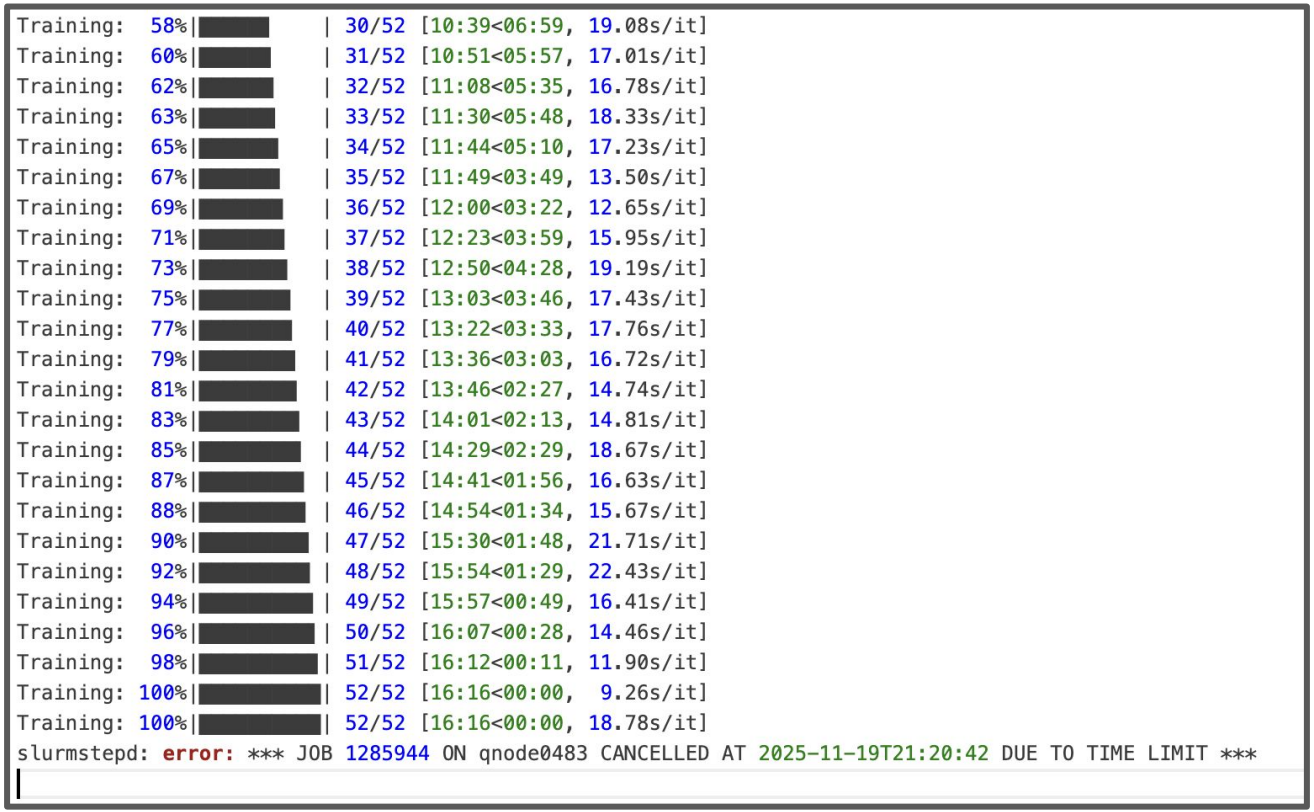
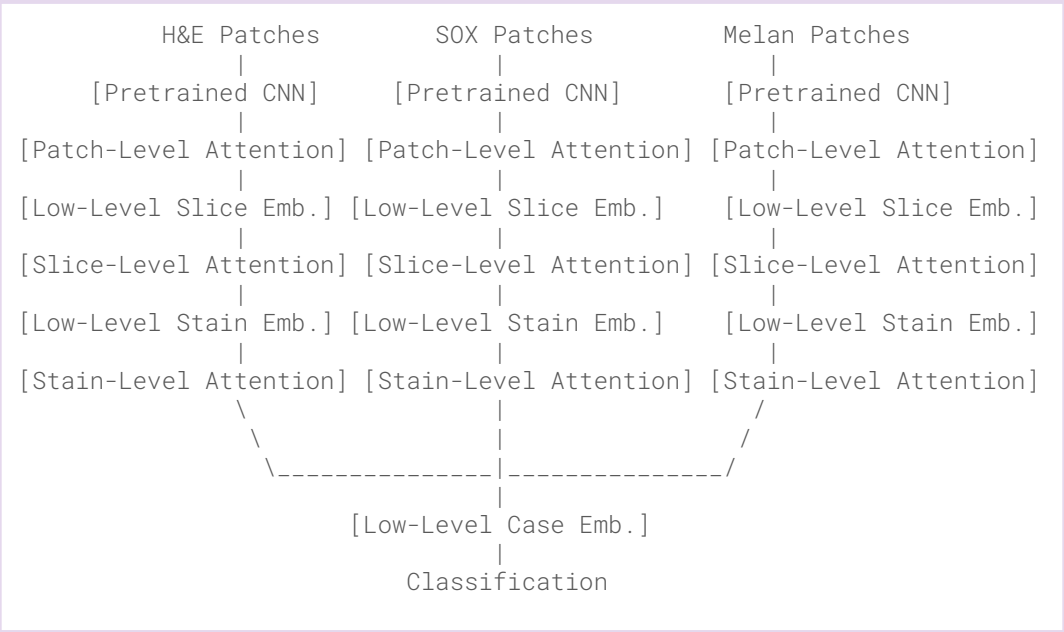
Presentation 4: Analyzing Multi-Stain Model Results

STAT 390 | Project 1 | Fall 2025


Training: 58%| | 30/52 [10:39<06:59, 19.08s/it]
Training: 60%| | 31/52 [10:51<05:57, 17.01s/it]
Training: 62%| | 32/52 [11:08<05:35, 16.78s/it]
Training: 63%| | 33/52 [11:30<05:48, 18.33s/it]
Training: 65%| | 34/52 [11:44<05:10, 17.23s/it]
Training: 67%| | 35/52 [11:49<03:49, 13.50s/it]
Training: 69%| | 36/52 [12:00<03:22, 12.65s/it]
Training: 71%| | 37/52 [12:23<03:59, 15.95s/it]
Training: 73%| | 38/52 [12:50<04:28, 19.19s/it]
Training: 75%| | 39/52 [13:03<03:46, 17.43s/it]
Training: 77%| | 40/52 [13:22<03:33, 17.76s/it]
Training: 79%| | 41/52 [13:36<03:03, 16.72s/it]
Training: 81%| | 42/52 [13:46<02:27, 14.74s/it]
Training: 83%| | 43/52 [14:01<02:13, 14.81s/it]
Training: 85%| | 44/52 [14:29<02:29, 18.67s/it]
Training: 87%| | 45/52 [14:41<01:56, 16.63s/it]
Training: 88%| | 46/52 [14:54<01:34, 15.67s/it]
Training: 90%| | 47/52 [15:30<01:48, 21.71s/it]
Training: 92%| | 48/52 [15:54<01:29, 22.43s/it]
Training: 94%| | 49/52 [15:57<00:49, 16.41s/it]
Training: 96%| | 50/52 [16:07<00:28, 14.46s/it]
Training: 98%| | 51/52 [16:12<00:11, 11.90s/it]
Training: 100%| | 52/52 [16:16<00:00, 9.26s/it]
Training: 100%| | 52/52 [16:16<00:00, 18.78s/it]
slurmstepd: error: *** JOB 1285944 ON qnode0483 CANCELLED AT 2025-11-19T21:20:42 DUE TO TIME LIMIT ***

