This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Exemplar Guided Deep Neural Network for Spatial Transcriptomics Analysis of Gene Expression Prediction

Yan Yang[1] Md Zakir Hossain[1] Eric A Stone [1] Shafin Rahman[2]
[1] BDSI, Australian National University, Australia [2] ECE, North South University, Bangladesh
{Yan.Yang, zakir.hossain, eric.stone}@anu.edu.au shafin.rahman@northsouth.edu

Abstract

Spatial transcriptomics (ST) is essential for understanding diseases and developing novel treatments. It measures gene expression of each fine-grained area (i.e., different windows) in the tissue slide with low throughput. This paper proposes an Exemplar Guided Network (EGN) to accurately and efficiently predict gene expression directly from each window of a tissue slide image. We apply exemplar learning to dynamically boost gene expression prediction from nearest/similar exemplars of a given tissue slide image window. Our EGN framework composes of three main components: 1) an extractor to structure a representation space for unsupervised exemplar retrievals; 2) a vision transformer (ViT) backbone to progressively extract representations of the input window; and 3) an Exemplar Bridging (EB) block to adaptively revise the intermediate ViT representations by using the nearest exemplars. Finally, we complete the gene expression prediction task with a simple attention-based prediction block. Experiments on standard benchmark datasets indicate the superiority of our approach when comparing with the past state-of-the-art (SOTA) methods.

1. Introduction

Based on an editorial report of the Natural Methods [23], Spatial Transcriptomics (ST) is the future of studying disease because of its capabilities in measuring gene expression of fine-grained areas (i.e., different windows) of tissue slides. However, ST is in low throughput due to limitations in concurrent analysis for the candidate windows [1]. To accurately predict gene expression from each window of a tissue slide image (Fig. 1), this paper proposes a solution named Exemplar Guided Network (EGN), while allowing efficient and concurrent analysis.

In literature, previous works adopt end-to-end neural networks, namely STNet [13] and NSL [8], to establish a mapping between gene expression and the slide image window. STNet is a transfer learning-based approach that finetunes a pretrained DenseNet for this task. On the contrary, NSL maps the color intensity of the slide image window to gene expression by a single convolution operation. Though amenable to high throughput because of using neural networks, their prediction performance is inferior.

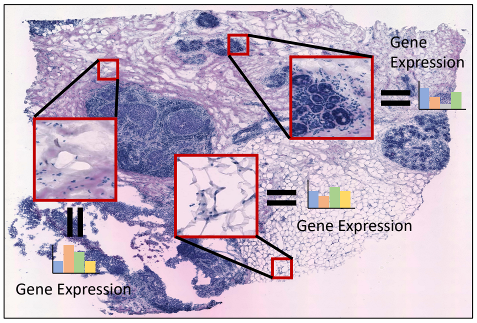

Figure 1: Overview of fields. Each fine-grained area (i.e., window) of a tissue slide image is with distinct gene expression. Here is an example, we have a tissue slide image with three windows, and each of the windows corresponds with expression of four different gene types. Our goal is to predict the gene expression of each window.

We investigate two important limitations of the existing approaches [13, 8]. *1)* Local feature aggregation: gene expression prediction can be considered as individually grouping the identified feature of each gene type from the slide image window. The long-range dependency among identified features is needed to reason about complex scenarios [35, 22], as those features are generally non-uniformly distributed across the slide image window (see Sec. 3 for details). STNet (i.e., a pure convolution approach) emphasizes local context during feature aggregation. It fails to bring interaction among features that are apart from each other. By experimenting with extensive SOTA architectures, we show that models with local feature aggregation achieve low performance when comparing to models with long-range dependencies. *2)* Vulnerable assumption: identifying gene expression by directly detecting the color intensity of the slide image window is vulnerable. By experimenting on standard benchmark datasets, we show this approach only works in extreme cases (for example, in the STNet dataset [13], tumour areas of slide image windows are usually purple, which benefits predicting tumour-related gene expression). This method shows a negative Pearson correlation coefficient (PCC) when evaluating the model with the least reliable gene expression prediction, i.e., PCC@F in Tab. 1.

In this paper, we propose an EGN framework to address the above limitations. EGN uses ViT [9] as a backbone and incorporates exemplar learning concepts for the gene expression prediction task. To enable unsupervised exemplar retrieval of a given slide image window, we use an extractor for defining a representation space to amount similarity between slide image windows. For brevity, we name the extractor output as a global view, which avoids confusion with the ViT representations. Note that, the representation ability of the global view is evidenced in the experiment section. Then, we have a ViT backbone to progressively construct presentations of the given slide image window under long-range dependency. Meanwhile, we have an EB block to revise intermediate ViT representations by the global views of the given slide image window, the global views of the exemplars, and the gene expression of exemplars. The global view of the given slide image window additionally serves as a prior to facilitating the long-range dependency, apart from the ViT backbone. To allow dynamic information propagation, we iteratively update the global views of the given slide image window and exemplars based on the status of the intermediate ViT representations. Semantically, the former update corresponds with 'what gene expression is known ?', and the latter corresponds with 'what is gene expression the model wants to know ?'. Finally, we have an attention-based prediction block to aggregate the exemplar-revised ViT representations and the global view of the given slide image window to predict gene expression.

Our contributions are summarised below:

- We propose an EGN framework, a ViT-based exemplar learning approach, to accurately predict gene expression from the slide image window.
- We propose an EB block to revise the intermediate ViT representation by using the nearest exemplars of the given slide image window.
- Experiments on two standard benchmark datasets demonstrate our superiority when comparing with SOTA approaches.

2. Related work

This section first review the study of gene expression prediction. Then, we summarise recent achievements of exemplar learning in both natural language processing and computer vision domains.

Gene Expression Prediction. Measuring gene expression is a basic process in developing novel treatments and monitoring human diseases [2]. To increase process accessibility, deep learning methods have been introduced to this task. Existing methods predict the gene expression either from DNA sequences [2] or slide images [13, 8, 30]. This paper explores the latter approach. Meanwhile, these image-based approaches are divided into two streams. First, Schmauch *et al.* [30] employs a multi-stage method including pretrained ResNet feature extractions [14] and a K-Means algorithm to model the Bulk RNA-Seq technique [21]. It measures the gene expression across cells within a large predefined area that is up to $10^5 \times 10^5$ pixels in a corresponding slide image [30]. However, this approach is ineffective for studies that require fine-grained gene expression information, such as tumour heterogeneity [11]. Second, on the contrary, He *et al.* [30, 15] and Dawood *et al.* [8] design a STNet and an NSL framework to predict the gene expression for each window (i.e., fine-grained area) of a slide image. This corresponds with the ST technique [24]. We model ST to predict gene expression, as this potentially solves the bulk RNA-Seq prediction task simultaneously [13]. For example, aggregation of gene expression predictions for each window across a slide image results in a bulk RNA-Seq prediction.

