Energy-based Modelling for Single-cell Data Annotation

Tianyi Liu **Philip Fradkin** **Lazar Atanackovic** **Leo J. Lee***
University of Toronto, Vector Institute
{tianyi, ljlee}@psi.toronto.edu
{phil.fradkin}@gmail.com
{l.atanackovic}@mail.utoronto.ca

Abstract

Single-cell sequencing has provided profound insights into understanding heterogeneous cellular activities by measuring sequence information at the individual cell resolution. Accurately annotating a single-cell RNA sequencing (scRNA-seq) dataset is a crucial step for the single-cell data analysis pipeline. In particular, previously unobserved cell types and cellular states frequently appear in scRNA-seq experiments and carry valuable information [1]. This highlights the need for reliable annotation tools with out-of-distribution (OOD) detection capability. Recent advances in energy-based modelling have made it possible to design and deploy EBMs for joint discriminative and generative tasks [2]. In this work, we introduced energy-based models (EBMs) for scRNA-seq annotation and investigated generative modelling for OOD detection, which result in more accurate, calibrated, and robust cell-type predictions. Specifically, we developed CLAMS, an EBM instance improved upon the previous joint energy-based model (JEM) [3], for single-cell data hybrid modelling. Our experiments reveal that hybrid modelling with EBMs maintains the strong discriminative power of baseline classifiers and outperforms the state-of-the-art by integrating generative capabilities in data annotation and OOD detection tasks. To the best of our knowledge, we are the first to apply EBMs for single-cell data modelling. The source code of this work is available at `https://github.com/tianyi21/clams`.

1 Introduction

The advent of single-cell high-throughput sequencing has enabled researchers to examine cellular activities and functionalities at unprecedented resolution. Single-cell genomics makes it possible to study a group of cells within its micro-environment and resolve its heterogeneity. Many downstream analyses, such as differential expression and trajectory inference, require a reliable annotation of cell types that appear in the dataset. Nevertheless, annotating single-cell data remains a challenging and laborious task, which usually requires both expert knowledge and well-established marker genes. A standard workflow consists of identifying highly expressed marker genes, which are typically associated with a specific cell type, and labelling the corresponding cells or cell clusters [4].

Another challenge in the single-cell annotation is that the current set of known cell types is not complete. An effective classification approach should be able to warn the user if a group of observed cells is significantly different from the set of previously observed cells. For example, if a researcher is studying a cell population, they should be warned if they are classifying a previously unseen cell type. A standard model with a softmax output would assign a class predictive probability when labelling previously unseen cells even though it could be outside the learned distribution. Thus,

*To whom correspondence should be addressed.

17th Conference on Machine Learning in Computational Biology (MLCB 2022).

an effective tool should have OOD detection capabilities, allowing it to alert the researcher when analyzing a fresh dataset.

Class predictive distribution $p(y \mid x)$ has become the mainstream method for single-cell OOD detection [5, 6]. Our work, instead, directly investigated generative data modelling with $p(x)$ for OOD detection. Taking advantage of the recent advances, EBMs can be easily integrated and reformulated to accomplish hybrid data modelling tasks, which simultaneously learn a discriminative classifier and a generative model. This brings the opportunity to incorporate generative OOD detection capability into discriminative cell-type classifiers. In this work, we presented a set of EBM-based data annotation tools and evaluated their performance through a diverse set of experiments on single-cell pancreas and peripheral blood mononuclear cell (PBMC) datasets. We found that EBMs produce well-calibrated cell type annotations and can simultaneously detect biological OOD samples. Our contributions in this work are three-fold:

- We introduced EBMs, a family of probabilistic models, to single-cell data modelling.
- We developed CLAMS, a customized EBM, and accomplished robust and well-calibrated results for scRNA-seq annotation.
- We explored generative data modelling with EBMs and demonstrated that our method outperforms state-of-the-art methods for OOD detection.

2 Background and Related Works

Our work resides in the cross-discipline of single-cell genomics and machine learning. Therefore, we discuss related prior works in both fields.

2.1 A Open Set Domain Adaption Problem

Single-cell data annotation can be framed as an open set domain adaption (OSDA) problem, where training data does not encode all information, and new classes may arrive in the deployment stage [7]. For example, a handwritten digit classifier was trained to recognize digits, but letters and punctuation marks may also be detected in the deployment. An OSDA problem has two distinct stages: a) align reference and query distributions, and b) learn a classifier [7, 8]. In the context of single-cell genomics, new cell types and cellular states are frequently detected in fresh single-cell datasets, which are often of interest to researchers given that they sometimes carry information about previously unknown cellular functions [1, 9]. Consequently, the OOD detection capability of a classifier is essential to flag certain groups of cells for further investigation.

2.2 scRNA-seq Annotation

State-of-the-art tools often address the problem in two stages: data alignment and cell-type classification. Various methods have been proposed to align multiple single-cell transcriptomes. ComBat uses a Bayesian network to adjust and integrate data [10]. It was originally developed for microarray data but shows the capability to generalize to single-cell data as well [11]. Mutual nearest neighbor (MNN) extracts MNN pairs in both reference and query datasets and integrates them with linear correction vectors [12]. Seurat also uses a MNN-based method to extract anchors for integration but filters MNNs with a shared neighbor graph [6]. Harmony iteratively learns correction vectors in a low-dimensional space with fuzzy clustering assignment [13]. It is reported that Harmony can be chosen as an initial attempt to integrate datasets due to its simplicity and efficiency [11].