Multi-Stain Model Overview



1. Initial Results

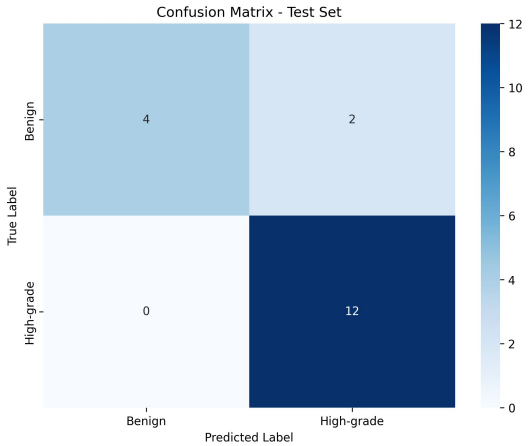
- **Improvements made to training**
  - Learning rate scheduler
    - Lowers LR by 0.3x when val loss plateaus
    - Intuition: escape plateaus and refine convergence
  - Early stopping
    - Stops training when val loss stops improving for 8 epochs
    - Intuition: prevents overfitting
  - Dropout
    - Randomly drops neurons during training
    - Intuition: prevents overfitting and increase generalization
  - Stronger weight decay
    - Stronger L2 regularization strength during training
    - Intuition: same as dropout

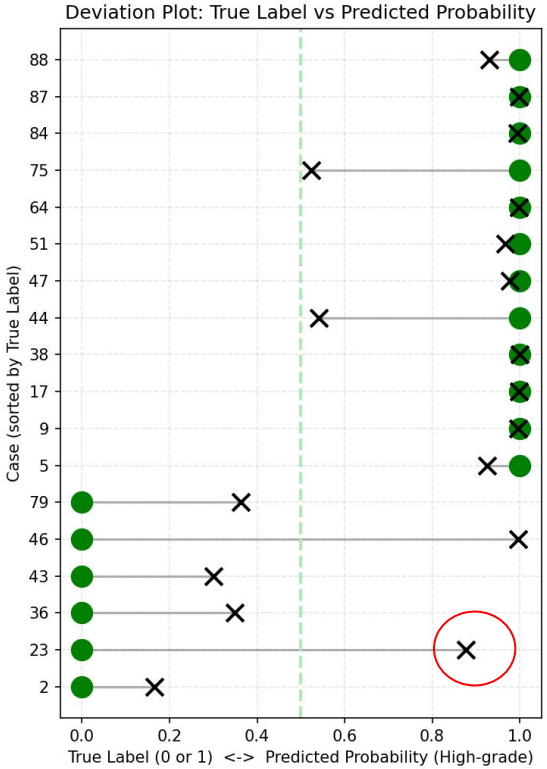
1. Initial Results

- **New losses** (Training still challenging given the small # of cases!!)

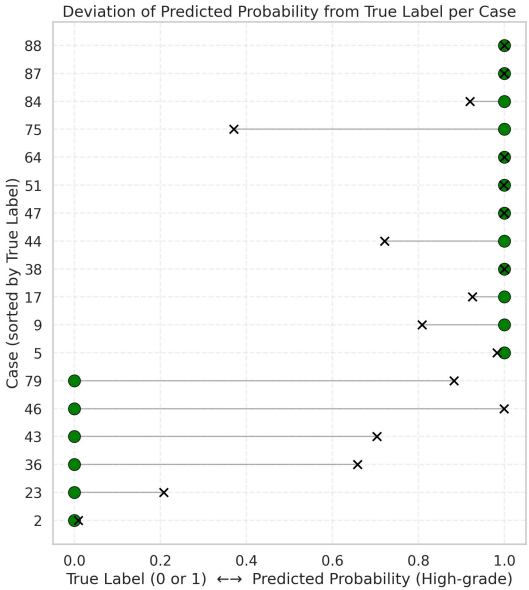
1. Initial Results

- New results
- Not cross validated





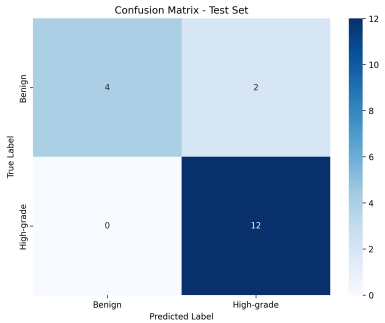
Previously…



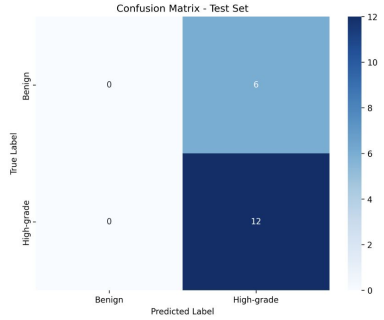
2. Changing Train-Validation-Test Split

- data_splits_01 (original split; see previous slides)
  - Subset | Benign | High-Grade
  - Train | 17 | 34
  - Val | 5 | 12
  - Test | 6 | 12 — test_cases: 2 5 9 17 23 36 38 43 44 46 47 51 64 75 79 84 87 88
- data_splits_02
  - Subset | Benign | High-Grade
  - Train | 16 | 35
  - Val | 6 | 11
  - Test | 6 | 12 — test_cases: 7 15 18 24 40 45 48 52 55 56 63 67 70 82 89 90 94 101
- data_splits_03
  - Subset | Benign | High-Grade
  - Train | 16 | 35
  - Val | 6 | 11
  - Test | 6 | 12 — test_cases: 16 25 28 49 50 59 60 68 72 78 83 85 86 91 92 93 95 100
- data_splits_04
  - Subset | Benign | High-Grade
  - Train | 17 | 35
  - Val | 6 | 12
  - Test | 5 | 11 — test_cases: 1 11 14 21 22 26 30 32 54 61 66 80 96 98 104 106
- data_splits_05
  - Subset | Benign | High-Grade
  - Train | 17 | 35
  - Val | 6 | 12
  - Test | 5 | 11 — test_cases: 3 4 12 19 27 34 41 42 53 57 58 69 73 77 97 99

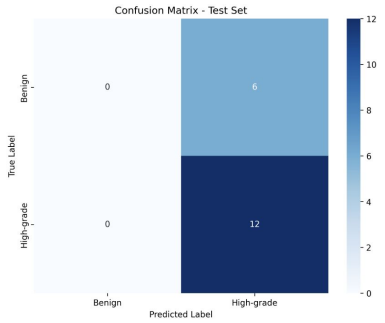
2. Changing Train-Validation-Test Split: Results



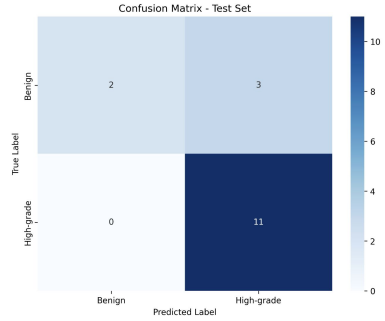
Split 1 | benign recall: 67% | high-grade recall: 100%



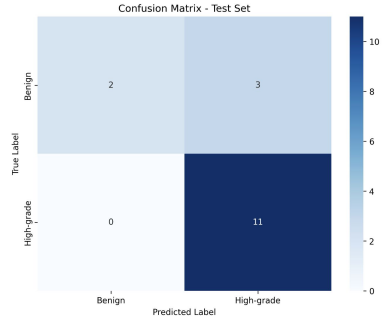
Split 2 | benign recall: 0% | high-grade recall: 100%



Split 3 | benign recall: 0% | high-grade recall: 100%



Split 4 | benign recall: 40% | high-grade recall: 100%



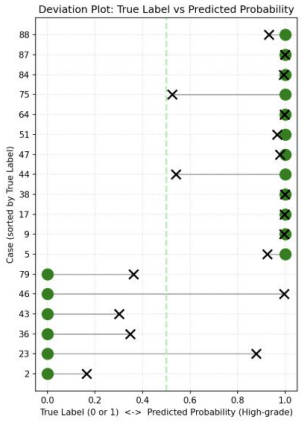
Split 5 | benign recall: 40% | high-grade recall: 100%