Exemplar Learning. The K-nearest neighborhood classifier is the most straightforward case of exemplar learning. It classifies the input by considering the class labels of the nearest neighbors. Exemplar learning is a composition of retrieval tasks and learning tasks [3]. It has been widely employed to increase the model capability by bringing in extra knowledge of similar exemplars [27, 7, 31, 10, 5, 33, 17, 12, 4, 20, 32, 28, 25]. Patro *et al.* [27], Chen *et al.* [7], Teney *et al.* [31], and Farazi *et al.* [10] refine the representation of given input with the closest match of text embedding and/or other visual embeddings to help the Visual Question Answering task. Borgeaud *et al.* [5], Wu *et al.* [33], Urvashi *et al.* [4], and Kelvin *et al.* [12] combine the examplar learning with transformers for language generation tasks. Their key idea is to enable a more knowledgeable attention mechanism, by picking exemplars as key and value representation. Similarly, Philippe *et al.* [4] incorporates the attention mechanism with exemplar learning to bring a real-time visual object tracking framework. Other applications of exemplar learning, including fact checking, fact completion, and dialogue, could be found in [20, 32, 28, 25]. However, most of the above approach does not apply to our task because of the domain shift. This paper investigates an application of exemplar learning in gene expression prediction from slide image windows. As a result, we devise an EB block to adapt exemplar learning into our task.

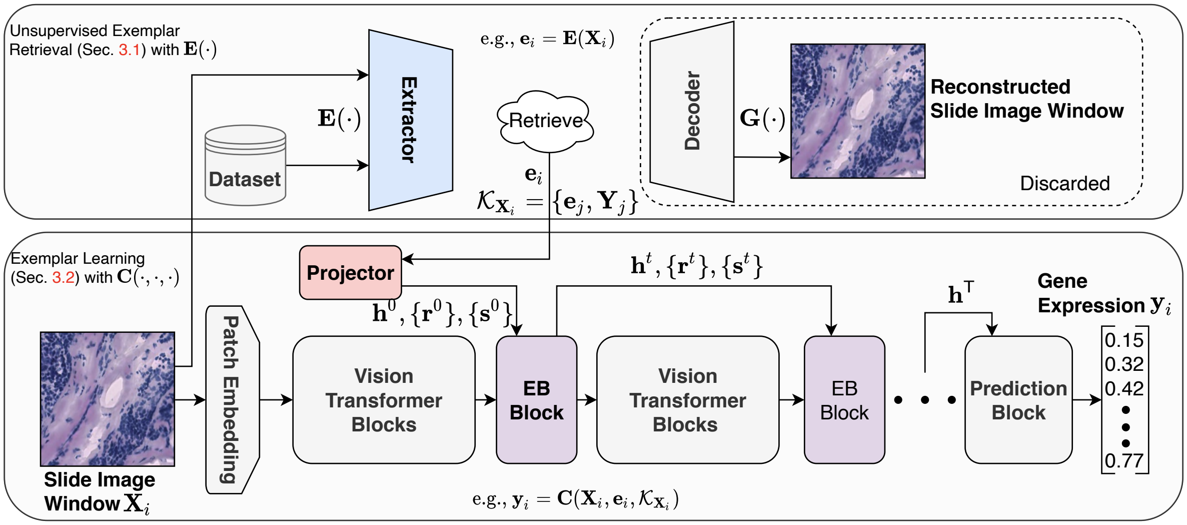
Unsupervised Exemplar Retrieval (Sec. 3.1) with $\mathbf{E}(\cdot)$
e.g., $\mathbf{e}_i = \mathbf{E}(\mathbf{X}_i)$
$\mathbf{E}(\cdot)$
Extractor
Dataset
Retrieve
$\mathbf{e}_i$
$\mathcal{K}_{\mathbf{X}_i} = \{\mathbf{e}_j, \mathbf{Y}_j\}$
Decoder
$\mathbf{G}(\cdot)$
Reconstructed Slide Image Window
Discarded
Exemplar Learning (Sec. 3.2) with $\mathbf{C}(\cdot,\cdot,\cdot)$
Projector
$\mathbf{h}^0, \{\mathbf{r}^0\}, \{\mathbf{s}^0\}$
$\mathbf{h}^t, \{\mathbf{r}^t\}, \{\mathbf{s}^t\}$
$\mathbf{h}^\mathsf{T}$
Gene Expression $\mathbf{y}_i$
Patch Embedding
Vision Transformer Blocks
EB Block
Vision Transformer Blocks
EB Block
Prediction Block
0.15
0.32
0.42
0.77
Slide Image Window $\mathbf{X}_i$
e.g., $\mathbf{y}_i = \mathbf{C}(\mathbf{X}_i, \mathbf{e}_i, \mathcal{K}_{\mathbf{X}_i})$

Figure 2: EGN framework. Our networks are trained in a two-stage manner. In the stage of unsupervised exemplar retrieval (Sec. 3.1), we learn an extractor $\mathbf{E}(\cdot)$ and a decoder $\mathbf{G}(\cdot)$ with image reconstruction objectives. After convergence, we use the extractor $\mathbf{E}(\cdot)$ with a distance metric for unsupervised exemplar retrieval. For example, given $\mathbf{X}_i$, we obtain the global view of $\mathbf{X}_i$, i.e., $\mathbf{e}_i$ and $\mathbf{e}_i = \mathbf{E}(\mathbf{X}_i)$, and construct the nearest exemplar set $\mathcal{K}_{\mathbf{X}_i} = \{\mathbf{e}_j, \mathbf{y}_j\}$, where $\mathbf{e}_j$ is the exemplar global view, and $\mathbf{y}_j$ is the exemplar gene expression. In the stage of exemplar learning (Sec. 3.2), we train a network $\mathcal{C}(\cdot,\cdot,\cdot)$ to predict gene expression $\mathbf{y}_i$ from $\mathbf{X}_i$, $\mathbf{e}_i$, and $\mathcal{K}_{\mathbf{X}_i}$. We use a vision transformer (ViT) as our backbone. The proposed EB block is interleaved with the vision transformer blocks. With a projector, we refine $\mathbf{e}_i$ and $\mathcal{K}_{\mathbf{X}_i} = \{\mathbf{e}_j, \mathbf{y}_j\}$ to $\mathbf{h}^0$, $\{\mathbf{r}^0\}$, and $\{\mathbf{s}^0\}$. Then, they are used by the EB block to revise the ViT patch representation and are updated to $\mathbf{h}^t$, $\{\mathbf{r}^t\}$, and $\{\mathbf{s}^t\}$, where $1 \le t \le \mathsf{T}$, and T is the number of layers. Finally, we have a prediction block that concatenates the refined global view $\mathbf{h}^\mathsf{T}$ of $\mathbf{X}_i$ and the attention-pooled ViT patch representation, to achieve the gene expression prediction task.

