Contrastive Pre-Training and Multiple Instance Learning for Predicting Tumor Microsatellite Instability

Ronald Nap, Mohammed Aburidi, and Roummel Marcia

University of California, Merced, CA, USA

July 15-19, 2024

This talk

Project goal: Predict tumor Microsatellite Instability from Whole Slide Images using Contrastive Learning and Multiple Instance Learning.

Whole Slide Image (WSI): A high-resolution digital scan of an entire microscope slide.

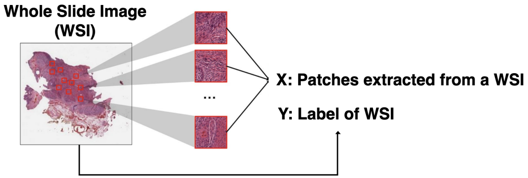
This talk

Project goal: Predict tumor Microsatellite Instability from Whole Slide Images using Contrastive Learning and Multiple Instance Learning.

Whole Slide Image (WSI): A high-resolution digital scan of an entire microscope slide.

Let

- $\mathbf{X} = \{x_i\}_{i=1}^{N}$: A set of N unlabeled patches extracted from a WSI.
- $\mathbf{Y} = \{y_j\}_{j=1}^{K}$: The set of K true WSI labels corresponding to different MSI statuses in a WSI.



Our approach

Challenges in Direct WSI Classification:

Our approach

Challenges in Direct WSI Classification:

- Substantial computational resources due to high dimensionality.
- **Tiny** details determine the classification of the **HUGE** image.

Our Approach:

Our approach

Challenges in Direct WSI Classification:

- Substantial computational resources due to high dimensionality.
- **Tiny** details determine the classification of the **HUGE** image.

Our Approach:

1. Use Contrastive Learning to learn robust feature representations of patches.

Our approach

Challenges in Direct WSI Classification:

- Substantial computational resources due to high dimensionality.
- **Tiny** details determine the classification of the **HUGE** image.

Our Approach:

1. Use Contrastive Learning to learn robust feature representations of patches.
2. Perform classification using feature vectors extracted from patches with Multiple Instance Learning.

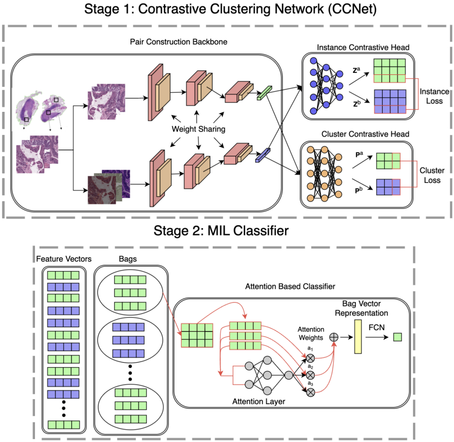
Our approach

Challenges in Direct WSI Classification:

- Substantial computational resources due to high dimensionality.
- **Tiny** details determine the classification of the **HUGE** image.

Our Approach:

1. Use Contrastive Learning to learn robust feature representations of patches.
2. Perform classification using feature vectors extracted from patches with Multiple Instance Learning.



1. Representation Learning with Contrastive Learning

Data Pair Creation:

Data pairs are generated for self-supervised representation learning. These pairs are created by applying various augmentations to the original image, resulting in two transformed versions, $\tilde{\mathbf{X}}^a$ and $\tilde{\mathbf{X}}^b$.

1. Representation Learning with Contrastive Learning

Data Pair Creation:

Data pairs are generated for self-supervised representation learning. These pairs are created by applying various augmentations to the original image, resulting in two transformed versions, $\tilde{\mathbf{X}}^a$ and $\tilde{\mathbf{X}}^b$.

