Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/86/20241606

CSIP: Contrastive learning with Graph Neural Network enables chemical structure to image mapping

Leo Liu

Georgetown Preparatory School, Maryland, USA

leoshiwuliu@outlook.com

Abstract. The development of advanced AI has been transforming the biomedical field. The emergence of multi-modal biomedical data such as imaging, sequencing, and other omics data further enabled the training of AI models for complex analytical tasks. However, such data analysis remains challenging due to the difficulty of integrating information from different modalities. In this paper, we aim to address this challenge by proposing a methodology to integrate image and molecular structure data. We present the Contrastive Structure-to-Image Pretraining (CSIP) framework, which leverages a self-supervised Graph Neural Network (GNN) to encode molecules and images into a joint feature embedding space. This direct mapping between the two modalities enables a wide variety of applications, including image profiling and clustering of molecules based on their effects on cell morphology. Image profiles generated by CSIP archived an average AUC of 0.708 on various biological activity prediction tasks, rivalling the state-of-the-arts and outperforming some fully supervised methodologies. Further, CSIP improved the accuracy of image-molecule matching by 29-folds from the random baseline after being trained on a small dataset, which demonstrated data efficiency. The code to reproduce our results can be found at https://github.com/LeoL18/CSIP.

Keywords: Image-based Profiling, Multi-modal Data, Graph Neural Network, Contrastive learning.

1. Introduction

Multi-modal biological data contains multiple modalities such as imaging data, sequencing data, or molecular data, providing rich information of biological entities. As modalities vary significantly in their statistical properties, integrating multi-modal data for biomedical applications such as diagnostics poses several challenges in need of further research. First, different modalities measure different sets of features with varying dimensions, which makes integration difficult. Furthermore, the distinct nature of each modality makes it difficult to generalize any method of analysis across modalities. The determination of the relative importance of each modality during integration also remains a topic to be researched. If these challenges are not addressed, studies of correlations between different modalities will be limited.

Thus, more effective methodologies for integrating multi-modal data will be very useful to the biomedical research community. Indeed, multiple beneficial applications may arise from new approaches to combine multi-modal data. For example, single-cell profiling with integrated multi-omics data was shown to have strong downstream performances [1]. Joint model of scRNA-seq and spatial

DOI: 10.54254/2755-2721/86/20241606

transcriptomics was capable of imputing missing gene expression measurements on a cell-level basis [2]. Integrating radiology, pathology, genomic, and clinical data also enabled deep learning methods to yield better cancer prognosis predictions than uni-modal models [3].

To leverage the potential of multi-modal data, many past studies focused on encoding data from different modalities to a joint embedding space. This approach bypasses the obstacle of dealing with data of non-uniform dimensions. However, there are some drawbacks associated with past methodologies. [4] used canonical correlation analysis to maximize the correlation of pairwise linear projections of features. However, this approach suffers from more complex data which cannot be mapped to a latent space through linear projection. [5] used a loss function that enforced the pairwise distance of points in the joint embedding space of scRNA-seq and scATAC-seq data to reflect the pairwise distances of the data in their respective modality space. However, this metric is computationally expensive, as it requires the calculation of diffusion distance up to 40 iterations, which makes it difficult to scale with larger batch size. [6] proposed CLOOME, which leveraged a contrastive learning paradigm to encode molecules and cell images to a joint embedding space. However, molecules are sent to the joint space by applying transformers to encode the corresponding SMILES strings, which hinders the extraction of molecular topology.

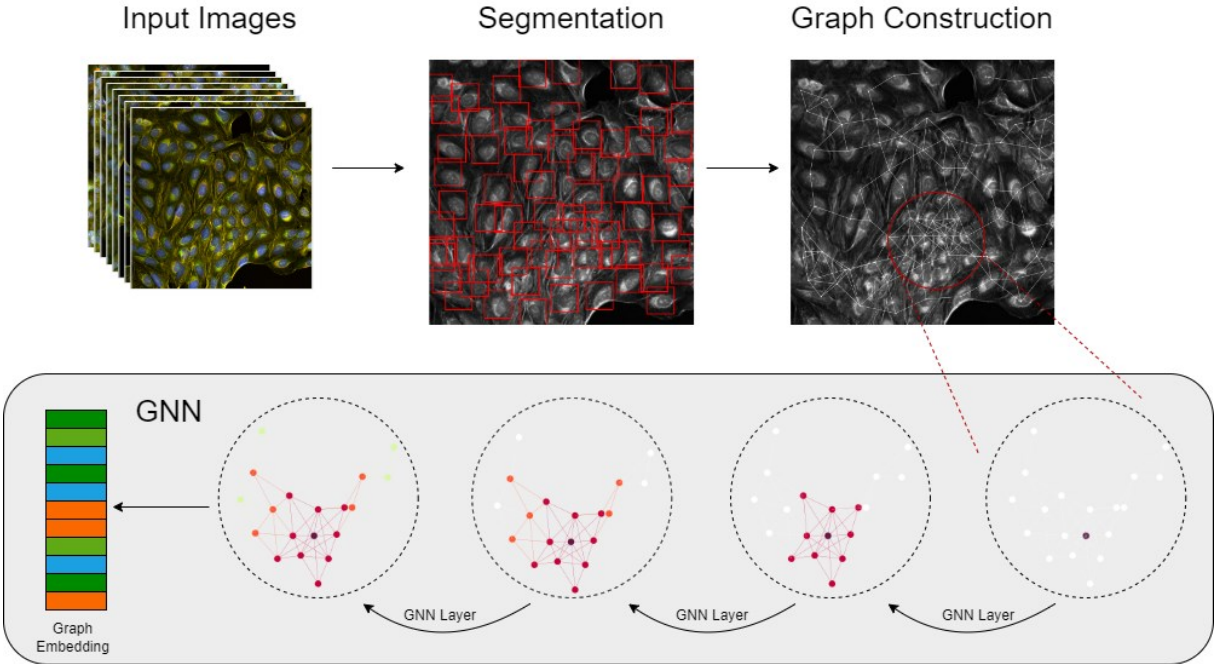

Figure 1. Overview of CSIP. We used CellProfiler to calculate the coordinate for each cell and constructed a cell-graph based on pairwise distances. For each cell, a morphological feature vector was computed using an autoencoder. GNN was then applied on the resulting cell-graph and features to obtain the final embedding.