Once datasets are aligned and integrated, they can be annotated with a classifier. Cell type annotation is a well-studied problem in single-cell genomics. There are various methods based on discriminative models that have shown promise for this task. Ingest uses a k nearest neighbor (kNN) classifier to assign a known label based on the kNN graph. scPred uses a calibrated support vector machine (SVM) classifier with a rejection option in a new feature space [5]. Seurat creates a binary classification matrix and uses the similarity matrix of query cells and anchors [6]. With the advancement of deep neural network (DNN) training, DNN-based methods have shown superior performance in many classical tasks such as image processing [14], data generation [15], natural language processing [16]. Such achievements have been gradually transferred to single-cell genomics data analysis.

scANVI [17] builds a variational auto-encoder for single-cell probabilistic modelling and data annotation on top of scVI data integration [18]. scDeepSort is a graph neural network based single-cell annotation tool with a pre-trained model which does not require additional references [19]. DNN-based models tend to integrate multiple functionalities and features into a single model for end-to-end optimizations, and survey says the scVI-scANVI workflow has particularly better performance on complicated tasks [20].

In this work, we designed a tool for scRNA-seq data annotation using an EBM. We chose Harmony for data integration and a neural network with residual connection [21] as the classifier for data hybrid modelling. With an EBM, we accomplished simultaneous discriminative and generative modelling for single-cell data, which yields an annotation tool and a data generator.

2.3 Energy-Based Models and Hybrid Data Modelling

EBMs are a very flexible class of probabilistic models inspired by statistical mechanics and physics. For a data point $\mathbf{x}$, its energy value is determined through a function $E(\mathbf{x})$ parameterized by θ. With the use of the Boltzmann distribution, EBMs can be interpreted as parameterizing a probabilistic distribution based on its energy value as:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}; \tag{1}$$

where $Z(\theta)$ is the normalizing constant that solely depends on θ, and $-E_\theta(\cdot)$ is defined as the negative energy function such that $E_\theta : \mathbb{R}^d \to \mathbb{R}$.

Since the energy could be parameterized by any functions, $E_\theta(\mathbf{x})$ can also be a softmax classifier, and thus EBMs can simultaneously produce a predictive model for the data and a generative model to capture the data distribution. One of the notable works in data hybrid modelling is JEM [3]. JEM defines a joint distribution of a data point $\mathbf{x}$ and its label y as:

$$p_\theta(\mathbf{x}, y) \triangleq \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}; \tag{2}$$

where $f_\theta(\mathbf{x})[y]$ is the y^{th} logit of $\mathbf{x}$'s class predictive distribution. By marginalizing out y, $p_\theta(\mathbf{x})$ can be re-expressed in the standard EBM form:

$$p_\theta(\mathbf{x}) = \frac{\exp\left[\log \sum_y \exp(f_\theta(\mathbf{x})[y])\right]}{Z(\theta)}. \tag{3}$$

Therefore, JEM uncovers a hybrid model by optimizing the joint distribution $p_\theta(\mathbf{x}, y)$. To jointly train the classifier and generative components, we can formulate the maximum likelihood objective as the sum of discriminative and generative losses:

$$\ell_{ML} = \log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y \mid \mathbf{x}). \tag{4}$$

JEM adopts the standard EBM training strategy by retaining a replay buffer with stochastic gradient Langevin dynamics (SGLD) sampling, a gradient-guided Monte-Carlo Markov chain method [2, 3, 22].

Although effective for hybrid modelling, JEMs can be difficult to train. For instance, training instability is one major challenge that can make achieving high-quality generative and discriminative performance difficult [3, 23, 24, 25]. The model may diverge prior to model convergence, which causes unsatisfactory performance in deployment. We observed that this issue is especially common when working with single-cell datasets compared with images, the original application domain of JEM, and it motivated us to develop CLAMS, an improved version of JEM. The authors of JEM used heuristic approaches to address the instability issue through check-pointing and re-initializing prior to divergence [3]. Their succeeding work VERA resolved this issue by replacing SGLD sampling with variational inference, which consequently accelerates the training procedure and enables exact sample generation [23]. HDGE, another extension of JEM, re-interpreted the generative loss as a contrasting learning problem and jointly optimized it with the discriminative loss [24]. Both VERA and HDGE further improve JEM by replacing SGLD with other sampling methods. Nevertheless, our experiments demonstrate that VERA is poorly calibrated despite its accelerated training speed, and HDGE has to be integrated with JEM, as HDGE + JEM, to generate class-conditional samples, which circles back to the JEM instability problem. As a result, we focused on improving the original JEM formulation by identifying and resolving the instability issue of the SGLD sampler.

3 Methods

We implemented EBMs including JEM, VERA, HDGE, and HDGE + JEM with author-suggested hyper-parameters, and they also served as benchmark models in this paper. In this section, we mainly describe our model CLAMS as a carefully engineered EBM for scRNA-seq annotation and OOD detection to improve upon JEM. Before doing so, we first provide a brief analysis of JEM instability.

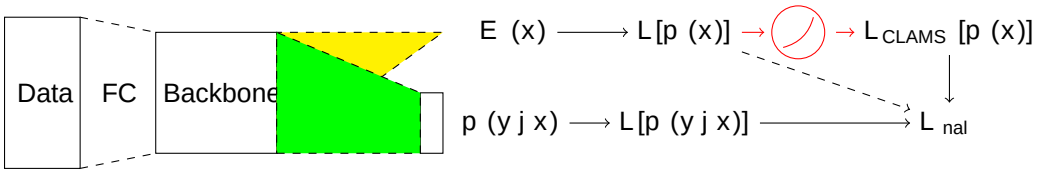


Figure 1: Schematic diagram of CLAMS training: elements in black are the original JEM formulation, and elements in red are our interventions. FC stands for fully-connected layer. We used a 1D residual network [21] as the backbone network architecture. The model has a two-headed output: when calculating the generative loss, the model parameters are frozen, and SGLD follows the gradient of energy function $E(x)$ to iteratively update the generated samples in the replay buffer; when calculating the discriminative loss, the model behaves similarly to other neural network classifiers.

JEM jointly optimizes the cross-entropy discriminative loss $L[p(y \mid x)]$ and the generative loss $L[p(x)]$ as shown in Figure 1. With state-of-the-art DNN training strategies, including dropout, weight decay, learning rate schedule, and early stopping, the training of a DNN discriminative classifier should be fairly stable. Therefore, we suspected that the generative loss was the source of training instability. In JEM, the generative loss is defined as:

$$L[p(x)] = E_{x \sim p_{fake}}[E(x)] - E_{x \sim p_{real}}[E(x)]; \tag{5}$$

where $E(\cdot)$ is the energy value assigned to a data point.

