# Supplementary Material: Active Data Enrichment by Learning What to Annotate in Digital Pathology

No Author Given

No Institute Given

## 1 Implementation Details

Resizing of any selected region from a WSI to a specific size in pixels was performed by choosing the closest larger pyramidal representation of the region using Python OpenSlide and then resizing using Python OpenCV resize method with default interpolation. All code was written Python using PyTorch, Scikit-learn, and SciPy libraries.

**Unsupervised Data Enrichment.** When using an ImageNet pre-trained feature extractors, we normalized the images using ImageNet mean (0.485, 0.456, 0.406) and standard deviation (0.229, 0.224, 0.225) constants before passing the images into the feature extractor. However, we followed the pre-processing steps described by Li et al. [1] when using their networks pre-trained on lung subset of TCGA: used Instance Normalization instead of Batch Normalization inside the ResNet, and did not normalize the inputs. In both cases, we resized the smaller side of the query region to 224 pixels and cropped the central $224 \times 224$ square to extract the features.

**Supervised Active Data Enrichment.** We only used ResNet-18 pre-trained on patches extracted from the lung subset of TCGA at 10x magnification by Li et al. [1]. We used the same pre-processing as described in Method 1 for this feature extractor. Before training or validation we resized the smaller side of the every region to 224 pixels. During training, we shuffled the dataset at every epoch, cropped a random $224 \times 224$ square from an image and used a random horizontal flip (p=0.5) to augment the data. During validation and testing we used the central $224 \times 224$ square without any flips.

## 2 Data Distribution

| Batch | WSIs | ROIs | Keratinization | Acinar |
|---|---|---|---|---|
| (1) Hamamatsu, 20x | 20 | 145 | 1 | 34 |
| (2) Ventana DP200, 20x | 17 | 120 | 15 | 20 |

**Table 1:** Data distribution between the first two batches. Two patterns were chosen for performing the experiments (unsupervised: keratinization, supervised: acinar).

| Set | Batch | Samples | Yes | No | Not Sure | Yes Proportion |
|---|---|---|---|---|---|---|
| Train | 1 | 86 | 20 | 31 | 35 | 0.233 |
| Validation | 1 | 59 | 14 | 21 | 24 | 0.237 |
| Test | 2 | 60 | 10 | 46 | 4 | 0.167 |
| 10 Ranked Pool | 2 | 10 | 4 | 5 | 1 | 0.4 |
| Train + 10 Ranked Pool | 1+2 | 96 | 24 | 36 | 36 | 0.25 |
| 20 Ranked Pool | 2 | 20 | 6 | 13 | 1 | 0.3 |
| Train + 20 Ranked Pool | 1+2 | 106 | 26 | 44 | 36 | 0.245 |
| 30 Ranked Pool | 2 | 30 | 8 | 21 | 1 | 0.267 |
| Train + 30 Ranked Pool | 1+2 | 116 | 28 | 52 | 36 | 0.241 |
| Pool | 2 | 60 | 10 | 46 | 4 | 0.167 |
| Train + Pool | 1+2 | 146 | 30 | 77 | 39 | 0.205 |

**Table 2:** Data distribution for the acinar pattern.

## 3 Mathematical Formulation of the Proposed Ranking Curves; Expected Normalized AUC Calculation

### 3.1 Ranking Curve Definition

- Let inputs $\{x_1, ..., x_N\}$ be already ranked by some ranking method that aims to put the inputs with the positive label (1) before (at lower indexes) the inputs with the negative label (0).
- Let $\{y_1, ..., y_N\}$ such that $y_i \in \{0, 1\}$ for all $i \in \{1, ..., N\}$ be the true labels with the total number of positive labels equal to $t$
- Let $p = t/N$ be the proportion of positive examples in our data.

$$t = \sum_{1}^{N} y_i \tag{1}$$

We define the ranking curve proportion $R(n)$ as follows for each $n \in \{1, ..., N\}$

$$R(n) = \frac{\sum_1^n y_i}{min\{n,t\}} \tag{2}$$

- The numerator is the number of positive examples in the top-$n$ ranked samples: $\sum_1^n y_i$
- The denominator is the total number of positive samples that we could have possibly retrieved by taking $n$ examples, which is $min\{n,t\}$ since we can neither retrieve more positive examples then there are in total ($t$), nor can we retrieve more examples than we took ($n$)

### 3.2 Expected Performance on a Random Sample

If the examples are not in any particular order, the positive examples will be uniformly distributed within the indexes $\{1,...,N\}$ giving each index a probability $p = t/N$ of having a positive example. This means that for any sample of $n$ randomly chosen examples we have:

$$\mathbf{E}[\sum_1^n y_i] = p \times n \tag{3}$$

Which results in:

$$\begin{aligned}
\mathbf{E}[R(n)] &= \mathbf{E}[\frac{\sum_1^n y_i}{min(n,t)}] \\
&= \frac{p \times n}{min\{n,t\}} \\
&= \frac{(t/N) \times n}{min\{n,t\}} \\
&= \begin{cases} t/N, & 1 \le n \le t \\ n/N, & t < n \le N \end{cases} \\
&= \begin{cases} p, & 1 \le n \le t \\ n/N, & t < n \le N \end{cases}
\end{aligned} \tag{4}$$

Plotted as a function of $n$ from 1 to $N$ this results in a horizontal line for $1 \le n \le t$ and a line with $1/N$ gradient for $t \le n \le N$. The two lines have the common point when $n = t$, where $\mathbf{E}[R(n)] = t/N = p$.

$$AUC = (1 \times (N-1))/2 + ((t-1) \times p)/2 \tag{5}$$

- $(N-1)/2$ is the area of the right triangle between (1, 0), (N, 0), (N, 1).
- $((t-1) \times p)/2$ is the area of the right triangle between (1, 0), (1, p), (t, p).

