Relational Classification of Biological Cells in Microscopy Images

Ping Liu and **Mustafa Bilgic**
Illinois Institute of Technology, Chicago, IL
pliu19@hawk.iit.edu, mbilgic@iit.edu

Abstract

We investigate the relational classification of biological cells in 2D microscopy images. Rather than treating each cell image independently, we investigate whether and how the neighborhood information of a cell can be informative for its prediction. We propose a Relational Long Short-Term Memory (R-LSTM) algorithm, coupled with auto-encoders and convolutional neural networks, that can learn from both annotated and unlabeled microscopy images and that can utilize both the local and neighborhood information to perform an improved classification of biological cells. Experimental results on both synthetic and real datasets show that R-LSTM performs comparable to or better than six baselines.

Introduction

Signaling variability of individual cells results in different responses when they are exposed to external stimuli such as proteins, drugs, and viruses. Single-cell analyses are carried out to understand cell heterogeneity (Shipp et al. 2002). Studying the cellular heterogeneity can help with understanding drug response (Sheng, Li, and Wong 2015), drug tolerance (Lee et al. 2014), tumor necrosis (Obraztsov et al. 2019), viral infection (Drayman et al. 2017), and more.

One approach to single-cell analysis is to use imaging. For example, cancer tissues can be stained, histology images can be generated, cell nuclei can be detected, and cell types can be manually annotated, as can be seen in Figure 1. Another example is studying viral infections where viruses are introduced to a sample and protein dynamics are monitored under a microscope, taking thousands of images during the process (Drayman et al. 2017).

Through high-content screening (Usaj et al. 2016), it is possible to generate thousands of images per day for studying cell biology. While it is possible to detect cell nuclei (Shifat-E-Rabbi et al. 2020), labeling the cells is a costly process, where experiments need to be carried out and the cells need to be labeled manually. Therefore, even though unlabeled data is abundant, labeled data is often scarce.

Machine learning models have been employed for analyses of these biological images. One simple modeling approach is to treat each cell independently. In this case, the cell nuclei are detected and a rectangular area around the nucleus can be cropped to represent the cells. One approach to cell label prediction is to extract features, such as SIFT, from the cell images and use vector-based machine learning models such as logistic regression and support vector machines (Loo, Wu, and Altschuler 2007). Recently, deep learning models have also been employed for cell classification (Kraus, Ba, and Frey 2016; Sirinukunwattana et al. 2016).

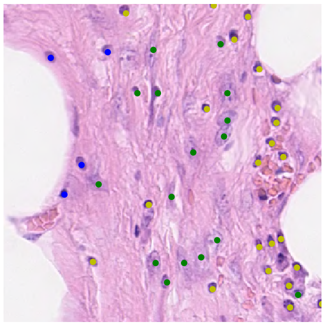

Figure 1: Examples microscopy images of cells. This figure is a cancer histology image from (Sirinukunwattana et al. 2016), where each cell is classified into four categories: epithelial, inflammatory, fibroblast, and miscellaneous.

An alternative approach is to study and model the cells and their neighborhood(Sirinukunwattana et al. 2016). This approach can be informative due to several reasons including i) cells often communicate with their neighborhood, ii) some stimuli diffuse locally in a neighborhood, iii) viral and bacterial infections can affect a local neighborhood, and iv) the labels of the cells within a local neighborhood might be correlated (e.g., tumor cells tend to be neighbors with other tumor cells.).

In this paper, we study relational classification of biological cells in 2D microscopy images. Rather than treating the classification of each cell as independent of its neighborhood, we develop a relational deep learning model that is able to utilize both the cell's own image as well as its neighborhood to improve the classification performance. Our contributions in this paper include:

- A relational Long Short-Term Memory (LSTM) model for classification of biological cells in microscopy images that can take into account both local and neighborhood information.

- A synthetic data generator to simulate various settings where informativeness of the local versus neighborhood information is controlled.
- A stratified cross-validation algorithm for object classification in images.
- Development and analysis of three non-relational and three relational baselines.

The rest of the paper is organized as follows. We first discuss related work in Section . In Section , we describe the proposed relational-LSTM model. We explain the synthetic data generator in Section . Section includes the discussion of the experimental methodology and the results. We conclude in Section .

Related Work

One of our main contributions in this paper is the relational classification of the cells. A related area is learning with graphs (Perozzi, Al-Rfou, and Skiena 2014; Jin et al. 2019; Aguinaga, Chiang, and Weninger 2019; Abu-El-Haija et al. 2018) where the data typically consists of vertices and edges. The graph can be heterogeneous where vertices and edges belong to different types. The vertices and the edges optionally have features that describe them. Typical classification tasks include node prediction, link prediction, and network prediction. In many domains, the links are often defined explicitly through an adjacency matrix. In some domains, however, the links are implicitly defined or mined from data (e.g., through a neighborhood similarity).