Average benign recall:
29%

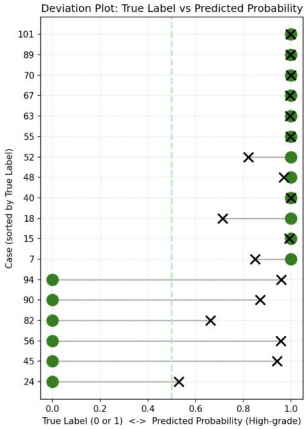
Average high-grade recall:
100%

(See MIL_trainer_17Nov_Paisley for .npz files)
job_id: 1338821 is split 2, last checkpoint is epoch 16
job_id: 1338838 is split 3, last checkpoint is epoch 15
job_id: 1338846 is split 4, last checkpoint is epoch 26
job_id: 1338851 is split 5, last checkpoint is epoch 19
^ checkpoints (at bottom of folder, labeled) can be found under
run_20251119_230053

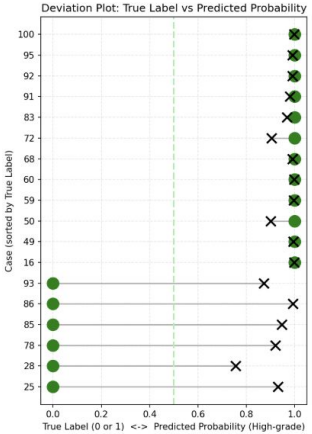
2. Changing Train-Validation-Test Split: Results



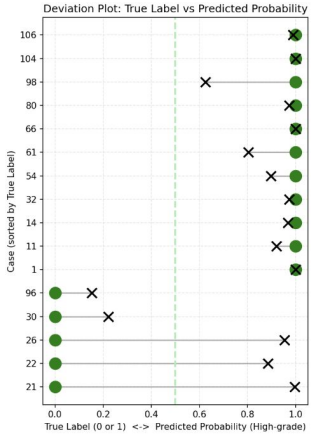
split_01



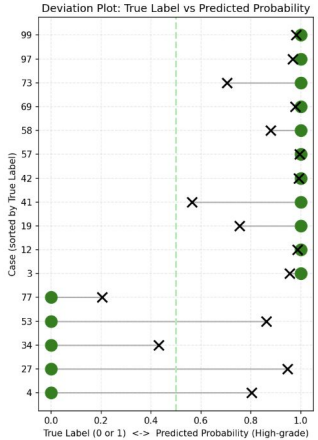
split_02



split_03



split_04

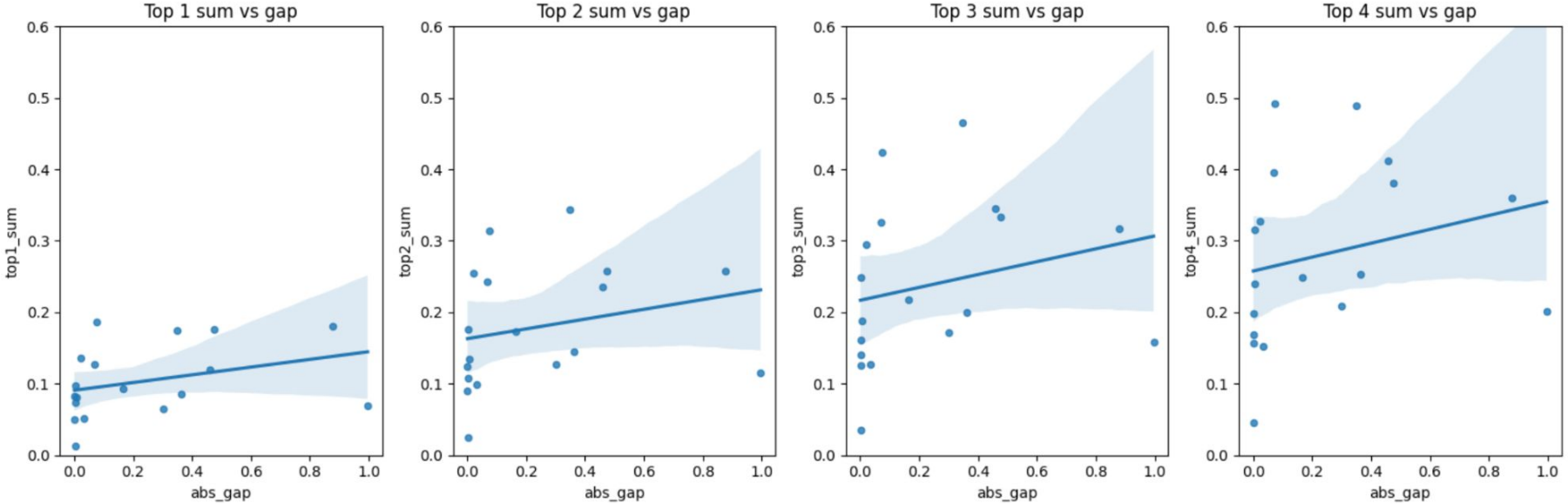


split_05

3. Effective Attention Analysis

- compute_effective_patch_attention() function to calculate effective attention for all patches
  - effective attention: stain_attn_weight * slice_attn_weight * patch_attn_weight
- plot_effective_patch_attention_distribution_per_case() function plot how effective attention weights are distributed (1 distribution plot per case)
- analyze_top_effective_patches_per_case() function shows the “identity” of the top n patches (in terms of effective attention) for each case. “Identity” means which stain & slice are these top effectively attending patches belongs to.
  - The idea behind this is to see if the effective attention is only focusing on patches in only a few high attending slices, or is it taking top attending patches from various slices. And how that contributes to the prediction
- visualize_case_effective_patches() function visualize the top and bottom n patches in terms of effective attention for each case