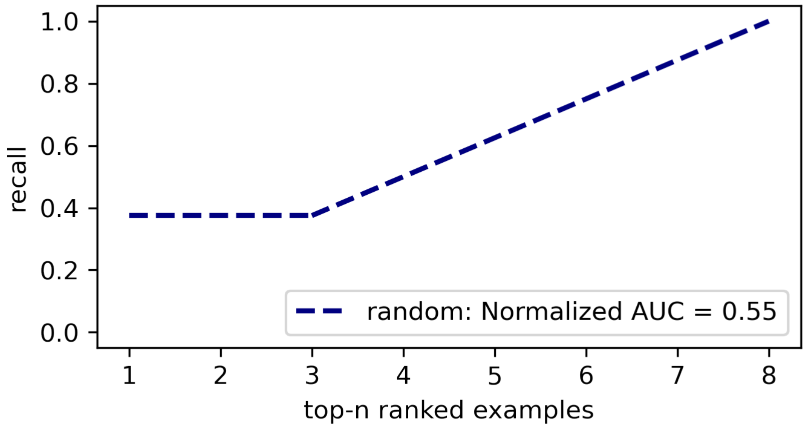


**Fig. 1:** Expected Ranking Curve for $N = 8, t = 3, p = 3/8 = 0.375$

To calculate the Normalised AUC we change the x-axis from $[1, N]$ to $[0, 1]$ which results in

$$AUC = \frac{(1 \times (N-1))/2 + ((t-1) \times p)/2}{N-1} = 0.5 + \frac{t-1}{N-1} \times \frac{p}{2}, \tag{6}$$

For the example on Figure 1, Normalized AUC $= 0.5 + (2/7) * (3/8)/2 \simeq 0.55$.

### 3.3 Lower Bound

If all the positive examples are ranked after all negative examples, i.e. $y_i = 0$ for $i \in \{1, ..., N-t\}$ and $y_i = 1$ for $i \in \{N-t+1, ..., N\}$, then the Normalized AUC = p/2 because:

$$\begin{aligned} R(n) &= \frac{\sum_1^n y_i}{min(n,t)} \\ &= \begin{cases} 0, & 1 \leq n \leq N-t \\ n-(N-t), & N-t+1 < n \leq N \end{cases} \end{aligned} \tag{7}$$

This forms a right triangle between (N-t, 0), (N, 0), and (N, 1) with an area of $(t \times 1)/2 = t/2$. When normalizing the x-axis from [1, N] to [0, 1], the area becomes $(t/2)/(N-1)$.

For the example on Figure 2, Normalized AUC $= (3/2)/(8-1) \simeq 0.21$.

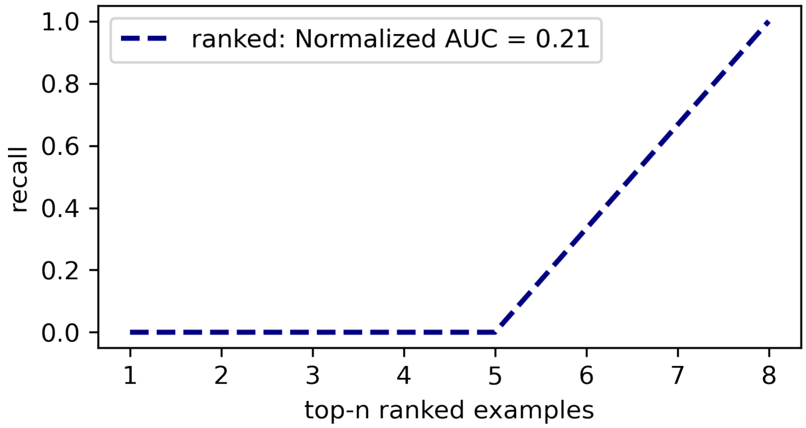


**Fig. 2:** Minimal Ranking Curve for $N = 8, t = 3, p = 3/8 = 0.375$

### 3.4 Upper Bound is 1

If all the positive examples are ranked before all negative examples, i.e. $y_i = 1$ for $i \in \{1, ..., t\}$ and $y_i = 0$ for $i \in \{t+1, ..., N\}$, then the Normalized AUC $= 1$ because:

$$\begin{aligned} R(n) &= \frac{\sum_1^n y_i}{min(n,t)} \\ &= \begin{cases} n/n, & 1 \leq n \leq t \\ t/t, & t < n \leq N \end{cases} \\ &= \begin{cases} 1, & 1 \leq n \leq t \\ 1, & t < n \leq N \end{cases} \\ &= 1 \end{aligned} \tag{8}$$

## References

1. Li, B., Li, Y., Eliceiri, K.W.: Dual-Stream Multiple Instance Learning Network for Whole Slide Image Classification With Self-Supervised Contrastive Learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 14318–14328 (Jun 2021). https://doi.org/10.1109/CVPR46437.2021.01409