The most closely-related work is the work on relational and collective classification (Neville and Jensen 2003; Sen et al. 2008; Fakhraei et al. 2015). In relational classification, the label of an object is determined based on its own features as well as its neighbors' features (Preisach and Schmidt-Thieme 2006; Zhu et al. 2017; Taskar, Segal, and Koller 2001). In collective classification, the labels of a set of related objects are predicted jointly as opposed to each one predicted independently. The iterative classification algorithm (Neville and Jensen 2003) first bootstraps the labels of the nodes of a graph using local information, and then predicts the labels of the nodes based on its features and the predicted labels of its neighbors. Recently, deep learning approaches, such as Long Short-Term Memory (LSTM) models, have been utilized for collective classification (e.g., (Moore and Neville 2017; Fan and Huang 2018). In this case, the LSTM model treats a node and its unordered neighbors as a sequence, and uses both local features and the predicted labels as an input. This is most similar to our work, except, in our case the neighborhood structure is not explicitly defined, the number of neighbors is fixed, there is a large amount of unlabeled data, and the input is an image that contains numerous cells and empty locations.

Another related area is node embedding where nodes are represented through a low-dimensional embedding that is learned based on the node itself and its neighbors. An example approach is Graph Convolutional Networks (Kipf and Welling 2016) that learn convolutional neural networks on the graph directly for node prediction. Similar work include (Grover and Leskovec 2016; Hamilton, Ying, and Leskovec 2017; Perozzi, Al-Rfou, and Skiena 2014; Jin et al. 2019; Abu-El-Haija et al. 2018) where the embedding representation for each node is obtained by optimizing a customized cost function. Such methods are able to take care of the varying-size neighborhoods typically using sampling methods such as random walks.

Relational classification via feature extraction and selection has been studied for biological experiments. For example, Snijder et al. (2009) discover the impact of population context on the prediction of virus infection using bootstrapped Bayesian networks. Toth et al. (2018) propose an approach to extract cell-based and neighbourhood features from segmentation of cells, where the neighbor features are aggregated by using K-nearest neighbours and the N-distance methods, and a classifier, such as Support Vector Machines and Random Forests, are trained and evaluated on the combined set of features. These approaches, however, require significant amount of feature engineering. In the mean time, Kraus, Ba, and Frey (2016) design and combine convolutional neural network with multiple instance learning to classify and segment microscopy images with only annotations in image level. Beck et al. (2011) construct objects classifier for superpixels in cancer images using contextual and relational features.

Lastly, a related area is object detection and segmentation (e.g., (Long, Shelhamer, and Darrell 2015; Girshick et al. 2014; Girshick 2015; Ren et al. 2015; He et al. 2017; Xu et al. 2017; Yang et al. 2018). The main differences between our work and this line of work are that in our case the type of objects is only one type (i.e. biological cells), we are interested in predicting a feature of the object, and the relationship between cells is defined via a local proximity.

Our Approach: Relational-LSTM

Let $\mathbf{I}$ represent the set of all microscopy images, where each microscopy image $\mathbf{I}_i \in \mathbf{I}$ represents the microscopy image of a biological sample that contain several (often 50 to 200) cells $c_i^j \in \mathbf{I}_i$. The cells are located at arbitrary locations in the image. A typical approach to represent each cell is to crop a rectangular area centered around its nucleus, after its nucleus has been automatically detected (Shifat-E-Rabbi et al. 2020). Thus, we assume that c_i^j are images within the larger image $\mathbf{I}_i$.

The labels of the cells are known for a subset of biological samples, $\mathbf{L} \subset \mathbf{I}$. Each $\mathbf{L}_l \in \mathbf{L}$ represents a biological sample where the labels of c_l^j are known. The labels are typically obtained through experimentation, imaging, and human annotation. Let $\mathbf{Y_L}$ represent the set of all known labels, $\mathbf{Y}_l$ represent the set of labels of all cells in the annotated biological sample $\mathbf{L}_l$, and y_l^k represent the label of the individual cell c_l^k. For the remaining subset of samples, $\mathbf{U} = \mathbf{I} \backslash \mathbf{L}$, only the images of the samples are taken but they are not annotated. That is, the images of the cells, c_u^k, are given and the labels, y_u^k, are unknown. These represent several biological samples that have been imaged but not annotated.

The objective is to train an accurate predictive model M using all images $\mathbf{I}$, which include both annotated samples $\mathbf{L}$ and the labels of the cells in those samples $\mathbf{Y_L}$, *and* the unlabeled $\mathbf{U}$ samples. We take a probabilistic approach and aim to maximize conditional likelihood of the known labels.

$$\arg\max_{M} P(\mathbf{Y_L} \mid \mathbf{L}, \mathbf{U}, M)$$

Because each microscopy image $\mathbf{I}_i \in \mathbf{I}$ represents an independent sample, $P(\mathbf{Y_L} \mid \mathbf{L}, \mathbf{U}, M)$ factorizes over individual images:

$$\arg\max_{M} \prod_{\mathbf{L}_l \in \mathbf{L}} P(\mathbf{Y}_l \mid \mathbf{L}, \mathbf{U}, M) \tag{1}$$

Equation 1 optimizes the joint distribution of labels. Most collective classification approaches optimize this objective function. A model that assumes the individual cells are independent and identically distributed (i.i.d.) but uses all information as input would further factorize the product to individual cell labels:

$$\arg\max_{M} \prod_{\mathbf{L}_l \in \mathbf{L}} \prod_{c_l^k \in \mathbf{L}_l} P(y_l^k \mid \mathbf{L}, \mathbf{U}, M) \tag{2}$$

Relational models typically uses the objective function in Equation 2 where the target variables are independently optimized, but the input from neighbors is still used as input. A model that further assumes the local information is sufficient and the neighborhood information is irrelevant would ignore the neighbors and instead optimize:

$$\arg\max_{M} \prod_{\mathbf{L}_l \in \mathbf{L}} \prod_{c_l^k \in \mathbf{L}_l} P(y_l^k \mid c_l^k, M) \tag{3}$$

Equation 3 refers to the traditional classification approaches where each object is classified using only its own features.