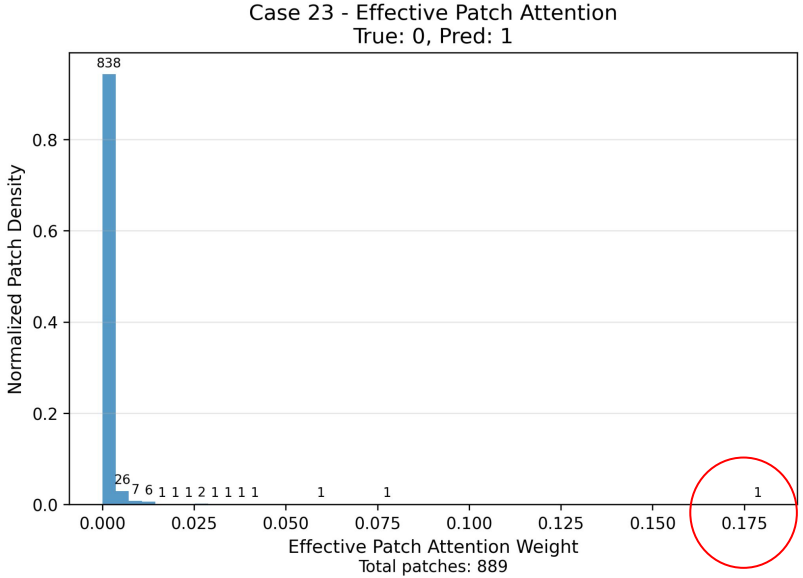
3. Effective Attention Distribution

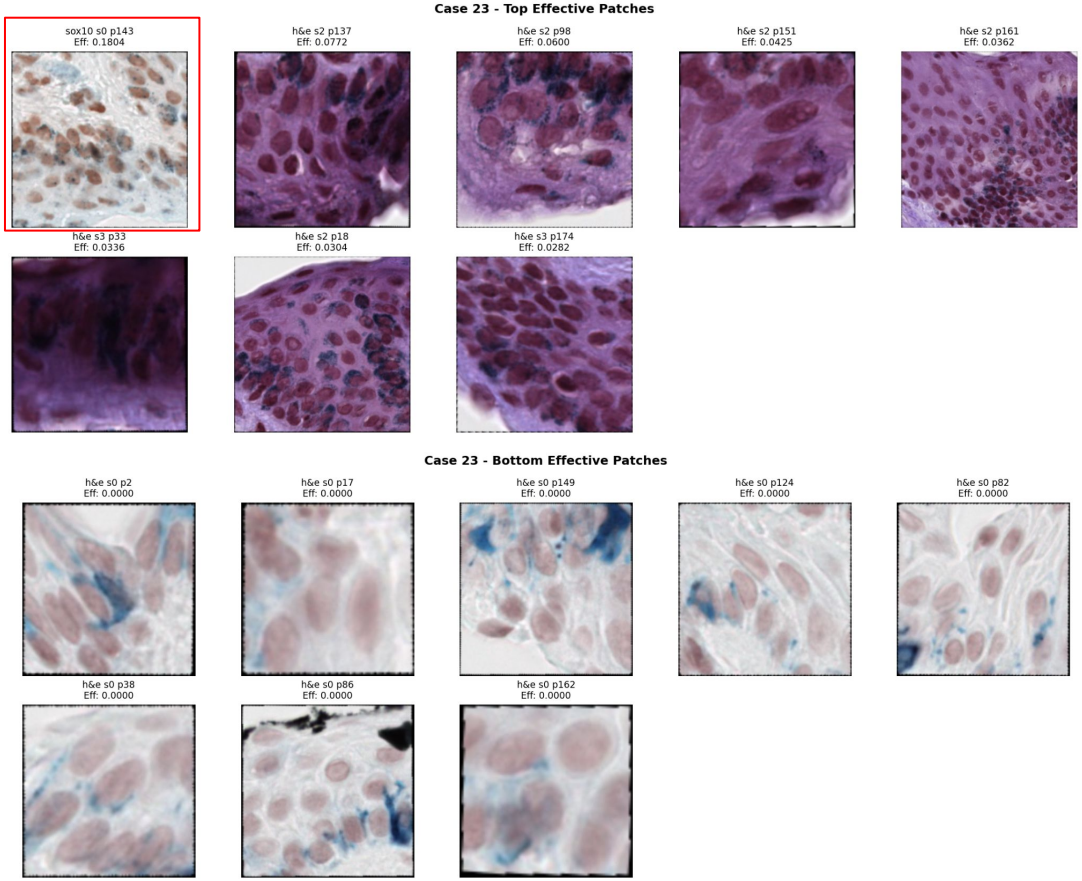


X-axis: absolute deviation of predicted probability from ground truth
Y-axis: Summation of effective attention scores of top-k patches in a case

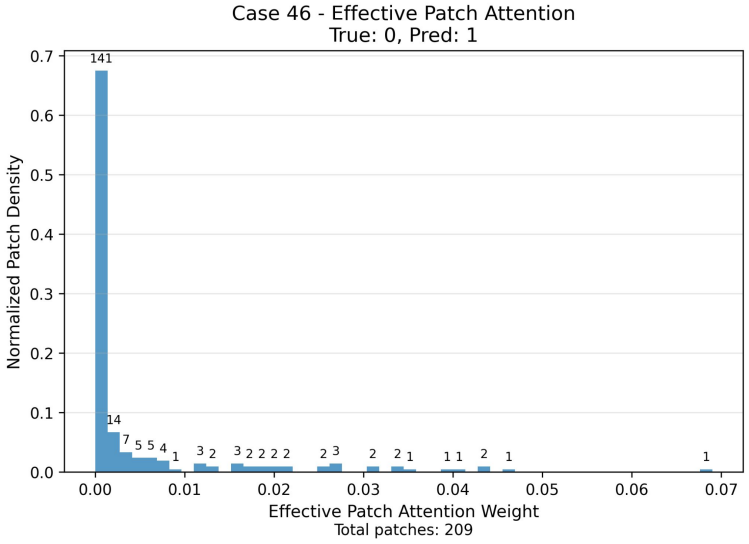
Insight: “hyper-focusing” to only a few patches does not help model’s predictive probability to get close to the true label

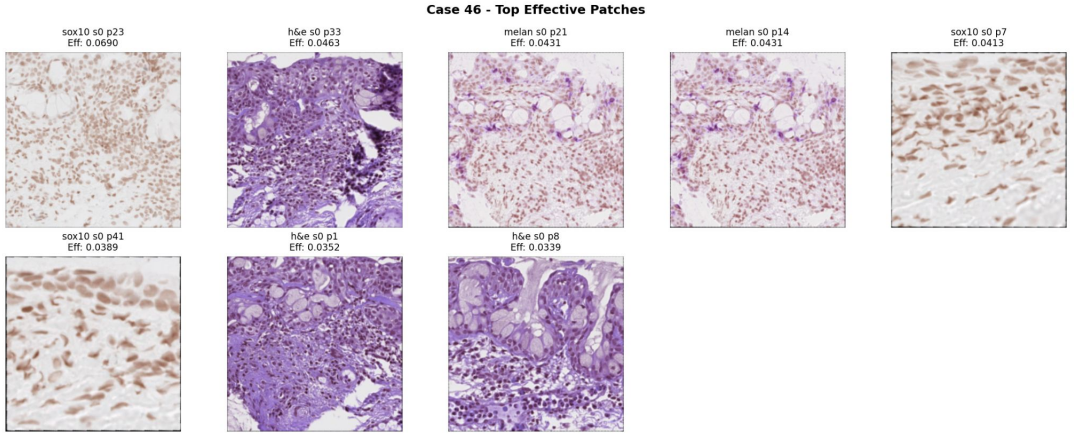
3. Effective Attention For Misclassified Cases

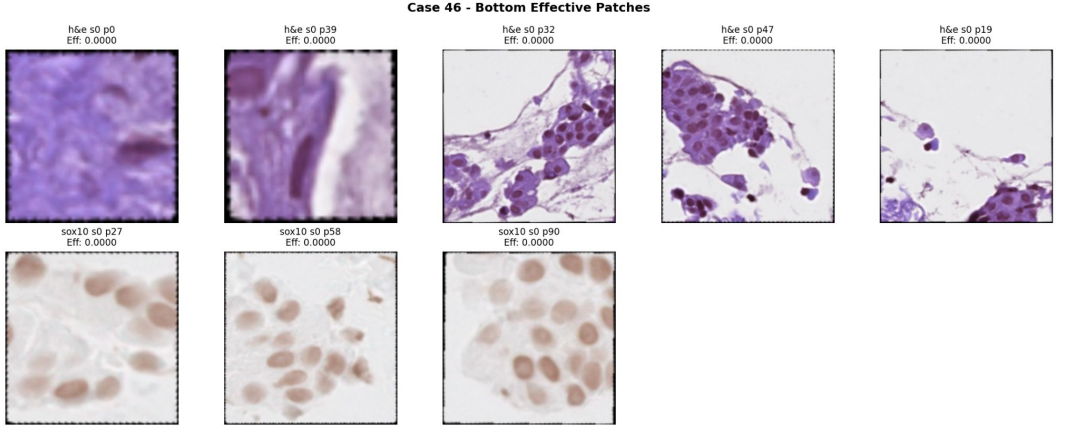




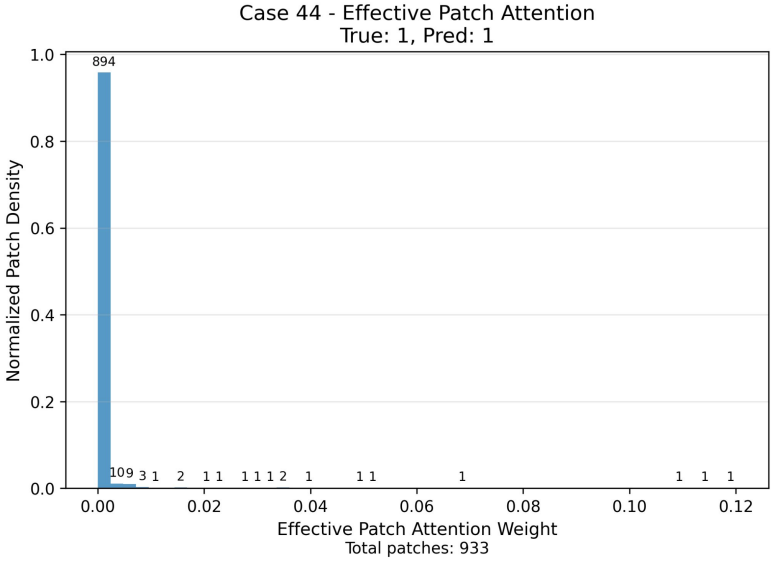
3. Effective Attention For Misclassified Cases

