. We calculate the **Adjusted rand Index (ARI) score** to measure the accuracy of our models.

For quanlitative evaluation of our models, we plot the embedding space in 2D using UMAP to ensure if structure is conserved or not in low dimension. We also visualize clusters with cell type labels to further verify the purity of clusters formed.

4.2 Inference

For performing label transfer from one modality $\mathbf{X}$ to modality $\mathbf{Y}$ (here RNA and Protein). We perform 3 steps:

(1) Get low dimensional embeddings ($\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$) of $\mathbf{X}$ and $\mathbf{Y}$.
(2) Find the nearest neighbour of each cell from modality $\mathbf{X}$ in $\mathbf{Y}$ according to suitable distance metric. We use $Distance2()$ in ProtSc-I and L1 distance in ProtSc-II between $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$.
(3) Transfer the label of nearest neighbours from $\mathbf{X}$ to $\mathbf{Y}$.

We have performed a fix set of experiments, to evaluate various ideas based on ARI score and the quality of UMAPs. As you can see in Table 1, we have presented results for the difference $Distance1()$ and $Distance2()$, while using 3 different approaches to sample cells for training the model.

(1) No Shuffle : We just use corresponding cells from RNA and Protein.
(2) Random Shuffle : We randomly chose cells from RNA and Protein.
(3) Probability Shuffle : We try to sample cells which are either very less correlated or very highly correlated based on Pearson Correlation Distance with high probability. We want to train our model with extreme case examples more frequently.

However, later, we found out from plotting the UMAPs of the embedding space, that the ARI score is not a good metric to guess the accuracy of our result. We can see from the Tabel 1 and Figure 4 that even though we have an ARI score of 0.4 there is very little overlapping of protein and RNA clusters. One reason we guessed was that the underlying distribution of both Protein and RNA latent spaces may not be the same. We also notice that initially keeping more weightage on cross loss and with iteration increasing weight on kl divergence losses and reconstruction losses results in better performance. The corresponding UMAP plot of label transfer and protein-RNA tensor concatenation can be seen in Fig. 4. From the Table 1 we can also confirm that shuffling on the basis of probability always works better than randomly sampling cells.

Column 1	Column 2	Column 3
Shuffle Distance	Pearson Distance	Euclidean Distance
No Shuffle	0.22/0.24	0.42/0.50
Random Shuffle	0.26/0.17	0.27/0.17
Probability Shuffle	0.22/0.33	0.32/0.24

Table 1. Best ARI score (Protein to RNA/RNA to Protein) achieved in all listed Experiments

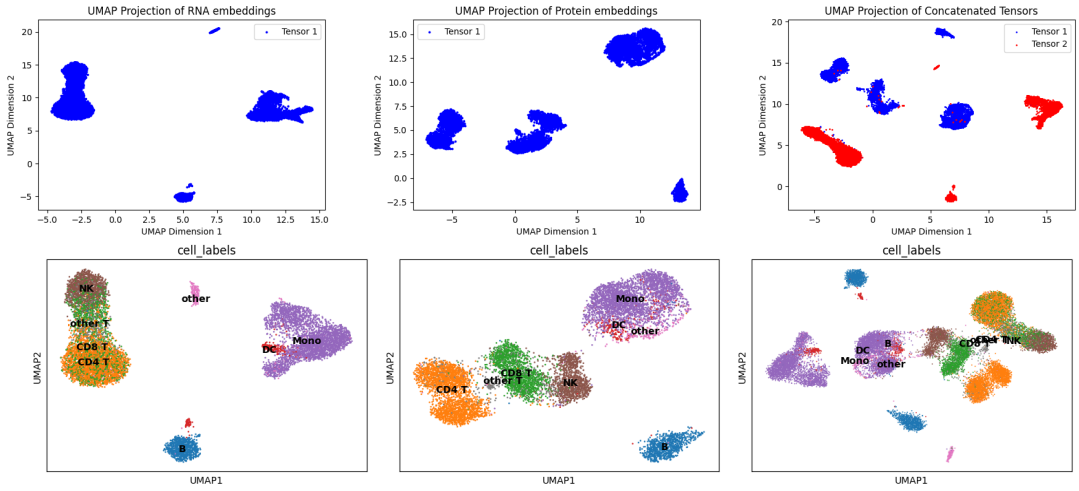


Fig. 3. (Top) Visualization of embedding space in 2D. (Bottom) Cells in embedding space marked with cell labels. (Last column) Project of latent vectors of Protein and RNA using a single UMAP transform. Experiment: Euclidean distance with No shuffle

4.4 Code description ProtSc I

The folders namely vineet-scvi contain the implementations for the ProtSc I. The environment yaml file is vineet-en for the same. There are 6 notebooks int the folder notebooks/experiments for all six experiments. For viewing the results for all experiments in folder fig1,fig2....fig6 to see the Ablation and loss function plots. Further some of the experiments were hosted on comet-ml whose shareble link is here All Results