3. EGN Framework

Problem Formulation. We have a dataset collection of tissue slide images, where each image contains multiple windows, and each window is annotated with gene expression. For brevity, we denote the dataset collection as pairs of a slide image window $\mathbf{X}_i \in \mathbb{R}^{3\times\mathsf{H}\times\mathsf{W}}$and gene expression $\mathbf{y}_i \in \mathbb{R}^\mathsf{M}$, i.e., $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{\mathsf{N}}$, where N is the collection size, H and W are the height and width of $\mathbf{X}_i$, and M is the number of gene types. We aim to train a deep neural network model to predict $\mathbf{y}_i$ from $\mathbf{X}_i$.

From the ST study, there exist two main challenges. *1)* Long-range dependency: gene expression-related features are non-uniformly distributed over the slide image window $\mathbf{X}_i$. Refer to Fig. 3 of [13] for evidence. The interactions among these features are needed to group expression of the same gene type. *2)* Skewed gene expression distribution: the expression of some gene types has a skewed distribution, similar to the imbalance class distribution problem. This skewed distribution (see Fig. 3 for an example) poses challenges in predicting expression of these gene types. In this paper, we attempt to mitigate them by learning from similar exemplars.

Model Overview. With these motivations, we have designed the EGN framework (Fig. 2) containing an extractor network for exemplar retrieval and a prediction network that learns from retrieved exemplars. Networks are trained on two stages. *1)* Unsupervised exemplar retrieval (Sec. 3.1): we train an extractor $\mathbf{E}(\cdot)$ to construct the global view of the slide image window $\mathbf{X}_i$, i.e., $\mathbf{e}_i = \mathbf{E}(\mathbf{X}_i)$ and $\mathbf{e}_i \in \mathbf{R}^\mathsf{D}$, where D is the representation dimension. This global view $\mathbf{e}_i$ is used for retrieving the nearest exemplar of $\mathbf{X}_i$ to form a set $\mathcal{K}_{\mathbf{X}_i} = \{[\mathbf{e}_j, \mathbf{y}_j] \mid j \in \Upsilon_i\}$, where Υ_i contains indexes of nearest exemplars, $\mathbf{e}_j$ is a global view of a nearest exemplar, and $\mathbf{y}_j$ is the paired gene expression of the exemplar. *2)* Exemplar learning (Sec. 3.2): we train a model $\mathbf{C}(\cdot,\cdot,\cdot)$ that maps $\mathbf{X}_i$, $\mathbf{e}_i$, and $\mathcal{K}_{\mathbf{X}_i}$, to $\mathbf{y}_i$. Our $\mathbf{C}(\cdot,\cdot,\cdot)$ is based on a ViT backbone. We bring interactions between $\mathbf{e}_i$ and $\mathbf{e}_j$ to leverage $\mathbf{y}_j$ with a proposed EB block. Then, for accurately predicting $\mathbf{y}_i$, the updated $\mathbf{e}_i$ is used to revise intermediate ViT representations of $\mathbf{X}_i$. Note that, $\mathbf{e}_i$ also serves as a prior to facilitating the long-range interactions. With the introduction of exemplars, our framework dynamically benefits when predicting gene expression.

3.1. Unsupervised Exemplar Retrieval

To retrieve the exemplar of a given slide image $\mathbf{X}_i$ in an unsupervised manner, we propose to train an extractor (i.e., an encoder) with a decoder for image reconstructions. After

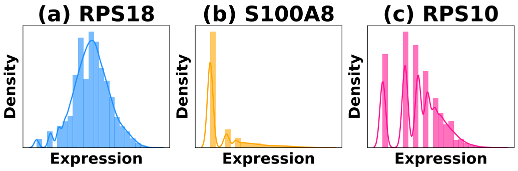

Figure 3: Gene expression distributions of STNet dataset [13]. Each gene expression is log-transformed. (a) is well distributed expression of gene RPS18. (b) and (c) are long-tail distributed expression of gene S100A8 and gene RPS10.

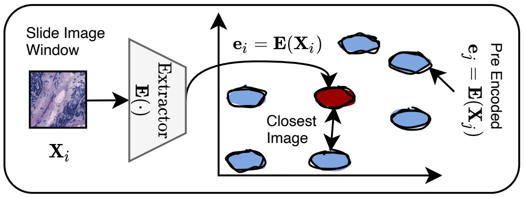
Slide Image Window
$\mathbf{X}_i$
Extractor $\mathbf{E}(\cdot)$
$\mathbf{e}_i = \mathbf{E}(\mathbf{X}_i)$
Closest Image
Pre Encoded $\mathbf{e}_j = \mathbf{E}(\mathbf{X}_j)$

Figure 4: Overview of the exemplar retrieval. For example, blue cycles are pre encoded global views. Given $\mathbf{X}_i$, we extract $\mathbf{e}_i$ with $\mathbf{E}(\cdot)$ and retrieve the nearest exemplars.

convergence, we discard the decoder, and couple the extractor with a distance metric to amount the similarity between each pair of slide image windows for the exemplar retrieval.

Method. The StyleGAN [16] applies a style code (i.e., a low dimension vector) to modulate convolution operations during image generations. As indicated by [34], the style code captures both fine-grained and high-level image attributes of a training dataset in an unsupervised manner. Meanwhile, in the style code, each scalar is highly disentangled, i.e., each of them tends to independently control an image attribute. This attribute independency is desired when amounting similarly between images. Thus, we borrow the style code for our unsupervised exemplar retrieval, where the style code is a global view of the given image input. Conventionally, to obtain the style code of a given slide image window, one needs to train a styleGAN and perform a GAN inversion [34] separately. Instead, we jointly train an encoder/extractor $\mathbf{E}(\cdot)$ (which replaces the GAN inversion) and a StyleGAN generator $\mathbf{G}(\cdot)$, with an image reconstruction objective. Afterwards, we can directly obtain the style code by a simple forward pass in the extractor $\mathbf{E}(\cdot)$.

Objective. We employ a least absolute deviation loss $\mathcal{L}_1$, a LPIPS loss $\mathcal{L}_{\text{LPIPS}}$, and a discriminator loss $\mathcal{L}_{\mathbf{F}}$ to optimize the image reconstruction, where $\mathbf{F}(\cdot)$ is a discriminator. Note that, the reconstruction indirectly impacts the representation ability of the style code. $\mathcal{L}_1$ constrains pixel-wise correspondence, while $\mathcal{L}_{\text{LPIPS}}$ and $\mathcal{L}_{\mathbf{F}}$ promotes a better reconstruction fidelity. We have

$$\begin{aligned}
&\mathcal{L}_1 = |\mathbf{X}_i - \mathbf{G}(\mathbf{E}(\mathbf{X}_i))|,\\
&\mathcal{L}_{\text{LPIPS}} = \|\phi(\mathbf{X}_i) - \phi(\mathbf{G}(\mathbf{E}(\mathbf{X}_i)))\|_2,\\
&\mathbf{L}_{\mathbf{F}} = u\Big(\mathbf{F}(\mathbf{X}_i)\Big) + u\Big(-\mathbf{F}\big(\mathbf{G}(\mathbf{E}(\mathbf{X}_i))\big)\Big),
\end{aligned}$$

where $\phi(\cdot)$ is a pretrained LPIPS network [37], and $u(\cdot)$ is a Softplus function [16]. Let $\mathbf{X}$ be a distribution of slide image windows. The overall objective is

$$\mathcal{L}_{\mathbf{E}} = \min_{\mathbf{G},\mathbf{E}} \max_{\mathbf{F}} \mathbb{E}_{\mathbf{X}_i \sim \mathbf{X}} \Big[\mathcal{L}_1 + \mathcal{L}_{\text{LPIPS}} + \mathcal{L}_{\mathbf{F}}\Big].$$

In a later section, we show $\mathbf{E}(\cdot)$ effectively encodes the gene expression-related features, by combining a pretrained $\mathbf{E}(\cdot)$ with a linear layer for gene expression prediction.