In this paper, we propose a relational classification approach (Getoor and Taskar 2007) and hence solve the optimization of Equation 2. In collective, relational, and neighborhood-based approaches, it is typically assumed that the relational information is local (Sen et al. 2008; Grover and Leskovec 2016). For example, in a social network, one's close neighborhood is assumed to be more informative than it is distant neighborhood. Similarly, we assume that the relationship between the biological cells is local: cells communicate with their immediate neighbors and a cells within a local vicinity might have correlated features and labels[1].

Because each cell c^k is represented as a rectangle image, we define its immediate neighborhood as rectangles in its immediate vicinity. Please see Figure 2 as an illustration of the neighborhood of the target cell.

Using the labeled data $\mathbf{L}$ and each cell c_l^k as the target, we optimize Equation 2 using a Long Short-Term Memory (LSTM) architecture, which is illustrated in Figure 3. The first eight inputs are the images of the eight neighbors and the last input is the image of the target cell. The output of the *relational-LSTM* (R-LSTM) is the label of the target cell. The input images are processed through an autoencoder convolutional neural network (AE-CNN), which is initialized using the unlabeled data $\mathbf{U}$.

Neighbor 1 *Cell*	Neighbor 2 *Empty*	Neighbor 3 *Cell*
Neighbor 8 *Empty*	**Target Cell**	Neighbor 4 *Empty*
Neighbor 7 *Cell*	Neighbor 6 *Cell*	Neighbor 5 *Empty*

Figure 2: The neighborhood of a target cell. Each location is represented as a rectangle image. The target cell is in the center. The target cell has eight neighbors some of which can be empty.

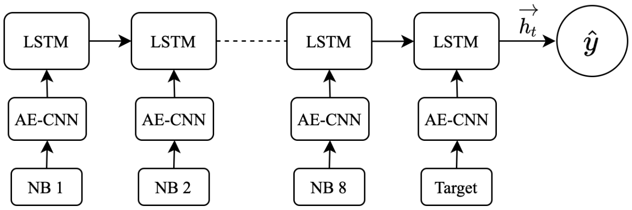

Figure 3: The relational-LSTM (R-LSTM) architecture. The images of the eight neighbors and the target cell, as shown in Figure 2, is treated as sequence of inputs. The AE-CNN is a convolutional encoder, which is initialized using the weights of an auto-encoder trained on unlabeled data $\mathbf{U}$ using unsupervised learning. The target of the R-LSTM is the label of the target cell.

Note that both during training and testing, the neighbor locations that do not contain cells are still used as input to the R-LSTM. The main motivation for such a choice is that even though a neighboring location might not contain a cell, the image of the neighborhood can still contain valuable information for biological reasons. For example, in tissue regeneration a biological cell that is far from blood vessels enters into a distress mode due to lack of oxygen and nutrients, and starts producing and releasing a distress signal protein (vascular endothelial growth factor) into its environment (Ferrara et al. 1992), and hence, both the image of the cell and the image of its vicinity are informative for predicting the cell's state.

Synthetic Data Generator

We first develop a synthetic data generator to simulate various settings where we can experiment with i) how informa-

[1] We do not require the cells in a local neighborhood to carry correlated information. Our model learns if such a correlation exists, learns the strength and the sign of the correlation, and utilizes it to improve prediction performance.

Algorithm 1 Synthetic data generation

Input: N – the number of images that will be generated; G – the size of the grid; p_e – probability of a grid being empty; p_a – probability that a non-empty grid contains an active cell; ac_l, ac_h, in_l, in_h – the lower and upper bounds for the intensities of the active and inactive cells; λ – parameter controlling how informative the neighbors are
Output: $\mathbf{I}$ – a set of synthetic images each of which contains multiple cells; $\mathbf{Y_I}$ – the labels of the cells

```
Let $\mathbf{I} = \{\}$.
while $|\mathbf{I}| < N$ do
    create a $G \times G$-grid image $\mathbf{I}_i$
    for entry in grid do
        if random.roll $< p_e$ then
            Leave it empty
        else if random.roll $< p_a$ then
            $y = active$
            Sample an intensity uniformly from $[ac_l, ac_h]$
        else
            $y = inactive$
            Sample an intensity object uniformly from $[in_l, in_h]$
        $\mathbf{Y}_{I_i}$.add($y$)
    $\mathbf{I}$.add($\mathbf{I}_i$)
    $\mathbf{Y_I}$.add($\mathbf{Y}_{I_i}$)
Train a binary Gaussian naïve Bayes $M$ on cells and their labels
for $\mathbf{I}_i$ in $\mathbf{I}$ do
    for $c_i^k \in \mathbf{I}_i$ do
        $probas = M$.predict($c_i^k$)
        $conf = \max(probas)$
        $d = \log(1/conf)/10 + \lambda$
        if random.roll $> conf - d$ then
            $c_i^k$'s label $y_i^k$ is voted by non-empty neighbors' probabilistic labels and updated accordingly
return $\mathbf{I}$ and $\mathbf{Y_I}$
```

tive each individual cell features are, ii) how informative the neighborhood information is, iii) the label distribution, and iv) the distribution of the cells in the images. The algorithm is described in Algorithm 1, which generates gray-scale images that contain cells whose labels are binary indicating whether they are activated by external stimuli.