The following augmentations are applied:

- Random cropping
- Color jittering
- Grayscale transformation
- Horizontal flipping
- Normalization

1. Representation Learning with Contrastive Learning

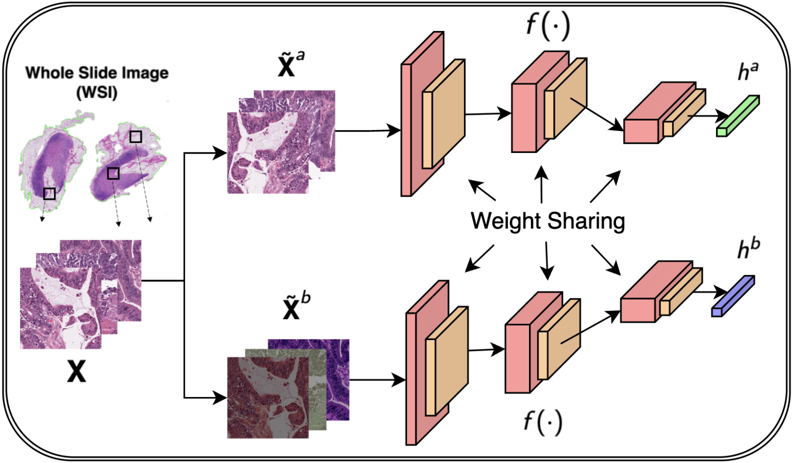
- **Feature Extraction:**
  - Features are extracted using a shared ResNet $f(\cdot)$ encoder.

1. Representation Learning with Contrastive Learning

- **Feature Extraction:**
  - Features are extracted using a shared ResNet $f(\cdot)$ encoder.
  - Resulting in feature vectors $h^a = f(\tilde{\mathbf{X}}^a)$ and $h^b = f(\tilde{\mathbf{X}}^b)$.

1. Representation Learning with Contrastive Learning

- **Feature Extraction:**
  - Features are extracted using a shared ResNet $f(\cdot)$ encoder.
  - Resulting in feature vectors $h^a = f(\tilde{\mathbf{X}}^a)$ and $h^b = f(\tilde{\mathbf{X}}^b)$.



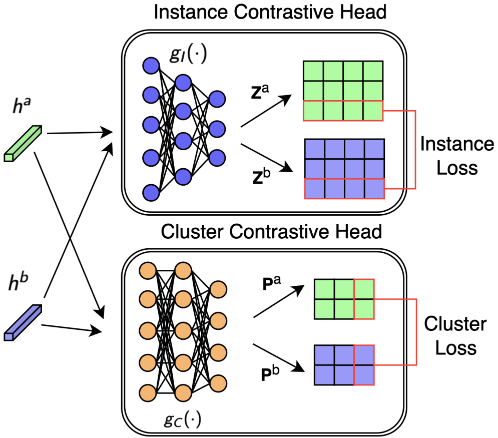
1. Representation Learning with Contrastive Learning

- **Projections:** Features are projected into row and column spaces using two separate MultiLayer Perceptions (MLPs) $g_I(\cdot)$ and $g_C(\cdot)$.

- **Projections:** Features are projected into row and column spaces using two separate MultiLayer Perceptions (MLPs) $g_I(\cdot)$ and $g_C(\cdot)$.
  - Instance space: $\mathbf{Z}^a = g_I(h^a)$, $\mathbf{Z}^b = g_I(h^b)$.
  - Cluster space: $\mathbf{P}^a = g_C(h^a)$, $\mathbf{P}^b = g_C(h^b)$.

1. Representation Learning with Contrastive Learning

- **Projections:** Features are projected into row and column spaces using two separate MultiLayer Perceptions (MLPs) $g_I(\cdot)$ and $g_C(\cdot)$.
  - Instance space: $\mathbf{Z}^a = g_I(h^a)$, $\mathbf{Z}^b = g_I(h^b)$.
  - Cluster space: $\mathbf{P}^a = g_C(h^a)$, $\mathbf{P}^b = g_C(h^b)$.



Instance-Level Loss

Instance-Level Loss

$$\mathcal{L}_I = -\sum_{i=1}^{N} \log \frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^{N} \exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)} \quad (1)$$

where $s(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is the cosine similarity.

Instance-Level Loss

$$\mathcal{L}_I = -\sum_{i=1}^{N} \log \frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^{N} \exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)} \tag{1}$$

where $s(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is the cosine similarity. τ_I and τ_C are the instance-level and cluster-level temperature parameters (scaling factor for cosine similarity).

Cluster-Level Loss

Instance-Level Loss

$$\mathcal{L}_I = -\sum_{i=1}^{N} \log \frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^{N} \exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)} \tag{1}$$

where $s(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is the cosine similarity. τ_I and τ_C are the instance-level and cluster-level temperature parameters (scaling factor for cosine similarity).