Exemplar Retrieval. After convergence, we use the representation space of the extractor $\mathbf{E}(\cdot)$ for the exemplar retrieval. Note that, we call the extractor output, the style code, as a global view. The slide image window $\mathbf{X}_i$ is encoded into a global view, by $\mathbf{e}_i = \mathbf{E}(\mathbf{X}_i)$. Then, we measure the similarity between each given pair of the slide image, for example, $\mathbf{X}_i$ and $\mathbf{X}_j$, with the Euclidean distance $\mathcal{L}_2$, for constructing the nearest exemplar set $\mathcal{K}_{\mathbf{X}_i}$. We empirically verify the optimal size of $\mathcal{K}_{\mathbf{X}_i}$ in Sec. 4.3. To generalize the model performance, we restrict that the candidate image pairs are from different patients. The overall process is presented in Fig. 4. In the experiment section, we compare the proposed exemplar retrieval with alternative retrieval approaches.

3.2. Exemplar Learning

Our model $\mathbf{C}(\cdot,\cdot,\cdot)$ is composed of a projector, a backbone, an EB block, and a prediction block. It maps a slide image window $\mathbf{X}_i$, a corresponding global view $\mathbf{e}_i$, and a retrieved nearest exemplar set $\mathcal{K}_{\mathbf{X}_i}$ to gene expression $\mathbf{y}_i$, i.e., $\mathbf{y}_i = \mathbf{C}(\mathbf{X}_i, \mathbf{e}_i, \mathcal{K}_{\mathbf{X}_i})$. Our model brings interactions between $\mathcal{K}_{\mathbf{X}_i}$ and $\mathbf{e}_i$ to progressively revise the intermediate representations of $\mathbf{X}_i$ in the ViT backbone.

Projector. The global views $\mathbf{e}_i$ and $\mathbf{e}_j \in \mathcal{K}_{\mathbf{X}_i}$ summarise a wide range of dataset-dependent attributes. We refine the global view to concentrate on the gene expression of interest by several multi-layer perceptrons (MLPs). Firstly, the global view $\mathbf{e}_i$ of the given slide image window $\mathbf{X}_i$ is projected by $\text{MLP}^0_h(\cdot)$. Secondly, for $\mathbf{e}_j \in \mathcal{K}_{\mathbf{X}_i}$, as the associated gene expression $\mathbf{y}_i \in \mathcal{K}_{\mathbf{X}_i}$ is available, we empower the refinement of $\mathbf{e}_j$ by $\mathbf{y}_j$. We concatenate $\mathbf{e}_j$ and $\mathbf{y}_j$ before feeding to $\text{MLP}^0_r(\cdot)$. Thirdly, with $\text{MLP}^0_s(\cdot)$, $\mathbf{y}_j$ is projected to the model dimension. We have

$$\mathbf{h}^0_i = \text{MLP}^0_h(\mathbf{e}_i),\ \ \mathbf{r}^0_j = \text{MLP}^0_r([\mathbf{e}^j, \mathbf{Y}j]),\ \ \mathbf{s}^0_j = \text{MLP}^0_s(\mathbf{y}_j),$$

where the superscript of $\mathbf{h}^0_i$, $\mathbf{r}^0_j$, and $\mathbf{s}^0_j$ denotes that they are initial refined global view, and $[\cdot,\cdot]$ is a concatenation operator. $\text{MLP}^0_h(\cdot)$ and $\text{MLP}^0_s(\cdot)$ are two layer perceptrons with a ReLU activation function. $\text{MLP}^0_r(\cdot)$ shares the same parameters with $\text{MLP}^0_h(\cdot)$ but has an extra linear layer on top of it.

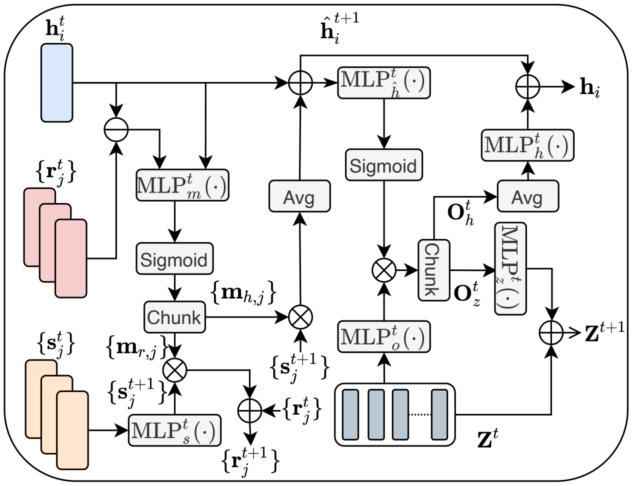

Figure 5: EB block architecture. The inputs are $\mathbf{h}_i$, $\{\mathbf{r}_j^t \mid j \in \Upsilon_i\}$, and $\{\mathbf{s}_j^t \mid j \in \Upsilon_i\}$, and $\mathbf{Z}^t$. Firstly, we project $\mathbf{s}_j$ to $\mathbf{s}_i^{t+1}$ and have interactions between $\mathbf{h}_i$ and $\mathbf{r}_j$. This interaction is assisted by $\mathbf{s}_i^{t+1}$ and obtains $\mathbf{h}^{\hat{t+1}}{}_i$ and $\mathbf{r}_i^{t+1}$. Secondly, $\hat{\mathbf{h}}_i^t$ is projected to the number of patches in the ViT backbone, to scale the magnitude of the ViT patch representation $\mathbf{Z}^t$. By their interactions, $\hat{\mathbf{h}}_i^{t+1}$ and $\mathbf{Z}^t$ are updated to $\mathbf{h}_i^{t+1}$ and $\mathbf{Z}^{t+1}$.

Backbone. We use the ViT as our backbone [9]. It has a patch embedding layer and vision transformer blocks. The patch embedding layer tiles the slide image window $\mathbf{X}_i$ into non-overlapping patches before flattening and feeding to a linear layer. The outputs are $\mathbf{Z}^0 = [\mathbf{z}_l^0 \mid l \in 1, \cdots, \mathsf{L}]$, where L is the total number of patches, and $\mathbf{Z}^0$ is a matrix representation of the projected patches. Each vision transformer block composes of a self-attention layer and a feed-forward layer, which performs global interaction among $\mathbf{Z}^0$. Assuming there are T layers, and let $t \in [1, \cdots, \mathsf{T}]$. We denote the l^{th} patch representation at t^{th} layer as $\mathbf{z}_l^t$.