The algorithm generates N images. Each image $\mathbf{I}_i$ is a GxG grid. Whether each grid is empty or occupied by a cell is decided by a random roll controlled by the p_e parameter, where p_e represents probability of being "empty." This parameter controls the cell density of the image. If a grid is occupied by a cell, we first sample the cell's label (active or inactive) using the parameter p_a, where p_a represents probability of being "active." Active cells' intensities are sampled from $[ac_l, ac_h]$ whereas inactive cells' intensities are sampled from $[in_l, in_h]$. The potential overlap between inactive and active cells' intensities control how much informative the individual cells' own intensities are.

Next, the algorithm decides how informative the cells' neighborhood is. The neighborhood plays a bigger role for cells whose local information is less discriminative. That is, if a cell's local information (in this case, the image intensity) is not very informative about the cell's label, then its neighborhood has a higher chance to play a more important role in the classification process. To simulate this behavior, we first learn a local model that predicts the cell's label using only its local image. Because we generated image intensities for active and inactive cells, the synthetic generator fits a Gaussian naïve Bayes (GNB) model whose input is the image intensity and the output is whether the cell is "active." Then, it uses GNB to predict each cell's label probabilistically and if the prediction is not confident enough, controlled by the λ parameter, the cell's label is determined by its neighborhood: the cell's neighbors vote what the target cell's label should be set to.

In summary, p_e controls the cell density of the microscopy image, p_a controls the likelihood that a cell will react to external stimuli, the overlap between the intensity parameters (ac_l, ac_h, in_l, in_h) controls the informativeness of the local information, and λ controls when neighborhood will play a role in determining the final label of the cell.

Stratified Cross Validation

A typical evaluation approach for classification is to use stratified cross validation where the data is divided into folds where each fold has similar label distribution (Tsamardinos, Greasidou, and Borboudakis 2018; Tibshirani and Tibshirani 2009). Achieving similar label distributions in each fold is straightforward when each cell can be assigned to any fold. However, because each microscopy image $\mathbf{I}_i$ corresponds to a different laboratory experiment, it is more appropriate to create the folds at the image level rather than at the cell level. Hence, when an image $\mathbf{I}_i$ is assigned to fold k, all cells $c_i^j \in \mathbf{I}_i$ are assigned to that fold. However, because each image $\mathbf{I}_i$ varies in the number of cells that they contain and the label distribution of those cells can vary from image to image, splitting the entire data into stratified folds is not trivial. We therefore introduce a stratified fold creation algorithm (Algorithm 2) for microscopy images.

Algorithm 2 describes our proposed approach for creating k folds where each fold has similar label distribution to the overall label distribution in the data. The algorithm works as follows: first, the overall label distribution in the data, O, is calculated. Then, k empty sets are created to hold each of the k folds. Finally, for each random image that has not been yet assigned a fold, the most appropriate fold is assigned by finding the fold that would result in the minimum Kullback-Leibler (KL) divergence between the overall distribution O and the candidate fold, if the image was added to that fold. This process is repeated until all images are assigned a fold.

Because this algorithm is a greedy algorithm, it is not guaranteed to result in the optimal split where each fold would have the closest label distribution to the overall label distribution. Thus, this entire fold generation process is repeated m times with a different random seed and the fold assignment that has the minimum of the maximum (i.e., min-max) KL divergence from the target distribution is returned.

Algorithm 2 Stratified fold generation

Input: O – the overall class distribution; k – the number of folds ; $\mathbf{L}$ – a set of labeled images; m – the number of trials
Output: $folds$ – a splitting sample that has the lowest KL-Divergence score with O

```
Let $R = \{\}$
while $|R| < m$ do
    Let $s$ be a list of sets, where $s^j = \varnothing$, for $0 \leq j < k$
    Let $f^j$ be the class distribution of $s^j$, for $0 \leq j < k$
    while $\mathbf{L} \neq \varnothing$ do
        Pick a random $\mathbf{L}_i \in \mathbf{L}$
        $z = \arg\min_{j} KL(O, f^j)$ if $s^j = s^j \cup \{\mathbf{L}_i\}$
        $s^z = s^z \cup \{\mathbf{L}_i\}$
        $\mathbf{L} = \mathbf{L} \setminus \{\mathbf{L}_i\}$
    $R$.add($s$)
$folds = \arg\min_{s \in R} \max_{0 \leq j < k} KL(O, f^j)$
return $folds$
```

Prior work on stratification of multi-label data uses similar greedy idea (Sechidis, Tsoumakas, and Vlahavas 2011). We adopt minimizing KL divergence on the label distribution, whereas the prior work greedily optimizes the desired number of labels for each set.