Cluster-Level Loss

$$\mathcal{L}_C = -\sum_{i=1}^{K} \log \frac{\exp(s(y_i^a, y_i^b)/\tau_C)}{\sum_{j=1}^{K} \exp(s(y_i^a, y_j^a)/\tau_C) + \exp(s(y_i^a, y_j^b)/\tau_C)} \tag{2}$$

Instance-Level Loss

$$\mathcal{L}_I = -\sum_{i=1}^{N} \log \frac{\exp(s(z_i^a, z_i^b)/\tau_I)}{\sum_{j=1}^{N} \exp(s(z_i^a, z_j^a)/\tau_I) + \exp(s(z_i^a, z_j^b)/\tau_I)} \tag{1}$$

where $s(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}|| ||\mathbf{v}||}$ is the cosine similarity. τ_I and τ_C are the instance-level and cluster-level temperature parameters (scaling factor for cosine similarity).

Cluster-Level Loss

$$\mathcal{L}_C = -\sum_{i=1}^{K} \log \frac{\exp(s(y_i^a, y_i^b)/\tau_C)}{\sum_{j=1}^{K} \exp(s(y_i^a, y_j^a)/\tau_C) + \exp(s(y_i^a, y_j^b)/\tau_C)} \tag{2}$$

Overall Loss Function

$$\mathcal{L} = \mathcal{L}_I + \mathcal{L}_C \tag{3}$$

1. Representation Learning with Contrastive Learning

Overall Workflow:

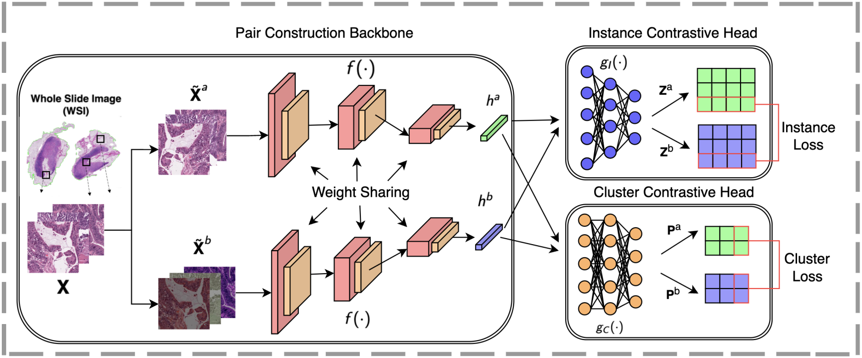
- Apply augmentations to create pairs of transformed patches.
- Use a shared ResNet encoder to extract feature vectors.
- Project feature vectors into instance and cluster spaces.
- Optimize the model to minimize the overall loss function.

1. Representation Learning with Contrastive Learning

Overall Workflow:

- Apply augmentations to create pairs of transformed patches.
- Use a shared ResNet encoder to extract feature vectors.
- Project feature vectors into instance and cluster spaces.
- Optimize the model to minimize the overall loss function.

Stage 1: Contrastive Clustering Network (CCNet)



Multiple Instance Learning (MIL)

- MIL is a form of **weakly supervised learning.**
- Suited for scenarios with **uncertainty in labeling** individual data points $\{x_1, \ldots, x_K\}$.
- Labels $Y \in \{0, 1\}$ are available at a **bag level** but individual labels instances $\{y_1, \ldots, y_K\}$ are **unknown**.

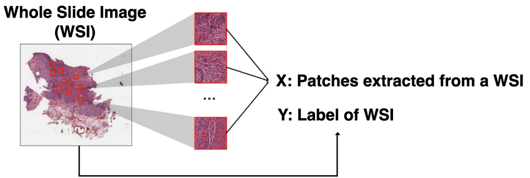
2. Multiple Instance Learning

Multiple Instance Learning (MIL)

- MIL is a form of **weakly supervised learning.**
- Suited for scenarios with **uncertainty in labeling** individual data points $\{x_1, \ldots, x_K\}$.
- Labels $Y \in \{0, 1\}$ are available at a **bag level** but individual labels instances $\{y_1, \ldots, y_K\}$ are **unknown**.

Context: Whole Slide Image (WSI) Analysis

- Treats each WSI as a collection of instances (patches).
- Effective due to the complexities and expansiveness of WSIs.
- Focus on **combined characteristics** of patches.