EB Block. This block is interleaved with the vision transformer blocks, which brings $\mathbf{s}_j^t$, i.e., knowledge about gene expression of the nearest exemplar, to each ViT patch representation $\mathbf{z}_l^t$. Firstly, we project $\mathbf{s}_j^t$ (i.e., a projection of gene expression of the exemplars) to $\mathbf{s}_j^{t+1}$. With $\mathbf{s}_j^{t+1}$, we have interactions between $\mathbf{h}_i^t$ and $\mathbf{r}_j^t$ (i.e., refined global views) to obtain $\hat{\mathbf{h}}_i^{t+1}$ and $\mathbf{r}_j^{t+1}$. As $\hat{\mathbf{h}}_i^{t+1}$ and $\mathbf{r}_j^{t+1}$ are initialized from the same extractor $\mathbf{E}(\cdot)$, their interactions could serve as a bridge for the revision of the ViT patch representation. Secondly, considering $\hat{\mathbf{h}}_i^{t+1}$ is initially extracted by the reconstruction based-network $\mathbf{E}(\cdot)$, $\hat{\mathbf{h}}_i^{t+1}$ is expected to contain spatial information about the input image distribution. With $\hat{\mathbf{h}}_i^{t+1}$, we revise the ViT patch representation $\mathbf{z}_l^t$, obtaining $\mathbf{h}_i^{t+1}$ and $\mathbf{z}_l^{t+1}$. This revision process brings interactions between $\hat{\mathbf{h}}_i^{t+1}$and $\mathbf{z}_l^t$ to propagate information about possible gene expression (which receives from the exemplars) and patch representation statuses. The overall process is shown in Fig. 5.

To do so, firstly, we concatenate $\mathbf{h}_i^t$ and the difference $\mathbf{h}_i^t - \mathbf{r}_j^t$ to bring interactions between $\mathbf{h}_i^t$ and $\mathbf{r}_j^t$, with $\mathrm{MLP}_m^t(\cdot)$. We chunk the outputs to $\mathbf{m}_{h,j}$ and $\mathbf{m}_{r,j}$. Then, they are fused with $\hat{\mathbf{s}}_j$ to retrieve gene expression from $\mathbf{s}_j^{t+1}$ (a projection of $\mathbf{s}_j^t$). Semantically, $\mathbf{m}_{h,j}$ summarises 'the existing knowledge of gene expression', and $\mathbf{m}_{r,j}$ tells 'the desired knowledge'. Mathematically, we have

$$\begin{aligned}
&\hat{\mathbf{h}}_i^{t+1} = \mathbf{h}_i^t + \mathrm{Avg}([\mathbf{m}_{h,j} \cdot \mathbf{s}_j^{t+1} \mid \forall j \in \Upsilon_i]),\\
&\mathbf{r}_i^{t+1} = \mathbf{r}_i^t + \mathbf{m}_{r,j} \cdot \mathbf{s}_j^{t+1},\\
&\mathbf{m}_{h,j}, \mathbf{m}_{r,j} = \mathrm{Chunk}(\sigma(\mathrm{MLP}_m^t(\mathbf{h}_i^t, \mathbf{h}_i^t - \mathbf{r}_j^t))),\\
&\mathbf{s}_j^{t+1} = \mathrm{MLP}_s^t(\mathbf{s}_j^t)
\end{aligned}$$

where $\mathrm{MLP}_m^t(\cdot)$ is a Multi-layer perceptron, the $\mathrm{Chunk}(\cdot)$ operator equally splits the input into two outputs, and $\sigma(\cdot)$ is a Sigmoid function. To avoid notation overloading, we remain to use $\mathrm{MLP}_s^t(\cdot)$ as a single-layer perceptron. Secondly, $\hat{\mathbf{h}}_i^{t+1}$ and $\mathbf{Z}^t$ reciprocate each other by scaling the magnitude of each patch representation $\mathbf{z}_l^t \in \mathbf{Z}^t$:

$$\begin{aligned}
&\mathbf{Z}^{t+1} = \mathbf{Z}^t + \mathrm{MLP}_z^t(\mathbf{O}_z^t),\\
&\mathbf{h}_i^{t+1} = \hat{\mathbf{h}}_i^{t+1} + \mathrm{MLP}_h^t(\mathrm{Avg}(\mathbf{O}_h^t)),\\
&\mathbf{O}_h^t, \mathbf{O}_z^t = \mathrm{Chunk}(\mathrm{MLP}_o^t(\mathbf{Z}^t) \cdot \sigma(\mathrm{MLP}_{\hat{h}}^t(\hat{\mathbf{h}}_i^{t+1}))),
\end{aligned}$$

where $\mathrm{MLP}_z^t(\cdot)$, $\mathrm{MLP}_h^t(\cdot)$, $\mathrm{MLP}_o^t(\cdot)$, and $\mathrm{MLP}_{\hat{h}}^t(\cdot)$ are single-layer perceptrons. $\mathbf{O}_h^t$ and $\mathbf{O}_z^t$ are the fusion of the ViT patch representations $\mathbf{Z}^t$ and $\hat{\mathbf{h}}_i^{t+1}$. Note that, the output dimension of $\mathrm{MLP}_o^t(\cdot)$ is the number of patches in the ViT, and each scalar of the output is used to scale a corresponding patch from $\mathrm{MLP}_o^t(\mathbf{Z}^t)$. By scaling the magnitudes of each ViT patch representation, we indirectly inject the knowledge about gene expression of the exemplars, while updating the refined global view $\hat{\mathbf{h}}_i^{t+1}$ to facilitate both interactions with the exemplars and the ViT representation in following layers. One may note that the current operation is single-head based, which can be extended to a multi-head operation by expanding the output dimension of $\mathrm{MLP}_o^t(\cdot)$ in practice.

Prediction Block. Following [38], each ViT patch representation shows different priorities toward the gene expression prediction task. We apply an attention pooling layer $\mathrm{AttPool}(\cdot)$ to aggregate $\mathbf{z}_l^{\mathsf{T}} \in \mathbf{Z}^{\mathsf{T}}$. Moreover, $\mathbf{h}_i^{\mathsf{T}}$ is a progressively refined global view of the slide image window $\mathbf{X}_i$, which captures high-level attributes of $\mathbf{X}_i$. We concatenate $\mathbf{h}_i^{\mathsf{T}}$ and $\mathrm{AttPool}(\mathbf{Z}^{\mathsf{T}})$ for the prediction. We have

$$\mathbf{y}_i = \mathrm{MLP}_g([\mathbf{h}_i^{\mathsf{T}}, \mathrm{AttPool}(\mathbf{Z}^{\mathsf{T}})]),$$

Table 1: Quantitative gene expression prediction comparisons with SOTA methods in STNet dataset and 10xProteomic dataset. We bold the best results. We use '-' to denote unavailable results. Models are evaluated by four-fold cross-validation and three-fold cross-validation in the above datasets. Our proposed EGN framework consistently outperforms the SOTA methods in $\text{MAE}_{\times 10^1}$, $\text{PCC@F}_{\times 10^1}$, $\text{PCC@S}_{\times 10^1}$ and $\text{PCC@M}_{\times 10^1}$ for both datasets. The CycleMLP finds the best $\text{MSE}_{\times 10^2}$ in the 10xProteomic dataset.