Once a stratified fold is generated, standard train-validate-test process is applied where $k-2$ folds are used for training, one fold is used for validation, and one fold is used for testing. This process is repeated k times where each fold gets to be the validation set exactly once and the test set exactly once, and average performance over the test sets is reported.

Experimental Methodology and Results

In this section, we describe the datasets, the baselines, the experimental methodology, the performance metrics, and the results. The code to replicate the experimental results in this paper is available at https://github.com/IIT-ML/AAAI21-relational-cell-classification.

Datasets

We experimented with several synthetic datasets, generated using Algorithm 1, and three real-world datasets.

Synthetic Data We experimented with three parameter configurations to test how models that ignore or utilize the neighborhood information would perform under various conditions. The parameter configurations are shown in Table 1. The first four parameters, N, G, p_e, and p_a are held constant for all three synthetic datasets. $N = 40$, resulting in 40 images for each dataset, each of which is a $G \times G = 25 \times 25$ grid. $p_e = 0.5$, resulting in approximately half of the grids in each image to be empty. $p_a = 0.5$, resulting in an approximately equal number of active and inactive cells. We varied the last three parameters to generate datasets with varying degree of importance of the local versus neighborhood information.

Parameter	Setting 1 (S1)	Setting 2 (S2)	Setting 3 (S3)
N	40	40	40
G	25	25	25
p_e	0.5	0.5	0.5
p_a	0.5	0.5	0.5
$[in_l, in_h]$	$[0.3, 0.8]$	$[0.3, 0.8]$	$[0.3, \mathbf{0.7}]$
$[ac_l, ac_h]$	$[0.5, 1.0]$	$[0.5, 1.0]$	$[\mathbf{0.6}, 1.0]$
λ	0.2	$\mathbf{0.4}$	0.2

Table 1: Parameter settings for synthetic datasets. To make it easy to spot the differences, the parameters that are different in one setting compared to the rest are highlighted in bold.

Local features: We used the amount of overlap between the intensities of the "active" and "inactive" cells to control how much a local classifier can be confident (and accurate) in its predictions. The overlap of the image intensities in S1 and S2 are the same ($[0.5, 0.8]$) whereas the overlap in S3 is smaller ($[0.6, 0.7]$). Hence, a local classifier is expected to be equally confident (and accurate) in S1 and S2, and more confident (and accurate) in S3.
Neighborhood: We used the λ parameter to control when a cell's neighbors can intervene and vote on its label. A higher λ means the local classifier needs to be more confident to prevent neighbors from determining the cell's label. Comparing S1 and S2 where the local classifier is expected to be equally confident, a higher λ for S2 means neighbors will intervene more in S2 than in S1. Comparing S1 and S3, λ is equal for both but because the local classifier is expected to be more confident in S3, the neighbors will intervene more in S1 than in S3. Comparison of S2 and S3 is now straightforward: the neighbors will intervene more in S2 than in S3.

Histology Images of Colorectal Cancer (CRC) This dataset is introduced by Sirinukunwattana et al. (2016) and it contains 100 H&E stained histology images of colorectal cancer (colorectal adenocarcinomas). All images have the same size of 500×500 pixels. The cancer cells are labeled as Epithelial, Inflammatory, Fibroblast, or Miscellaneous, and the number of cells in these categories are 5745, 4942, 4025, and 1440 respectively. One example image is shown in Figure 1 in Section .

Human MCF7 Breast Cancer Cells (MCF-7) MCF-7 is a public image set that was collected and labeled using a typical set of morphological labels and a relevant p53-wildtype breast-cancer model system (MCF-7) provided by Broad Bioimage Benchmark Collection[2]. The microscopy images are recorded in 24 hours with a collection of 113 small molecules. For our experiments, we follow the single-cell annotation presented and provided by Piccinini et al. (2017); Toth et al. (2018). We classify cells as "debris" versus "non-debris", and the corresponding number of cells are 250 versus 1137.

Urinary Bladder Cancer Tissue Sections (UBC) UBC image dataset (Toth et al. 2018) is a collection of mi-

[2] https://data.broadinstitute.org/bbbc/BBBC021/

croscopy images that record urinary bladder cancer tissue. The histopathologic process was applied to generate Hematoxylin-Eosin (HE) staining of slides of the urinary cancer tissue. This dataset contains 24 images and a total of $1,494$ cells. We classified the cells as "cancer" versus "non-cancer." There are 178 cancer cells and $1,316$ non-cancer cells annotated in the UBC dataset.

In practice, a grid can contain more than one cell. For three real datasets, we the analyzed the mean number of cells in each of the non-empty locations. On average CRC has 1.76 cells and MCF-7 and UBC have 1.0 cell on non-empty locations. Our method does not explicitly need each grid to have at most one cell. It is possible, and indeed expected that, if a grid has multiple cells, the image could contain stronger signals (if cells have similar features) or a mixed signal (if cells have varying features).

In all of the following experiments, we treat half of the dataset as the labeled set $\mathbf{L}$ and the remaining half as the unlabeled set $\mathbf{U}$. Rather than randomly assigning the individual cells c_i^J to labeled and unlabeled sets, we first assign the full microscopy images $\mathbf{I}_i$ into the annotated and not-annotated sets. We then treat the cells in the annotated images as labeled and the cells in the not-annotated images as unlabeled. The choice of splitting the dataset into labeled and unlabeled sets based on images rather than individual cells is aligned with experimental practice: several microscopy images can be taken before the introduction of the stimuli and such images are not annotated. Annotation requires one to introduce the stimuli, observe the cells, and annotate them as active or inactive. Hence, whether a cell is labeled or not depends on whether the microscopy image that contains that cell is annotated or not.