To address the above shortcomings, we present the CSIP framework, which leverages GNN to generate joint embeddings for molecules and cell images. GNN is inherently suitable for encoding molecules as it can operate directly on the molecular topology, which other frameworks (e.g. transformers) have to learn from scratch. In addition, GNN also requires less computational resources to encode images than more conventional methods such as Convolutional Neural Network (CNN). To train the model, contrastive learning is employed so that the model learns to generate similar latent embeddings for positive, or matched molecule-image pairs. Cosine similarity, which is computationally inexpensive, is used to measure distance in the latent space to calculate the contrastive loss.

DOI: 10.54254/2755-2721/86/20241606

We envision three applications for this framework. CSIP can generate cell image embeddings for downstream tasks such as mechanism-of-action classification and bio-activity prediction. From the chemical structure of the molecule, CSIP can predict the change in cell morphology when the molecule is used to treat the cell. In addition, the ability to infer the structure of a molecule from image of cells treated by that molecule enables CSIP to identify replacement molecules with similar effect on cells. On a Cell Painting dataset [7] with around one million 5-channels cell images and 30, 616 different compound perturbations, CSIP correctly identified the compound used to treat the cells in the input image with top-5 accuracy and top-10 accuracy 120 times and 220 times higher than random guessing, respectively. Further, across 209 bio-activity prediction tasks, image profiles generated by CSIP achieved an average AUC of 0.708 and F1 of 0.373. These results rivalled the previously best-performing models and even outperformed some fully supervised methodologies.

2. Related Work

Analyzing biomedical multi-modal data With a variety of modalities, multi-modal data provides both complementary information to aid prediction and redundant features shared by multiple modalities to guard against noisy data [8]. However, the extraction of this rich information poses a challenge to traditional methods used for uni-modal data. Past studies developed a variety of new methodologies for leveraging the rich information in multi-modal data. A straightforward way to integrate multi-modal data is to concatenate the feature vectors from each modality, which [8] termed "Early Fusion". [9] used concatenated features selected from the gene expression and DNA methylation data as input to a feed-forward network to predict Alzheimer's disease. Similarly, [10] combined multi-modal gene profiles as input to a deep neural network to yield prediction for cancer survival. On the other hand, some studies proposed the use of multiple independent discrete encoders, each modelling data from a specific modality. [11] trained multiple autoencoders to model RNA-Seq, miRNA-Seq, and methylation data. Likewise, [12] used a concatenation of latent features from modal-specific variational autoencoders as input to a feed-forward network. In addition, some studies used unsupervised learning to train models with multi-modal data. In particular, [13] used Spearman's correlation to determine the pairwise correlation of over a thousand features from various modalities and leveraged Louvain community detection to identify biologically functional subnetworks. [14] utilized an image registration network to model the noise in biomedical images to improve the performance of Generative Adversarial Network (GAN) on image translation tasks.

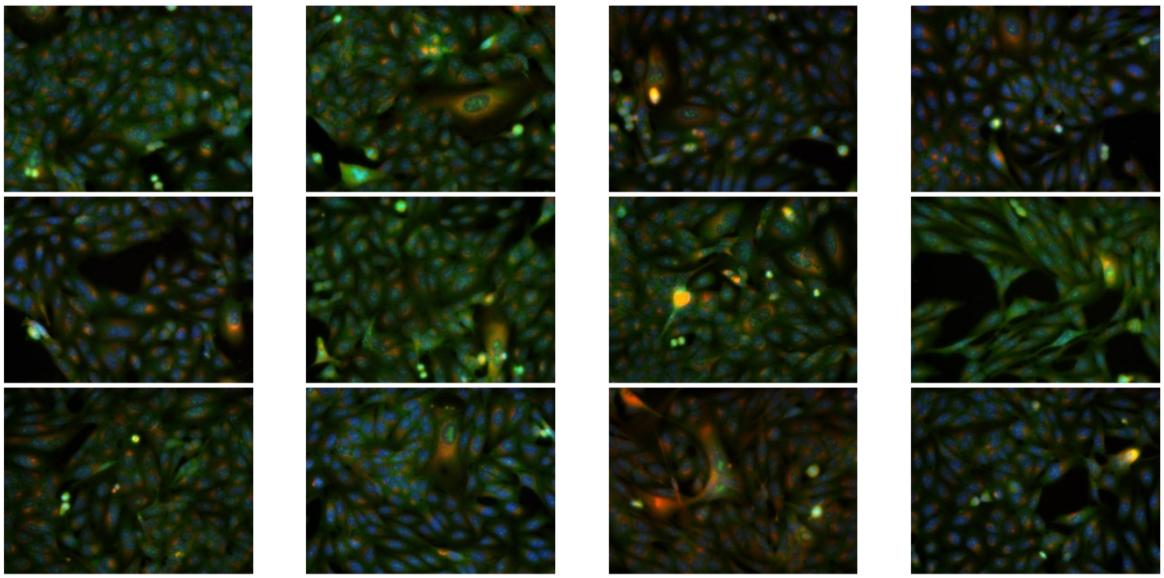

Figure 2. Sample dyed images from the dataset we used [7]. The original images in the dataset have five channels with resolution 696×520.