Through a series of diagnostic experiments, we confirmed that this was indeed the case and identified that model divergence occurred when the expected energy value becomes negative, which results in gradient growth. This establishes a positive feedback loop, which eventually causes gradient explosion, resulting in poorly performed models. To prevent the initiation of the feedback loop, we employed two regularization strategies. First, we clipped the generative loss using a clipping function, capping the magnitude of the negative values the energy function can take on. Second, we introduced stochastic regularization units for the generative loss, referencing the dropout unit [26]. Our final model CLAMS is a JEM model with ELU clipping and stochastic regularization of $p_{ON} = 0.25$. Details about our analysis and interventions are provided in Appendix A.1.

4 Experiments

We started with a scRNA-seq annotation experiment under a reference-query setup with different EBMs. In this experiment, we evaluated our models by annotation accuracy, calibration performance, and OOD detection performance. We further developed a designated one-class OOD detection experiment by leaving out classes in the dataset to inspect our model. Finally, we conducted an ablation study by model decomposition to assess the effectiveness of our regularization methods.

4.1 Datasets

We ran scRNA-seq annotation experiment with the following reference-query dataset pairs.

Tissue/Sample	Reference	Query	Source
pancreas	Baron [27], Segerstolpe [28], Wang [29]	Muraro [30]	[31]
PBMC	Zheng [32] (10x NextSeq)	10x NovaSeq	[33, 34]

Table 1: Datasets used in this work

We downloaded the processed pancreas dataset from [2] and processed the PBMC dataset with standard ScanPy [35] work ow, and selected top highly variable genes as an initial feature selection. To start, we ran Harmony to integrate reference and query datasets prior to the model training. The reference dataset was split into 8:1:1 for training, validation, and testing. Under our experimental setup, we had access to the labels of query datasets and split the query datasets into two subgroups: in-distribution data, where cell types appear in the reference dataset, and OOD data, where cell types do not appear in the reference dataset, i.e., new cell types.

4.2 Evaluation Criteria

We benchmarked CLAM against three groups of models: EBMs (JEM [3], VERA [23], HDGE [24], HDGE + JEM [24]); classi ers (ResNet [21], SVM$_{reject}$ [36], scANVI [17] (where applicable)); and biological tools (Ingest, scPred [5], Seurat [37]).

The core evaluation metrics are the annotation accuracy for in-distribution data and the score's area under the receiver operating characteristic (AUROC) for OOD data. Annotation accuracy re ects how well the model annotates known cell types, and score AUROC measures how well a score differentiates OOD data from in-distribution data. We also analyzed the calibration performance of a series of models, which re ects how well the model's prediction con dence corresponds to its prediction accuracy. It has been shown that, although modern DNN-based methods have accomplished exceptional performance, they tend to be poorly calibrated [3, 38].

ECE score is a popular metric to quantify the calibration performance of a classi er. It calculates the expected difference between prediction con dence and the corresponding prediction accuracy as follows:

$$\mathrm{ECE} = \frac{1}{n}\sum_{i=1}^{m} |B_i|\,|\mathrm{avg\_acc}(B_i) - \mathrm{avg\_conf}(B_i)|; \qquad (6)$$

where n data points are split into m equal-sized bins B_i based on their prediction con dence. Figure 6 shows a histogram for ECE calculation. We expect a well-calibrated classi er to have a lower ECE score such that its prediction con dence is close to its prediction accuracy (along the red dashed diagonal line).

Figure 2: ECE histogram

4.3 scRNA-seq Annotation

In this section, we provide an overview of CLAMS annotation capabilities, while we expand on ablation results in Appendix A.2. We trained all models for a xed number of epochs and reported results on the query dataset for the model with the best validation accuracy achieved during training. To estimate the uncertainty of our results due to stochasticity, we averaged performance over ve experiments with varying random seeds. We summarized our results in Table 2.

We assessed a model trained on a reference dataset by evaluating its ability to generalize to a query dataset with a partially overlapping set of cell types. The heatmap in Figure 3 demonstrates the classi cation accuracy of a model trained on cell types indicated as rows and evaluated on cell types as columns from the query dataset. This allows us to assess model accuracy, as well as its OOD detection capabilities. We observed that CLAMS almost perfectly identi ed B cells, T cells, dendritic cells, and NK cells. For rarer cell types such as HSPC that are not present in the reference dataset, the model labelled them as OOD data and rejected them. This is the desired behavior since it allows a hypothetical user to further investigate the nature of these cells. We further assessed the model's ability to identify these OOD cells by calculating AUROC values of $p(x)$ and $\max_y p(y \mid x)$. We noticed that EBMs are consistently able to effectively identify OOD cell types using the generative score as indicated by the relatively high AUROC values in Table 2.

Furthermore, we observed a clear improvement of CLAMS over JEM in terms of almost all evaluation metrics. This evidences the effectiveness of our regularization methods and con rms our hypotheses regarding JEM training instability. Our model has slightly better performance over

[2] https://www.dropbox.com/s/qj1jlm9w10wmt0u/pancreas.h5ad?dl=1

Figure 3: An annotation heatmap of PBMC dataset: rows are labels of the reference dataset and columns are labels of the query dataset. The heatmap is normalized column-wise.

HDGE, which reformulates the hybrid data modelling as a contrastive learning problem [24]. Lastly, we found that the un-normalized density $p(x)$ outperforms maximum class predictive probability $\max p(y \mid x)$, which is the classical score for OOD detection in the single-cell annotation. $p(x)$ also produces more reliable and more stable OOD detection, as a result of higher mean AUROC and lower standard deviation. Visually, Figure 4 shows that our model maps $p(x)$ scores into multi-modal distributions with clear separations, which consequently provides convenience in real applications where cell type labels are not available.