	Methods	STNet [13]	NSL [8]	ViT [9]	CycleMLP [6]	MPViT [19]	Retro [5]	ViTExp	Ours
STNet Dataset	$\text{MSE}_{\times 10^2}$	4.52	-	4.28	4.41	4.49	4.53	4.46	**4.10**
	$\text{MAE}_{\times 10^1}$	1.70	-	1.67	1.68	1.70	1.71	1.69	**1.61**
	$\text{PCC@F}_{\times 10^1}$	0.05	-0.71	0.97	1.11	0.91	0.99	0.87	**1.51**
	$\text{PCC@S}_{\times 10^1}$	0.92	0.25	1.86	1.95	1.54	1.74	1.72	**2.25**
	$\text{PCC@M}_{\times 10^1}$	0.93	0.11	1.82	1.91	1.69	1.79	1.74	**2.02**
10xProteomic Dataset	$\text{MSE}_{\times 10^2}$	12.40	-	7.54	**4.69**	5.45	5.25	5.04	5.49
	$\text{MAE}_{\times 10^1}$	2.64	-	2.27	**1.55**	1.56	1.65	1.66	**1.55**
	$\text{PCC@F}_{\times 10^1}$	1.25	-3.73	5.11	5.88	6.40	5.46	5.59	**6.78**
	$\text{PCC@S}_{\times 10^1}$	2.26	1.84	4.64	6.60	7.15	6.35	6.36	**7.21**
	$\text{PCC@M}_{\times 10^1}$	2.15	0.25	4.90	6.32	6.84	6.04	6.00	**7.07**

where MLP_g is a single-layer perception.

Objective. We optimize $\mathbf{C}(\cdot,\cdot,\cdot)$ with mean squared loss (i.e., the Euclidean distance) $\mathcal{L}_2$ and batch-wise PCC $\mathcal{L}_{ppc}$. We have

$$\mathcal{L}_{total} = \mathcal{L}_2 + \mathcal{L}_{ppc}$$

4. Experiments

Datasets. We perform experiments in the publicly available STNet dataset [13] and 10xProteomic datasets[1]. The STNet dataset contains roughly 30,612 pairs of the slide image window and the gene expression. This dataset covers 68 slide images from 23 patients. Following [13], we target predicting expression of 250 gene types that have the largest mean across the dataset. The 10xProteomic dataset has 32,032 slide image windows and gene expression pairs from 5 slide images. We select target gene types in the same way as the STNet dataset. We apply log transformation and min-max normalization to the target gene expression. Note that, our normalization method is different from [13] (they use log transformation and a custom normalization, i.e., dividing the expression of each gene type by the sum of expression from all gene types, for each slide image window). Our normalization method allows independent analysis of gene expression prediction.

4.1. Experimental set-up

Baseline methods. We compare with extensive SOTA methods from gene expression prediction, ImageNet classification benchmarks, and exemplar learning.

- STNet [13] and NSL [8]. They are the SOTA methods in gene expression prediction.
- ViT [9], MPViT [19] and CycleMLP [6]. We use the SOTA ImageNet classification methods in our task. They are strong baselines in our task. Specifically, we use ViT-B, MPViT-Base, and CycleMLP-B2. Please refer to [9, 19, 6] for details.
- Retro [5] and ViTExp. We explore the SOTA exemplar learning methods. However, Retro is originally developed for natural language processing. We adapt it by providing the extractor output $\mathbf{e}$ as the exemplar representations. ViTExp directly concatenates the exemplar representations to the ViT patch representation. The exemplar representations are added with an embedding to be differentiated from the patch representation of the slide image window. Both Retro and ViTExp are based on the ViT-B architecture [9].

Evaluation metrics. We evaluate the proposed methods and alternative baselines with PCC, mean squared error (MSE), and mean absolute error (MAE). We use PCC@F, PCC@S, and PCC@M to denote the first quantile, median, and mean of the PCC. The PCC@F verifies the least performed model predictions. The PCC@S and PCC@M measure the median and mean of correlations for each gene type, given predictions and GTs for all of the slide image windows. Meanwhile, the MSE and MAE measure the sample-wise deviation between predictions and GTs of each slide image window for each gene type. Note that, for MSE and MAE, the lower value means better performance. In contrast, for PCC@F, PCC@S, and PCC@M, the higher value indicates better performance.

Implementation details[2]. The architectures and optimization settings for our extractor follow [36]. For the exemplar learning, we implement EGN by using the *Pytorch* frame-

[1] https://www.10xgenomics.com/resources/datasets

[2] Codes are available at: https://github.com/Yan98/EGN

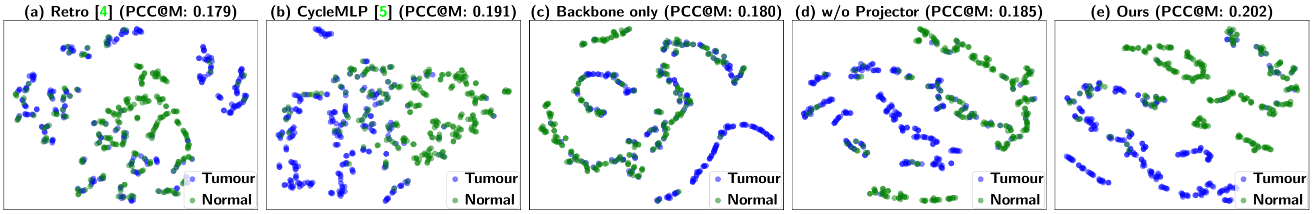

Figure 6: Quantitative evaluation of the top performed models from Tab. 1 and Tab. 4. We employ t-SNE [29] for dimension reduction of model latent space. We use the extra labels (i.e., tumour and normal) from the STNet dataset for annotations.

work [26]. EGN is trained from scratch for 50 epochs with batch size 32. We set the learning rate to 5×10^{-4} with a cosine annealing scheduler. Our weight decay is 1×10^{-4}. We use a ViT backbone with the following settings: patch size 32, embedding dimension 1024, feedforward dimension 4096, attention heads 16, and depth 8. We interleave the proposed EB block with the ViT backbone at frequency 2, where each block has 16 heads and dimension 64, namely the block has 16 unique operations with dimension 64. Our EB block uses 9 nearest exemplars. All the experiments are conducted in 2 NVIDIA Tesla P100 GPUs.