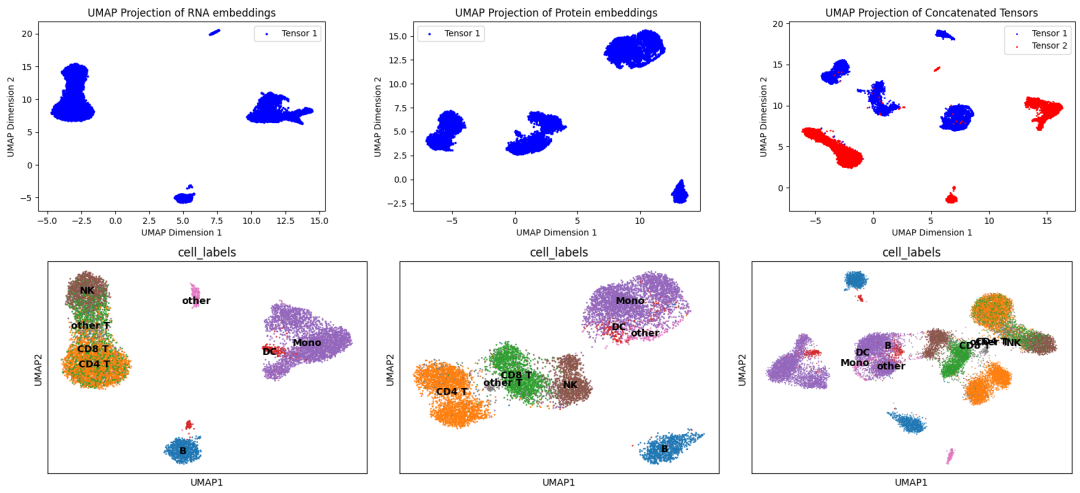


Fig. 4. (Top) Visualization of embedding space in 2D. (Bottom) Cells in embedding space marked with cell labels. (Last column) Project of latent vectors of Protein and RNA using a single UMAP transform. Experiment: Pearson Correlation distance with Probability shuffle.

4.5 Code description ProtSc II

The folders namely tejas-scVI contain the implementations for the ProtSc II. The environment yaml file is tejas-env for the same. There are three notebooks namely Final code,Final code with warmup,Final code without batch paradigm.The first one is without warmup and the second one is with warmup and the third one is with the batch shuffling paradigm where we do restrict the dataloader to sample a randomly linked protein-RNA pair every epoch. For viewing the results go to the figures folder to see the Ablation and loss function plots. Further some of the experiments were hosted on comet-ml whose shareble link is here All Results The original code has been modified from the link is here Code

4.6 ProtSc II

In this method also we use the standard label transfer task, in which we transfer cell-type labels from RNA to Protein and vice-versa with coarse cell-type annotations only. For Label transfer, we used euclidean distance as a metric to find similar cells for clustering. ARI (Adjusted Rand Score) is again used to evaluate our label transfer predictions and also plot the UMAPs of the embedding space for better visualization of clusters. We can see the dependence of different experiment with their corresponding ARI score in Table Fig. 2. Finally, the we shows various UMAP at various epochs, UMAP of protein-rna concatenated and Lossplots for different Experiments.

We then train the three networks simultaneously in three experiments.

(1) The first experiment consists of using a warmup with prioritizing the reconstruction loss in the initial epochs and then linearly increasing the weight of KL-divergence loss, during the end of the training paradigm.
(2) The second experiment consists of using without warmup and fixing the coefficients of loss across epochs.
(3) The third experiment consists of changing the batch paradigm,sending the same randomly linked protein and RNA in the Shared encoder
(4) The fourth experiment consists of using SGD as an optimizer

Experiment	ARI Score
Batch effects	0.46
Without Warmup	0.39
With Warmup	0.49
SGD (Stochastic Gradient Descent)	-0.002

Table 2. Best ARI score (Protein to RNA/RNA to Protein) achieved in all listed Experiments
*These all ARI score are from shuffled dataset comparison

4.6.1 Comparitives. The results and figures corresponding to this section is in section 9 Fig. 5

4.6.2 Batch Change. The results and figures corresponding to this section is in section 9 Fig. 6

4.6.3 SGD. The results and figures corresponding to this section is in section 9 Fig. 8

4.6.4 With Warmup. The results and figures corresponding to this section is in section 9 Fig. 9

5 ABALATION STUDIES

5.1 Batch Change

The results and figures corresponding to this section is in section 9 Fig. 10

5.2 SGD

The results and figures corresponding to this section is in section 9 Fig. 11

5.3 With Warmup

The results and figures corresponding to this section is in section 9 Fig. 12

5.4 Without Warmup

The results and figures corresponding to this section is in section 9 Fig. **??** and Fig. 7

5.5 ProtSc II

6 OBSERVATIONS AND FUTURE DIRECTIONS

6.1 ProtSc II

(1) We observe that SGD significantly underperforms and is not able to identify meaningful clusters in the RNA.
(2) Moreover the warmup paradigm does not work as in the results the clustering requires a more finer reconstruction.
(3) The best result is achieved by using hardcoded coefficients for the reconstruction loss that give it an edge.
(4) Propagating the matches using a fix threshold does increase the ARI score from 0.45 to 0.49 and accuracy to 0.7 from 0.68 and similar trends are observed in all the methods.
(5) We are looking for future directions pertaining to similarities in max-fuse where we do modify the prior and ideally make closer cell matches and employ a similar iterative learning scheme
(6) Look for sharing more embeddings across the networks and aim for a better method for cross loss.

7 CONTRIBUTIONS

(1) **Vineet Kumar:**
- Completely formulate the architecture of ProtSc-I from scratch using the codes of ProteinVI and TotalVI modified by Tejas.
- Wrote dataset class and preprocessing codes for most of the experiments.
- Also designed the probability based shuffling based on correlation for cell sampling.
- Performing experiments and evaluation on ProtSc-I.

(2) **Tejas Chikoti:** Completely formulated the ProtSc-II from scratch and helped in the initial code of ProtSc-I in creating ProteinVI,TotalVI.

8 (3) **Prasoon Patel:**

- Did the complete pre-processing on citeseq-PBMC data and break it into data compatible with ScVI, TotalVI, ProteinVI.
- Helped in the construction of Loss function for both ProtSc-I and ProtSc-II from scratch.
- Helped in creation and debugging of class ProteinRNADataset with Tejas
- Performed and Implemented complete data analysis such as Label Transfer, studied and Implemented various metric distances(Euclidean, Minkowski, Pearson Correlation, L1) to calculate the distance between RNA and protein features and subsequent UMAP construction.
- Prepared the slides for presentation and documented the steps of various experiments done by us in the final report.