PBMC		n	Div.	In Acc		In ECE		$p(x)$		$\max p(y \mid x)$	
EBM	CLAMS (Ours)	5	0	92.89	0.72	5.39	0.46	0.92	0.01	0.77	0.02
	JEM	5	100	90.32	4.93	2.61	0.66	0.61	0.28	0.86	0.00
	VERA	5	0	88.76	2.04	6.56	2.06	0.87	0.05	0.80	0.03
	HDGE	5	0	93.58	0.30	3.45	0.11	0.81	0.01	0.78	0.01
	HDGE + JEM	5	100	92.29	0.87	6.25	2.34	0.42	0.23	0.70	0.09
Classifier	ResNet	5	0	91.53	1.41	4.73	0.25	–		0.79	0.02
	$\text{SVM}_{\text{reject}}$	1	–	91.55		22.82		–		0.72	
	scANVI	1	–	97.81		1.93		–		0.83	
Bio Tool	Ingest	1	–	81.55		–		–		–	
	scPred	1	–	93.01		5.42		–		0.92	
	Seurat	1	–	93.09		11.44		–		0.81	

Pancreas		n	Div.	In Acc		In ECE		$p(x)$		$\max p(y \mid x)$	
EBM	CLAMS (Ours)	5	0	88.38	0.62	5.81	0.75	0.89	0.03	0.71	0.07
	JEM	5	100	86.61	2.07	7.57	0.86	0.92	0.06	0.54	0.13
	VERA	5	0	89.14	0.86	9.59	1.12	0.88	0.13	0.57	0.19
	HDGE	5	0	89.05	0.21	6.34	0.45	0.79	0.02	0.78	0.02
	HDGE + JEM	5	100	87.02	1.63	6.42	0.73	0.69	0.13	0.57	0.17
Classifier	ResNet	5	0	88.13	0.53	8.69	0.42	–		0.67	0.04
	$\text{SVM}_{\text{reject}}$	1	–	91.53		13.28		–		0.59	
Bio Tool	Ingest	1	–	89.37		–		–		–	
	scPred	1	–	90.45		8.82		–		0.09	
	Seurat	1	–	89.81		3.31		–		0.87	

Table 2: Annotation results: Div. is the chance of divergence (%); In Acc and In ECE are annotation accuracy and ECE score for in-distribution data; $p(x)$ and $\max p(y \mid x)$ are AUROCs of the corresponding score for OOD detection.

Meanwhile, we found that scANVI has exceptionally good performance on the cell-type annotation experiment and tends to be calibrated as well. The pancreas dataset we downloaded was pre-processed by the authors, which doesn't in compliance with scANVI data format (scran normalized and log-transformed counts). For this reason, we were only able to run scANVI appropriately on the PBMC dataset. We further examined the results and presumed that the integration algorithm we chose before could be the bottleneck of performance for our models.

Figure 4: OOD scores assigned by CLAMS: un-normalized density $p(x)$ (left), maximum class predictive probability $\max p(y \mid x)$ (right). In-distribution data are colored in green; OOD data are colored in red.

4.4 One-class OOD Detection

We further investigated the OOD detection capability of CLAMS by designing this follow-up OOD detection experiment: for a labelled dataset $\mathcal{D}$ with n classes, we first generated sub-datasets by leaving out each class. This yielded n sub-dataset pairs $\mathcal{D}_i^{n-1}$ and $\mathcal{D}_i^{n-1}, i \in [1, n]$. The leave-out class $\mathcal{D}_i^{n-1}$ contains samples from an unseen class in the training procedure and will serve as OOD samples. The models were trained on $\mathcal{D}_i^{n-1}$ and evaluated on the leave-out class $\mathcal{D}_i^{n-1}$. We ran five repetitions by setting matched seeds.

PBMC	n	Div.	$p(x)$		$\max p(y \mid x)$	
CLAMS (Ours)	40	0	0.78	0.13	0.68	0.12
JEM	40	100	0.64	0.18	0.58	0.16
VERA	40	0	0.65	0.22	0.63	0.14
HDGE	40	0	0.66	0.15	0.68	0.10
HDGE + JEM	40	100	0.70	0.17	0.69	0.11
ResNet	40	0	–		0.63	0.13
$\text{SVM}_{\text{reject}}$	8	–	–		0.75	0.12

Pancreas	n	Div.	$p(x)$		$\max p(y \mid x)$	
CLAMS (Ours)	75	0	0.90	0.12	0.88	0.10
JEM	75	100	0.80	0.14	0.79	0.13
VERA	75	0	0.82	0.18	0.82	0.11
HDGE	75	0	0.88	0.10	0.89	0.10
HDGE + JEM	75	100	0.83	0.11	0.84	0.11
ResNet	75	0	–		0.87	0.05
$\text{SVM}_{\text{reject}}$	14	–	–		0.90	0.05

Table 3: One-class OOD detection results: we did not include results for biological tools since they align the reference and query dataset by default, in which there is no overlapped cell type between them. Forcing such alignment will consequently cause over fitting and poor performance.

To reduce the effect of AUROC dilution, we selected a subset of cell types by removing some highly similar ones and summarized the results in Table 3. Our one-class OOD experiment shows that $p(x)$ possesses better OOD detection power compared to classical $\max p(y \mid x)$ score. Our model also shows a clear improvement over JEM and has overall the best results. The annotation OOD experiment in Section 4.3 focuses on an end-user case where the dataset to be annotated has a mix of in-distribution and OOD data. Whereas, this one-class OOD experiment is a follow-up experiment under an artificial setting to analyze the model's capability to detect biological OOD data.

4.5 Data Generation

In our work, we mainly focused on using generative data modelling for the task of OOD detection rather than data generation. Nevertheless, we found that our model could produce meaningful single-cell data from the training distribution. In Figure 5, we included a t-SNE plot for the generated data from the PBMC dataset. Through visualizing samples from the replay buffer, we can visually inspect the learned class distribution over the sample space. As opposed to class-balanced datasets in the vision domain, single-cell datasets can have dramatic differences in sample sizes between classes, and we observed that classes with fewer cells were not learned as well as those with many samples.

