From Noise to Knowledge: Probabilistic Diffusion-Based Neural Inference of Gene Regulatory Networks

Hao Zhu[1[0000−0002−3386−6076]] and Donna K. Slonim[1[0000−0003−3357−437X]]

Tufts University, Medford MA 02155, USA
{hao.zhu,donna.slonim}@tufts.edu

Abstract. Understanding gene regulatory networks (GRNs) is crucial for elucidating cellular mechanisms and advancing therapeutic interventions. Original methods for GRN inference from bulk expression data often struggled with the high dimensionality and inherent noise in the data. The prevalence of dropout events and background noise inherent in single-cell RNA sequencing have only increased network-inference challenges. Here we introduce RegDiffusion, a novel neural network structure inspired by Denoising Diffusion Probabilistic Models but focusing on the regulatory effects among feature variables. Unlike other GRN methods for single-cell RNA sequencing data, RegDiffusion introduces Gaussian noise to the input data following a diffusion schedule. It is subsequently trained to predict the added noise using a neural network with a parameterized adjacency matrix. We show that using this process, GRNs can be learned effectively with a surprisingly simple model architecture. In our benchmark experiments, RegDiffusion shows superior performance compared to several baseline methods in multiple datasets. We also demonstrate that RegDiffusion can infer biologically-meaningful regulatory networks from real-world single-cell data sets with over 10,000 genes in under 5 minutes. This work not only introduces a fresh perspective on GRN inference but also highlights the promising capacity of diffusion-based models in the area of single-cell analysis. The RegDiffusion software package and experiment data are available at https://github.com/TuftsBCB/RegDiffusion.

Keywords: Regulatory genomics · single-cell analysis · diffusion models · single-cell RNA sequencing · gene regulatory network inference.

1 Introduction

Gene regulatory networks (GRNs) describe the underlying regulatory network governing transcription and control of various cellular functions [5,25,40]. Understanding how these networks work can shed light on processes as diverse as organ development, response to inflammation, or oncogenesis. Moreover, understanding gene regulation from a systems perspective may help identify key points of regulation that could be amenable to therapeutic modulation [8].

Despite increasing experimental data and a growing recognition of the binding sites of many regulatory proteins, GRNs are often inferred primarily from transcriptomic data [34,56]. Such data sets offer unique functional *in vivo* snapshots of how genes are expressed in particular cells or under various conditions. Previously, such inference was based on microarray or bulk RNA sequencing data sets [34,47,18]. More recently, the increasing accessibility of single-cell RNA sequence data has provided a higher resolution view of expression in particular cells and cell states [37]. However, inferring GRNs from single-cell data is still challenging [41]. In part, this is because GRN inference methods struggle with the dimensionality of the data. For data sets consisting of thousands of genes, there are potentially millions of edges to consider. Many algorithms do not scale well at this level. Noisy data further amplifies the challenge. Key among the recognized patterns of noise in single-cell data is "dropout" [14], where transcripts, disproportionately those with low or moderate expression levels, are lost from the expression counts. Other work has shown that even nonzero counts can be subject to technological background noise [23].

Numerous methods, ranging from Bayesian Networks (BNs), mutual information (MI), to tree-based methods, have previously been proposed for GRN inference. BN-based methods, such as G1DBN [28] and ebdbNet [42], model GRNs as causal inference problems in a directed acyclic graph. MI-based methods, such as ARACNE [32], CLR [9], MRNET [35], and PIDC [4], measure the statistical dependence between two random variables and sort the edges according to these dependencies. Tree-based methods, such as GENIE3

[22], dynGENIE3 [21], and GRNBoost2 [36], rely on variable importance, a metric to rank variables while creating trees, to sort corresponding edges.

Recent advances in deep learning have offered new approaches to GRN inference. A recently published approach known as DeepSEM [48] learns the adjacency matrix by reconstructing the expression data through a modified Variational Auto-Encoder (VAE). The authors found that DeepSEM runs efficiently and outperforms many commonly used GRN inference methods on multiple benchmarks. Its network structure uses a parameterized adjacency matrix; an encoder, which transforms gene expression data to latent variables; and a decoder, which reconstructs the expression data from latent variables. The normality of the latent variables is enforced using the Kullback–Leibler (KL) divergence.

Yet, noise in single-cell data still presents challenges. Previously, we introduced the idea of dropout augmentation (DA) and presented a novel method called DAZZLE [60]. The DA idea establishes a bridge between the "dropout" events in single-cell data and the deep learning regularization method also called "dropout" [52]. Instead of trying to eliminate all spurious zeros in single-cell data, DAZZLE demonstrates the advantage of training the model with added zeros and explicitly predicting the (partially augmented) noise. Since a modest amount of augmented zeros can simulate dropout events, they increase model robustness and improve the benchmark performance of GRN inference. Encouraged by our positive findings, here we further explore the value of noise injection for GRN inference methods.

One emerging methodology that has shown potential in various domains, specifically computer vision, is that of probabilistic diffusion models [19,50]. By simulating the process of diffusion, or the spread of information, diffusion models are likelihood-based methods that aim to restore data from Gaussian corruption by an iterative process. In Denoising Diffusion Probabilistic Models (DDPMs) [19], a probabilistic diffusion model consists of a non-parameterized forward process, which adds small amounts of noise to the data for each of a number of steps, and a parameterized reverse process, which reconstructs a less-noisy version of the data. The reverse process is trained on predicting the added noise so that the de-noised data can be recovered by subtracting the predicted noise from the noisy data. Recent studies have suggested similarity between diffusion models and a generalized form of VAE with many latent spaces [31]. While a classic VAE enforces normality in the center latent space with KL divergence, diffusion models enforce a trajectory to normality by the diffusion process. In addition, some studies further connect diffusion models with score matching models via annealed Langevin dynamics sampling [51]. This finding puts the diffusion process within the framework of stochastic differential equations (SDE) and enables continuous diffusion steps. To date, there are a few very recent preprints using diffusion models for single-cell data [54,44], but none of these considered modeling the regulatory relationships among genes.

Here, we introduce RegDiffusion, a novel diffusion probabilistic model focusing on the interactions among variables. To our knowledge, this is the first time a diffusion probabilistic model has been applied for this purpose. We demonstrate that GRNs can be learned effectively using the objective of predicting added noise in a diffusion process. The network architecture of our model is surprisingly simple, yet it outperforms several benchmark GRN inference methods. Overall, RegDiffusion has the following advantages:

- Compared to previous VAE-based models, the runtime of RegDiffusion improves from $O(m^3n)$ to $O(m^2)$, where m is the number of genes and n is the number of cells. One of the most important theoretical breakthroughs is the utilization of a more restrictive assumption allowing elimination of the costly adjacency matrix inversion step. Inference of networks with more than ten thousand genes now takes less than 5 minutes.
- RegDiffusion combines high-performance GRN inference with high stability. Compared to VAE-based deep learning models, RegDiffusion enforces a trajectory to normality by its diffusion process, which helps stabilize the learning process.
- Although presented as a deep learning method, RegDiffusion can be viewed as a form of Bayesian Network and is based on widely-used assumptions in gene regulatory research [46].
- The success of RegDiffusion also illustrates the potential of applying diffusion models to single-cell and other kinds of noisy tabular data.