4.2. Experimental results

We compare our EGN framework with the baselines on the STNet dataset and the 10xProteomic dataset (Tab. 1). As the gene expression prediction task emphasizes capturing the relativity variation, we bias on the PCC-related evaluation metrics, i.e., PCC@F, PCC@S, and PCC@M. Our EGN consistently achieves the SOTA performance in MAE, PCC@F, PCC@S, and PCC@M. Our findings are as follows: *1)* It's worth noting that Retro and ViTExp use the ViT as the backbone. However, their performance is lower than the ViT. Both Retro and ViTExp use attention-based operations to directly have interactions between the ViT representations and the global views of the exemplars, when comparing with our EB block. This validates the necessity of using the global view of the input slide image window as a bridge to introduce the knowledge of exemplars to the ViT representations; *2)* CycleMLP and MPViT, the SOTA methods in the ImageNet classification task, lead to the second-best performance in the STNet dataset and 10xProteomic dataset in PCC-related evaluations. Meanwhile, CycleMLP finds the best MSE in 10xProteomic. Note that, PCC-related evaluation metrics are most important in our task. By using the exemplars, our model that uses the vanilla ViT backbone outperforms them with a reason marginal in PCC-related evaluation metrics, while overall achieving similar MSE and MAE with them; *3)* The PCC@F of our model significantly outperforms the baseline methods. This metric evaluates the worst model capability, by calculating the first quantile of PCC across all gene types. Note that, the majority of gene types covered by the first quantile have skewed expression distributions, which are the most challenging part of the prediction task. Our method has 0.038 - 0.040 higher than the second-best method in PCC@F; and *4)* STNet and NSL fail to achieve good performance. Again, the gene expression-related feature is usually non-uniformly distributed across the slide image window input. They do not have long-range interactions to capture the expression of the same gene type from these features. Moreover, NSL even shows a negative correlation with PCC@F. This validates our claims that predicting gene expression directly with the color intensity is vulnerable, and it is only feasible in extreme cases (recall the example of tumour-related gene expression in Sec. 1).

Quantitative Evaluation. We present the latent space visualization (Fig. 6), by considering the top performed models from Tab. 1 and Tab. 4 (see details of 'Backbone only' and 'w/o Projector' settings in Sec. 4.3). Note that, the 'Backbone only' setting has a smaller number of parameters than a regular ViT-B. The extra labels (i.e., tumour and normal) from the STNet dataset are used for annotations. To enable a clean visualization, we randomly sample 256 representations of the slide image window for each label, i.e., tumour and normal. By gradually using the proposed components (from Fig. 6 (c) to Fig. 6 (e)), i.e., introduction exemplars to our model, our method sufficiently separates the tumour representations from the normal representations, by having an improved gene expression prediction than alternative approaches.

4.3. Ablation study

We study the capability of each model component by conducting a detailed ablation study in the STNet dataset.

Extractor Capability. We explore alternative approaches for retrieving exemplars (Tab. 2). We compare with representations from AlexNet [18] and ResNet50 [14]. Note that, they are pretrained on ImageNet, and our $\mathbf{E}(\cdot)$ is learned in an unsupervised manner. We explore diverse distance matrices including LPIPS [37], $\mathcal{L}_2$, $\mathcal{L}_1$, and $cosine$ (i.e., cosine similarity). We have the following findings: *1)* LPIPS distance retrieved exemplars lead to bad model performance. This distance is trained based on the human perception of regular images, where these images are different from the

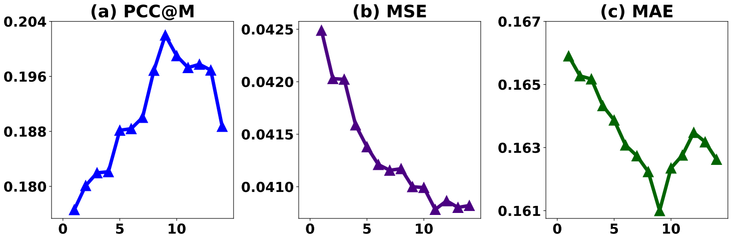

Figure 7: Ablation study on number of exemplar used in EB block. This number is varied from one to fifteen, and we present PCC@M, MSE, and MAE in sub-figure (a), (b), and (c).

Table 2: Ablation study on exemplar retrieval. We compare with representation from AlexNet and ResNet50 for exemplar retrieval.

Feature Space	Distance Metrics	$\text{MSE}_{\times 10^2}$	$\text{MAE}_{\times 10^1}$	$\text{PCC@M}_{\times 10^1}$
$\mathbf{E}(\cdot)$	$\mathcal{L}_2$	**4.10**	**1.61**	**2.02**
$\mathbf{E}(\cdot)$	$\mathcal{L}_1$	4.13	1.64	1.80
$\mathbf{E}(\cdot)$	*cosine* .	4.24	1.63	1.83
AlexNet	LPIPS	4.46	1.70	1.74
ResNet50	$\mathcal{L}_2$	4.12	1.62	1.89
ResNet50	$\mathcal{L}_1$	4.13	1.62	1.83
ResNet50	*cosine*	4.16	1.64	1.94

Table 3: Ablation study on the EB block.

Heads	Head Dim	Frequency	$\text{PCC@M}_{\times 10^1}$
4	32	2	1.84
4	64	2	1.76
4	64	3	1.78
8	32	2	1.85
8	64	2	**2.02**
8	64	3	1.88
16	32	2	1.86
16	64	2	1.82
16	64	3	1.80

Table 4: Ablation study on model architectures.

Settings	$\text{MSE}_{\times 10^2}$	$\text{MAE}_{\times 10^1}$	$\text{PCC@M}_{\times 10^1}$
Pretrained $\mathbf{E}(\cdot)$	4.85	1.76	1.56
Backbone only	4.31	1.67	1.80
w/o EB Block	4.65	1.72	1.70
w/o projector	4.22	1.63	1.85
Ours	**4.10**	**1.55**	**2.02**

slide image windows. Thus, it retrieves bad exemplars and damages the model performance; *2)* Retrieving exemplars with ResNet50 lead to decent performance, though the pretrained ResNet50 lacks gene expression-related knowledge. This retrieval method complements the knowledge about image textures learned from the ImageNet for our task, which enhances the versatility of our model representation and verifies our model robustness; and *3)* Using the extractor $\mathbf{E}(\cdot)$ with $\mathcal{L}_2$ distance achieves the best performance.

EB Block Settings. Firstly, we present PCC@M (Fig. 7(a)), MSE (Fig. 7(b)), and MAE (Fig. 7(c)), by varying the number of exemplars used in the EB block from one to fifteen. Having nine exemplars find the best PCC@M and MAE, and we have the best MSE by using ten exemplars. Again, our task emphasizes capturing relative changes. Thus, our final recommendation is to use nine exemplars. Secondly, we ablate the EB block architectures and the frequency of interleaving with the vision transformer blocks (Tab. 3). This balances the model's capability in structuring the ViT representations and receiving knowledge from the exemplars. We have the best PCC@M, by setting heads, head dimension, and frequency to 8, 64, and 2.