Moreover, we proposed a modified SGLD sampler with an imbalanced replay buffer to address this class-imbalance issue (see Appendix A.3). We did not adopt the modified version here but will further discuss this in our future work.

Figure 5: Generated data in t-SNE space: we generated 300 samples per class and the reference dataset was 0.25 sub-sampled for better visualization. On the left, data points are colored based on the source of data; on the right, data points are colored by cell types.

5 Discussion

Overall our results support that EBM-based methods have consistently good performance in annotation and OOD detection tasks with both PBMC and pancreas datasets, in addition to being robust, accurate, and well-calibrated. Commonly used state-of-the-art biological tools typically achieve competitive classi cation results however are often poorly calibrated and have unreliable OOD detection capabilities in the annotation experiments, as shown in Table 2. In this work, we narrowed our attention to detecting biological OOD samples rather than technical OOD samples (due to the batch effect). When a model assigns consistently low scores to samples in the query set, one should consider the technical OOD effect and re-examine the batch effect correction procedure as described in Section 4.1.

Although we observed excellent performance of scANVI in Section 4.3, we think such comparisons could be unfair since it was developed under considerably different assumptions. scANVI extended scVI by encapsulating auxiliary tasks, such as dimensional reduction and batch effect correction, into a single variational auto-encoder to jointly optimize multiple tasks simultaneously, in contrast to our independent optimizations. As a result, the data fed into scANVI were log-transformed counts, without being corrected by Harmony [13] beforehand. The alignment performed by Harmony is likely not as good as that of the scVI-scANVI work ow [18, 17], and it could very well explain the performance gap in Table 2. Nevertheless, we still observed a clear advantage of using generative modelling for single-cell OOD detection.

6 Conclusions and Future Work

In this work, we utilized EBMs and data hybrid modelling techniques to address scRNA-seq annotation and OOD detection problems, which are two essential steps in the single-cell data analysis pipeline. We explored the generative power of EBMs for OOD detections while maintaining strong annotation accuracy and calibration performance. We also developed CLAMS, a speci c instance of EBM, which yields accurate and robust single-cell annotation performance. Moreover, we pinpointed the source of JEM training instability being the problematic loss formulation and demonstrated that our regularization methods are simple yet effective in improving training and deployment performance. We hope that our work demonstrates the advantages of EBMs for single-cell data annotation under the presence of reference datasets. In our future works, we will consider employing the class-imbalanced buffer as well as introducing a more integrated work ow similar to scVI-scANVI. As an initial attempt, we demonstrated the possibility of using a vanilla EBM for single-cell technical batch effect correction in Appendix A.4. Furthermore, CLAMS also has the potential to be deployed for other relevant computational biology problems, such as single-nuclei RNA sequencing (snRNA-seq) and single-cell assay for transposase-accessible chromatin sequencing (scATAC-seq) data annotation given the underlying similarity between these data formats.

Acknowledgement

We acknowledge the nancial and computing support from the Vector Institute. We would like to thank Dr. Peter W. S. Hill (Imperial College London) for providing helpful interpretation and feedback for our results. T.L. was supported by University of Toronto Data Science Institute (UofT DSI) Doctoral Student Fellowships; P.F. was supported by Ontario Graduate Scholarship; L.A. was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) PGS-D scholarships.

References

[1] H. Pan, C. Xue, B. J. Auerbach, J. Fan, A. C. Bashore, J. Cui, D. Y. Yang, S. B. Trignano, W. Liu, J. Shi, et al., "Single-cell genomics reveals a novel cell state during smooth muscle cell phenotypic switching and potential therapeutic targets for atherosclerosis in mouse and human," Circulation, vol. 142, no. 21, pp. 2060–2075, 2020.

[2] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," Advances in Neural Information Processing Systems, vol. 32, 2019.

[3] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classi er is secretly an energy based model and you should treat it like one," pp. 1–22, 12 2019.

[4] Z. A. Clarke, T. S. Andrews, J. Atif, D. Pouyabahar, B. T. Innes, S. A. MacParland, and G. D. Bader, "Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods," Nature Protocols, vol. 16, pp. 2749–2764, June 2021. Number: 6 Publisher: Nature Publishing Group.

[5] J. Alquicira-Hernandez, A. Sathe, H. P. Ji, Q. Nguyen, and J. E. Powell, "scpred: accurate supervised method for cell-type classi cation from single-cell rna-seq data," Genome biology, vol. 20, no. 1, pp. 1–17, 2019.

[6] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," Cell, vol. 177, no. 7, pp. 1888–1902, 2019.

[7] P. Panareda Busto and J. Gall, "Open set domain adaptation," in Proceedings of the IEEE international conference on computer vision, pp. 754–763, 2017.

[8] S. Garg, S. Balakrishnan, and Z. C. Lipton, "Domain adaptation under open set label shift," arXiv preprint arXiv:2207.13048, 2022.

[9] F. A. Vieira Braga, G. Kar, M. Berg, O. A. Carpaij, K. Polanski, L. M. Simon, S. Brouwer, T. Gomes, L. Hesse, J. Jiang, et al., "A cellular census of human lungs identi es novel cell states in health and in asthma," Nature medicine, vol. 25, no. 7, pp. 1153–1163, 2019.

[10] W. E. Johnson, C. Li, and A. Rabinovic, "Adjusting batch effects in microarray expression data using empirical bayes methods," Biostatistics, vol. 8, no. 1, pp. 118–127, 2007.

[11] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, "A benchmark of batch-effect correction methods for single-cell rna sequencing data," Genome biology, vol. 21, no. 1, pp. 1–32, 2020.

[12] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, "Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors," Nature biotechnology, vol. 36, no. 5, pp. 421–427, 2018.

[13] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P.-r. Loh, and S. Raychaudhuri, "Fast, sensitive and accurate integration of single-cell data with harmony," Nature methods, vol. 16, no. 12, pp. 1289–1296, 2019.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention, pp. 234–241, Springer, 2015.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, vol. 63, no. 11, pp. 139–144, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[17] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef, "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models," Molecular systems biology, vol. 17, no. 1, p. e9620, 2021.