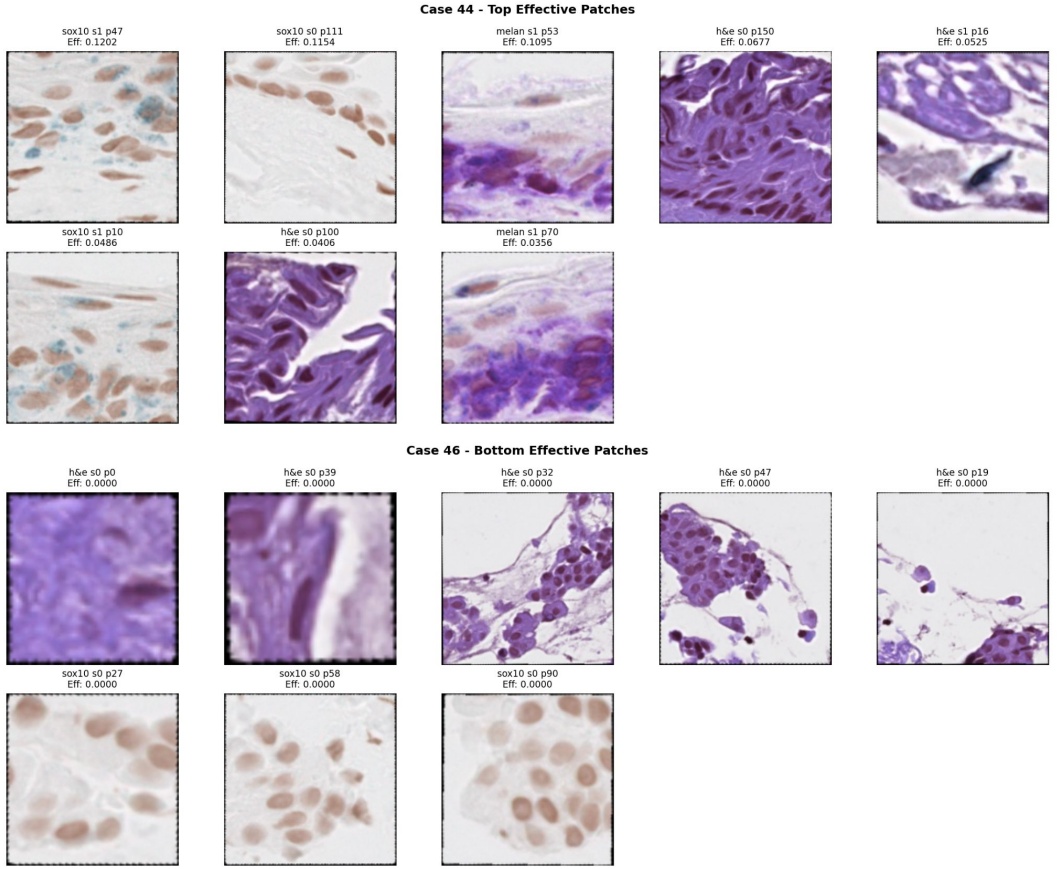




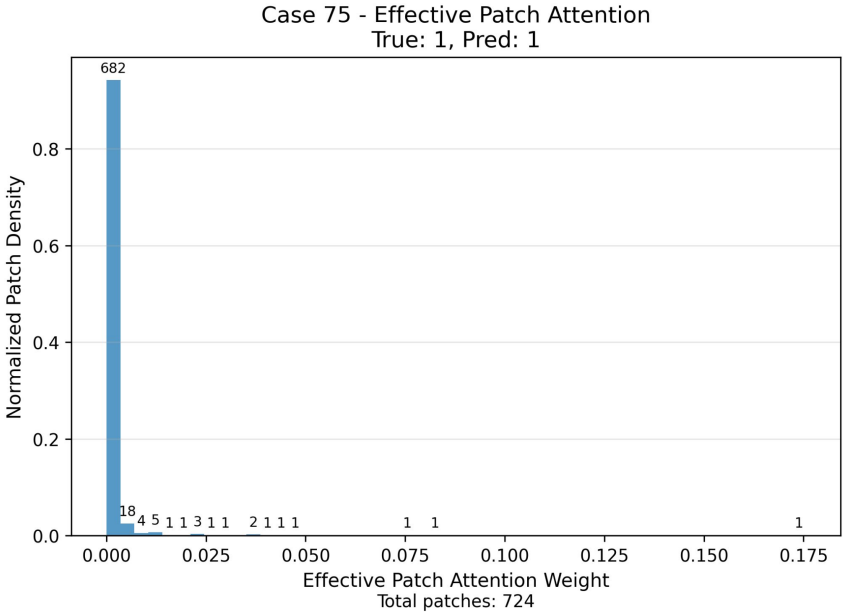


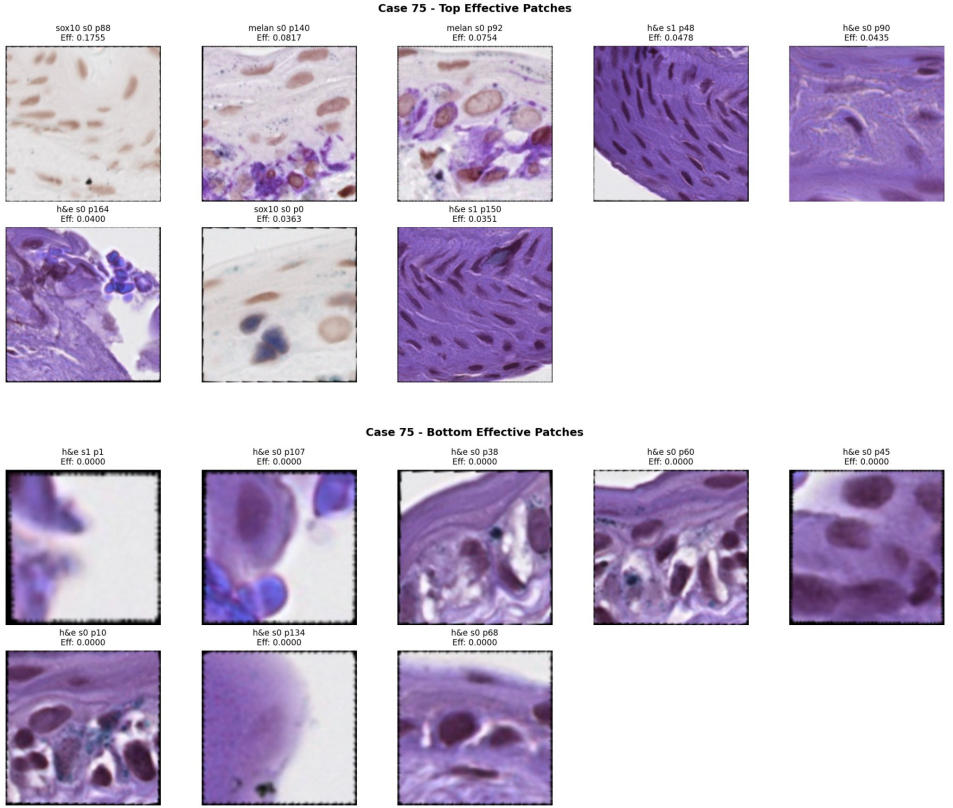
3. Effective Attention For Near-Threshold Cases