2. Multiple Instance Learning

MIL classifiers utilize a trainable attention mechanism to concentrate on the most informative instances within each bag.

2. Multiple Instance Learning

MIL classifiers utilize a trainable attention mechanism to concentrate on the most informative instances within each bag. This process involves an MLP attention network equipped with parameters **W**, **V**, and **U** to allocate a weight a_k for each embedded instance $\mathbf{z}_k$.

2. Multiple Instance Learning

MIL classifiers utilize a trainable attention mechanism to concentrate on the most informative instances within each bag. This process involves an MLP attention network equipped with parameters $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{U}$ to allocate a weight a_k for each embedded instance $\mathbf{z}_k$.

$$a_k = \frac{\exp\{\mathbf{W}^\top(\tanh(\mathbf{V}\mathbf{z}_k) \odot \mathrm{sigm}(\mathbf{U}\mathbf{z}_k))\}}{\sum_{j=1}^{N} \exp\{\mathbf{W}^\top(\tanh(\mathbf{V}\mathbf{z}_j) \odot \mathrm{sigm}(\mathbf{U}\mathbf{z}_j))\}} \tag{4}$$

where sigm$(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise product, N is the number of the embedded instance vectors in a bag.

2. Multiple Instance Learning

MIL classifiers utilize a trainable attention mechanism to concentrate on the most informative instances within each bag. This process involves an MLP attention network equipped with parameters **W**, **V**, and **U** to allocate a weight a_k for each embedded instance $\mathbf{z}_k$.

$$a_k = \frac{\exp\{\mathbf{W}^\top(\tanh(\mathbf{V}\mathbf{z}_k) \odot \mathrm{sigm}(\mathbf{U}\mathbf{z}_k))\}}{\sum_{j=1}^{N} \exp\{\mathbf{W}^\top(\tanh(\mathbf{V}\mathbf{z}_j) \odot \mathrm{sigm}(\mathbf{U}\mathbf{z}_j))\}} \tag{4}$$

where sigm$(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise product, N is the number of the embedded instance vectors in a bag.

$$\mu = \frac{1}{N} \sum_{k=1}^{N} a_k \mathbf{z}_k \tag{5}$$

The overall bag representation is computed by the weighted mean μ of these instance embeddings as shown above.

2. Multiple Instance Learning

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.
    - δ - Smoothness parameter ensuring differentiability.
  - Smooth SVM Loss for bag x_j is defined as:

$$l(y_j, f(x_j), \delta) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\delta}\xi_i^2 & \text{if } \xi_i \leq \delta \\ \frac{1}{N}\sum_{i=1}^{N}(\xi_i - \frac{\delta}{2}) & \text{otherwise} \end{cases}$$

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.
    - δ - Smoothness parameter ensuring differentiability.
  - Smooth SVM Loss for bag x_j is defined as:

$$l(y_j, f(x_j), \delta) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\delta}\xi_i^2 & \text{if } \xi_i \leq \delta \\ \frac{1}{N}\sum_{i=1}^{N}(\xi_i - \frac{\delta}{2}) & \text{otherwise} \end{cases}$$

- **Kullback Leibler (KL) Divergence Regularization:**
  - Acts as a regularizer applied with SVM Loss.

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.
    - δ - Smoothness parameter ensuring differentiability.
  - Smooth SVM Loss for bag x_j is defined as:

$$l(y_j, f(x_j), \delta) = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\delta} \xi_i^2 & \text{if } \xi_i \leq \delta \\ \frac{1}{N} \sum_{i=1}^{N} (\xi_i - \frac{\delta}{2}) & \text{otherwise} \end{cases}$$

- **Kullback Leibler (KL) Divergence Regularization:**
  - Acts as a regularizer applied with SVM Loss.
    - M - Total number of bags, N - Number of instances in bag j.
    - A_{ji} - Attention weight of instance i in bag j, U_{ji} - Uniform distribution value for instance i in bag j.