DOI: 10.54254/2755-2721/86/20241606

Graph-based learning for identifying biological correlation Guilt by-association is a popular principle in the biomedical field, which states that associated genes are more likely to share similar function. This principle inspires analysis of biomedical knowledge graph to discover new associations between biological entities. Often times, such analysis is conducted with graph-based learning. [15] modelled drug-drug interaction (DDI) by applying GNN on biomedical knowledge graphs to learn generic node embeddings, which were then fine-tuned through sub-graph learning to predict DDI. [16] used random walks on knowledge graph to generate drug and disease embeddings for drug repurposing. To address the lack of drug and disease nodes in knowledge graphs, the authors proposed the use of teleportation guided by semantic information obtained from hierarchical similarity. [17] used GNN to model molecular associations. Node embeddings were fused at each propagation step, which the authors claimed to have improved the classification results. In addition, past studies showed that graph-based learning can also be used in image segmentation tasks. For example, [18] used Graph Convolutional Network (GCN) for semantic segmentation. Specifically, the authors first leveraged CNN to transform image into graph, after which GCN was used for node classification to determine the label for each pixel of the original image.

Contrastive learning for multi-modal data In contrastive learning, unlabelled data points are compared against each other to teach the model to recognize similar (positive) and different (negative) pairs. This method has recently achieved success in learning multi-modal data. One notable example is CLIP [19], which used contrastive learning to learn a joint image-text embedding space. When contrastive learning is used for analyzing multi-modal data, a separate encoder is trained for each modality to map inputs into a joint embedding space where modality data belonging to the same sample have adjacent embeddings. Since contrastive learning is well-suited to large, unlabelled dataset, it is very promising in the medical field, where labelled data is rarely available. [6], inspired by [19] and [20], leveraged a contrastive learning paradigm to learn joint image-molecule embeddings. Similarly, [21] used contrastive learning to match chest X-rays and their corresponding reports. Sometimes, however, data might not come with a clear division into positive and negative pairs. In that case, positive pairs can be created by generating new samples from each sample of the original data, where each new sample represents a feature of the original sample. For example, [22] extracted radiomic features and deep features from histopathological image to form positive pairs. [23] applied small perturbations on images to create positive pairs and trained an encoder to minimize the distance between embeddings of positive pairs. Metadata from MRI can also be used to form positive pairs [24].

3. Materials and Methods

3.1. Dataset

We used a Cell Painting dataset with a variety of compound perturbations [7] as our dataset (Figure 2). The dataset consists of human U2OS osteosarcoma cells imaged over five channels (RNA, ER, AGP, Mito, and DNA) in 384-wells plates. Each well received a different form of compound perturbation targeting a specific gene in the cells, with some wells receiving DMSO to be used as controls. In each well, 6 images were taken at different sites. After applying quality control on the dataset, which filtered out blurry and saturated images, we obtained around $950{,}000$ 5-channels images with resolution 696×520 representing $30{,}616$ different compound perturbations targeting a variety of genes. We removed the top 0.02% of the brightest pixels before normalizing each image to the $[0,1]$ range. Illumination correction was then applied to the normalized images (i.e. dividing the image by the corresponding brightfield mask, which was obtained by taking an image over the light source after removing the cell specimen).

DOI: 10.54254/2755-2721/86/20241606

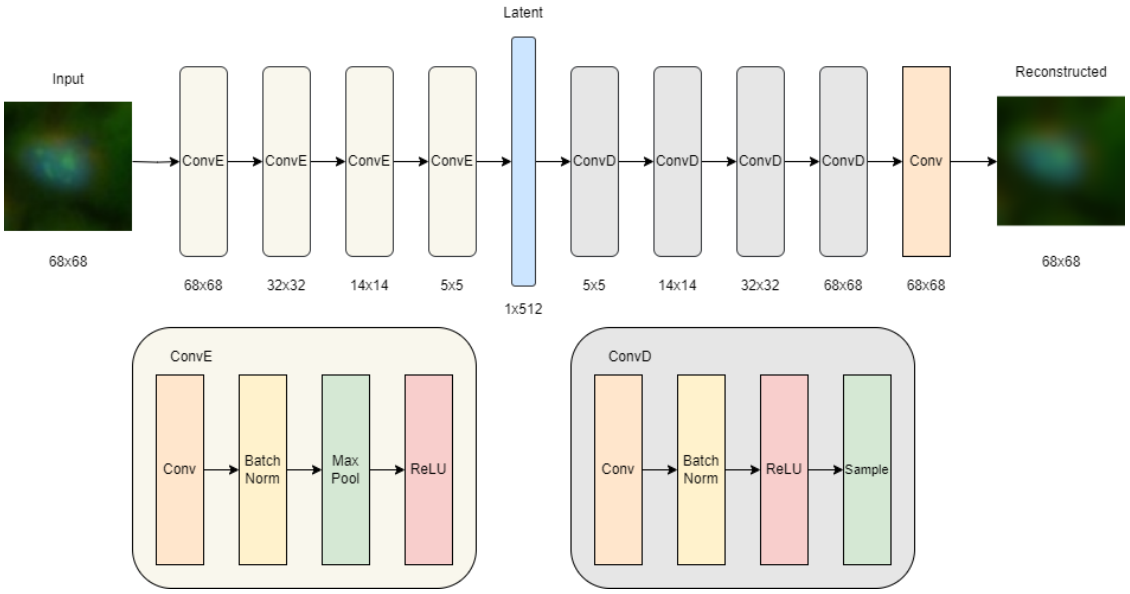

Figure 3. The CNN autoencoder we used to extract embeddings for each individual cell. The encoder consists of four identical blocks, with each block composed of a convolutional layer, a batch normalization layer, a pooling layer, and an activation function. Similarly, the decoder also has four blocks, with an extra convolutional layer at the end.

3.2. Data Preprocessing

CSIP consists of an image encoder and a molecular encoder mapping inputs to a joint space, with both encoders using GNN as backbone. The use of GNN for image encoder necessitates a pipeline converting images to graphs. During this conversion, we sought to preserve two types of information: phenotypic features of cells and topology (cell-cell interactions). A graph is defined as a set of nodes and a set of edges connecting the nodes to each other. In our case, we used the cells in the original image as the nodes of the graph. This construction preserves phenotypic features of cells via node feature vectors. We then added edges between adjacent cells in the graph, thus retaining the cell-cell interactions.