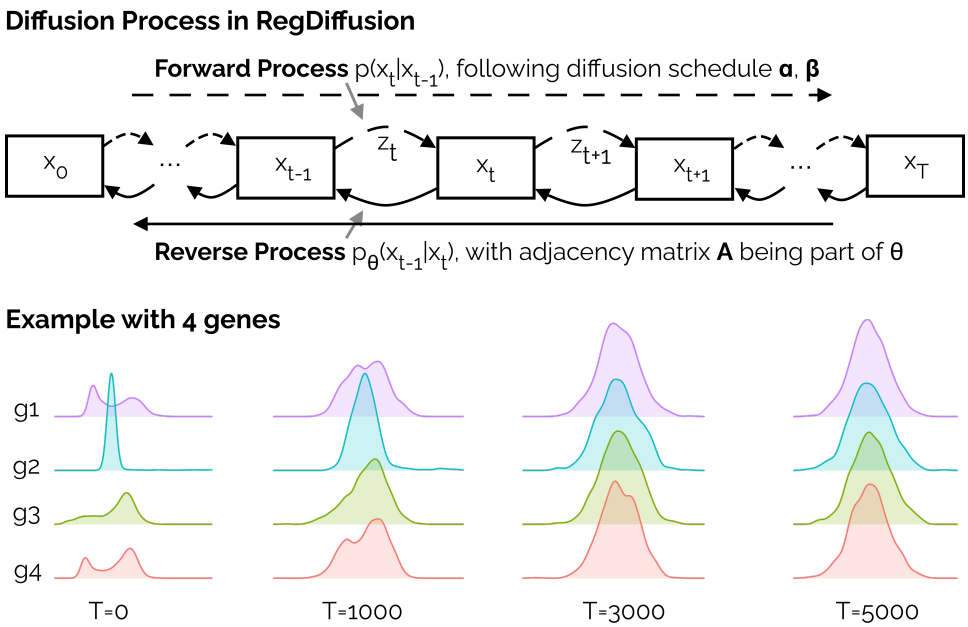

Fig. 1: The Diffusion Process is an iterative process that consists of both the forward diffusion pass and the reverse pass. The top panel (based on [19]) shows the forward diffusion process transforming all data into Gaussian noise in T steps following a diffusion schedule, while the reverse process aims to recover the original distribution from the near-Normal distribution. We are modeling the reverse process with the adjacency matrix being part of the model. The bottom panel shows four representative genes and how their distributions change over time. Each gene begins with a unique distribution over all cells, but by the final time step the distribution is nearly Normal.

2 Methods

2.1 Problem Statement

Given a single-cell expression count table $\mathbf{X} \in \mathbb{R}^{n \times m}$, where n is the number of cells and m is the number of genes, our objective is to infer the weighted adjacency matrix of the underlying gene regulatory network $\mathbf{A} \in \mathbb{R}^{m \times m}$. It is reasonable to assume that the count table $\mathbf{X}$ reflects the typical zero-inflation and background noise affecting most single-cell expression technologies. We use $\mathbf{x} \in \mathbb{R}^m$ to denote the expression counts measured in one cell.

2.2 Denoising Diffusion Probabilistic Models

Forward process As outlined in the Introduction, a DDPM consists of a forward process and a reverse process. The forward process, denoted by $q(\mathbf{x}_T|\mathbf{x}_0)$, is a non-parameterized Markov Chain that transforms an original unperturbed gene expression vector $\mathbf{x}_0$ to Gaussian noise $\mathbf{x}_T$ in T steps (eq. 1). This process generates a series of noisy samples $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T$ and as it proceeds, $\mathbf{x}$ gradually loses its distinct features as illustrated in Figure 1.

$$q(\mathbf{x}_T|\mathbf{x}_0) = \prod_{t=0}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

In each step of the forward process, x_{t-1} was perturbed under the restrain of a constant $\beta_t \in (0, 1)$. This β_t is part of a pre-generated noise schedule β. The value of β_t is usually very small. Intuitively, β_t represents the amount of information loss at each step. In the original DDPM paper [19], the authors proposed using the transformation function shown below (eq. 2; following their notation, the first parameter is simply the

noise at time t, while the second and third parameters are the mean and variance of the Normal distribution at that time). This transformation uses β_t to control both the mean and variance of the sampling process. The fact that $\sqrt{1-\beta_t}$ is always less than 1 ensures that after a number of steps, the mean of the final distribution will approach 0:

$$(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2}$$

In practice, however, it is not feasible to do iterative sampling during training. To solve this problem, the authors of DDPM proposed a reparameterization trick that enables the generation of $\mathbf{x}_t$ from $\mathbf{x}_0$, or $q(\mathbf{x}_t|\mathbf{x}_0)$, in one single step. We briefly explain this technique here, as it is crucial for our method.

Let $\alpha_t = 1-\beta_t$ and $\overline{\alpha}_t = \prod_{i=0}^{t}\alpha_i$. The reparameterization approach generates standard Gaussian noise $\mathbf{z} = \mathcal{N}(0,1)$. We can rewrite the transformation with the following equations:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{(1-\alpha_t)+\alpha_t(1-\alpha_{t-1})}\mathbf{z}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\mathbf{z}_{t-2} \\
&= ... \\
&= \sqrt{\overline{\alpha_t}}\mathbf{x}_0 + \sqrt{1-\overline{\alpha_t}}\mathbf{z}
\end{aligned} \tag{3}$$

As mentioned above, the diffusion schedule β is pre-calculated. Therefore, the the forward process at each iteration could be simplified as a non-paramerized function $f_{forward} : \mathbf{x_0} \times \mathbf{z} \times t \rightarrow \mathbf{x_t}$. To be more specific, at each iteration of the training process, for every observation (cell), we randomly generated a sampled time step t and a standard Gaussian noise $\mathbf{z}$. We put these two random variables together with the the unperturbed data $\mathbf{x}_0$ into the forward process and generate the perturbed data $\mathbf{x}_t$ at the time step t.

Reverse Process Recall that the reverse process is a Markov Chain aiming to recover the original input $\mathbf{x}_0$ from $\mathbf{x}_T$. Since computing the exact reverse step $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is intractable, we use a parameterized model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to approximate this process.

$$p_\theta(\mathbf{x}_0) = p(\mathbf{x}_T)\prod_{t=T}^{0} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(x_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t))$$

For simplicity, we chose to use the simplified loss function proposed in the DDPM paper:

$$L_{simple}(\theta) = \mathbb{E}_{\mathbf{x}_0,t,\mathbf{z}}\left[||\mathbf{z} - \mathbf{z}_\theta(\sqrt{\overline{\alpha_t}}\mathbf{x}_0 + \sqrt{1-\overline{\alpha_t}}\mathbf{z}, t)||^2\right] \tag{4}$$

In other words, we need to create a model $p_\theta(\mathbf{z}|\mathbf{x_t}, t)$ to estimate the standard Gaussian noise $\mathbf{z}$ used in the forward pass (Eq. 3). In computer vision, this challenge problem is usually resolved using image segmentation models, such as U-Net [43], to 'segment' the noise. Such approach is not applicable in single cell data, where the data is tabular and the columns are unordered genes. Here, to resolve this problem, we propose a new way of calculating the noise using the linear additive assumption in Bayesian Networks.

2.3 RegDiffusion

Graph noise estimating model Our proposed RegDiffusion method relies on the linear additive assumption, which is commonly used in many GRN inference methods, especially in those based on Bayesian Networks [11,46]. This assumption requires that the log probability of a particular gene is a linear combination of all of the regulator genes.

$$\mathbf{X} = \mathbf{X}\mathbf{A} \tag{5}$$

Note that this assumption is more restrictive than the one used in DeepSEM, which includes an addition term for random noise. The benefit of removing that term will become apparent later.

During the diffusion process, the gene expression matrix $\mathbf{X}$ is under slight Gaussian perturbation. The perturbed $\mathbf{XA}$ term on the RHS of Equation 5 will become a good approximation of the unperturbed $\mathbf{X}$ since the sum of the noise on all neighboring nodes will most likely trend back to zero.

$$\mathbf{X} \approx \mathbf{X}_t\mathbf{A}$$

This means that the perturbed gene expression $\mathbf{X}_t$ can be written in the following form:

$$\mathbf{X}_t \approx \mathbf{X}_t\mathbf{A} + \lambda_t\mathbf{Z},$$

where $\mathbf{Z}$ is the Gaussian noise and λ_t is some constant for time step t. This equation can be easily transformed into a conceptual equation to estimate $\mathbf{Z}$:

$$\begin{aligned} \mathbf{Z} &\propto \mathbf{X}_t - \mathbf{X}_t\mathbf{A} \\ &\propto \mathbf{X}_t(\mathbf{I} - \mathbf{A}) \end{aligned} \tag{6}$$

Equation 6 is exactly what we need to replace U-Net in the reverse modeling part of a diffusion model. This simple form describes a more straightforward network design compared with VAE-based models such as DeepSEM and DAZZLE. Most importantly, it is no longer necessary to perform matrix inversion on the $(\mathbf{I} - \mathbf{A})$ matrix. Matrix inversion, which is often solved by Gaussian elimination, runs in cubic time, and is one of the major bottlenecks in VAE-based models for large numbers of genes. At the same time, there is no need for separate encoders and decoders. All we need to do is to parameterize $\mathbf{X}_t$ and integrate the time step t.