Baselines

The following baselines treat each cell independently, ignoring the relational information, and simply use the cropped images of the cells as input.

Support Vector Machines (SVM) We train SVMs with individual cells' cropped images as input and their labels as output. We use SVM with an 'rbf' kernel and use grid search for optimizing the complexity 'C' (for regularization) and the 'gamma' (for kernel) parameters using the validation set.

CNN The setup for this model is the same as SVM, except a convolutional neural netwok (CNN) is used as the supervised model. We use the same network structure used by the CRC dataset paper (Sirinukunwattana et al. 2016): two convolutional layers with RELU activations followed by a max-pooling operation, which is then followed by two dense layers and a final softmax function.

AE-CNN This is the same model as the CNN model above, except the convolutional layers of the CNN is initialized as follows: an auto-encoder (AE) with the same convolutional layer structure, followed by deconvolutional layers, is trained on the unlabeled set of images $\mathbf{U}$. The convolutional layers of AE-CNN are then initialized with the weights of the convolutional layers of the AE. AE-CNN is then trained, validated, and tested using the stratified cross-validation on the labeled set.

	Models	S1	S2	S3
Non-relational	SVM	.647 (.00)	.597 (.00)	.778 (.00)
	CNN	.648 (.00)	.595 (.00)	.780 (.00)
	AE-CNN	.655 (.00)	.596 (.00)	.782 (.00)
Relational	AE-SVM-F	.766 (.00)	.800 (.00)	.828 (.00)
	AE-SVM-A	.778 (.00)	.826 (.36)	.825 (.00)
	LCNN	.797 (.07)	.828 (.44)	.836 (.00)
	R-LSTM	.802 (–)	.828 (–)	.849 (–)

Table 2: Accuracy and p-value comparisons for the synthetic datasets.

The following baselines utilize both the target cell and its neighborhood for training, validation, and testing.

AE-SVM-F We apply the AE's encoder portion to encode the target cell's image and its eight neighbor images, and then concatenate the nine embedding vectors. We train an SVM where the input is the concatenated vectors and the output is the target cell's label.

AE-SVM-A This is similar to AE-SVM-F, except the embedding vectors of the neighbors are aggregated through averaging the eight neighbor vectors, and then the embedding of the target cell is concatenated to the average neighbor vector.

LCNN This model is the larger version of the CNN model above, except instead of just the target cell, a larger image that contains the target cell and its eight neighbors is used as a single image and fed to LCNN.

For the deep learning models (AE, CNN, LCNN, R-LSTM) we used PyTorch (Paszke et al. 2017). For SVM, we used scikit-learn (Pedregosa et al. 2011).

Performance Metrics

We perform 10-fold stratified cross validation, where eight folds are used for training, one fold for validation, and one fold for testing. Average performance is reported. For synthetic datasets, we report accuracy because the class distribution is approximately even. For the three real datasets, we report accuracy, macro-F1, and weighted-F1. We also report one-tailed paired t-test results comparing R-LSTM to the six baselines. The pairings are done through the test folds. The p values are listed in parentheses after each metric.

Results and Discussion

The performance metrics and p values of the the t-tests are presented in Tables 2 through 5. Each entry has two numbers: the first number is the average performance and the second number, shown in parentheses, is the p-value of the t-test comparing R-LSTM to the respective baseline.

Synthetic Data In the synthetic datasets, the local feature structure is simple: cells are represented as circles with intensities sampled from an interval corresponding to their label. Therefore, the non-relational models (SVM, CNN, and

	Models	Accu (p)	M-F1 (p)	W-F1 (p)
Non-relational	SVM	.697 (.00)	.588 (.01)	.697 (.00)
	CNN	.730 (.08)	.619 (.16)	.727 (.05)
	AE-CNN	.722 (.04)	.618 (.14)	.723 (.04)
Relational	AE-SVM-F	.730 (.13)	.610 (.06)	.724 (.05)
	AE-SVM-A	.693 (.00)	.604 (.05)	.707 (.00)
	LCNN	.721 (.05)	.605 (.02)	.721 (.03)
	R-LSTM	.745 (–)	.634 (–)	.743 (–)

Table 3: Accuracy, Macro-F1, Weighted-F1, and p-value comparisons for the CRC dataset.

AE-CNN) have similar results. Comparing S1, S2, and S3 results for the non-relational models, S3 results in the highest accuracy as expected, because active and inactive cells have the lowest intensity overlap. Comparing S1 and S2, even though the intensity overlap is the same in both S1 and S2, the neighbors intervene more in S2 and hence the non-relational accuracy is lower for S2.