More specifically, we first used CellProfiler [25] to calculate the bounding box of each cell in the image. The center of each bounding box was treated as a node in the graph (Figure 1). We then trained a CNN autoencoder [26] to extract features from the 68×68 pixels neighborhood of each cell (Figure 3). We noted that the crop box sometimes missed the cell partially due to discrepancies in cell sizes. Thus, we enhanced the autoencoder by training it to repair the cell from an incomplete crop. CNN profiles have shown strong performances in the downstream tasks.

To model cell-cell interactions in the original image, we computed the pairwise distances between cell centers and constructed edges between adjacent pairs. We used two parameters to control the size of the graph: L and n. L denotes the maximum distance between two nodes connected by an edge, and n denotes the maximum degree of each node. In our experiment, we set $L = 120$ pixels and $n = 7$.

The processing of molecules, on the other hand, is relatively straightforward. We processed molecules using the procedure in [27]. Specifically, for each atom, several measures were extracted from the SMILES string using the Mol class of the rdkit module, including atom type, number of covalent bond (further divided into single bond, double bond, and triple bond), number of radical electrons, formal charge, hybridization, aromaticity, number of connected hydrogens, and chirality. Note that bond-related information was included in the extracted atom features. Then, a graph was constructed based on the chemical structure of the molecule, and atom features were assigned to the corresponding nodes on the graph.

DOI: 10.54254/2755-2721/86/20241606

3.3. GNN Backbone for Both Encoders

In this section, we define the GNN backbone used for both encoders in mathematical formulation. A graph G is defined by its node set V and edge set E, together denoted as $G = (V, E)$. The edges can be described by the adjacency matrix A, where $A_{ij} = 1$ if there is an edge between node i and node j. By default, $A_{ii} = 1$ for all nodes $1 \leq i \leq |V|$. At every timestep $t \geq 0$, each node $u \in V$ has an embedding $h_u^{(t)}$. The initial set of embeddings $\{h_u^{(0)} | u \in V\}$ are obtained from CNN autoencoder and then clamped to $[-4,4]$ to prevent slow training induced by large numbers. If $(u, v) \in E$, then u is connected to v by an edge with a weight W_{uv}, which is defined as

$$W_{uv} = \begin{cases} \max\left(\frac{1}{2}, 1 - \frac{\text{dist}(u,v)}{L}\right) & \text{dist}(u,v) \leq L \\ 0 & \text{dist}(u,v) > L \end{cases} \quad (1)$$

with $\text{dist}(u, v)$ being the distance between u and v.

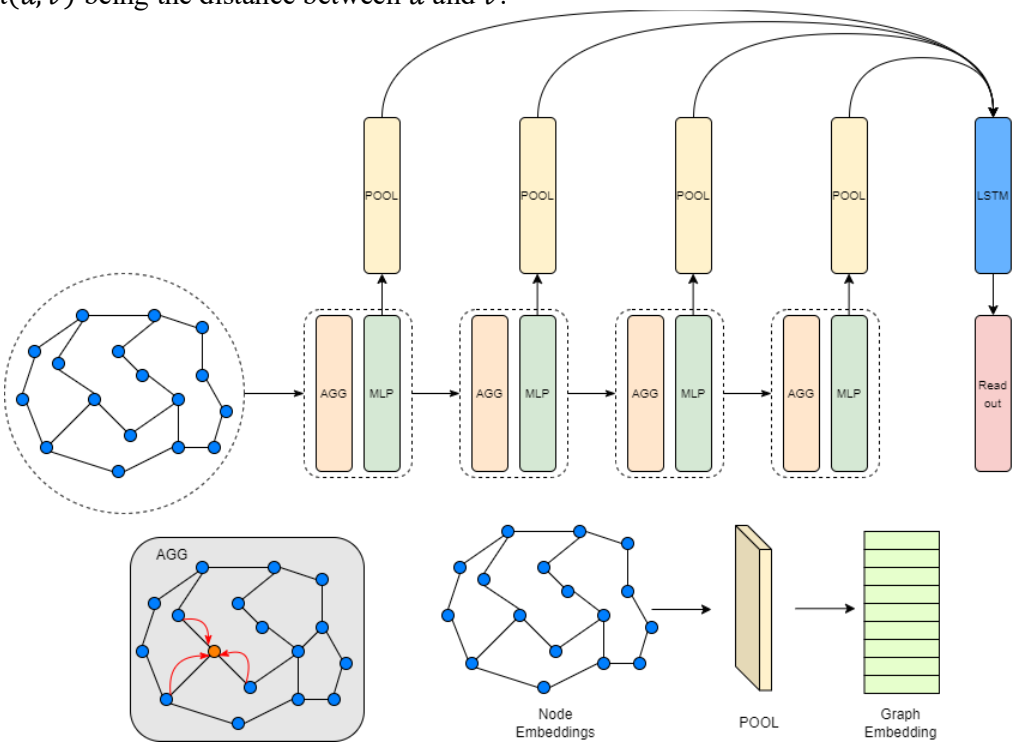

Figure 4. An overview of the image encoder. Each cell embedding is aggregated with the embeddings of neighboring cells, and the resulting cell embedding is mapped to the next layer of the model through a MLP block. Graph embeddings were generated at each timestep, and an importance score was computed for each graph embedding by an LSTM block. The final embedding is calculated by concatenating the graph embeddings scaled by their respective importance scores.