The detailed structure of our proposed network, RegDiffusion, is shown in Figure 2. We explain each section in the following paragraphs.

Data Preprocessing and Normalization RegDiffusion requires a single-cell count table as input. We suggest users first follow common data preprocessing steps to address their specific needs. In experiments described here, we simply transform the data using the log2(count-plus-one) transformation and apply a standard quality assurance workflow (removing cells with high mitochondrial content or low UMI or gene counts). In terms of gene filtering, we removed genes that are not expressed at all. There is no need to restrict consideration to only the most variable genes, because RegDiffusion with all detected genes runs fairly quickly on modern GPUs.

We first normalize the expression table by cells using min-max normalization to balance inter-cell differences. Then, we z-score normalize the genes.

Embedding As shown in Figure 2, our model starts with an embedding process that transforms the input gene expression vector x_t, diffusion time step t, and (optionally designated) cell type ct to corresponding embeddings h_x^0, h_t^0, and h_{ct}^0. For each cell, the gene embedding h_x^0 is a 2D array in the shape of $\mathbb{R}^{m \times d_{gene}}$, where m is the number of genes and d_{gene} is the size of the gene embedding. The first column is always the normalized expression values. The other columns are trainable gene features that are shared across cells. Next, following the standard practice for Diffusion Probabilistic Models, we use a Sinusoidal Positional Embedding to embed the time steps t. This process transforms an ordinal time step to a meaningful vector $h_t^0 \in \mathbb{R}^{d_{time}}$ using sine and cosine transformation. Finally, the cell type embedding h_{ct}^0 is anther 2D array in the shape of $\mathbb{R}^{c \times d_{celltype}}$, where c is the number of cell types and $d_{celltype}$ is the dimension of the cell type embedding. The provided cell type index is used to pull up the corresponding cell type feature during training. Note that here, all the feature dimensions (d_{gene}, d_{time}, and $d_{celltype}$) are hyperparameters of the model. In the current implementation, we set d_{gene} to be the same as the size of the first block.

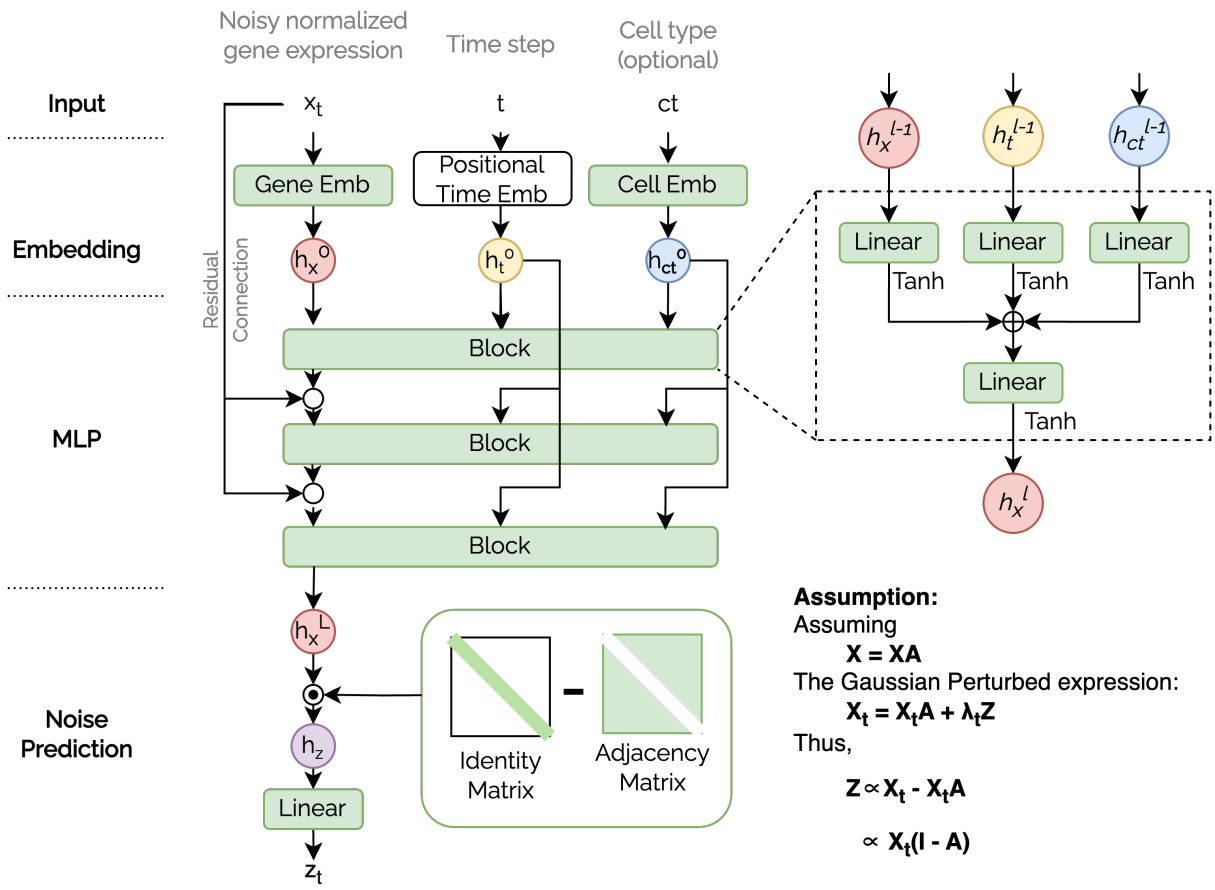

Fig. 2: Graphic illustration of the RegDiffusion model. Green areas represent trainable parameters within the model. RegDiffusion starts with embedding the model inputs (gene expression x_t, diffusion time step t, and cell type ct) into corresponding feature matrices (h_x^0, h_t^0, and h_{ct}^0). These feature vectors are then then integrated through 3 layers of training blocks. The integrated feature h_x^L then multiplies with the $(I - A)$ matrix and the outcome is used to predict the added noise z_t.

Feature Learning MLPs In RegDiffusion, we use a set of multilayer perceptrons (MLPs) to aggregate feature matrices (h_x, h_t, and h_{ct}) into a representation that is ready for noise prediction, as in Equation 6. To achieve this goal, we design an MLP building block as shown in Figure 2. This building block consumes the previous gene feature h_x^{l-1}, the original gene feature h_x^0, the time step embedding h_t^0, and the cell type embedding h_{ct}^0. These features are converted into the same dimension using three separate linear models, and the activated values are summed together followed by another linear layer to build the new gene feature h_x^l. After L layers of the learning block, we are able to build a combined feature h_x^L.

Adjacency Matrix Initialization and Regularization The adjacency matrix in RegDiffusion is an $m \times m$ matrix of trainable model parameters, where m is the number of genes. Here we assume the average effect of a random gene to another, or the "regulation norm", is $\frac{1}{m-1}$. The initialization value of the adjacency matrix is set to five times the regulation norm to help the model explore both potential links and non-links. During the training process, unless otherwise specified, the learning rate on the adjacency matrix is set to one fiftieth of the regulation norm to ensure the learning steps on the adjacency matrix scale with the number of genes. To help the model converge faster, we also employ a soft threshold function (as shown in Equation 7) of half the regulation norm on the absolute values of the edge weights, so tiny edge weights are clipped during the training process. At the same time, the diagonal entries of the adjacency matrix are set to zeros to discourage self-regulation. To increase the sparsity of the adjacency matrix, we applied a combination of

L1 and L2 losses on the matrix (weights controlled by two hyper parameters). To prevent overfitting and increase model robustness, we also apply dropout to the adjacency matrix; specifically, a proportion of edge weights are randomly set to zeros while the rest are scaled accordingly during training.

$$S_\tau(x) = \text{sign}(x) \cdot \max(0, |x| - \tau) \tag{7}$$

Noise Prediction Following equation 6, with the learned feature matrix h_x coming out of the stacked blocks and the adjacency matrix, we can calculate the feature of the added noise h_z using matrix multiplication. This feature is then converted into the prediction of noise z using a linear layer.