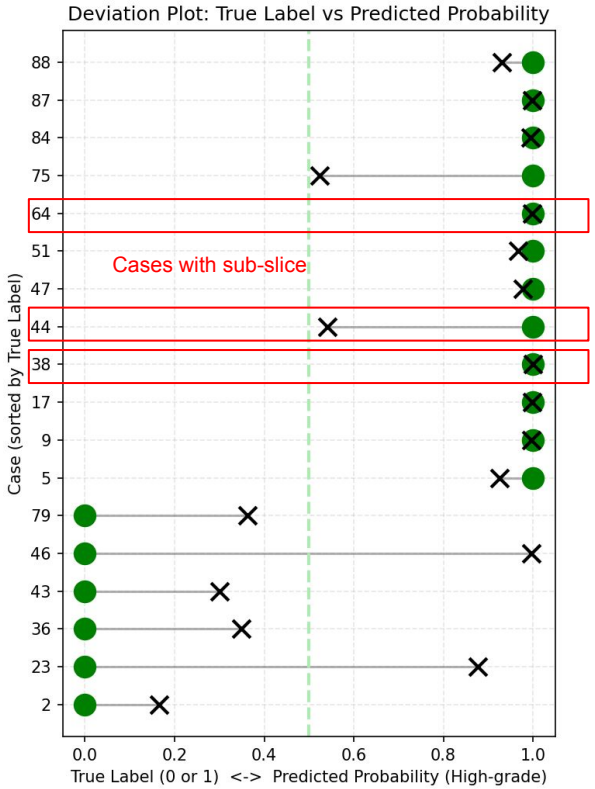


3. Effective Attention For Near-Threshold Cases





3. Effective Attention for Subslice Cases



Case 64:
- True label: 1
- Pred label: 1
- # total patches: 1622
- # top patches (per-case): 81
- # total slices: 12
- slice ratio: 1.0000
- Top-patch counts by stain:
  - h&e: 34
  - melan: 31
  - sox10: 16
- Top-patch counts by (stain, slice_idx):
  - (h&e, slice 0): 3
  - (h&e, slice 1): 7
  - (h&e, slice 2): 23
  - (h&e, slice 3): 1
  - (melan, slice 0): 7
  - (melan, slice 1): 13
  - (melan, slice 2): 9
  - (melan, slice 3): 2
  - (sox10, slice 0): 6
  - (sox10, slice 1): 2
  - (sox10, slice 2): 7
  - (sox10, slice 3): 1

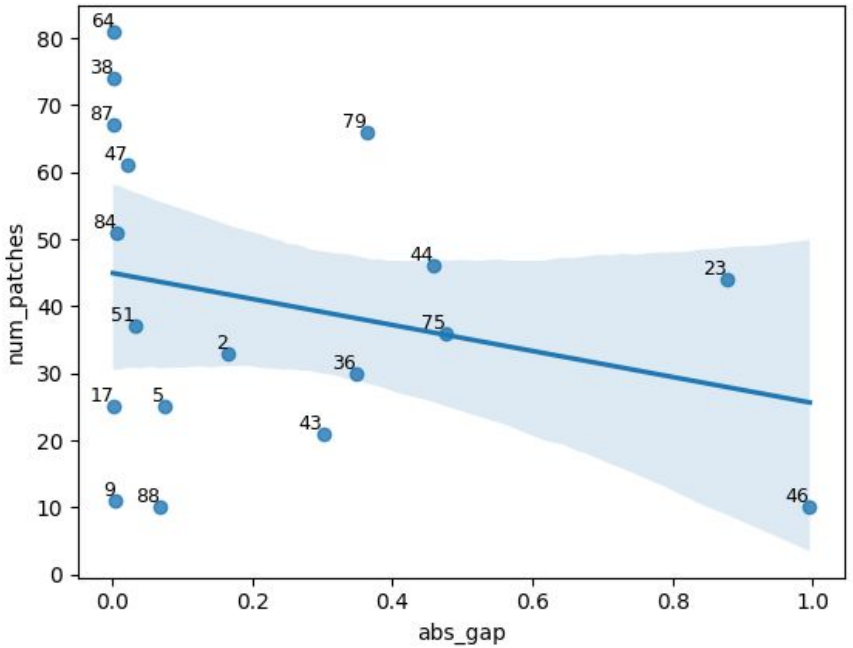
Case 38:
- True label: 1
- Pred label: 1
- # total patches: 1499
- # top patches (per-case): 74
- # total slices: 15
- slice ratio: 0.7333
- Top-patch counts by stain:
  - h&e: 28
  - melan: 31
  - sox10: 15
- Top-patch counts by (stain, slice_idx):
  - (h&e, slice 0): 3
  - (h&e, slice 1): 2
  - (h&e, slice 2): 9
  - (h&e, slice 3): 3
  - (h&e, slice 4): 11
  - (melan, slice 0): 7
  - (melan, slice 2): 16
  - (melan, slice 3): 3
  - (melan, slice 4): 5
  - (sox10, slice 2): 13
  - (sox10, slice 3): 2

Case 44:
- True label: 1
- Pred label: 1
- # total patches: 933
- # top patches (per-case): 46
- # total slices: 6
- slice ratio: 1.0000
- Top-patch counts by stain:
  - h&e: 19
  - melan: 9
  - sox10: 18
- Top-patch counts by (stain, slice_idx):
  - (h&e, slice 0): 13
  - (h&e, slice 1): 6
  - (melan, slice 0): 2
  - (melan, slice 1): 7
  - (sox10, slice 0): 11
  - (sox10, slice 1): 7

3. Top 5% Effective Attention and Model Performance

Insight: cases generally with more patches have better prediction -> get attention to focus on more patches for cases with less total patches

```
Case 46:
  True label: 0
  Pred label: 1
  # total patches: 209
  # top patches (per-case): 10
  # total slices: 3
  slice ratio: 1.0000
  Top-patch counts by stain:
    h&e: 4
    melan: 3
    sox10: 3
  Top-patch counts by (stain, slice_idx):
    (h&e, slice 0): 4
    (melan, slice 0): 3
    (sox10, slice 0): 3
```



X-axis: Number of patches in top 5% effective attention scores for each case

Y-axis: absolute deviation of predicted probability from ground truth

4. Effective Attention for Patches at Slice Level

Goal: Determine which regions of slices have high effective attention

Steps:

1. Extract patch index from file name
    a. This file verifies accurate extraction:

    /projects/e32998/MIL_training/pres_4_runs/run_20251118_115222/

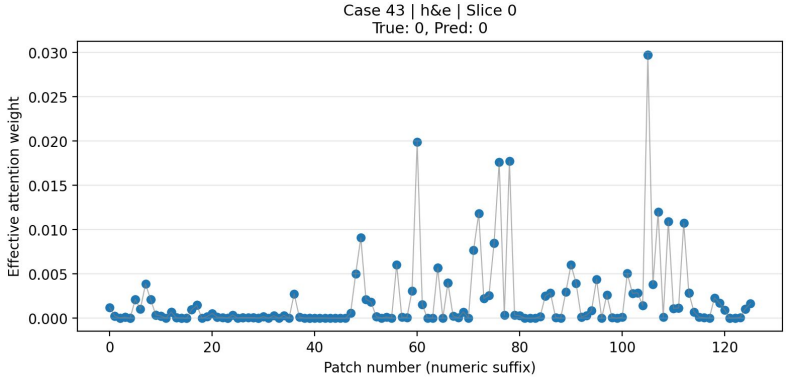
    attention_analysis/plots/patch_attention_selected_slices.csv
2. Use top 20 slices with highest attention
3. Plot patch attention vs patch index