[18] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," Nature methods, vol. 15, no. 12, pp. 1053–1058, 2018.

[19] X. Shao, H. Yang, X. Zhuang, J. Liao, P. Yang, J. Cheng, X. Lu, H. Chen, and X. Fan, "scdeepsort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network," Nucleic acids research, vol. 49, no. 21, pp. e122–e122, 2021.

[20] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, et al., "Benchmarking atlas-level data integration in single-cell genomics," Nature methods, vol. 19, no. 1, pp. 41–50, 2022.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

[22] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in Proceedings of the 28th international conference on machine learning (ICML-11), pp. 681–688, Citeseer, 2011.

[23] W. Grathwohl, D. Duvenaud, and J. Kelly, "No mcmc for me: A mortized sampling for fast and stable training of energy - based models," pp. 1–35, 2021.

[24] H. Liu and P. Abbeel, "Hybrid discriminative-generative training via contrastive learning," arXiv preprint arXiv:2007.09070, 2020.

[25] X. Yang, H. Ye, Y. Ye, X. Li, and S. Ji, "Generative max-mahalanobis classi ers for image classi cation, generation and more," Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 67–83, Springer, 2021.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from over tting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai, "A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure," Cell Systems, vol. 3, pp. 346–360.e4, 2016.

[28] Asa Segerstolpe, A. Palasantza, P. Eliasson, E. M. Andersson, A. Céline Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. M. Smith, M. Kasper, Ämmälä, and R. Sandberg, "Single-cell transcriptome pro ling of human pancreatic islets in health and type 2 diabetes," Cell Metabolism, vol. 24, pp. 593–607, 2016.

[29] Y. J. Wang, J. Schug, K. J. Won, C. Liu, A. Naji, D. Avrahami, M. L. Golson, and K. H. Kaestner, "Single-cell transcriptomics of the human endocrine pancreas," Diabetes, vol. 65, pp. 3028–3038, 2016.

[30] M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. de Koning, and A. van Oudenaarden, "A single-cell transcriptome atlas of the human pancreas," Cell Systems, vol. 3, pp. 385–394.e3, 2016.

[31] ScanPy, "Integrating data using ingest and bbknn," 2022.

[32] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas, "Massively parallel digital transcriptional pro ling of single cells," Nature Communications, vol. 8, 2017.

[33] 10x Genomics Inc, “Fresh 68k pbmcs (donor a),” 2016.

[34] 10x Genomics Inc, “10k pbmcs from a healthy donor (v3 chemistry),” 2018.

[35] F. A. Wolf, P. Angerer, and F. J. Theis, “Scanpy: large-scale single-cell gene expression data analysis,” *Genome biology*, vol. 19, no. 1, pp. 1–5, 2018.

[36] G. Fumera and F. Roli, “Support vector machines with embedded reject option,” in *International Workshop on Support Vector Machines*, pp. 68–82, Springer, 2002.

[37] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, “Integrating single-cell transcriptomic data across different conditions, technologies, and species,” *Nature biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[38] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *International conference on machine learning*, pp. 1321–1330, PMLR, 2017.

[39] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.

[40] X. Yang and S. Ji, “Jem++: Improved techniques for training jem,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6494–6503, 2021.

A Appendix

A.1 JEM Training Instability

Theoretically, the generative loss $L_{gen}[p_\theta(x)]$ should oscillate around 0, which produces generated samples whose energy values are on a similar scale as the real samples. However, we noticed that this doesn't always hold, and we presumed that this could be the implicit problem of JEM formulation [25]. As shown in Equation (5), the generative loss is defined as the difference between the expectation of fake energy values and real energy values. When $L_{gen}[p_\theta(x)]$ is negative, $E_{x \sim p_{fake}}[E_\theta(x)] < E_{x \sim p_{real}}[E_\theta(x)]$. The model parameters will be adjusted to either increase fake energy values or decrease the real energy values, vice versa. Empirically, we observed that the model tended to behave as the former. As a result, such a training strategy closes a feedback loop, and the energy values will keep increasing in the training procedure, which will eventually cause gradient explosion. Under our assumption, to stabilize the training, we must introduce additional regularization methods to break the loop.

Figure 6: Energy values of CLAMS and JEM: with the presence of the feedback loop, the expectations of energy value tended to increase, which would eventually explode. With ELU clipping, the curve became smoother, and training was more stable.

Coincidentally, activation functions used in neural network training can be employed as a clipping function to break the feedback loop. We tested three of them that have different tail behavior in the negative half-space as shown in Figure 7: ELU: exponentially suppresses negatives values with an asymptote $y = -1$; LeakyReLU linearly suppresses negative values without any asymptote; ReLU eliminates all negative values. Empirically, we ran a Bayesian hyper-parameter sweep to evaluate the model and clipping function under a pool of hyper-parameter combinations. We used the results shown in Table 4 to evaluate the robustness of clipping functions and selected the best combinations of hyper-parameters. We did not observe any divergence in the training when the model was regularized with ReLU and ELU and observed a lower chance of divergence when regularized with LeakyReLU. In consideration of other metrics, we selected ELU and conducted additional experiments based on it.

Figure 7: Clipping functions

Clipping function	# of experiments	Chance of divergence (%)	Valid Acc
None (JEM)	189	70.90	82.79
ELU	415	0	85.46
ReLU	91	0	84.44
LeakyReLU	509	37.29	85.38

Table 4: Empirical evaluation of JEM training stability on PBMC dataset with a Bayesian hyper-parameter sweep, including model architecture, batch size, weight decay, dropout rate, learning rate, buffer size, SGLD step size, and SGLD standard deviation. We found that the selected clipping functions could stabilize the training procedure. ELU and ReLU were robust to a large set of combinations of hyper-parameters.