Comparing the relational models to the non-relational ones, the relational ones outperform the non-relational ones in all settings, as expected. Notable results are as follows. **i)** Even though non-relational accuracy is lower in S2 than in S1, the relational accuracy is higher in S2 than in S1, because neighbors are more informative in S2 than in S1. **ii)** The SVM method that aggregates the neighborhood information, AE-SVM-A, performs comparable to or better than the SVM that concatenates the neighborhood information, AE-SVM-F. This result is expected because the order of neighbors is not important in the synthetic data generation process. **iii)** LCNN performs comparable to or better than the SVM approaches. **iv)** R-LSTM performs comparable to or better than all baselines. The most competitive baseline is LCNN as expected; R-LSTM has comparable performance to LCNN in S1 and S2, and R-LSTM outperforms LCNN in S3, as the p values in the parentheses show.

Histology Images of Colorectal Cancer (CRC) The accuracy, macro-F1 (M-F1), weighted-F1 (W-F1), and p values of the the t-tests for the CRC dataset are presented in Table 3. Unlike the synthetic data where the local cell information was simple, SVM performs worse than CNN for the CRC data. Similarly, the comparison of relational versus non-relational for the CRC data is not as clear cut as the synthetic data. For example, LCNN performs a little worse than CNN, most likely due to insufficiency of the size of the training data and the complexity of the cell neighborhood. R-LSTM, in contrast, performs comparable to or better than all baselines (both relational and non-relational) for all three performance metrics.

Human MCF7 Breast Cancer Cells (MCF-7) The experimental results of MCF-7 are reported in Table 4. MCF-7 has fewer data points than synthetic and the CRC datasets. In general, relational models perform better than non-relational models in all three metrics. AE-SVM-F is better than all other models except for R-LSTM. R-LSTM achieves the highest results for all three metrics.

	Models	Accu (p)	M-F1 (p)	W-F1 (p)
Non-relational	SVM	.836 (.00)	.685 (.00)	.825 (.00)
	CNN	.869 (.07)	.724 (.09)	.852 (.09)
	AE-CNN	.866 (.06)	.716 (.08)	.848 (.07)
Relational	AE-SVM-F	.877 (.25)	.730 (.07)	.856 (.09)
	AE-SVM-A	.854 (.08)	.720 (.08)	.843 (.07)
	LCNN	.859 (.00)	.734 (.09)	.850 (.02)
	R-LSTM	.885 (–)	.770 (–)	.873 (–)

Table 4: Accuracy, Macro-F1, Weighted-F1, and p-value comparisons for the MCF-7 dataset.

	Models	Accu (p)	M-F1 (p)	W-F1 (p)
Non-relational	SVM	.968 (.23)	.873 (.18)	.977 (.39)
	CNN	.952 (.02)	.832 (.06)	.953 (.03)
	AE-CNN	.956 (.08)	.853 (.10)	.962 (.05)
Relational	AE-SVM-F	.942 (.06)	.806 (.06)	.958 (.06)
	AE-SVM-A	.936 (.17)	.814 (.09)	.956 (.22)
	LCNN	.930 (.05)	.809 (.13)	.923 (.09)
	R-LSTM	.980 (–)	.925 (–)	.980 (–)

Table 5: Accuracy, Macro-F1, Weighted-F1, and p-value comparisons for the UBC dataset.

Urinary Bladder Cancer Tissue Sections (UBC) Table 5 shows the results for UBC. Unlike the previous two datasets, all non-relational models have better results than AE-SVM-F, AE-SVM-A, and LCNN. R-LSTM outperforms all other models in a large margin in all three measures. The reason may lie in the fact that many cells in UBC are located on the boundaries among different kinds of tissues. Thus neighbors' often contain several different kinds of cells. R-LSTM was able to handle the cell diversity better than other relational model baselines.

Conclusions

We investigated the relational classification of biological cells in microscopy images. We proposed a relational LSTM (R-LSTM) model that take both the target cell's image and neighboring images into account to make a prediction on the target cell. We also proposed a synthetic data generator and an algorithm for stratified cross-validation on microscopy images. The experimental results on three synthetic datasets and three real datasets showed that R-LSTM performed comparable to or better than both non-relational and relational baselines.

Acknowledgments

This material is based upon work supported by the National Science Foundation under CAREER grant No. 1350337.

References

Abu-El-Haija, S.; Perozzi, B.; Al-Rfou, R.; and Alemi, A. A. 2018. Watch Your Step: Learning Node Embeddings via Graph Attention. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 9180–9190. Curran Associates, Inc. URL http://papers.nips.cc/paper/8131-watch-your-step-learning-node-embeddings-via-graph-attention.pdf.

Aguinaga, S.; Chiang, D.; and Weninger, T. 2019. Learning Hyperedge Replacement Grammars for Graph Generation. *IEEE transactions on pattern analysis and machine intelligence* 41(3): 625–638.

Beck, A. H.; Sangoi, A. R.; Leung, S.; Marinelli, R. J.; Nielsen, T. O.; Van De Vijver, M. J.; West, R. B.; Van De Rijn, M.; and Koller, D. 2011. Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. *Science translational medicine* 3(108): 108ra113–108ra113.

Drayman, N.; Karin, O.; Mayo, A.; Danon, T.; Shapira, L.; Rafael, D.; Zimmer, A.; Bren, A.; Kobiler, O.; and Alon, U. 2017. Dynamic proteomics of herpes simplex virus infection. *MBio* 8(6).

Fakhraei, S.; Foulds, J.; Shashanka, M.; and Getoor, L. 2015. Collective spammer detection in evolving multi-relational social networks. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, 1769–1778. ACM.