Data Loading and Model Convergence RegDiffusion considers data from each cell as an observation. In reality, the number of cells in different single-cell data varies significantly. For different datasets, if the model sees the entire dataset in every training iteration, it would take different numbers of iterations to converge. To overcome this issue, in each training iteration, we sample a fixed number of cells with replacement. This bootstrap resampling strategy also helps make RegDiffusion more robust.

The convergence point of RegDiffusion could be identified using the "amount of change" of the adjacency matrix with the help of the iteration training loss. The "amount of change" score is defined as the average change in value of the adjacency matrix after one iteration of training, in units of the regulation norm ($\frac{1}{m-1}$). Eventually, this score will approach 0, reflecting that at this point in the training process, the optimizer is only making tiny updates on the adjacency matrix, showing that the model has converged.

As shown later in the result section, with a fixed sampling size per iteration, the convergence of RegDiffusion on different datasets follows a similar pattern, regardless of the number of genes and cells. The default number of training iteration is selected based on these observations.

Network Interpretation and Neighbor/Neighborhood Extraction After convergence, the next task is to interpret the inferred network. However, the difficulty of this task increases quadratically with the number of genes. Here, for the purpose of validating the inferred networks in their biological contexts, we propose visualizing the 2-hop local neighborhoods of specific interesting genes. For a given gene, we extract its top-k neighbors (sorted by the absolute values of their edge weights) as the first hop; then, for each gene in that set, we extract its top-k neighbors and add them to the "second hop" set. Finally, we scan through the neighbors of the second-hop genes, to add any remaining edges to nodes in either the first hop or second hop node sets. Thus, multiple genes that work together as a group are pulled closer in the visualization space, making their biological relationship more apparent.

2.4 Experiment Design

Experiments were run on the BEELINE benchmark data sets, where there are (multiple, different) putative "ground truth" networks to which to compare inferred regulatory edges. In addition, we inferred networks from two published sets of single-cell data in microglia, which can be compared to our knowledge about regulation in these cell types and which demonstrate the method's feasibility in a more realistic context. Experimental details appear in the following subsections.

Beeline Benchmark Data Sets The BEELINE single-cell benchmark [41] comprises 7 distinct preprocessed single-cell RNAseq datasets from both human (hESC, hHep) and mouse (mDC, mESC, mHSC-E, mHSC-GM, mHSC-L), offering a broad scope for evaluation. With the provided benchmark-generation scripts in BEELINE, users can specify the number of target genes. BEELINE selects the most variable genes, based on the variance and p values returned by a fitted generalized additive model (GAM). BEELINE also suggests evaluating with respect to three sets of "ground truth" networks, namely the STRING v11 database [53], non-cell-type specific transcriptional networks, and cell-type specific networks. The non-cell-type specific network combines ChipSeq-based data collected from DoRothEA[13], RegNetwork[30], and TRRUST v2[17], while the cell-type specific networks were experimentally generated by the BEELINE authors. Here, we use

the STRING network as the ground truth and set the number of target genes to be 1,000. The exact numbers of genes/cells/true edges for each dataset are reported in Table 1. Our results on non-cell-type specific networks, yielding similar findings, are included in the Supplement.

The cell-type-specific networks, though intuitively appealing, proved problematic for evaluation in several ways. First, *all* of the unsupervised GRN inference methods we tested report performance near that of random guessing when compared to the cell-type specific data for the corresponding network. Second, although DeepSEM reports slightly better results than we found with the other methods (though still effectively no better than random guessing), that performance required a different set of hyperparameters than was otherwise suggested, one that minimizes the regularization on the adjacency matrix and minimizes the regulations of the KL divergence. The underlying cause of this pattern is still unclear, so we do not feel comfortable with the comparisons to these networks. Nonetheless, the performance of RegDiffusion (under minimal adjacency matrix regularization) compared to this data set is still reported in the Supplement, to be interpreted with these caveats in mind.

We compared the performance of RegDiffusion on these seven data sets to four other methods: GENIE3, GRNBoost2, DeepSEM, and our previously proposed method DAZZLE. GENIE3 and GRNBoost2 are two tree-based machine learning methods that have been found to perform well on single-cell data according to recent benchmarks [24,41]. Particularly, as previously reported, with the same BEELINE benchmark we use, they are among the best three methods (together with PIDC [4]) and have been found to outperform SCODE [33], SINCERITIES [39] and PPCOR [26]. DeepSEM and DAZZLE are both variational autoencoder based neural networks. For GENIE3 and GRNBoost2, we used the implementation from the Python package arboreto v0.1.6 [36]. For DeepSEM and DAZZLE, we used our implementation from the Python package GRN-dazzle v0.0.2 [60]. Note that as described by the authors, DeepSEM is an ensemble algorithm that combines results from 10 runs. Because of its speed, we include the single-run version, denoted by DeepSEM*, in our comparison as well as the ensemble version.

Evaluation Metrics for Benchmark Data Following the BEELINE paper, we use Area Under the Precision Recall Curve *Ratio* (AUPRCR) as the main evaluation metric. For the link prediction task as we have here, there is a huge class imbalance between the positive and negative group. As argued in [6], AUPRC is a better metric than Area Under the Receiver Operating Characteristic (AUROC) when dealing with class imbalance. AUPRC Ratio is simply the AUPRC score divided by that of a random predictor. Since the number of all possible edges and the number of true edges are usually very large in actual GRNs, the values of the AUPRC itself tend to be small. Converting them to ratios make it easier to understand the performance across benchmarks.

We also report the ratio value for Early Precision (EP), which is essentially the top-k precision where k is the number of true edges. Results for the EP ratio are provided in the Supplement. Because it is a more familiar metric, the AUPRC ratio results are shown in the main text.

Hammond Microglia Dataset To assess RegDiffusion's capacity on real-world single-cell data, we tested it on two different mouse brain microglia datasets. The first one is taken from a published single-cell data set on mouse microglia across the lifespan, by Hammond, *et al.*[16] (Gene Expression Omnibus [7] accession GSE121654). For our analysis, we used data from all four healthy male adults (P100) and filtered out cells with fewer than 400 or more than 3,000 unique genes, cells with more than 10,000 UMIs, and cells with over 3% of reads mapping to mitochondrial genes. For clarity and interpretability of results, we further removed mitochondrial genes, ribosomal genes, and genes with only "model" RefSeq entries (genes predicted by automated annotation pipelines only, with "GM_" RefSeq prefixes ([38])). Finally, we removed genes that were not expressed at all in any of the cells. The final processed data consists of 8,258 cells and 14,065 genes. Note that this data set potentially includes many different microglia subtypes.

Cerebellum Atlas Microglia Dataset The second real-world microglia single-cell data is derived from a published transcriptomic atlas of mouse cerebellar cortex consisting 611,034 cells of many types [27]. The authors assigned cells to into known cell types based on specific gene markers in cell clusters. For microglia cells, the marker genes used were *Prkcd* and *C1qa*. From this dataset, we didn't remove any cells from those

designated as microglia, but we applied the same criteria as above to filter genes. We then log-plus-one transformed the count data. The final processed data set consists of 1,296 cells and 15,547 genes.

3 Experimental Results

3.1 RegDiffusion is Highly Accurate on BEELINE Benchmarks

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
# of Genes	1410	1448	1321	1620	1204	1132	692
# of Cells	758	425	383	421	1071	889	847
# of True Edges	5,149	9,000	5,898	8,479	1,826	1,311	154
GENIE3	1.98 (0.01)	1.86 (0.01)	**1.72 (0.01)**	2.05 (0.01)	4.19 (0.05)	6.27 (0.04)	7.02 (0.08)
GRNBoost2	1.67 (0.01)	1.50 (0.01)	1.43 (0.01)	1.88 (0.02)	3.65 (0.05)	5.16 (0.11)	7.07 (0.08)
DeepSEM* (single-run)	1.95 (0.08)	1.66 (0.10)	1.57 (0.05)	2.20 (0.03)	5.00 (0.49)	5.83 (0.81)	6.69 (1.81)
DeepSEM (ensemble)	2.11 (0.02)	1.85 (0.02)	*1.67 (0.02)*	*2.31 (0.03)*	5.67 (0.13)	*6.76 (0.17)*	*7.50 (0.12)*
DAZZLE	**2.50 (0.05)**	*1.87 (0.02)*	1.60 (0.02)	2.26 (0.06)	**6.02 (0.15)**	6.39 (0.16)	**7.65 (0.12)**
RegDiffusion	*2.48 (0.01)*	**1.99 (0.01)**	1.66 (0.01)	**2.45 (0.02)**	*5.96 (0.07)*	**7.14 (0.07)**	7.28 (0.07)

Table 1: RegDiffusion outperforms most GRN inference methods on BEELINE benchmark datasets (Metric: AUPRC Ratio; Ground truth: STRING; Number of target genes: 1000). Numbers reported are mean and std. of AUPRC Ratio compared with random guess over 10 runs. Higher ratios indicate better performance. Here, bold **Red** and italicized *orange* text indicate the best and 2nd best algorithms. In this experiment, we ran each algorithm on all available data with the default settings.