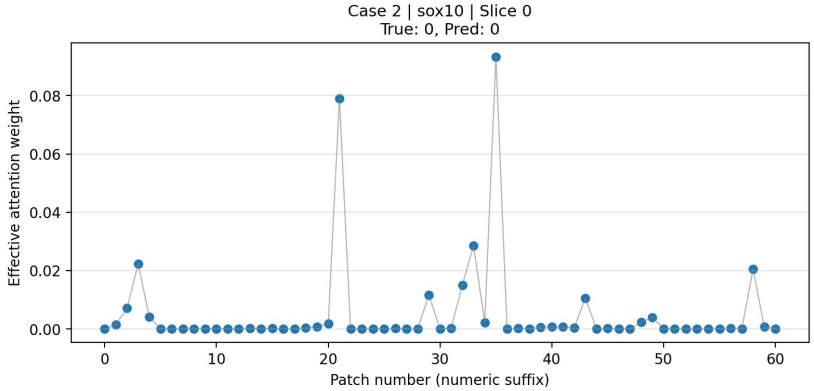
Selected Patches

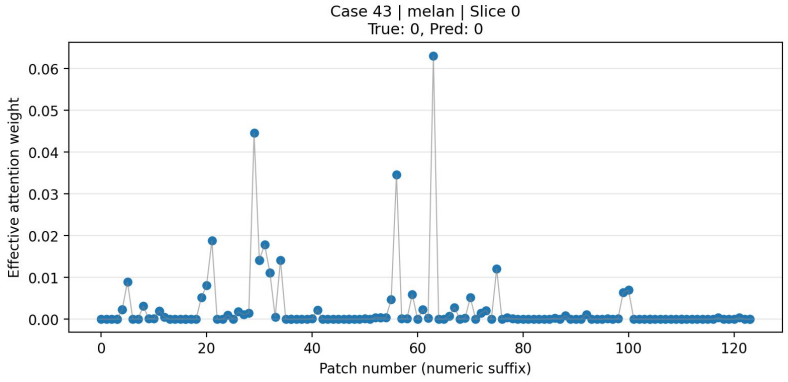
Case #	Stain	Slice #	Effective attention
46	h&e	0	0.574512
47	h&e	0	0.427708
23	h&e	2	0.378971
84	melan	0	0.375497
87	melan	1	0.350803
43	melan	0	0.345458
43	h&e	0	0.335724
43	sox10	0	0.318818
88	melan	0	0.315857
23	sox10	0	0.313861
2	sox10	0	0.312554
75	melan	0	0.307292
87	h&e	0	0.303412
17	sox10	0	0.297756
47	melan	0	0.296922
9	h&e	0	0.284751
75	sox10	0	0.283576
88	sox10	0	0.280459
5	melan	0	0.276566
47	sox10	0	0.275369

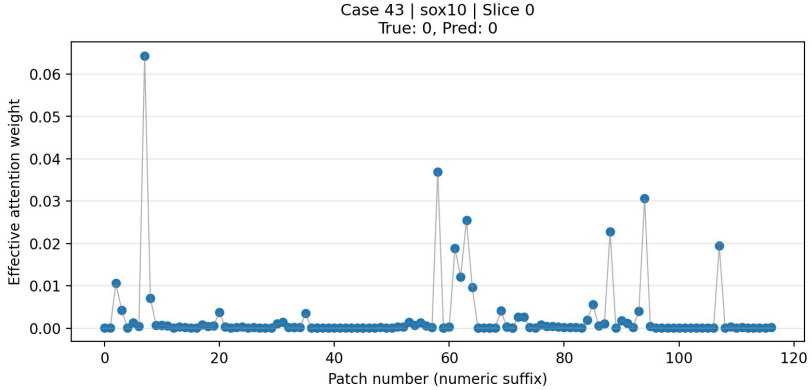
4. Effective Attention for Patches at Slice Level

Benign cases: predicted correctly



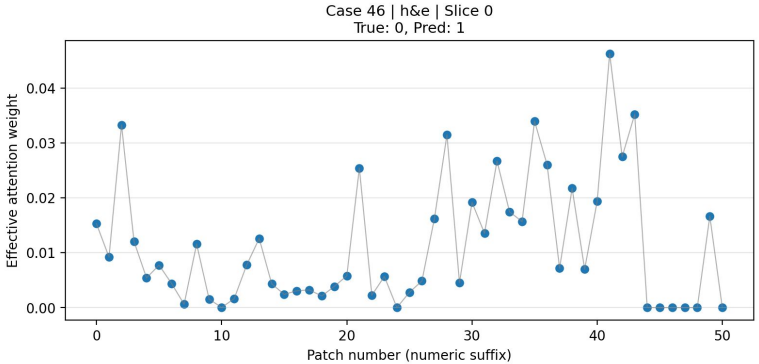


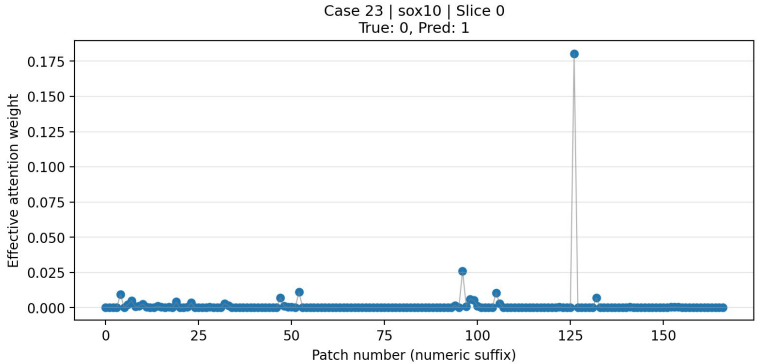


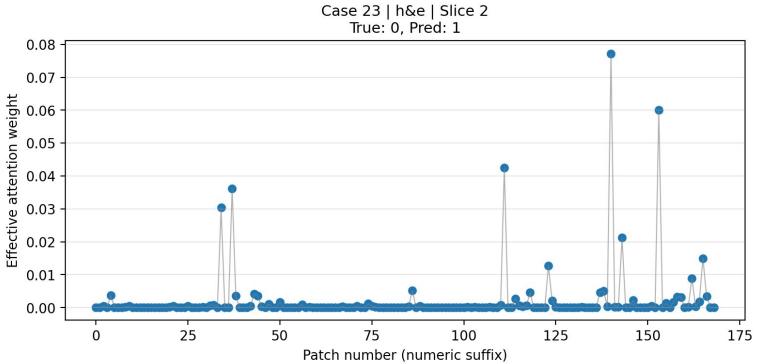


4. Effective Attention for Patches at Slice Level

Benign cases: predicted incorrectly

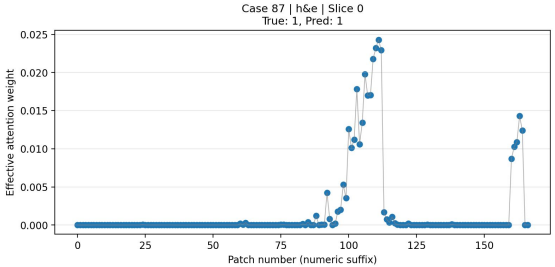


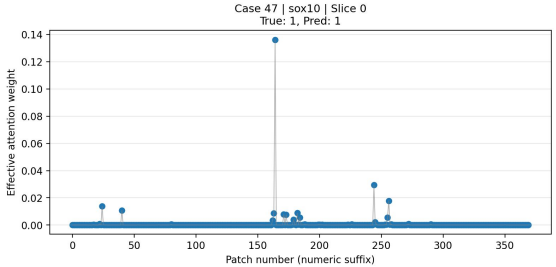


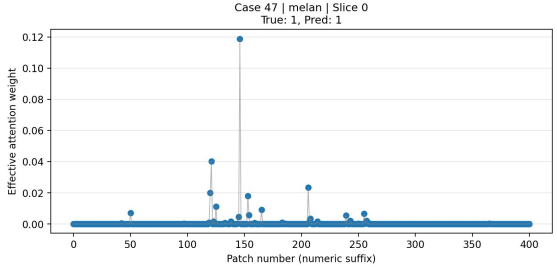


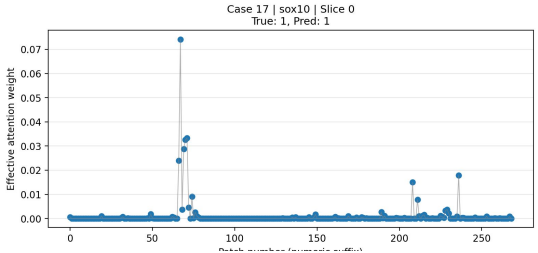
4. Effective Attention for Patches at Slice Level