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.
    - δ - Smoothness parameter ensuring differentiability.
  - Smooth SVM Loss for bag x_j is defined as:

$$l(y_j, f(x_j), \delta) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\delta}\xi_i^2 & \text{if } \xi_i \leq \delta \\ \frac{1}{N}\sum_{i=1}^{N}(\xi_i - \frac{\delta}{2}) & \text{otherwise} \end{cases}$$

- **Kullback Leibler (KL) Divergence Regularization:**
  - Acts as a regularizer applied with SVM Loss.
    - M - Total number of bags, N - Number of instances in bag j.
    - A_{ji} - Attention weight of instance i in bag j, U_{ji} - Uniform distribution value for instance i in bag j.
  - Given by:

$$\text{KL Divergence} = \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N} A_{ji}\log\left(\frac{A_{ji}}{U_{ji}}\right)$$

2. Multiple Instance Learning

MIL Loss + Regularizer:

- **Support Vector Machine (SVM) Loss:**
  - For each bag x_j with label y_j, define $\xi_i = \max(0, 1 - \zeta_i \times y_j)$, where:
    - ζ_i - The predicted logit for instance i within the bag.
    - ξ_i - The hinge loss term for instance i.
    - δ - Smoothness parameter ensuring differentiability.
  - Smooth SVM Loss for bag x_j is defined as:

$$l(y_j, f(x_j), \delta) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\delta}\xi_i^2 & \text{if } \xi_i \leq \delta \\ \frac{1}{N}\sum_{i=1}^{N}(\xi_i - \frac{\delta}{2}) & \text{otherwise} \end{cases}$$

- **Kullback Leibler (KL) Divergence Regularization:**
  - Acts as a regularizer applied with SVM Loss.
    - M - Total number of bags, N - Number of instances in bag j.
    - A_{ji} - Attention weight of instance i in bag j, U_{ji} - Uniform distribution value for instance i in bag j.
  - Given by:

$$\text{KL Divergence} = \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N} A_{ji} \log\left(\frac{A_{ji}}{U_{ji}}\right)$$

  - We found that promoting uniform attention within each bag helped prevent overfitting to a few negative instances.

2. Multiple Instance Learning

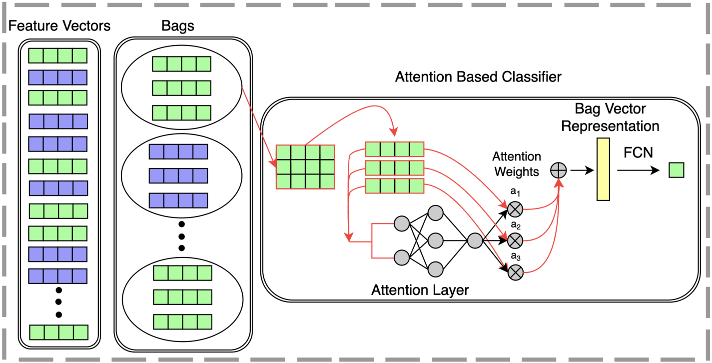
Overall Workflow:

- Extract patches from WSIs.
- Learn instance-level features using Stage 1 (CCNet).
- Aggregate instance features to form a bag-level representation.
- Classify the bag-level representation to predict MSI status.

Overall Workflow:

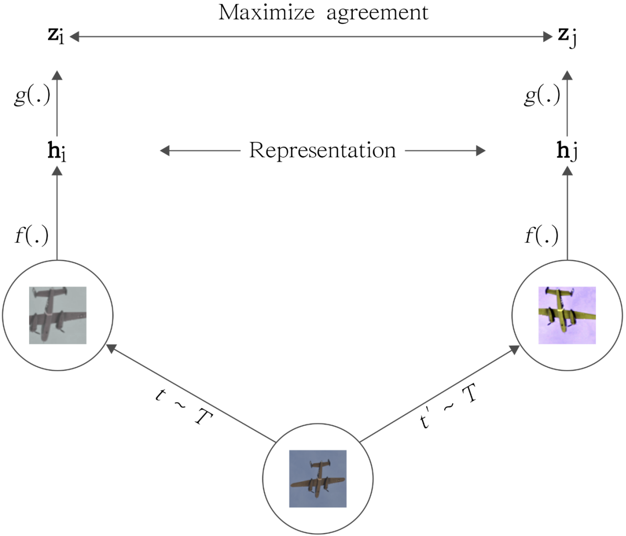
- Extract patches from WSIs.
- Learn instance-level features using Stage 1 (CCNet).
- Aggregate instance features to form a bag-level representation.
- Classify the bag-level representation to predict MSI status.