Table 1 presents a performance comparison of AUPRC ratios for several GRN inference methods on the BEELINE benchmark data sets. The seven data sets differ in terms of the number of cells, genes, and edges in the ground truth networks - details appear in the table header.

The proposed RegDiffusion method generally outperforms most other GRN inference techniques across the different datasets, as indicated by higher AUPRC Ratios. It achieves top results for the hHep, mESC, and mHSC-GM data sets, and it is either the second or third best for the others, with performance above 95% of the top score in all cases.

Results for the EP Ratio metric also used in the DeepSEM paper appear in the Supplement. RegDiffusion performs even better by that metric, where it is either best or second best for all data sets. The Supplement contains results for different "ground truth" data sets and EPR or AUPRCR, but in all cases, RegDiffusion is either the clear winner, or one of the top performers along with DAZZLE and DeepSEM (ensemble).

Furthermore, the performance of RegDiffusion is very stable across runs, as indicated by the small standard deviations of AUPRC Ratios in 10 repeated runs. It has even better stability than the ensemble version of DeepSEM and is almost as stable as GENIE3 and GRNBoost2.

3.2 RegDiffusion Runs Faster that Competitors

As mentioned in section 2, one of the most important breakthroughs in RegDiffusion is the elimination of the matrix inversion step on the adjacency matrix, which is required by previous VAE-based models. Matrix inversion requires cubic time and is the key computational bottleneck when the number of genes is large. By incorporating bootstrap sampling with a fixed pool size, the running time of RegDiffusion is independent of the number of cells. In Figure 3a, by inspecting the amount of change in the adjacency matrix per iteration in the unit of the regulation norm ($\frac{1}{m-1}$), we show that across all of our experiments, whether on benchmark or real data sets, the training processes follow a similar pattern and the models converge after a fixed number of steps.

Compared to previous VAE-based models, the run time of RegDiffusion drops from $O(m^3n)$ to $O(m^2)$. A comparison of algorithm execution time is provided in Table 2. Across all benchmarks, RegDiffusion runs

	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L	Hammond	Atlas
# of Genes	1,410	1,448	1,321	1,620	1,204	1,132	692	14,065	15,547
# of Cells	758	425	383	421	1,071	889	847	8,258	1,296
GENIE3	6,462	2,931	2,709	4,173	7,723	5,674	2,493	-	-
GRNBoost2	2,055	992	962	2,171	2,442	1,382	379	-	-
DeepSEM* (single run)	29	15	12	17	30	23	15	4hr 21min	57min
DeepSEM (ensemble)	251	147	116	168	298	229	139	43hr 33min	9hr 34min
DAZZLE	31	16	13	18	34	27	19	4hr 47min	1hr 10min
RegDiffusion	7	8	7	9	6	5	5	3min 28s	4min 10s

Table 2: Algorithm execution time (reported in seconds unless noted). Time cost on BEELINE data sets are averages over 10 runs. All code was executed on HPC computing nodes with 4 cores and 12GB memory. DeepSEM, DeepSEM*, DAZZLE, and RegDiffusion are executed with an additional NVIDIA A100 card.

in a fraction of the time needed by any previous algorithm. This speedup has greater impact on larger data sets. For the two real-world single-cell data sets, each with tens of thousands of genes and thousands of cells, RegDiffusion taes only minutes, while previous VAE-based methods require hours of computation.

3.3 Characteristics of RegDiffusion

Since RegDiffusion is trained only on the objective of predicting added noise, the ground truth GRN network never leaks into the training process. It works in a unsupervised manner, in the same way as GENIE3 or GRNBoost2. To ensure there was no overfitting, we further investigated the training behavior by applying a train/test split on cells. Figure 3b illustrates the training losses of a typical training process on the mESC data set when we applied a 4:1 train/test split. As shown in the figure, the loss curves for the train and test splits are never separated too much, showing that the model was not overfitted. Also, note that since the diffusion steps are stochastic and it is usually harder to learn the fine details, it is expected to see training loss oscillating even after the model converges.

In terms of the number of parameters used in the model, we tested a number of choices for the dimension of the hidden layer in the MLP building block; results are shown in Figure 3c. For all dimensions greater than 8, the performance of the model was rather stable. Since we would like to reduce the number of parameters in the model as much as possible, we set the default model dimension to 16.

Finally, we investigated the performance of RegDiffusion with limited training data. We gradually down-sampled the training data and then compared the AUPRC Ratios of results from RegDiffusion, GENIE3, and GRNBoost2. Figure 3d shows that even with 10% of the mESC data, which has around 42 cells, RegDiffusion still works thanks to the bootstrap resampling methods. It also outperforms tree-based methods, which are usually considered to be robust. In practice, some cell subtypes may only have a few sampled cells. This analysis shows RegDiffusion can be effective even in such cases.

3.4 Regulatory Networks in Microglia

In addition to benchmark testing, we further examined the biological interpretability of the inferred GRNs in mouse microglia using data from the two complementary studies described in Section 2.4. For each study, we inferred a GRN and analyzed the local regulation neighborhood around particular genes. For this discussion, we focus on *ApoE* (apolipoprotein E) and visualize the gene's RegDiffusion-inferred network neighborhoods from both studies' data in Figure 4. *ApoE* is a well studied lipid regulator that plays a role in several diseases and disorders, but it is probably best known for the E4 allele's association with Alzheimer's Disease (AD) [10].

The ApoE neighborhood inferred from the Hammond dataset is essentially a superset of the neighborhood inferred from the Cerebellum Atlas data. This is possibly because the microglia from the Atlas dataset were selected from clusters marked by expression of (*C1qa* and *Prkcd*). Some microglia subtypes may have been filtered out in this process, or there might have been fewer microglial cells because of the focus on cerebellum and the large number of other cell types considered in the atlas.