8 CITATIONS AND BIBLIOGRAPHIES

[1] Shuxiao Chen, Bokai Zhu, Sijia Huang, John W Hickey, Kevin Z Lin, Michael Snyder, William J Greenleaf, Garry P Nolan, Nancy R Zhang, and Zongming Ma. 2023. Integration of spatial and single-cell data across modalities with weakly linked features. *Nature Biotechnology* (2023), 1–11.

[2] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. 2021. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature methods* 18, 3 (2021), 272–282.

[3] Shila Ghazanfar, Carolina Guibentif, and John C Marioni. 2022. StabMap: mosaic single cell data integration using non-overlapping features. *bioRxiv* (2022), 2022–02.

[4] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. 2018. Deep generative modeling for single-cell transcriptomics. *Nature methods* 15, 12 (2018), 1053–1058.

9 FIGURES

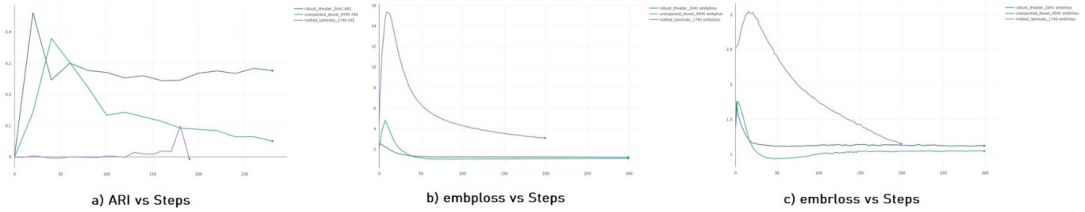


Fig. 5. Comparitives

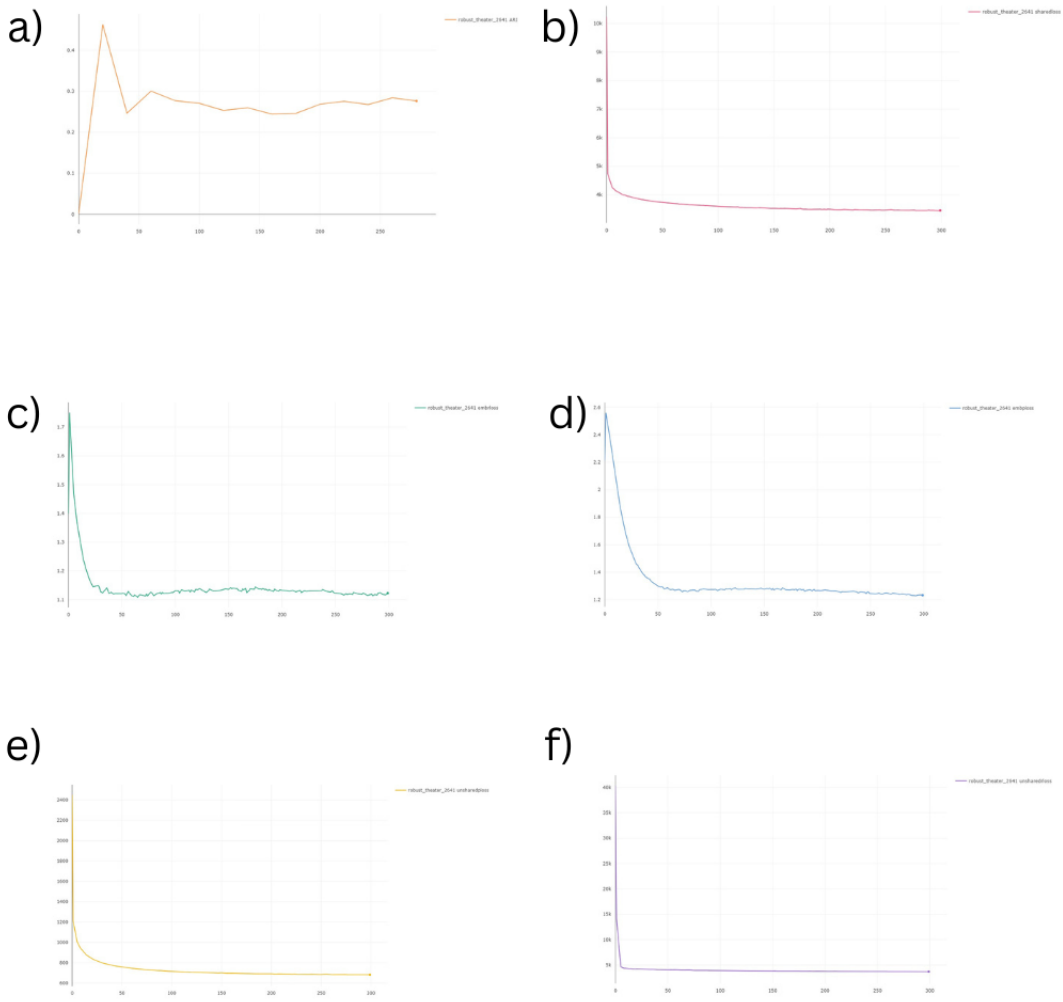
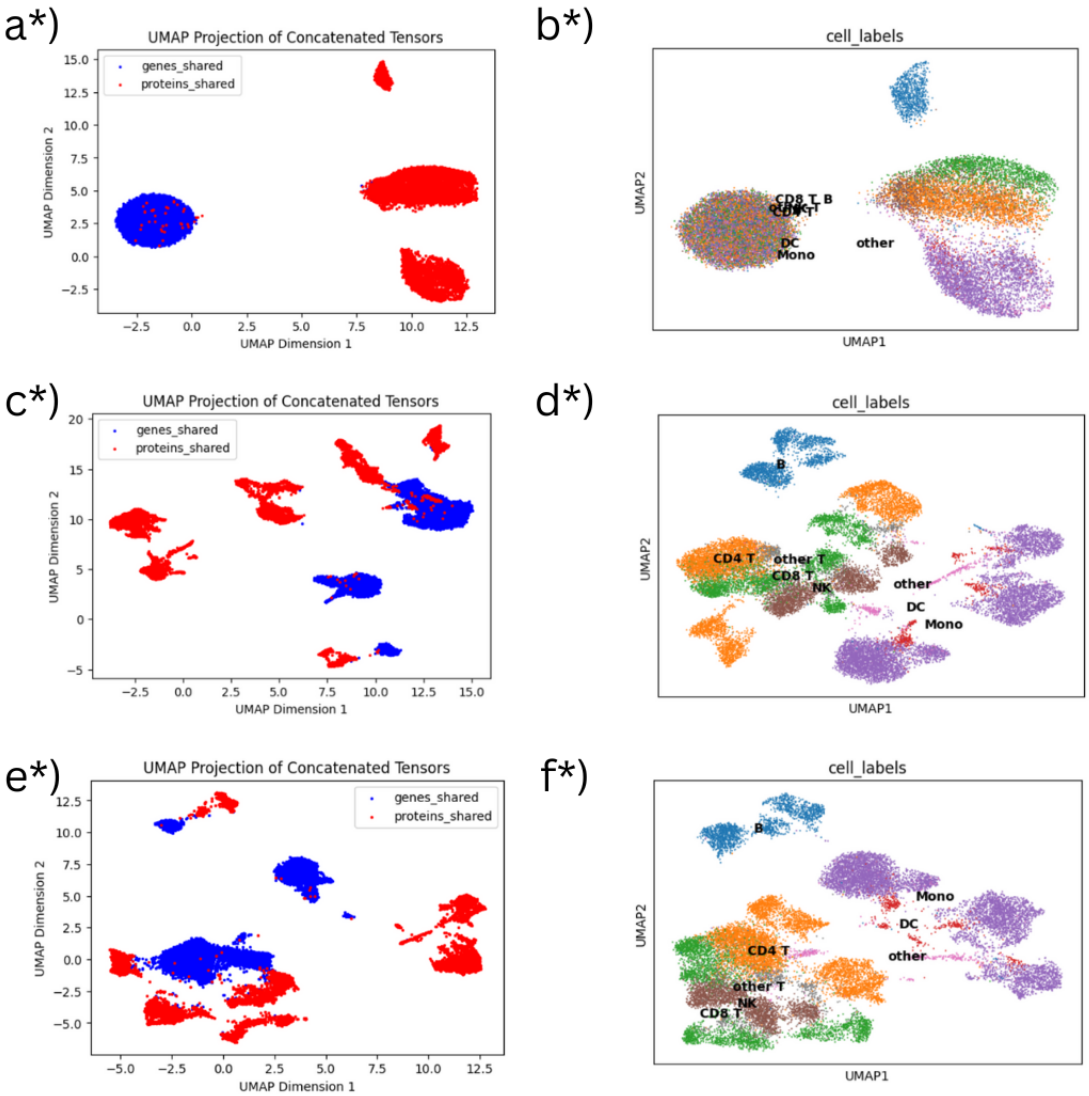