Stage 2: MIL Classifier



3. Numerical results: SimCLR

Contrary to our feature extractor (CCNet), SimCLR only utilizes an instance-level projection head for training.



3. Numerical results: Datasets

Dataset:

- We utilize two image datasets obtained from The Cancer Genome Atlas (TCGA) cohort:
  - The **Colorectal Cancer (CRC) dataset** was utilized for comparative analysis.
  - The **Stomach Adenocarcinoma (STAD) dataset** was employed to externally validate our model.

3. Numerical results: Datasets

Dataset:

- We utilize two image datasets obtained from The Cancer Genome Atlas (TCGA) cohort:
  - The **Colorectal Cancer (CRC) dataset** was utilized for comparative analysis.
  - The **Stomach Adenocarcinoma (STAD) dataset** was employed to externally validate our model.

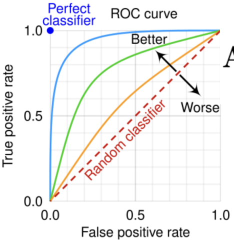
Evaluation:

- We partitioned the training data into 5-folds for cross-validation and reserved the testing split for final validation.

Dataset	Label	# of WSIs		# of Patches		# of Bags	
		Train	Test	Train	Test	Train	Test
CRC	MSI	39	26	46,704	29,335	1850	1122
	MSS	221	74	46,704	70,569	1757	2787
STAD	MSI	35	25	50,285	27,904	N/A	N/A
	MSS	150	74	50,285	90,104	N/A	N/A

3. Numerical results: Evaluation metrics

Area Under the Receiver Operating Characteristic (AUROC)

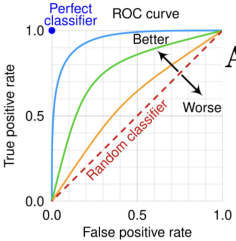


$$\text{AUROC} = \frac{1}{2}\sum_i (\mathsf{TPR}_i + \mathsf{TPR}_{i-1}) \times (\mathsf{FPR}_i - \mathsf{FPR}_{i-1})$$

$$\mathsf{TPR} = \frac{TP}{TP + FN} \qquad \mathsf{FPR} = \frac{FP}{TN + FP}$$

3. Numerical results: Evaluation metrics

Area Under the Receiver Operating Characteristic (AUROC)



$$\text{AUROC} = \frac{1}{2}\sum_{i}(\mathsf{TPR}_i + \mathsf{TPR}_{i-1}) \times (\mathsf{FPR}_i - \mathsf{FPR}_{i-1})$$

$$\mathsf{TPR} = \frac{TP}{TP + FN} \qquad \mathsf{FPR} = \frac{FP}{TN + FP}$$

F1 Score (F1)

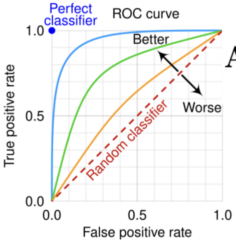
True positive (TP)	**False negative (FN)**
False positive (FP)	**True negative (TN)**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

3. Numerical results: Evaluation metrics

Area Under the Receiver Operating Characteristic (AUROC)



$$\text{AUROC} = \frac{1}{2}\sum_i (\text{TPR}_i + \text{TPR}_{i-1}) \times (\text{FPR}_i - \text{FPR}_{i-1})$$

$$\text{TPR} = \frac{TP}{TP + FN} \qquad \text{FPR} = \frac{FP}{TN + FP}$$

F1 Score (F1)

True positive (TP)	**False negative (FN)**
False positive (FP)	**True negative (TN)**

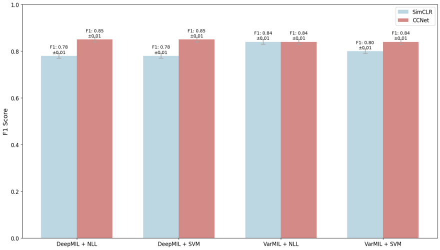
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

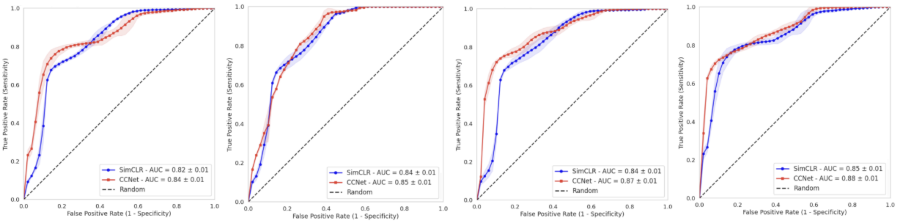
$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

Both measures values between 0 and 1 with higher values indicating better results.