At each timestep $t \geq 1$, the node embedding $h_u^{(t)}$ is updated by itself $h_u^{(t-1)}$ and all of it neighbors $\{h_v^{(t-1)} | (u, v) \in E\}$. For each update, we scale the features from neighboring nodes with respect to W_{uv}. Specifically, we compute the features in each update by the formula

$$\hat{h}_u{}'^{(t)} = \text{Agg}\left(\{W_{uv} \cdot h_v^{(t-1)} | (u, v) \in E\}\right) \quad (2)$$

$$\hat{h}_u^{(t)} = (1 + \epsilon) \cdot h_u^{(t-1)} + \hat{h}_u{}'^{(t)} \quad (3)$$

where ϵ is a trainable parameter determining the importance of $h_u^{(t-1)}$. For the Agg operation, we used the sum of neighboring features. Then, node features are updated by

DOI: 10.54254/2755-2721/86/20241606

$$h_u^{(t)} = \text{Norm}\left(\text{GeLU}\left(\text{MLP}\left(\hat{h}_u^{(t)}\right)\right)\right) \tag{4}$$

where Norm denotes the graph-wise normalization from [28]. More conveniently, the update of node features can be written as

$$H^{(t)} = \text{Norm}(\text{GeLU}\left(\text{MLP}\left(WH^{(t-1)} + \epsilon H^{(t-1)}\right)\right) \tag{5}$$

where $H^{(t)}$ denotes the concatenation of all node features $\{h_u^{(t)} | u \in V\}$ at the timestep t.

After each update, we pool the updated node features and generate the graph embedding at the timestep t

$$h_G^{(t)} = \text{Pool}\left(\{h_u^{(t)} | u \in V\}\right) \tag{6}$$

One naive way to pool the nodes is summation. However, this approach fails to consider the continuous nature of node embeddings. When summation is used for the Pool operator, the graph embeddings become almost identical even for different graph inputs, which makes the embeddings not representative of the original inputs. This is because the node features are sampled from a continuous distribution, and variance decreases as more node features are summed. Thus, we use element-wise maximum for pooling to obtain distinguishable graph embeddings. We compute $h_G^{(1)}, h_G^{(2)}, \cdots, h_G^{(m)}$, where m is the number of layers.

It has been previously noted that GNN tends to converge to fixed node embeddings with a large number of layers [29,30]. This problem is known as over-smoothing in the literature [31,32]. To alleviate this issue, we use graph embeddings from all timesteps to compute a final readout. Previous methodology suggests the use of concatenation [33]. We further use a LSTM block to learn an importance score p for graph embedding at each timestep. Graph embeddings are sequentially passed into the LSTM block, which generates an importance score p_t corresponding to the graph embedding $h_G^{(t)}$ at each timestep $0 \leq t \leq m$. We then obtain the final embedding of the graph by concatenating

$$h_G = \text{Concatenate}\{h_G^{(t)} p_t | 0 \leq t \leq m\} \tag{7}$$

3.4. Contrastive Loss Function

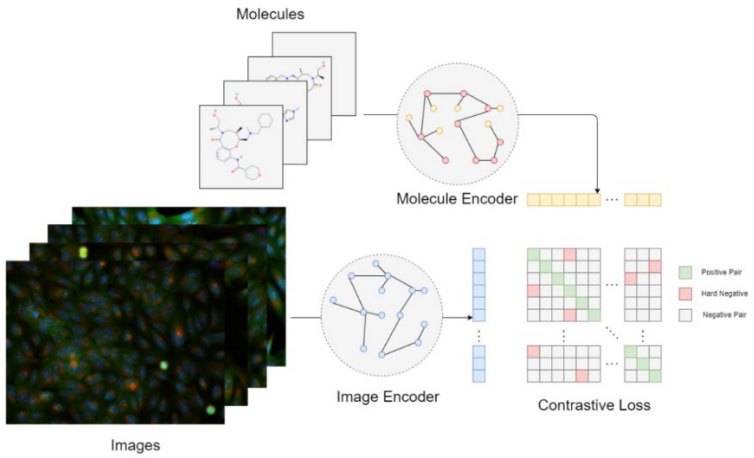

Figure 5. Training pipeline of CSIP. Each image forms a positive pair with its corresponding molecule. Of the rest of the molecules in the batch, the ones with highest latent similarity with the image form hard negative pairs. The model is trained to maximize the similarity of latent representations between positive pairs and to minimize the similarity between hard negative pairs.

DOI: 10.54254/2755-2721/86/20241606

The goal of our framework is to map molecules and images to a shared, latent embedding space. Ideally, positive pairs of molecule and image should be close to each other in this space, and negative pair should be separate. Further, it is desired to have a structured embedding space for generalization. To accomplish this, we adopted a similar strategy to OpenAI's CLIP [19]. We first normalized the image embeddings $\mathbf{x} \in \mathbb{R}^{n \times d}$ and molecule embeddings $\mathbf{y} \in \mathbb{R}^{n \times d}$ to unit vectors, where n is the number of samples and d is the dimension of the embedding space. Next, we employ the cosine similarity, represented by $\text{Softmax}(\mathbf{x} \cdot \mathbf{y}^T)$, to compute the similarity of each image with respect to the molecules. Finally, the model is trained to maximize the similarity of positive pairs

$$L_P = -\frac{1}{n}\sum_{i=1}^{n} \tau^{-1}\, \mathbf{x}_i \mathbf{y}_i^T \tag{8}$$

where τ^{-1} is a temperature parameter controlling the model's confidence in its predictions.

We also have to account for negative pairs in the loss function to keep them separate. However, optimizing with respect to all negative pairs within a batch might lead to noisy gradients. Thus, we sample negative pairs with the highest similarity, known as hard negative pairs [34], to be used in optimization. This sampling can be accomplished by

$$L_N = -\frac{1}{n}\sum_{i=1}^{n} lse\left(\tau^{-1}\left(\mathbf{x}\mathbf{y}^T - diag(\mathbf{x}\mathbf{y}^T)\right)_i\right) \tag{9}$$

where lse denotes the log-sum-exponent function, which is used as an approximation of max to select the hard negative pair. Combining the two objectives of training, we obtain the loss function

$$L = L_P + L_N = -\frac{1}{n}\sum_{i=1}^{n} \ln \frac{\exp\left(\tau^{-1}\mathbf{x}_i {\mathbf{y}_i}^T\right)}{\sum_{j \neq i} \exp\left(\tau^{-1}\mathbf{x}_j {\mathbf{y}_j}^T\right))} \tag{10}$$

which agrees with the InfoLOOB loss as derived in [20]. This training objective ensures positive pairs to be close to each other and negative pairs to be separate. Further, embeddings are normalized so that they all lie on a unit hypersphere, which ensures a structured embedding space.