Fig. 6. Batch Change
a) ARI score b) Shared Loss c)Embedded RNA Loss d) Embedded Protein Loss f) Unshared


a*)
UMAP Projection of Concatenated Tensors
genes_shared
proteins_shared
UMAP Dimension 2
UMAP Dimension 1
b*)
cell_labels
CD8 T
B
DC
Mono
other
UMAP2
UMAP1
c*)
UMAP Projection of Concatenated Tensors
genes_shared
proteins_shared
UMAP Dimension 2
UMAP Dimension 1
d*)
cell_labels
B
CD4 T
other T
CD8 T
NK
other
DC
Mono
UMAP2
UMAP1
e*)
UMAP Projection of Concatenated Tensors
genes_shared
proteins_shared
UMAP Dimension 2
UMAP Dimension 1
f*)
cell_labels
B
Mono
DC
CD4 T
other
other T
NK
CD8 T
UMAP2
UMAP1

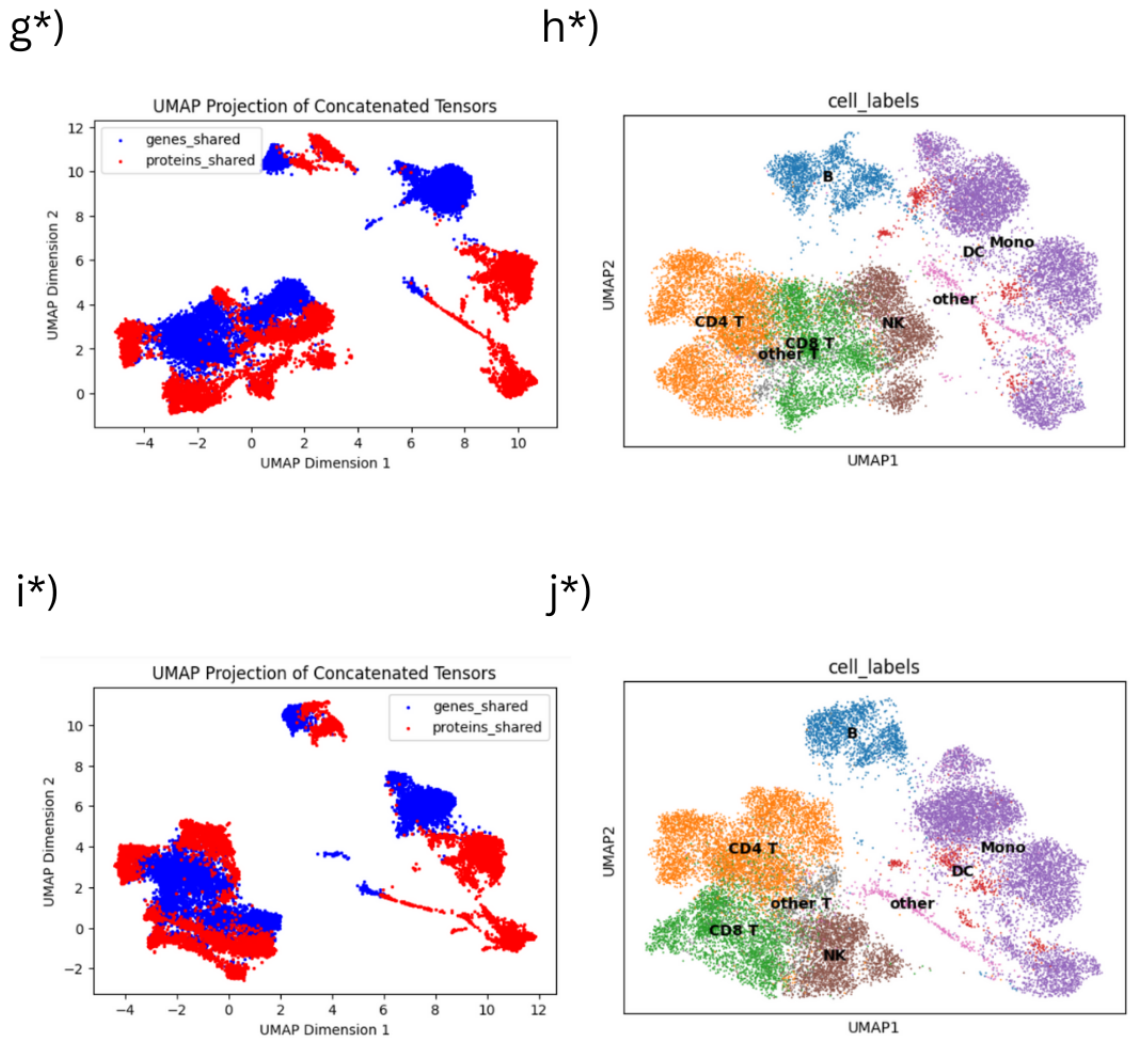


Fig. 7. Without Warmup
a*) 0 Epochs Concatenated Tensors b*) 0 Epochs Cell Labels c*)60 Epochs Concatenated Tensors d*) 60 Epochs Cell Labels e*)160 Epochs Concatenated Tensors f*) 160 Epochs Cell Labels g*) 280 Epochs Concatenated Tensors h*) 280 Epochs Cell Labels i*)340 Epochs Concatenated Tensors j*)340 Epochs Cell Labels