3. Numerical results: CRC Dataset

- To ensure a comprehensive evaluation, we compute the mean and standard deviation of AUROC and F1 scores across five folds.
- We compare the performance of SimCLR with our CCNet extractor.
- We evaluate the efficacy of SVM loss compared to the conventional Negative Log-Likelihood (NLL) loss.
- We employ two attention-based MIL classifiers, DeepMIL and VarMIL, and assess the performance of each model configuration.





3. Numerical results: Transfer Learning

Motivation for Transfer Learning:

- In medical imaging, labeled data scarcity limits model training.
- Transfer learning leverages pre-trained models to enhance performance on related tasks.

3. Numerical results: Transfer Learning

Motivation for Transfer Learning:

- In medical imaging, labeled data scarcity limits model training.
- Transfer learning leverages pre-trained models to enhance performance on related tasks.

Experiment:

- We explore the performance of ResNet18 pretrained on ImageNet, SimCLR pre-trained on STAD and CCNet pre-trained on STAD.
- We utilize these pre-trained extractors to generate feature vectors for input for MIL Classification.

3. Numerical results: Transfer Learning

Motivation for Transfer Learning:

- In medical imaging, labeled data scarcity limits model training.
- Transfer learning leverages pre-trained models to enhance performance on related tasks.

Experiment:

- We explore the performance of ResNet18 pretrained on ImageNet, SimCLR pre-trained on STAD and CCNet pre-trained on STAD.
- We utilize these pre-trained extractors to generate feature vectors for input for MIL Classification.

Key Findings:

- CCNet, achieved better AUROC and F1 scores than SimCLR, demonstrating effective feature extraction.

Extractor	Classifier	Loss	AUROC	F1
ResNet18	DeepMIL	NLL	0.58 ± 0.01	0.67 ± 0.01
		SVM	0.58 ± 0.01	0.69 ± 0.01
	VarMIL	NLL	0.59 ± 0.01	0.69 ± 0.01
		SVM	0.59 ± 0.01	0.69 ± 0.01
SimCLR	DeepMIL	NLL	0.81 ± 0.01	0.82 ± 0.01
		SVM	0.82 ± 0.01	0.81 ± 0.01
	VarMIL	NLL	0.78 ± 0.01	0.82 ± 0.01
		SVM	0.81 ± 0.01	0.81 ± 0.01
CCNet	DeepMIL	NLL	0.81 ± 0.01	0.81 ± 0.01
		SVM	0.83 ± 0.01	0.82 ± 0.01
	VarMIL	NLL	0.81 ± 0.01	0.83 ± 0.01
		SVM	0.83 ± 0.01	0.81 ± 0.01

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.
2. Model incorporates Contrastive Learning and Multiple Instance Learning.

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.
2. Model incorporates Contrastive Learning and Multiple Instance Learning.
3. Conducted comparisons to state-of-the-art models.

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.
2. Model incorporates Contrastive Learning and Multiple Instance Learning.
3. Conducted comparisons to state-of-the-art models.
4. Conducted evaluation on two real-world histopathology cancers datasets: Colorectal (CRC) and Stomach (STAD) cancers.

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.
2. Model incorporates Contrastive Learning and Multiple Instance Learning.
3. Conducted comparisons to state-of-the-art models.
4. Conducted evaluation on two real-world histopathology cancers datasets: Colorectal (CRC) and Stomach (STAD) cancers.
5. Results from our proposed model improve upon existing methods.

4. Summary

1. Proposed a framework for predicting tumor Microsatellite Instability from Whole Slide Images.
2. Model incorporates Contrastive Learning and Multiple Instance Learning.
3. Conducted comparisons to state-of-the-art models.
4. Conducted evaluation on two real-world histopathology cancers datasets: Colorectal (CRC) and Stomach (STAD) cancers.
5. Results from our proposed model improve upon existing methods.

Thank You!