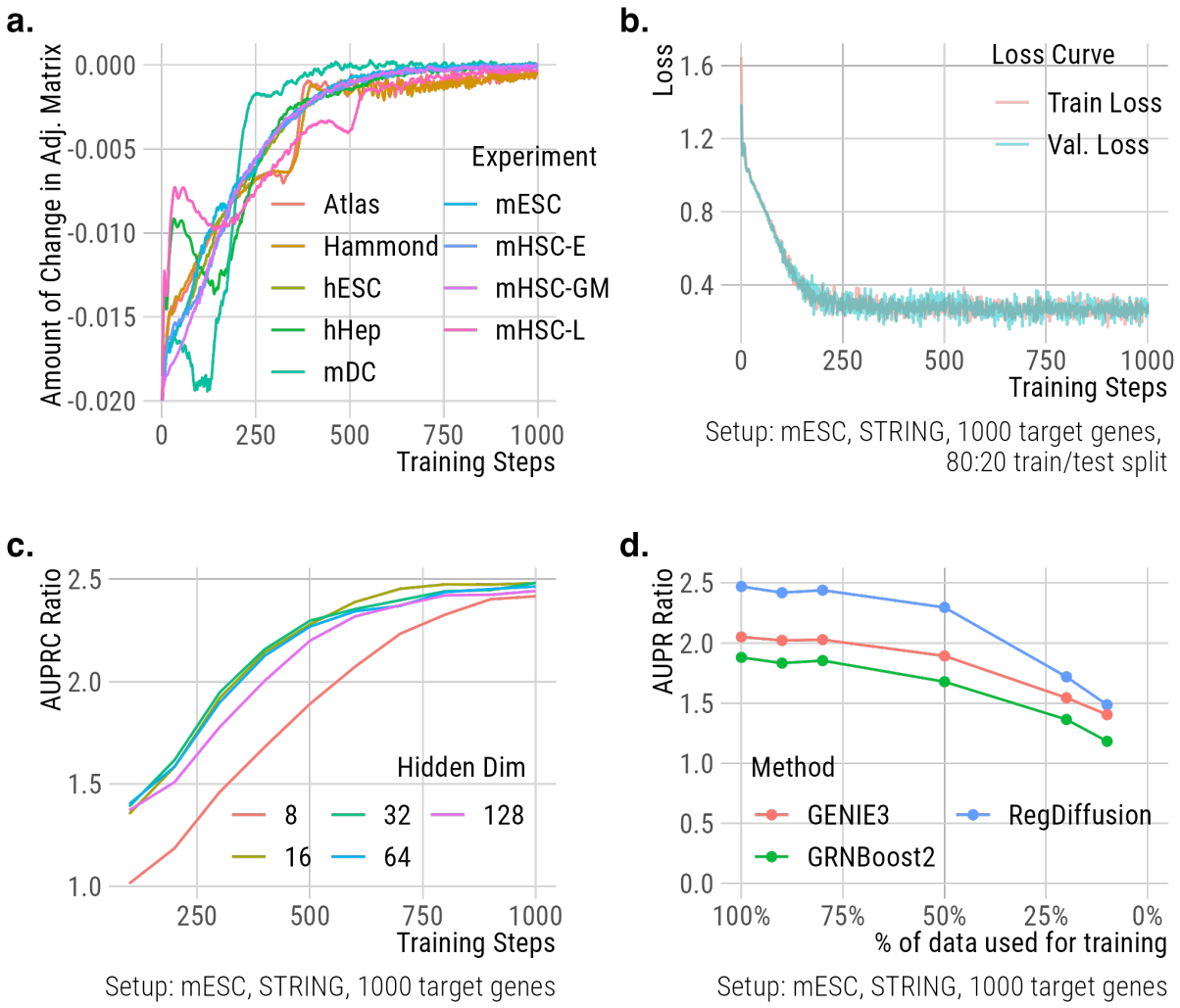

Fig. 3: Properties of RegDiffusion. **a.** By measuring the amount of change in the adjacency matrix, we show RegDiffusion converges after a fixed number of steps in all experiments; **b.** Loss curves on train/test splits (4:1) show that the model was not overfitted. Also, it's expected to see the training loss oscillating even after the model converges; **c.** Model performance is rather stable regardless of the hidden dimensions in the neural networks; **d.** Downsampling shows that RegDiffusion performs better than bulk RNAseq methods even when the sample size is limited.

Nonetheless, many key relationships are shared across the data sets in Figure 4, and many inferred links have previously been experimentally confirmed. For example, the regulatory relationship between *ApoE* and MHC (major histocompatibility complex) type I genes, such as *H2.K1* and *H2.D1* has been recently confirmed in [58] via an ApoE perturbation experiment. Activated Response Microglia (ARMs), a microglia subtype discovered in [12], have been found to have overexpressed MHC type II genes, including *H2-Aa*, *H2-Ab1*, and *Cd74*. RegDiffusion even further identified *H2.Eb1* as another potential biomarker for this cell type. On the right side of the MHC II complex in Figure 4, we have another pair of genes *Mrc1* and *Ms4a7*. These two genes have been identifed in [16] as biomarkers of a unique microglial state during the embryonic phase. While the data we analyzed is from adult mouse, the topological position of these two groups suggest a possible relationship between the ARMs cell type and that embryonic cell state. We further identified connections between *ApoE* and a set of antiviral response genes, including *Ifitm3*, which connects directly

ApoE Regulation Neighborhoods

Gene Group	Marker Genes	Refs
C1q Complex	C1qa, C1qb, C1qc	[12]
Cathepsins	Ctsa, Ctsb, Ctsd, Ctsh, Ctss, Ctsz	[26],[43],[56]
Ferritin	Ftl1, Fth1	[1]
Microglia Homeostatic genes	Itm2b, Trem2, Hexa, Hexb	[2],[52],[54]
MHC I	H2.D1, H2.K1	[55]
MHC II	H2.Aa, H2.Ab1, Cd74	[9]
Antiviral function	Ifitm3, Ifi27l2a, Ifi204, Oasl2, Bst2	[17]

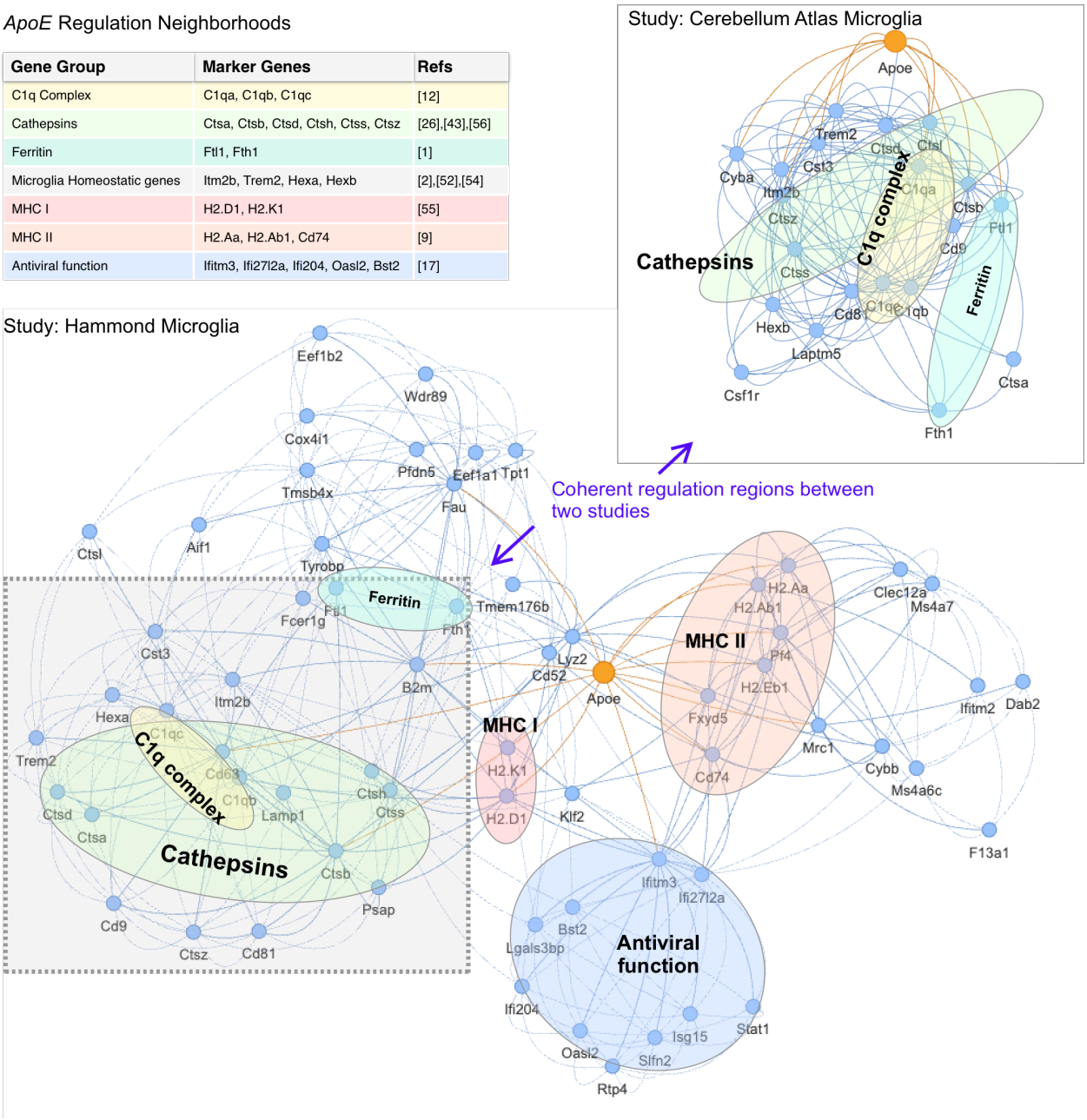

Fig. 4: Inferred 2-Hop Regulatory Networks (k=20) around *ApoE* based on single-cell data from two studies of microglia. Consistent, plausible networks are identified.

to *ApoE* based on our model. *Ifitm3* has previously been identified as a γ-secretase modulatory protein that increases the production of amyloid-β in AD ([20]).