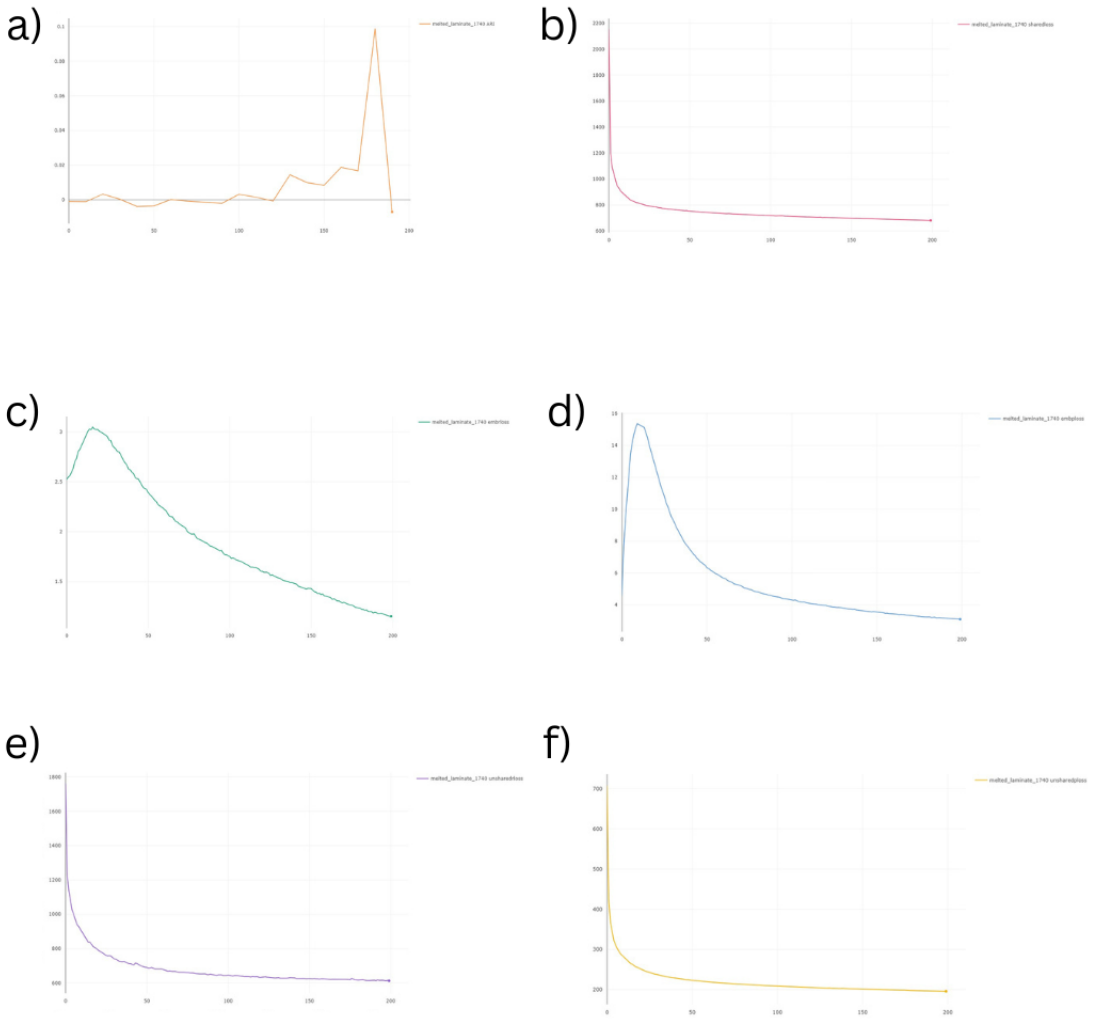


Fig. 8. SGD
a) ARI score b) Shared Loss c)Embedded RNA Loss d) Embedded Protein Loss f) Unshared

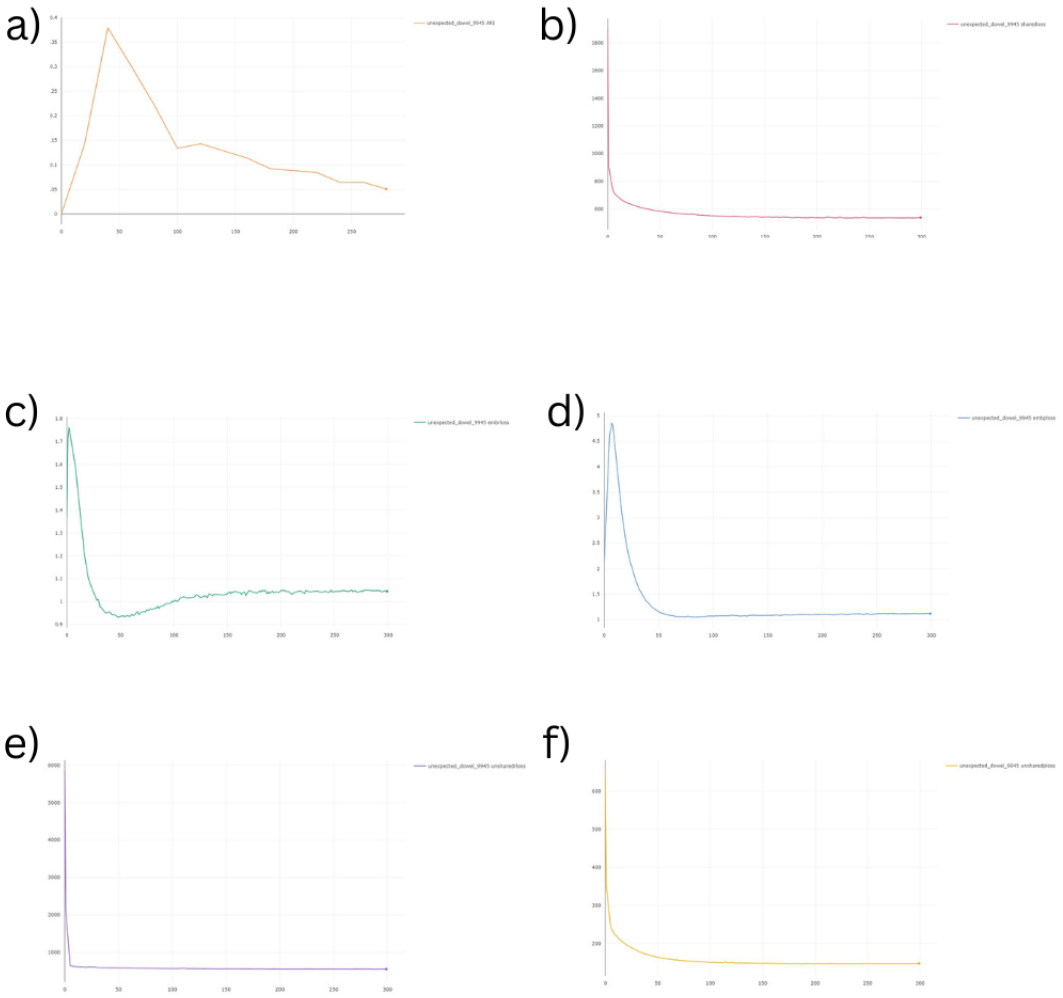


Fig. 9. With Warmup
a) ARI score b) Shared Loss c)Embedded RNA Loss d) Embedded Protein Loss f) Unshared