4. Experiments and Results

4.1. Implementation Details

We used three layers of GNN, thus enabling each node to aggregate information from a 3-hop neighborhood. Adam optimizer [35] with decoupled weight decay regularization [36] and a cosine scheduler [37] was used to train the model. Edges were randomly dropped with a probability of $p = 0.3$ to alleviate over-fitting and over-smoothing [38]. We searched for the suitable hyperparameters using a combination of manual tuning and bayesian optimization by training a smaller version of the model on a subset of the data. We used size 64 for node feature vector, which produced a latent space with 192 dimensions. A base learning rate of $5e-3$ was used to train the model for 5 warm-up epochs, after which a cosine scheduler was applied. Gradient clipping of 1.0 was applied to all parameters of the model. A large mini-batch size of 2048 was used to alleviate the problem of gradient bias [39]. Mixed precision was also utilized to save memory and to accelerate training.

4.2. Molecule Retrieval

In this experiment, we tested the ability of CSIP to correctly identify the molecules used to treat the cells in the input images. Note that matching molecules to images is a very difficult task, since cells often exhibit very subtle morphological changes when treated by molecules. Further, due to the large number of molecules in the dataset, it is possible that there exists multiple of them inducing similar morphological effects on the cells. Thus, low accuracy is expected for this task.

DOI: 10.54254/2755-2721/86/20241606

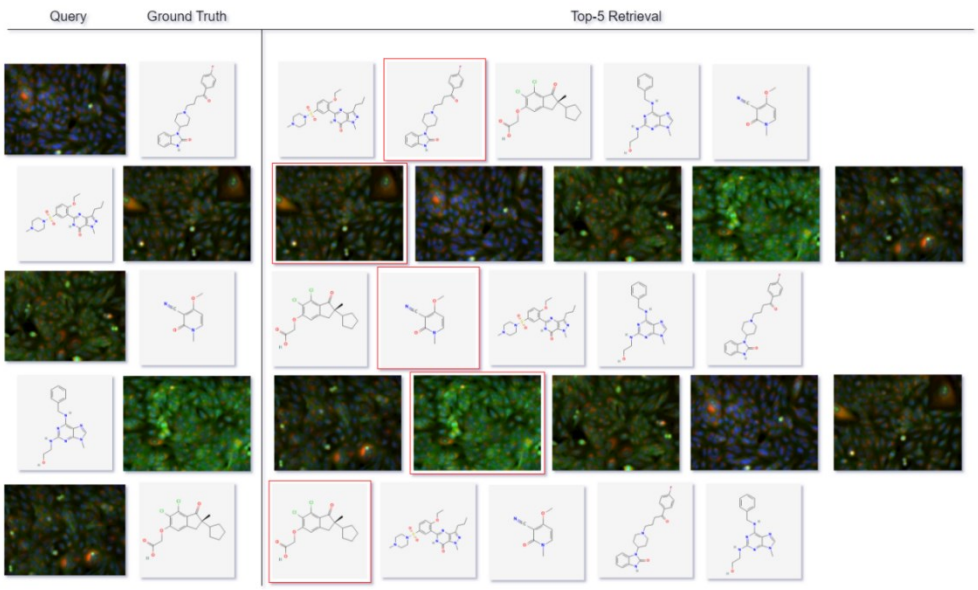

Figure 6. CSIP can be used to retrieve molecular structures from images. On the evaluation set, CSIP achieved a 15% top-5 accuracy, which is a strong performance. Indeed, the reverse is also possible due to the joint latent space.

To evaluate the accuracy of CSIP on the retrieval task, we first partitioned a dataset of 200,000 randomly sampled image-molecule pairs into five equal subsets. We then left out one subset for evaluation and trained the model on the rest. This process was repeated for four more times, with a different set being left out for evaluation each time. Average top-1 accuracy, top-5 accuracy, and top-10 accuracy were recorded. We then re-sampled the dataset and applied the same evaluation procedure, for a total of 5 trials.

Table 1. Top-1, top-5, and top-10 retrieval accuracy for CSIP and other baseline methods. 95% confidence interval for the accuracy of each method were calculated through 5 trials of evaluation. CSIP delivered a strong performance when compared against the baselines.

Method	Top-1	Top-5	Top-10
Random	0.001 ± 0.000	0.006 ± 0.002	0.012 ± 0.004
1-MLP	0.023 ± 0.088	0.140 ± 0.092	0.263 ± 0.011
CSIP	0.034 ± 0.008	0.148 ± 0.011	0.296 ± 0.028

Table 2. Folds of improvement of each method from random retrieval.

Method	Top-1	Top-5	Top-10
1-MLP	19.52	118.54	222.02
CSIP	29.01	124.68	250.19

We also tested three different baselines on the same task for comparison. First, we reported the accuracy of random matching. Next, we tested an alternative version of CSIP with the GNN backbone for the image encoder replaced by a one-layer MLP. Finally, we compared CSIP to a previous methodology.

We noted that our model achieved a 29-folds improvement in top-1 accuracy from the random baseline and a 1.5 -folds improvement from the 1-MLP baseline (Table 2). These significant improvements validated the use of GNN backbone for the image encoder. The result also suggested that modelling cell-cell interactions through graph-based learning improved retrieval accuracy, since CSIP

DOI: 10.54254/2755-2721/86/20241606

performed significantly better when GNN was used. Lastly, our model rivaled CLOOME [6], which achieved a 34-folds improvement in top-1 accuracy from the random baseline. It should be noted, however, that their model was trained on a larger dataset. Our model, in comparison, was trained on only 160,000 samples for the sake of evaluation.