If we compare the inferred *ApoE* neighborhoods between the two studies, we can identify coherent shared patterns. These recaptulate previously identified links among *ApoE* and several functional gene groups, including the *C1q* (Complement component 1q) complex (*C1qa*, *C1qb*, and *C1qc* [15]), Cathepsins (such as *Ctsd*, *Ctsb*, and *Ctss* [59], [45], [29]), Ferritins (*Ftl1* and *fth1* [1]), and several microglia homeostatic genes (such as *Trem2* [57], *Itm2b*[3], and *Hexb*. (While we didn't find prior evidence explicitly showing ApoE regulating β-hexosaminidase (*Hexa*, *Hexb*), there is evidence that they are both implicated in, and have correlated expression in, AD pathology [55,49], suggesting such a relationship is plausible). Further, β-hexosaminidase

plays a role in lysosomal function and autophagy. Our recent work examining fetal microglia in the setting of maternal immune activation[2] shows that the CLEAR (Coordinated Lysosomal Expression and Regulation) pathway, particularly *Hexa*, *Hexb*, and several cathepsins, is differentially regulated in microglia exposed to inflammation. From the consistency and plausibility of these networks, we conclude that the results from running RegDiffusion on current single-cell data sets are reasonable and consistent.

For comparison purposes, the local networks inferred by DeepSEM are provided in the supplements. Overall, DeepSEM yields similar findings on the Hammond dataset. However, the MHC I group and the antiviral group are missing. There are more microglia homeostatic genes but there are also more orphan links that are probably false predictions. For the Atlas dataset, DeepSEM generates a lot of orphan links but it still captures the *C1q* complex and the cathepsins among the noise. In addition to the local neighborhood around *ApoE*, we also provide visualizations for 10 other microglia signature genes on our project website: `https://tuftsbcb.github.io/RegDiffusion/supplements.html`.

4 Conclusion

We have presented RegDiffusion, a novel diffusion probabilistic model designed to infer GRNs from single-cell RNAseq data. RegDiffusion simulates the information diffusion process by iteratively adding Gaussian noise following a fixed schedule. Based on the benchmarking results, compared to other top methods, RegDiffusion demonstrates consistent superiority in GRN inference capacity while remain stable across runs. It also achieves a significant runtime reduction over comparable methods. Even with limited data, RegDiffusion still outperforms competitors. At the same time, by visualizing a local gene neighborhood in the inferred GRNs, we illustrate that GRNs inferred by RegDiffusion are consistent with current knowledge of molecular function.

Following the idea of Dropout Augmentation [60], which simulates the dropout phenomenon in single cell data by inserting zero values into expression data using Bernoulli sampling, RegDiffusion is our second attempt on leveraging noise injection in single cell analysis. The success of RegDiffusion in GRN inference also suggests several potential future research paths. First, it would be very interesting to explore the effects of RegDiffusion and its learned embeddings on other single-cell analysis tasks. Also, to our knowledge, RegDiffusion is the first diffusion model working on tabular data for causal inference. It would be interesting to see if it can be extended as a general causal inference method. Another important future research direction is understanding the inferred networks. Our proposed visualization method works well for inspecting individual genes, but it would be nice to investigate ways to understand the graphs more systematically. Finally, it would be helpful to find a solution to learn detailed features in single-cell data given the existence of background noise. In our experiments, we observed that the models are good at learning crude features (at high diffusion steps) but still have some difficulties learning fine features at low diffusion steps, a phenomenon that has been observed many times in other diffusion models.

Diffusion models in general have sparked a lot of research interest in recent years thanks to their huge success in computer vision. Together with a few recent pre-prints in 2023 [54], RegDiffusion presents as one of the initial attempts to solve the problems in single-cell data with diffusion models and is the first one that considers gene regulatory relationships. We anticipate that RegDiffusion many not only offer a reliable and convenient tool for GRN inference, but may also advance our understanding of regulatory genomics and single-cell analysis.

References

1. Ayton, S., Faux, N.G., Bush, A.I.: Ferritin levels in the cerebrospinal fluid predict alzheimer's disease outcomes and are regulated by apoe. Nature communications **6**(1), 6760 (2015)
2. Batorsky, R., Ceasrine, A.M., Shook, L.L., Kislal, S., Bordt, E.A., Devlin, B.A., Perlis, R.H., Slonim, D.K., Bilbo, S.D., Edlow, A.G.: Hofbauer cells and fetal brain microglia share transcriptional profiles and responses to maternal diet-induced obesity. bioRxiv (Dec 2023)
3. Biundo, F., Ishiwari, K., Del Prete, D., D'Adamio, L.: Interaction of apoe3 and apoe4 isoforms with an itm2b/bri2 mutation linked to the alzheimer disease-like danish dementia: Effects on learning and memory. Neurobiology of learning and memory **126**, 18–30 (2015)

4. Chan, T.E., Stumpf, M.P., Babtie, A.C.: Gene regulatory network inference from single-cell data using multivariate information measures. Cell systems **5**(3), 251–267 (2017)
5. Davidson, E., Levin, M.: Gene regulatory networks. Proceedings of the National Academy of Sciences **102**(14), 4935–4935 (2005)
6. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240 (2006)
7. Edgar, R., Domrachev, M., Lash, A.: Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. **30**(1), 207–10 (Jan 2002)
8. Emmert-Streib, F., Dehmer, M., Haibe-Kains, B.: Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. Frontiers in cell and developmental biology **2**, 38 (2014)
9. Faith, J.J., Hayete, B., Thaden, J.T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J.J., Gardner, T.S.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS biology **5**(1), e8 (2007)
10. Fernández-Calle, R., Konings, S.C., Frontiñán-Rubio, J., García-Revilla, J., Camprubí-Ferrer, L., Svensson, M., Martinson, I., Boza-Serrano, A., Venero, J.L., Nielsen, H.M., et al.: Apoe in the bullseye of neurodegenerative diseases: Impact of the apoe genotype in alzheimer's disease pathology and brain diseases. Molecular Neurodegeneration **17**(1), 62 (2022)
11. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using bayesian networks to analyze expression data. In: Proceedings of the fourth annual international conference on Computational molecular biology. pp. 127–135 (2000)
12. Frigerio, C.S., Wolfs, L., Fattorelli, N., Thrupp, N., Voytyuk, I., Schmidt, I., Mancuso, R., Chen, W.T., Woodbury, M.E., Srivastava, G., et al.: The major risk factors for alzheimer's disease: age, sex, and genes modulate the microglia response to aβ plaques. Cell reports **27**(4), 1293–1306 (2019)
13. Garcia-Alonso, L., Holland, C.H., Ibrahim, M.M., Turei, D., Saez-Rodriguez, J.: Benchmark and integration of resources for the estimation of human transcription factor activities. Genome research **29**(8), 1363–1375 (2019)
14. Ghazanfar, S., Bisogni, A.J., Ormerod, J.T., Lin, D.M., Yang, J.Y.: Integrated single cell data analysis reveals cell specific networks and novel coactivation markers. BMC systems biology **10**(5), 11–24 (2016)
15. Habenicht, L.K., Wang, Z., Zhang, X., Li, Y., Mogler, C., Huspenina, J.S., Schmid, R.M., Weber, C., Mohanta, S.K., Ma, Z., et al.: The c1q-apoe complex: A new hallmark pathology of viral hepatitis and nonalcoholic fatty liver disease. Frontiers in Immunology **13**, 970938 (2022)
16. Hammond, T.R., Dufort, C., Dissing-Olesen, L., Giera, S., Young, A., Wysoker, A., Walker, A.J., Gergits, F., Segel, M., Nemesh, J., et al.: Single-cell rna sequencing of microglia throughout the mouse lifespan and in the injured brain reveals complex cell-state changes. Immunity **50**(1), 253–271 (2019)
17. Han, H., Cho, J.W., Lee, S., Yun, A., Kim, H., Bae, D., Yang, S., Kim, C.Y., Lee, M., Kim, E., et al.: Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. Nucleic acids research **46**(D1), D380–D386 (2018)
18. Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., Guthke, R.: Gene regulatory network inference: data integration in dynamic models—a review. Biosystems **96**(1), 86–103 (2009)
19. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
20. Hur, J.Y., Frost, G.R., Wu, X., Crump, C., Pan, S.J., Wong, E., Barros, M., Li, T., Nie, P., Zhai, Y., et al.: The innate immunity protein ifitm3 modulates γ-secretase in alzheimer's disease. Nature **586**(7831), 735–740 (2020)
21. Huynh-Thu, V.A., Geurts, P.: dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. Scientific reports **8**(1), 3384 (2018)
22. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PloS one **5**(9), e12776 (2010)
23. Janssen, P., Kliesmete, Z., Vieth, B., Adiconis, X., Simmons, S., Marshall, J., McCabe, C., Heyn, H., Levin, J.Z., Enard, W., et al.: The effect of background noise and its removal on the analysis of single-cell expression data. Genome Biology **24**(1), 140 (2023)
24. Kang, Y., Thieffry, D., Cantini, L.: Evaluating the reproducibility of single-cell gene regulatory network inference algorithms. Frontiers in genetics **12**, 362 (2021)
25. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. Nature reviews Molecular cell biology **9**(10), 770–780 (2008)
26. Kim, S.: ppcor: an r package for a fast calculation to semi-partial correlation coefficients. Communications for statistical applications and methods **22**(6), 665 (2015)
27. Kozareva, V., Martin, C., Osorno, T., Rudolph, S., Guo, C., Vanderburg, C., Nadaf, N., Regev, A., Regehr, W.G., Macosko, E.: A transcriptomic atlas of mouse cerebellar cortex comprehensively defines cell types. Nature **598**(7879), 214–219 (2021)
28. Lèbre, S.: Inferring dynamic genetic networks with low order independencies. Statistical applications in genetics and molecular biology **8**(1) (2009)