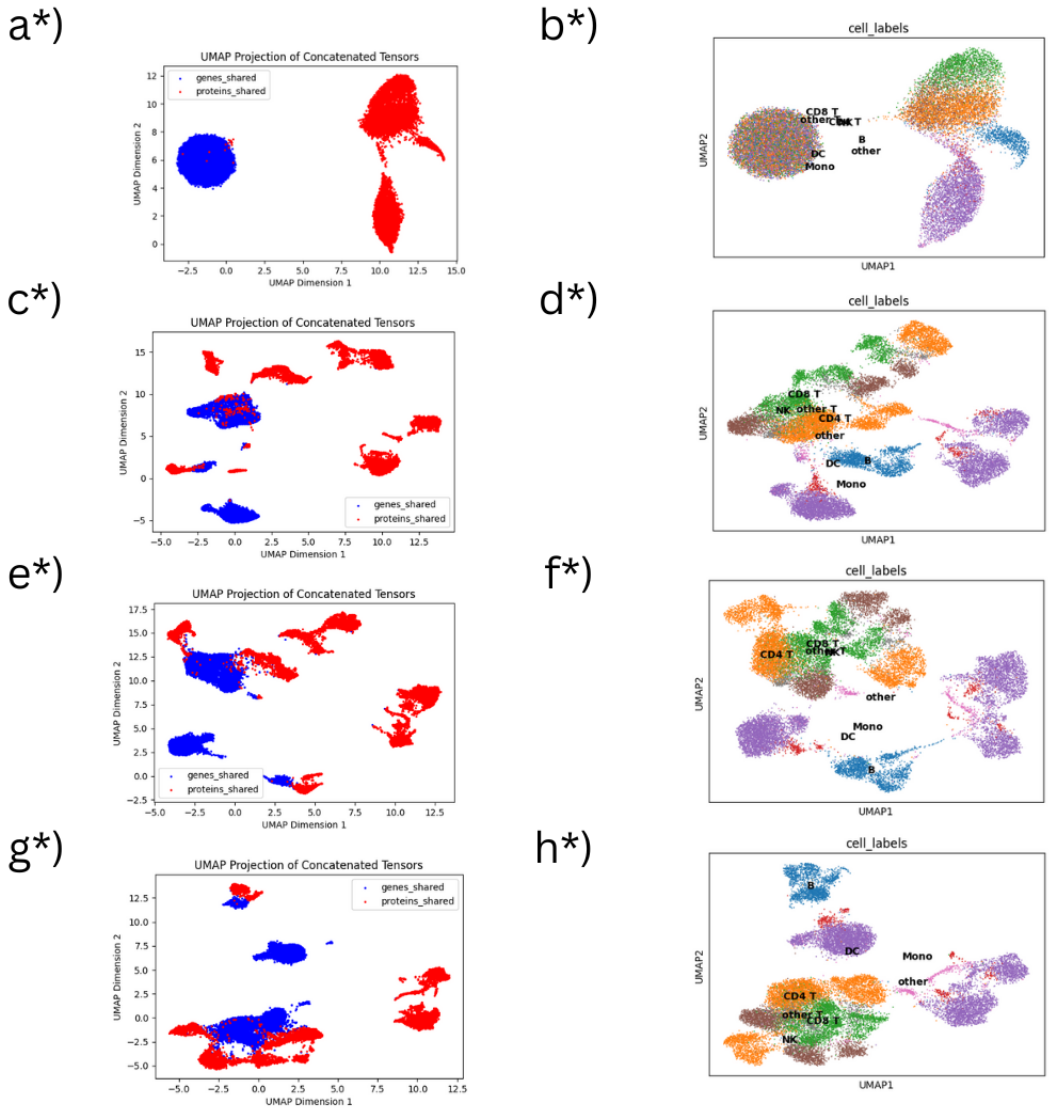


Fig. 10. Batch Change
a*) 0 Epochs Concatenated Tensors b*) 0 Epochs Cell Labels c)60 Epochs Concatenated Tensors d*) 60 Epochs Cell Labels e*)160 Epochs Concatenated Tensors f*) 160 Epochs Cell Labels g*) 260 Epochs Concatenated Tensors h*) 260 Epochs Cell Labels