4.3. Image Embedding

We adopted the evaluation procedure in [27] to determine the performance of CSIP image embeddings in downstream tasks. For each of the 10,574 molecules in the dataset, its chemical activities in 209 different assays were collected and represented as binary values, indicating whether the molecule was active or inactive in a specific assay, thus creating a label matrix. However, over 97% of the entries were missing in this matrix as relevant information were unavailable on the ChEMBL database. To address this issue, all empty activity labels were masked out in both training and evaluation. The task of predicting the label matrix from the embeddings can be interpreted as 209 independent downstream binary prediction tasks, and the performance of the embeddings on these tasks indicates their quality of representation.

During evaluation, the use of complex models must be avoided to ensure that the classification performance is indicative of the representation quality of the embeddings, since complex models might deliver strong performance even if the representation quality of the embeddings were poor. Thus, we used a simple bilinear model

$$\hat{y} = \sigma(\mathbf{x}^T \mathbf{W} \mathbf{y}) \tag{11}$$

to predict compound activities from the embeddings, where σ is the sigmoid function and W is a trainable weight matrix. For this experiment, we used the same dataset as [27]. First, we splitted the dataset into a training set, a validation set, and a testing set. The model was then trained on the training set until validation accuracy stopped increasing. After training, average AUC score on the testing set across all tasks was reported. A high AUC score on a specific task indicates strong transferability of the embeddings to that task. We also reported the F1-score to account for unbalanced classification tasks. Further, we recorded the number of the 209 tasks for which an AUC score above 0.7, 0.8, or 0.9 was observed.

For baselines, we adopted the models in [27], which consists of various CNN architectures trained in a fully supervised setting. In addition, we used a contrastive learning framework [6] as another baseline.

Table 3 suggests that CSIP image embeddings transferred well to the prediction of bioassay activities. CSIP achieved an AUC of 0.708 ± 0.13 and a F1 of 0.373 ± 0.22 across all prediction tasks, which is an improvement from CellProfiler and CNN autoencoder features. We also noted that CSIP outperformed some fully supervised CNN in the experiment. This result is very surprising, since CSIP was trained without labels.

Table 3. Experimental results from the 209 binary classification tasks. CSIP achieved a strong performance comparing to the baselines.

Type	Method	AUC	F1	AUC > 0.9	AUC > 0.8	AUC > 0.7
Transfer	CSIP	0.708 ± 0.13	0.373 ± 0.22	49	79	105
	CLOOME	0.714 ± 0.20	0.395 ± 0.32	57	84	109
	CellProfiler	0.655 ± 0.20	0.273 ± 0.32	35	63	84
	Autoencoder	0.634 ± 0.31	0.254 ± 0.36	28	59	76
Supervised	ResNet	0.731 ± 0.19	0.508 ± 0.30	68	94	119
	DenseNet	0.725 ± 0.19	0.530 ± 0.30	61	98	121
	GapNet	0.711 ± 0.18	0.510 ± 0.29	63	94	117
	MIL-Net	0.705 ± 0.19	0.445 ± 0.32	61	81	105
	M-CNN	0.705 ± 0.19	0.482 ± 0.31	57	78	105
	SC-CNN	0.705 ± 0.20	0.362 ± 0.29	61	83	109
	FNN	0.675 ± 0.20	0.361 ± 0.31	55	71	90

DOI: 10.54254/2755-2721/86/20241606

5. Discussion

We present CSIP, a contrastive learning framework capable of mapping molecules and images to a joint embedding space. CSIP achieved strong performances in both molecule retrieval and biological activity predictions. In particular, the use of contrastive learning eliminates the need of labelled data, which is rarely available in high-throughput projects. Furthermore, since CSIP was trained without explicit labels, image embeddings generated by CSIP hold potential for zero-shot predictions. This enables CSIP to make inference on a large set of compounds without extremely extensive training. Thus, CSIP serves as a preliminary step towards the development of query systems based on cell morphology for large compound databases.

The use of GNN for the image encoder rather than a more conventional choice such as CNN serves two purposes. First, GNN can be easily scaled up for larger graphs by adding more layers, enabling them to learn extremely large images such as whole slide images. These images can be as large as $100{,}000 \times 100{,}000$ pixels, which is very difficult to be learned by CNN. In addition, GNN is very light in both memory and computation. CSIP has only $120{,}000$ trainable parameters in total, most of which coming from fully connected layers. On the other hand, most deep CNNs have well over a million parameters and require extensive training on multiple GPU cores.

The use of GNN to learn molecules is easily justified. Molecules are inherently graphs, which makes graph-based learning a natural choice. On the other hand, using architectures such as transformer, which looks at relationships between all atoms at once, might lead to the learning of unwanted relationships between atoms. GNN does not suffer from this issue, since information is passed between atoms through the edges representing bonds. Thus, GNN is well-suited to learning molecular structures.

However, we acknowledge some limitations with our approach. At the cost of lightness, CSIP requires data preprocessing including image segmentation and extraction of cell features. Furthermore, a very large batch size is needed during the training of CSIP, since a large number of negative pairs must be sampled to effectively identify hard negative pairs. Nevertheless, there has been a recent interest in redesigning contrastive learning to overcome the problem of a large batch size [40][41], which might address this issue in the future. We acknowledge these limitations and leave them for further studies.

References

[1] Tao Peng, Gregory M. Chen, and Kai Tan. Gluer: integrative analysis of single cell omics and imaging data by deep neural network. *bioRxiv*, 2021.

[2] Romain Lopez, Achille Nazaret, Maxime Langevin, Jules Samaran, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. A joint model of unpaired data from scrna-seq and spatial transcriptomics for imputing missing gene expression measurements, 2019.

[3] Nathaniel Braman, Jacob W. H. Gordon, Emery T. Goossens, Caleb Willis, Martin C. Stumpe, and Jagadish Venkataraman. Deep orthogonal fusion: Multimodal prognostic biomarker discovery integrating radiology, pathology, genomic, and clinical data, 2021.

[4] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019.

[5] Ziqi Zhang, Chengkai Yang, and Xiuwei Zhang. scDART: integrating unmatched scRNA-seq and scATAC-seq data and learning cross-modality relationship simultaneously. *Genome Biology*, 23(1):139, June 2022.