Fan, S.; and Huang, B. 2018. Recurrent collective classification. *Knowledge and Information Systems* ISSN 0219-3116. doi:10.1007/s10115-018-1260-4. URL https://doi.org/10.1007/s10115-018-1260-4.

Ferrara, N.; Houck, K.; Jakeman, L.; and Leung, D. W. 1992. Molecular and biological properties of the vascular endothelial growth factor family of proteins. *Endocrine reviews* 13: 18–32.

Getoor, L.; and Taskar, B. 2007. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN 0262072882.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864. ACM.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980–2988. IEEE.

Jin, D.; Heimann, M.; Rossi, R.; and Koutra, D. 2019. node2bits: Compact Time-and Attribute-aware Node Representations for User Stitching. *arXiv preprint arXiv:1904.08572* .

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Kraus, O. Z.; Ba, J. L.; and Frey, B. J. 2016. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics* 32(12): i52–i59.

Lee, M.-C. W.; Lopez-Diaz, F. J.; Khan, S. Y.; Tariq, M. A.; Dayn, Y.; Vaske, C. J.; Radenbaugh, A. J.; Kim, H. J.; Emerson, B. M.; and Pourmand, N. 2014. Single-cell analyses of transcriptional heterogeneity during drug tolerance transition in cancer cells by RNA sequencing. *Proceedings of the National Academy of Sciences* 111(44): E4726–E4735.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Loo, L.-H.; Wu, L. F.; and Altschuler, S. J. 2007. Image-based multivariate profiling of drug responses from single cells. *Nature methods* 4(5): 445–453.

Moore, J.; and Neville, J. 2017. Deep Collective Inference. In *AAAI*, 2364–2372.

Neville, J.; and Jensen, D. 2003. Collective classification with relational dependency networks. In *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, 77–91. Citeseer.

Obraztsov, I. V.; Shirokikh, K. E.; Obraztsova, O. I.; Shapina, M. V.; Wang, M.-H.; and Khalif, I. L. 2019. Multiple cytokine profiling: a new model to predict response to tumor necrosis factor antagonists in ulcerative colitis patients. *Inflammatory bowel diseases* 25(3): 524–531.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710. ACM.

Piccinini, F.; Balassa, T.; Szkalisity, A.; Molnar, C.; Paavolainen, L.; Kujala, K.; Buzas, K.; Sarazova, M.; Pietiainen, V.; Kutay, U.; et al. 2017. Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data. *Cell systems* 4(6): 651–655.

Preisach, C.; and Schmidt-Thieme, L. 2006. Relational ensemble classification. In *Sixth International Conference on Data Mining (ICDM'06)*, 499–509. IEEE.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Sechidis, K.; Tsoumakas, G.; and Vlahavas, I. 2011. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 145–158. Springer.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine* 29(3): 93.

Sheng, J.; Li, F.; and Wong, S. T. 2015. Optimal drug prediction from personal genomics profiles. *IEEE journal of biomedical and health informatics* 19(4): 1264–1270.

Shifat-E-Rabbi, M.; Yin, X.; Fitzgerald, C. E.; and Rohde, G. K. 2020. Cell image classification: a comparative overview. *Cytometry Part A* 97(4): 347–362.

Shipp, M. A.; Ross, K. N.; Tamayo, P.; Weng, A. P.; Kutok, J. L.; Aguiar, R. C.; Gaasenbeek, M.; Angelo, M.; Reich, M.; Pinkus, G. S.; et al. 2002. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine* 8(1): 68–74.

Sirinukunwattana, K.; Ahmed Raza, S. E.; Tsang, Y.-W.; Snead, D. R. J.; Cree, I. A.; and Rajpoot, N. M. 2016. Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images. *IEEE transactions on medical imaging* 35(5): 1196—1206.

Snijder, B.; Sacher, R.; Rämö, P.; Damm, E.-M.; Liberali, P.; and Pelkmans, L. 2009. Population context determines cell-to-cell variability in endocytosis and virus infection. *Nature* 461(7263): 520.

Taskar, B.; Segal, E.; and Koller, D. 2001. Probabilistic classification and clustering in relational data. In *International joint conference on artificial intelligence*, 870–878.

Tibshirani, R. J.; and Tibshirani, R. 2009. A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics* 822–829.

Toth, T.; Balassa, T.; Bara, N.; Kovacs, F.; Kriston, A.; Molnar, C.; Haracska, L.; Sukosd, F.; and Horvath, P. 2018. Environmental properties of cells improve machine learning-based phenotype recognition accuracy. *Scientific reports* 8(1): 10085.

Tsamardinos, I.; Greasidou, E.; and Borboudakis, G. 2018. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Machine Learning* 107(12): 1895–1922.

Usaj, M. M.; Styles, E. B.; Verster, A. J.; Friesen, H.; Boone, C.; and Andrews, B. J. 2016. High-content screening for quantitative cell biology. *Trends in cell biology* 26(8): 598–611.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *European Conference on Computer Vision*, 690–706. Springer.

Zhu, X.; Suk, H.-I.; Wang, L.; Lee, S.-W.; Shen, D.; Initiative, A. D. N.; et al. 2017. A novel relational regularization feature selection method for joint regression and classification in AD diagnosis. *Medical image analysis* 38: 205–214.