Model Architectures. We present the performance of 'Pretrained $\mathbf{E}(\cdot)$', 'Backbone only', 'w/o EB block', and 'w/o projector' in Tab. 4. 'Pretrained $\mathbf{E}(\cdot)$' adds one trainable linear layer to $\mathbf{E}(\cdot)$, while freezing weights of $\mathbf{E}(\cdot)$ for gene expression prediction. This is also known as linear probing in the literature. 'Backbone only' uses the ViT backbone architecture. Note that, we use a non-regular ViT architecture which is determined by a binary search. 'w/o EB block' is equivalent to concatenating the pooled ViT representations with the global view for predicting gene expression. 'w/o projector' replaces the projector with a linear layer to unify the dimension. Our findings are as follows: *1)* The 'Pretrained $\mathbf{E}(\cdot)$' setting is a strong baseline. Note that, this setting has better performance than the STNet and NSL (Tab. 1). Our $\mathbf{E}(\cdot)$ captures gene expression-related features in an unsupervised manner, showing the potential of our unsupervised exemplar retrieval; *2)* The 'w/o EB block' setting achieves worse performance than the 'Backbone only' setting. Concatenating the global view for gene expression prediction with a single linear layer disables interactions between the global view and the ViT representations. This potentially causes inconsistent behaviors and results in poor performance; and *3)* With all proposed components, we have the best performance.

5. Conclusion

This paper proposes an EGN framework to accurately predict gene expression from each fine-grained area of tissue slide image, i.e., different windows. EGN uses the ViT as a backbone while integrating with exemplar learning concepts. We first have an extractor to retrieve the exemplars of the given tissue slide image window in an unsupervised manner. Then, we propose an EB block to progressively revise the ViT representation by reciprocating with the nearest exemplars. With extensive experiments, we demonstrate the superiority of the EGN framework over the SOTA methods. EGN is promising to facilitate studies on diseases and novel treatments with accurate gene expression prediction.

References

[1] Michaela Asp, Joseph Bergenstråhle, and Joakim Lundeberg. Spatially resolved transcriptomes—next generation tools for tissue exploration. *BioEssays*, 42:1900221, 05 2020.

[2] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph Ledsam, Agnieszka Grabska-Barwinska, Kyle Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18:1196–1203, 10 2021.

[3] Miguel Bautista, Artsiom Sanakoyeu, Ekaterina Sutter, and Björn Ommer. Cliquecnn: Deep unsupervised exemplar learning. 08 2016.

[4] Philippe Blatter, Menelaos Kanakis, Martin Danelljan, and Luc Gool. Efficient visual tracking with exemplar transformers. 12 2021.

[5] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. 12 2021.

[6] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *CoRR*, abs/2107.10224, 2021.

[7] Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Pan, Zonggang Yuan, and Huajun Chen. *Zero-Shot Visual Question Answering Using Knowledge Graph*, pages 146–162. 09 2021.

[8] Muhammad Dawood, Kim Branson, Nasir Rajpoot, and Fayyaz ul Amir Afsar Minhas. All you need is color: Image based spatial gene expression prediction using neural stain learning. 08 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[10] Moshiur Farazi, Salman Khan, and Nick Barnes. From known to the unknown: Transferring knowledge to answer questions about novel visual and semantic concepts. *Image and Vision Computing*, 103:103985, 08 2020.

[11] Marco Gerlinger, Andrew Rowan, Stuart Horswell, James Larkin, David Endesfelder, Eva Gronroos, Pierre Martinez, Nik Matthews, Aengus Stewart, Patrick Tarpey, Ignacio Varela, Benjamin Phillimore, Sharmin Begum, Neil Mcdonald, Adam Butler, David Jones, Keiran Raine, Calli Latimer, Claudio Santos, and Charles Swanton. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *The New England journal of medicine*, 366:883–92, 03 2012.

[12] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. 02 2020.

[13] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Ake Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature Biomedical Engineering*, 4:1–8, 08 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[15] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Weinberger. Densely connected convolutional networks. 07 2017.

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. Computer Vision Foundation / IEEE, 2020.

[17] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. 10 2019.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.

[19] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. *CoRR*, abs/2112.11010, 2021.

[20] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. 05 2020.

[21] Xinmin Li and Cun-Yu Wang. From bulk, single-cell to spatial rna sequencing. *International Journal of Oral Science*, 13, 12 2021.

[22] Hui Lin, Zhiheng Ma, Rongrong Ji, Yaowei Wang, and Xiaopeng Hong. Boosting crowd counting via multifaceted attention. *CoRR*, abs/2203.02636, 2022.

[23] Vivien Marx. Method of the year: spatially resolved transcriptomics. *Nature Methods*, 18:9–14, 01 2021.

[24] Vivien Marx. Method of the year: spatially resolved transcriptomics. *Nature Methods*, 18:9–14, 01 2021.

[25] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh Khapra. Towards exploiting background knowledge for building conversation systems. 09 2018.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett,

editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[27] Badri Patro and Vinay Namboodiri. Deep exemplar networks for vqa and vqg. 12 2019.

[28] F Petroni, PSH Lewis, A Piktus, Tim Rocktäschel, Yuxiang Wu, AH Miller, and Sebastian Riedel. How context affects language models' factual predictions. 06 2020.

[29] Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing time-dependent data using dynamic t-sne. In Enrico Bertini, Niklas Elmqvist, and Thomas Wischgoll, editors, *18th Eurographics Conference on Visualization, EuroVis 2016 - Short Papers, Groningen, The Netherlands, June 6-10, 2016*, pages 73–77. Eurographics Association, 2016.

[30] Benoit Schmauch, Alberto Romagnoni, Elodie Pronier, Charlie Saillard, Pascale Maillé, Julien Calderaro, Aurélie Kamoun, Meriem Sefta, Sylvain Toldo, Mikhail Zaslavskiy, Thomas Clozel, Matahi Moarii, Pierre Courtiol, and Gilles Wainrib. A deep learning model to predict rna-seq expression of tumours from whole slide images. *Nature Communications*, 11, 08 2020.

[31] Damien Teney and Anton Hengel. Zero-shot visual question answering. 11 2016.

[32] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. 04 2018.

[33] Yuhuai Wu, Markus Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. 03 2022.

[34] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12863–12872. Computer Vision Foundation / IEEE, 2021.

[35] Shuo-Diao Yang, Hung-Ting Su, Winston H. Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 869–877. IEEE, 2021.

[36] Yan Yang, Md. Zakir Hossain, Tom Gedeon, and Shafin Rahman. S2FGAN: semantically aware interactive sketch-to-face translation. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 3162–3171. IEEE, 2022.

[37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018.

[38] Ke Zhu and Jianxin Wu. Residual attention: A simple but effective method for multi-label recognition, 10 2021.