Besides, when the generative loss becomes large and negative, the ELU clipping function suffers from the saturation problem [39], which results in almost no learning progress (see $\frac{\partial \mathrm{ELU}}{\partial x} \approx 0$ in

Figure 8a, 8b). We, therefore, introduced additional regularizations to prevent the clipping unit from being saturated. Referencing the idea of a dropout unit [26], we tested a stochastic regularization unit and a hard-threshold unit on the clipped generative loss such that the generative loss is back-propagated only a fraction of the time. Stochastic regularization behaves the same as a dropout unit acting on the generative loss $L[p_\theta(x)]$; the hard-threshold unit acts as a binary switch and only allows the generative loss to pass through when $L[p_\theta(x)] > t$. Our final model CLAMS is a JEM with ELU clipping and stochastic regularization $p_{ON} = 0.25$.

(a) Generative loss $L([p_\theta(x)])$ of pancreas dataset (MA100)

(b) Generative loss $L([p_\theta(x)])$ of PBMC dataset (MA100)

Figure 8: Generative loss $L([p_\theta(x)])$: we observed that generative loss has an impact on the training stability. The accumulation of negative loss throughout training tended to lead to divergence. Applying clipping with or without additional regularization can evidently stabilize the training. Also, the usage of additional regularization reduced the effect of the saturation problem, which was reflected in higher generative loss values.

A.2 Ablation Study

Our model CLAMS consists of multiple regularization methods acting on the original JEM model, including clipping, stochastic regularization, and hard-threshold. To investigate the effect of these techniques, we broke down our model and conducted an ablation study.

In general, we found that any combination of the proposed regularization methods could improve JEM's performance in terms of almost all evaluation metrics in our experiments. We observed a positive additive effect by combining loss clipping with other regularizations, which necessarily reduces the effect of saturation. Stochastic regularization with a large p_{ON} (higher pass-through chance) might not be sufficient to stabilize the training, but we still observed a boost in results.

An alternative interpretation to this is from the perspective of generative loss. In Figure 8, we observed that our regularization methods accomplished this by maintaining the loss value in a reasonable range. Compared with the baseline JEM, the generative loss stayed at higher values, and combinations of these regularization methods further pushed the generative loss upwards. Such observations confirm our hypothesis made in Appendix A.1 that the generative loss is the source of instability, and breaking the implicit feedback loop will resolve the problem.

In the ablation study, we mainly examined the effectiveness rather than explicitly differentiating these methods. Therefore, although we chose ELU and $p_{ON} = 0.25$ for our model, we are not arguing that other combinations of regularization are not as effective as ours. In summary, we claim that additional regularizations are necessary to stabilize the JEM training procedure and improve JEM's performance in the single-cell domain.

PBMC	n	Div.	In Acc	In ECE	$p(\mathbf{x})$	$\max p(y \mid \mathbf{x})$
ELU $\mathbf{p = 0.25}$ (Ours)	5	0	92.89 ± 0.72	5.39 ± 0.46	$\mathbf{0.92} \pm 0.01$	0.77 ± 0.02
JEM	5	100	90.32 ± 4.93	$\mathbf{2.61} \pm 0.66$	0.61 ± 0.28	0.86 ± 0.00
$p = 0.25$	5	100	$\mathbf{93.35} \pm 1.05$	6.59 ± 1.51	0.79 ± 0.14	0.78 ± 0.04
$p = 0.1$	5	0	91.81 ± 0.92	6.14 ± 0.50	0.73 ± 0.10	0.75 ± 0.03
$T = -1$	5	0	92.06 ± 0.93	5.06 ± 0.37	0.46 ± 0.15	0.77 ± 0.02
ELU	5	0	93.09 ± 0.33	4.87 ± 0.55	0.36 ± 0.11	0.80 ± 0.01
ELU $p = 0.1$	5	0	92.21 ± 0.87	4.83 ± 0.23	0.68 ± 0.05	0.78 ± 0.28
ELU $T = -1$	5	0	92.03 ± 1.18	5.04 ± 0.85	0.73 ± 0.15	0.77 ± 0.01

(a) Annotation

PBMC	n	Div.	$p(\mathbf{x})$	$\max p(y \mid \mathbf{x})$
CLAMS (Ours)	40	0	0.78 ± 0.13	0.68 ± 0.12
JEM	40	100	0.64 ± 0.18	0.58 ± 0.16
$p = 0.25$	40	92	0.76 ± 0.16	0.66 ± 0.13
$p = 0.1$	40	0	0.79 ± 0.11	0.67 ± 0.11
$T = -1$	40	0	$\mathbf{0.81} \pm 0.14$	0.69 ± 0.12
ELU	40	0	0.75 ± 0.20	0.69 ± 0.11
ELU $p = 0.1$	40	0	0.76 ± 0.13	0.67 ± 0.11
ELU $T = -1$	40	0	0.80 ± 0.15	0.69 ± 0.12

(b) One-class OOD detection

Table 5: Ablation study of PBMC dataset

Pancreas	n	Div.	In Acc	In ECE	$p(\mathbf{x})$	$\max p(y \mid \mathbf{x})$
ELU $\mathbf{p = 0.25}$ (Ours)	5	0	88.38 ± 0.62	5.81 ± 0.75	0.89 ± 0.03	0.71 ± 0.07
JEM	5	100	86.61 ± 2.07	7.57 ± 0.86	$\mathbf{0.92} \pm 0.06$	0.54 ± 0.13
$p = 0.25$	5	20	87.31 ± 1.21	6.02 ± 1.46	$\mathbf{0.92} \pm 0.05$	0.78 ± 0.09
$p = 0.1$	5	0	87.88 ± 0.56	5.73 ± 1.16	0.90 ± 0.04	0.76 ± 0.04
$T = -1$	5	0	88.50 ± 0.39	$\mathbf{4.37} \pm 0.55$	0.86 ± 0.03	0.63 ± 0.10
ELU	5	0	86.89 ± 0.50	6.19 ± 0.52	0.91 ± 0.05	0.73 ± 0.03
ELU $p = 0.1$	5	0	$\mathbf{88.85} \pm 0.74$	4.50 ± 0.80	0.88 ± 0.05	0.71 ± 0.04
ELU $T = -1$	5	0	88.55 ± 0.37	4.96 ± 0.68	0.87 ± 0.04	0.71 ± 0.04