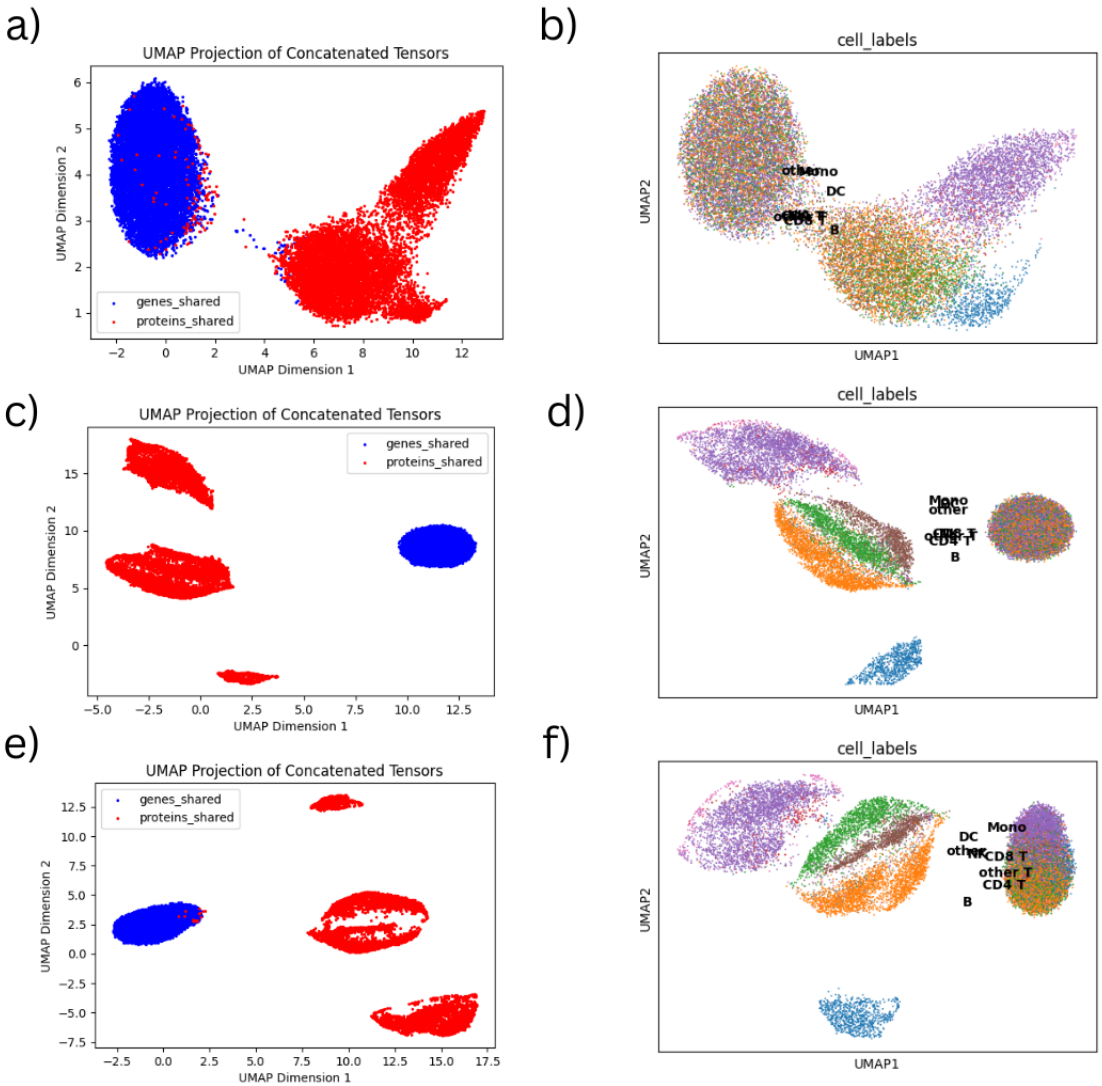


Fig. 11. SGD
a) 0 Epochs Concatenated Tensors b) 0 Epochs Cell Labels c)60 Epochs Concatenated Tensors d) 60 Epochs Cell Labels e)160 Epochs Concatenated Tensors f) 160 Epochs Cell Labels

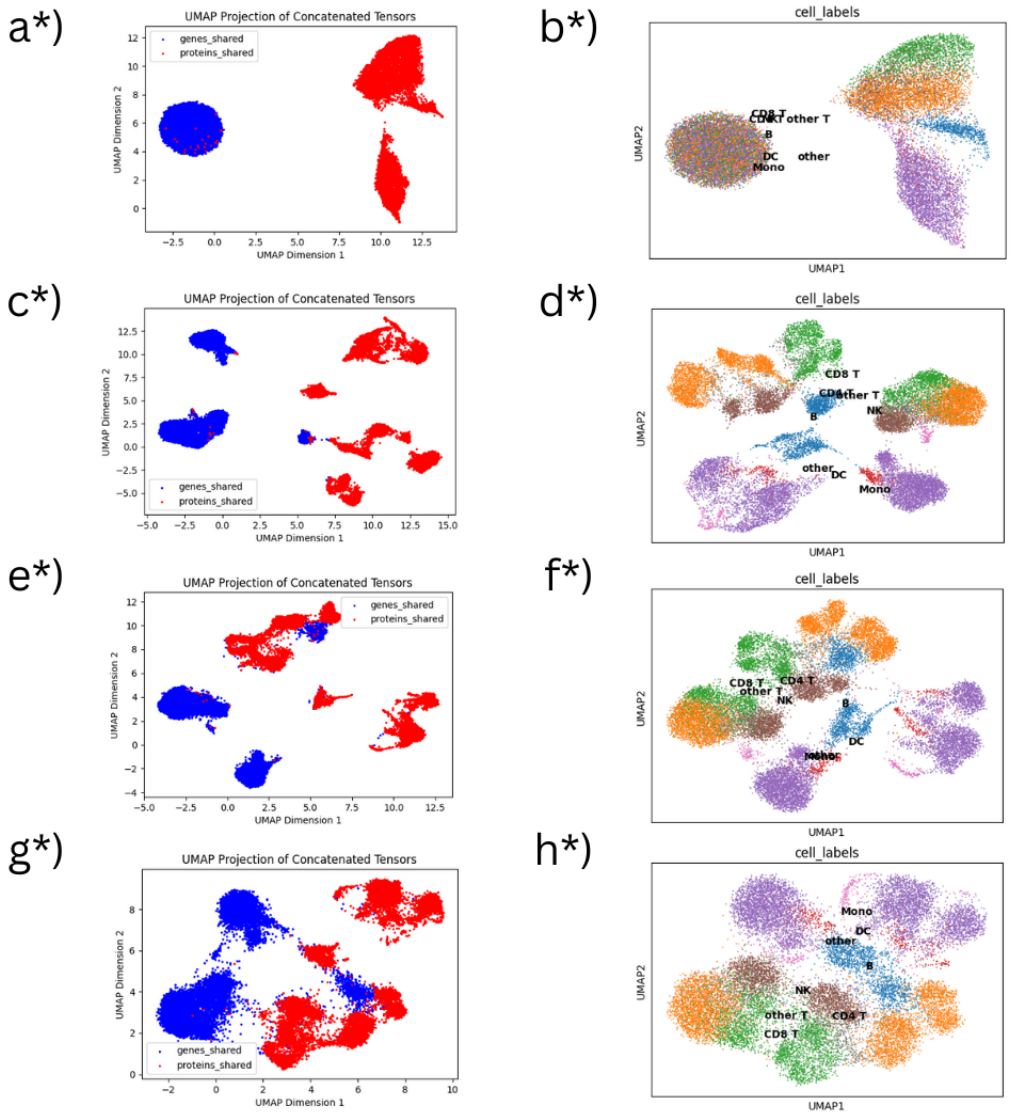


Fig. 12. With Warmup
a*) 0 Epochs Concatenated Tensors b*) 0 Epochs Cell Labels c)60 Epochs Concatenated Tensors d*) 60 Epochs Cell Labels e*)160 Epochs Concatenated Tensors f*) 160 Epochs Cell Labels g*) 260 Epochs Concatenated Tensors h*) 260 Epochs Cell Labels