29. Li, H., Zhao, Q., Liu, D., Zhou, B., Liao, F., Chen, L.: Cathepsin b aggravates atherosclerosis in apoe-deficient mice by modulating vascular smooth muscle cell pyroptosis through nf-κb/nlrp3 signaling pathway. Plos one **19**(1), e0294514 (2024)
30. Liu, Z.P., Wu, C., Miao, H., Wu, H.: Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. Database **2015** (2015)
31. Luo, C.: Understanding diffusion models: A unified perspective. arXiv preprint arXiv:2208.11970 (2022)
32. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D., Califano, A.: Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In: BMC bioinformatics. vol. 7, pp. 1–15. BioMed Central (2006)
33. Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M.S., Ko, S.B., Gouda, N., Hayashi, T., Nikaido, I.: Scode: an efficient regulatory network inference algorithm from single-cell rna-seq during differentiation. Bioinformatics **33**(15), 2314–2321 (2017)
34. Mercatelli, D., Scalambra, L., Triboli, L., Ray, F., Giorgi, F.M.: Gene regulatory network inference resources: A practical overview. Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms **1863**(6), 194430 (2020)
35. Meyer, P.E., Kontos, K., Lafitte, F., Bontempi, G.: Information-theoretic inference of large transcriptional regulatory networks. EURASIP journal on bioinformatics and systems biology **2007**, 1–9 (2007)
36. Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., Aerts, S.: Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. Bioinformatics **35**(12), 2159–2161 (2019)
37. Nguyen, H., Tran, D., Tran, B., Pehlivan, B., Nguyen, T.: A comprehensive survey of regulatory network inference methods using single cell rna sequencing data. Briefings in bioinformatics **22**(3), bbaa190 (2021)
38. O'Leary, N.A., Wright, M.W., Brister, J.R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., et al.: Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. Nucleic acids research **44**(D1), D733–D745 (2016)
39. Papili Gao, N., Ud-Dean, S.M., Gandrillon, O., Gunawan, R.: Sincerities: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. Bioinformatics **34**(2), 258–266 (2018)
40. Penfold, C.A., Wild, D.L.: How to infer gene networks from expression profiles, revisited. Interface focus **1**(6), 857–870 (2011)
41. Pratapa, A., Jalihal, A.P., Law, J.N., Bharadwaj, A., Murali, T.: Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nature methods **17**(2), 147–154 (2020)
42. Rau, A., Jaffrézic, F., Foulley, J.L., Doerge, R.W.: An empirical bayesian method for estimating biological networks from temporal microarray data. Statistical Applications in Genetics and Molecular Biology **9**(1) (2010)
43. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
44. Sadria, M., Layton, A.: The power of two: integrating deep diffusion models and variational autoencoders for single-cell transcriptomics analysis. bioRxiv pp. 2023–04 (2023)
45. Samokhin, A.O., Lythgo, P.A., Gauthier, J.Y., Percival, M.D., Brömme, D.: Pharmacological inhibition of cathepsin s decreases atherosclerotic lesions in apoe-/-mice. Journal of cardiovascular pharmacology **56**(1), 98–105 (2010)
46. Sanchez-Castillo, M., Blanco, D., Tienda-Luna, I.M., Carrion, M., Huang, Y.: A bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. Bioinformatics **34**(6), 964–970 (2018)
47. Schaffter, T., Marbach, D., Floreano, D.: Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics **27**(16), 2263–2270 (2011)
48. Shu, H., Zhou, J., Lian, Q., Li, H., Zhao, D., Zeng, J., Ma, J.: Modeling gene regulatory networks using neural network architectures. Nature Computational Science **1**(7), 491–501 (2021)
49. Sierksma, A., Lu, A., Mancuso, R., Fattorelli, N., Thrupp, N., Salta, E., Zoco, J., Blum, D., e, L., De Strooper, B., Fiers, M.: but not to TAU pathology. EMBO Mol Med **12**(3), e10606 (Mar 2020)
50. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
51. Song, Y., Durkan, C., Murray, I., Ermon, S.: Maximum likelihood training of score-based diffusion models. Advances in Neural Information Processing Systems **34**, 1415–1428 (2021)
52. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
53. Szklarczyk, D., Gable, A.L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N.T., Morris, J.H., Bork, P., et al.: String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic acids research **47**(D1), D607–D613 (2019)
54. Tang, W., Liu, R., Wen, H., Dai, X., Ding, J., Li, H., Fan, W., Xie, Y., Tang, J.: A general single-cell analysis framework via conditional diffusion generative models. bioRxiv pp. 2023–10 (2023)

55. Tiribuzi, R., Orlacchio, A., Crispoltoni, L., Maiotti, M., Zampolini, M., De Angeliz, M., Mecocci, P., Cecchetti, R., Bernardi, G., Datti, A., et al.: Lysosomal β-galactosidase and β-hexosaminidase activities correlate with clinical stages of dementia associated with alzheimer's disease and type 2 diabetes mellitus. Journal of Alzheimer's Disease **24**(4), 785–797 (2011)
56. Van Dam, S., Vosa, U., van der Graaf, A., Franke, L., de Magalhaes, J.P.: Gene co-expression analysis for functional classification and gene–disease predictions. Briefings in bioinformatics **19**(4), 575–592 (2018)
57. Wolfe, C.M., Fitz, N.F., Nam, K.N., Lefterov, I., Koldamova, R.: The role of apoe and trem2 in alzheimer's disease—current understanding and perspectives. International journal of molecular sciences **20**(1), 81 (2018)
58. Zalocusky, K.A., Najm, R., Taubes, A.L., Hao, Y., Yoon, S.Y., Koutsodendris, N., Nelson, M.R., Rao, A., Bennett, D.A., Bant, J., et al.: Neuronal apoe upregulates mhc-i expression to drive selective neurodegeneration in alzheimer's disease. Nature neuroscience **24**(6), 786–798 (2021)
59. Zhou, W., Scott, S., Shelton, S., Crutcher, K.: Cathepsin d-mediated proteolysis of apolipoprotein e: possible role in alzheimer's disease. Neuroscience **143**(3), 689–701 (2006)
60. Zhu, H., Slonim, D.: Improving gene regulatory network inference using dropout augmentation. bioRxiv pp. 2023–01 (2023)