(a) Annotation

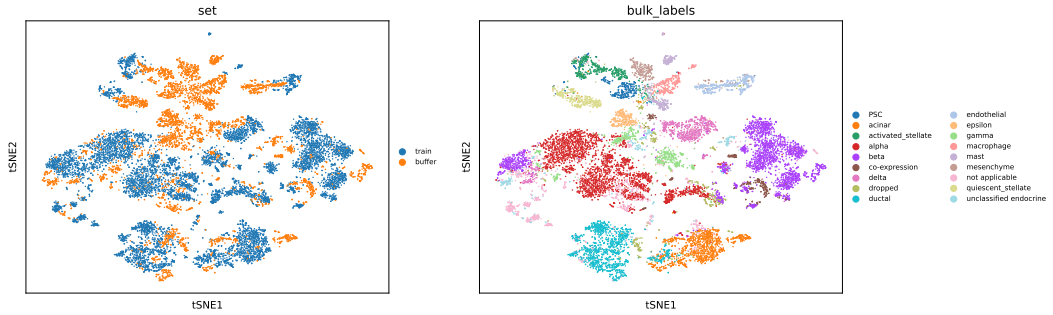
Pancreas	n	Div.	$p(\mathbf{x})$	$\max p(y \mid \mathbf{x})$
CLAMS (Ours)	75	0	0.90 ± 0.12	0.88 ± 0.10
JEM	75	100	0.80 ± 0.14	0.79 ± 0.13
$p = 0.25$	75	25	0.89 ± 0.12	0.87 ± 0.11
$p = 0.1$	75	0	0.89 ± 0.12	0.88 ± 0.10
$T = -1$	75	0	$\mathbf{0.91} \pm 0.11$	0.89 ± 0.10
ELU	75	0	0.90 ± 0.10	0.89 ± 0.08
ELU $p = 0.1$	75	0	0.90 ± 0.12	0.89 ± 0.10
ELU $T = -1$	75	0	$\mathbf{0.91} \pm 0.11$	0.89 ± 0.10

(b) One-class OOD detection

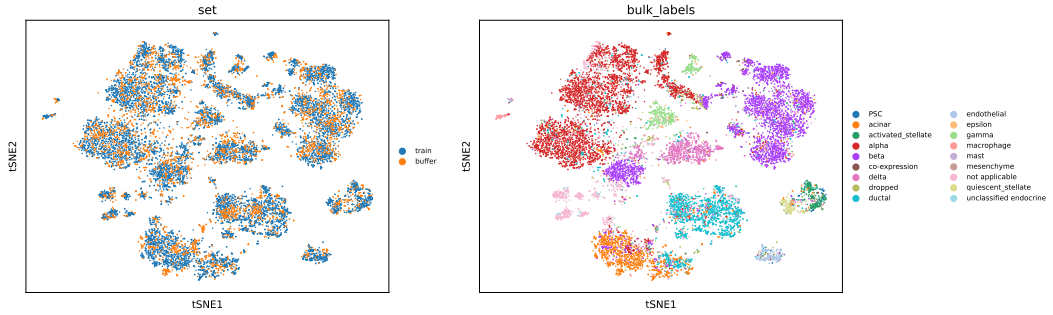
Table 6: Ablation study of pancreas dataset

A.3 Modified SGLD Sampler

We noticed that the SGLD sampler with a class-balanced replay buffer used in JEM may produce some artifacts when the training data is class-imbalanced (e.g., top orange clusters in Figure 9a). We further probed this issue and found that those classes belong to small subgroups that only take up 5% in the training set. In other words, we were trying to generate more data points than the dataset has. Works done by [2, 3, 40] mostly conducted experiments on CIFAR datasets where classes are balanced, and no significant artifacts were observed. For class-imbalanced datasets, we tested SGLD sampling with a modified class-imbalanced replay buffer, where classes that have more samples in the training set, will have more samples in the replay buffer as well. This can also be considered as diluting minority classes in the replay buffer.



(a) Data generation with SGLD sampler and a balanced buffer

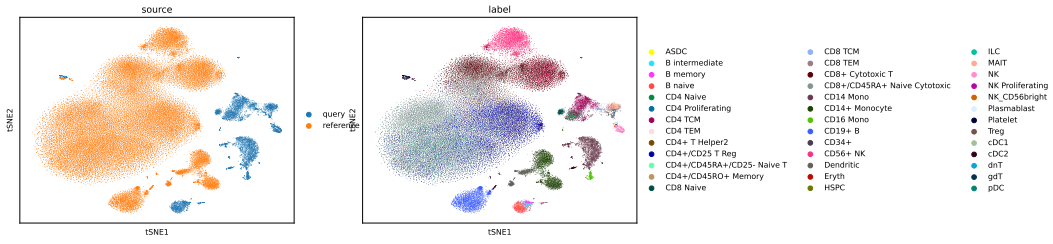


(b) Data generation with SGLD sampler and a imbalanced buffer (ours)

Figure 9: Data generation with an imbalanced replay buffer, the mixing was improved (left) and, class distributions were also well-preserved (right).

A.4 EBM for Batch Effect Correction

Here, we illustrated the possibility of using a vanilla EBM with a SGLD sampler for single-cell batch effect correction. We initialized the replay buffer with the query dataset and trained an EBM to model the reference dataset. In the training procedure, the model will iteratively update samples in the replay buffer and minimize the energy values assigned to samples in the training set and the replay buffer. This can be considered as mapping the query dataset onto the reference dataset. From Figure 10, we observed that the biological variations were well kept during the training process, and cell types in the query dataset were mapped onto the corresponding cell types in the reference dataset. This displays the possibility to be integrated with CLAMS to accomplish a more integrated analysis workflow.



(a) Uncorrected reference and query datasets in t-SNE space