Presentation 2

- Links to presentation(s) and code(s) on GitHub
    - Presentation
    - Code
- What did you do?
    - I modularized the Hierarchical Attention MIL model from Jupyter notebook into Python code for HPC deployment. I resolved critical memory issues by implementing sequential stain processing. I also fixed broken gradient flow that caused underfitting by properly managing detach operations and properly freezing only DenseNet features while keeping attention modules trainable. I implemented comprehensive evaluation framework with attention visualization, command-line interface for hyperparameter control, and reproducible data splitting. The model now successfully learns from the training examples, achieving reasonable validation performance (AUC-ROC = 0.82) instead of being stuck at baseline.
- How does it help the project?
    - Successfully utilized Quest's computational power for training and classification. The modularized codebase is now easily expandable and executable on Quest.
- Issues faced (if any)
    - Model exhibited stagnant gradient descent with training loss plateauing and validation accuracy stuck at majority class baseline (broken gradient flow through hierarchical attention layers)
- Attempts to resolve issues (if any)
    - I tested higher learning rates and adjusted class weights to overcome training stagnation. Debugged gradient flow by systematically checking detach operations and frozen layer configurations throughout model.
- Issues resolved (if any)
    - Fixed gradient flow by removing inappropriate detach calls. No longer stagnant. Training now progresses with higher learning rates, though training and validation losses remains unsmooth requiring further resolution (such as regularization?).

- Next steps
  - Implement regularization techniques (dropout, weight decay) to stabilize training. Add early stopping and learning rate scheduling. Conduct hyperparameter tuning.
  - Maybe: add more layers to the attention modules and final classifier to improve performance
- References (Mention if you built up on someone else’s work)
  - ChatGPT & Claude (coding)