[6] Ana Sanchez-Fernandez, Elisabeth Rumetshofer, Sepp Hochreiter, and Günter Klambauer. Cloome: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature Communications*, 14(1):7339, 2023.

[7] Marzieh Haghighi, Juan C. Caicedo, Beth A. Cimini, Anne E. Carpenter, and Shantanu Singh. High-dimensional gene expression and morphology profiles of cells across 28,000 genetic and chemical perturbations. *Nat Methods*, 2022.

DOI: 10.54254/2755-2721/86/20241606

[8] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2):bbab569, 01 2022.

[9] Chihyun Park, Jihwan Ha, and Sanghyun Park. Prediction of alzheimer's disease based on deep neural network by integrating gene expression and dna methylation dataset. *Expert Systems with Applications*, 140:112873, 2020.

[10] Lianhe Zhao, Qiongye Dong, Chunlong Luo, Yang Wu, Dechao Bu, Xiaoning Qi, Yufan Luo, and Yi Zhao. Deepomix: A scalable and interpretable multi-omics deep learning framework and application in cancer survival analysis. *Computational and Structural Biotechnology Journal*, 19:2719–2725, 2021.

[11] Olivier B Poirion, Zheng Jing, Kumardeep Chaudhary, Sijia Huang, and Lana X Garmire. DeepProg: an ensemble of deep-learning and machine-learning models for prognosis prediction using multi-omics data. *Genome Medicine*, 13(1):112, July 2021.

[12] Chunlei Liu, Hao Huang, and Pengyi Yang. Multi-task learning from multimodal single-cell omics with Matilda. *Nucleic Acids Research*, 51(8):e45–e45, 03 2023.

[13] Ilan Shomorony, Elizabeth T Cirulli, Lei Huang, Lori A Napier, Robyn R Heister, Michael Hicks, Isaac V Cohen, Hung-Chun Yu, Christine Leon Swisher, Natalie M Schenker-Ahmed, Weizhong Li, Karen E Nelson, Pamila Brar, Andrew M Kahn, Timothy D Spector, C Thomas Caskey, J Craig Venter, David S Karow, Ewen F Kirkness, and Naisha Shah. An unsupervised learning approach to identify novel signatures of health and disease from multimodal data. *Genome Medicine*, 12(1):7, January 2020.

[14] Lingke Kong, Chenyu Lian, Detian Huang, Zhenjiang Li, Yanle Hu, and Qichao Zhou. Breaking the dilemma of medical image-to-image translation, 2021.

[15] Yaqing Wang, Zaifei Yang, and Quanming Yao. Accurate and interpretable drug drug interaction prediction enabled by knowledge subgraph learning. *Communications Medicine*, 4(1):59, March 2024.

[16] Dongmin Bang, Sangsoo Lim, Sangseon Lee, and Sun Kim. Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers. *Nature Communications*, 14(1):3570, June 2023.

[17] Chuanze Kang, Han Zhang, Zhuo Liu, Shenwei Huang, and Yanbin Yin. LR-GNN: a graph neural network based on link representation for predicting molecular associations. *Briefings in Bioinformatics*, 23(1):bbab513, 12 2021.

[18] Yi Lu, Yaran Chen, Dongbin Zhao, and Jianxin Chen. Graph-fcn for image semantic segmentation, 2020.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[20] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela BittoNemling, and Sepp Hochreiter. Cloob: Modern hopfield networks with infoloob outperform clip, 2022.

[21] Zhanghexuan Ji, Mohammad Abuzar Shaikh, Dana Moukheiber, Sargur N Srihari, Yifan Peng, and Mingchen Gao. Improving joint learning of chest X-Ray and radiology report by word region alignment. *Mach Learn Med Imaging*, 12966:110–119, September 2021.

[22] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. Transpath: Transformer-based self-supervised learning for histopathological image classification. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27 – October 1, 2021, Proceedings, Part VIII*, page 186–195, Berlin, Heidelberg, 2021. Springer-Verlag.

[23] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations, 2020.

[24] Benoit Dufumier, Pietro Gori, Julie Victor, Antoine Grigis, Michel Wessa, Paolo Brambilla, Pauline Favre, Mircea Polosan, Colm McDonald, Camille Marie Piguet, and Edouard Duchesnay. Contrastive learning with continuous proxy meta-data for 3d mri classification, 2021.

[25] Claire McQuin, Allen Goodman, Vasiliy Chernyshev, Lee Kamentsky, Beth A. Cimini, Kyle W. Karhohs, Minh Doan, Liya Ding, Susanne M. Rafelski, Derek Thirstrup, Winfried Wiegraebe, Shantanu Singh, Tim Becker, Juan C. Caicedo, and Anne E. Carpenter. Cellprofiler 3.0: Next-generation image processing for biology. *PLOS Biology*, 16(7):1–17, 07 2018.

[26] Maxime W. Lafarge, Juan C. Caicedo, Anne E. Carpenter, Josien P.W. Pluim, Shantanu Singh, and Mitko Veta. Capturing single-cell phenotypic variation via unsupervised representation learning. In M. Jorge Cardoso, Aasa Feragen, Ben Glocker, Ender Konukoglu, Ipek Oguz, Gozde Unal, and Tom Vercauteren, editors, *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, volume 102 of Proceedings of Machine Learning Research*, pages 315–325. PMLR, 08–10 Jul 2019.

[27] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2020. PMID: 31408336.

[28] Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training, 2021.

[29] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2021.

[30] Qimai Li, Zhichao Han, and Xiao-ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[31] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs, 2020.

[32] Nicolas Keriven. Not too little, not too much: a theoretical analysis of graph (over)smoothing, 2022.

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2018.

[34] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples, 2021.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.

[38] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification, 2020.

[39] Changyou Chen, Jianyi Zhang, Yi Xu, Liqun Chen, Jiali Duan, Yiran Chen, Son Dinh Tran, Belinda Zeng, and Trishul Chilimbi. Why do we need large batchsizes in contrastive learning? a gradient-bias perspective. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[40] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022.

[41] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup, 2021.