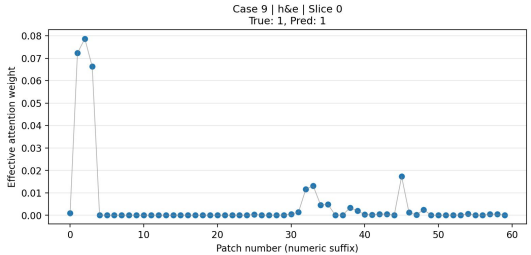
High grade cases predicted correctly: regions of high attention

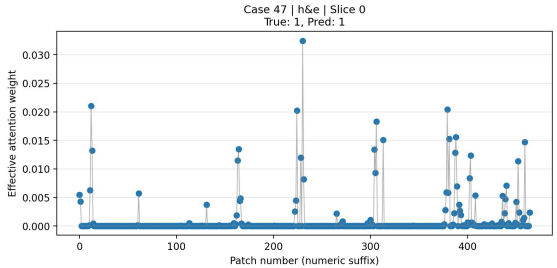


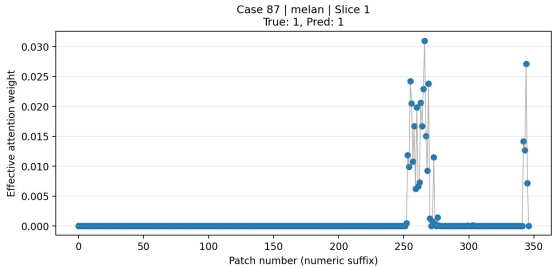


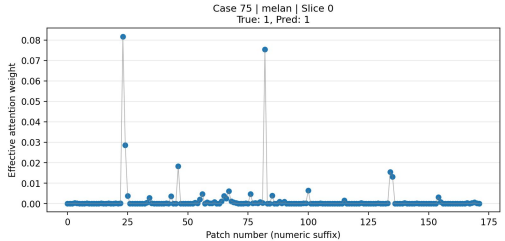


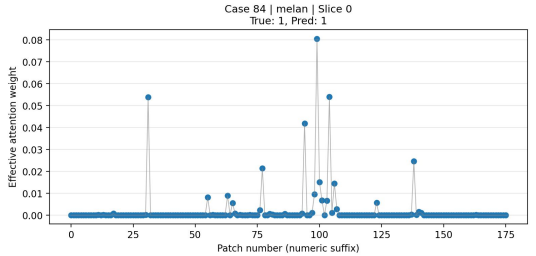






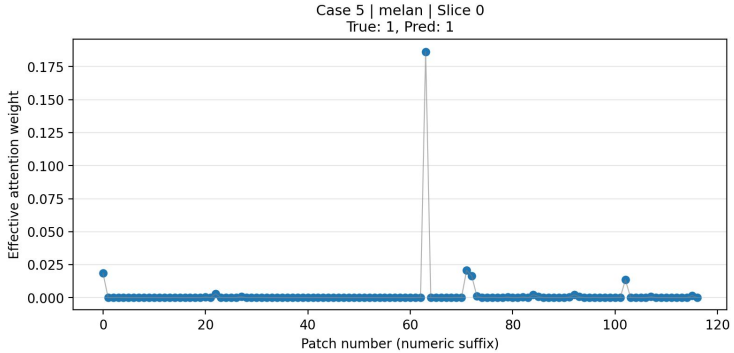


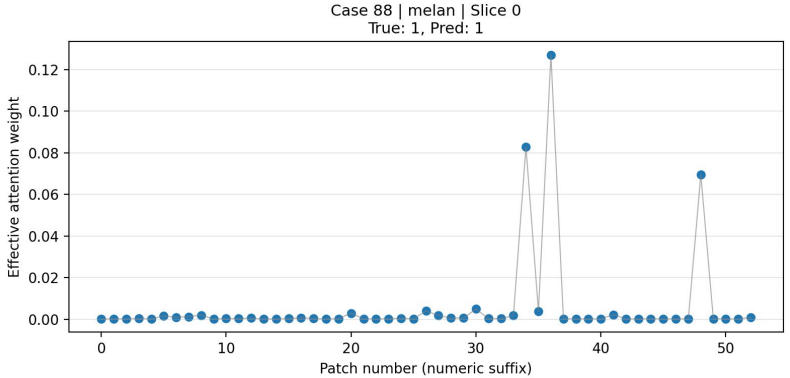


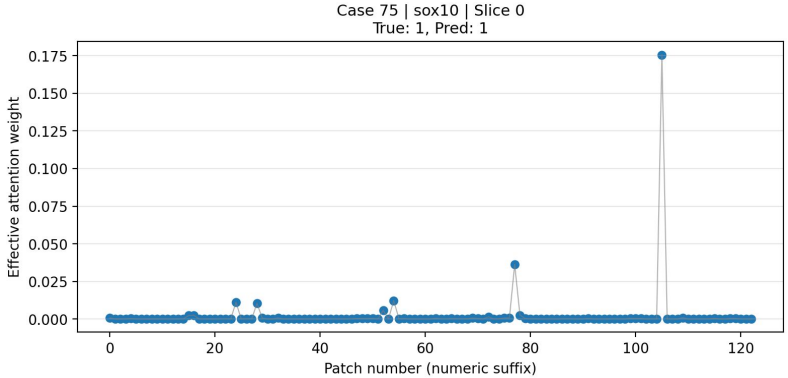


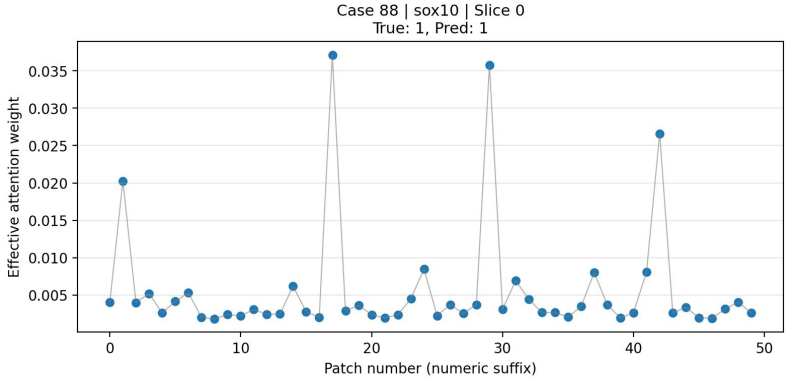
4. Effective Attention for Patches at Slice Level

High grade cases predicted correctly: Spikes of high attention with one patch

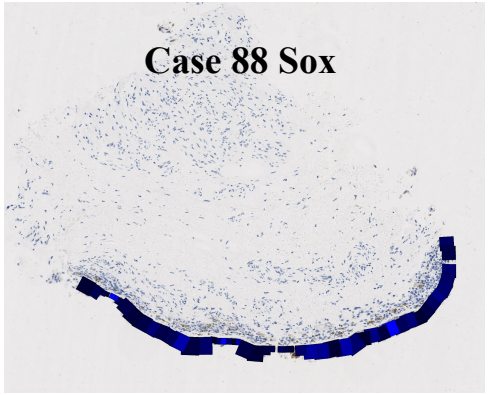


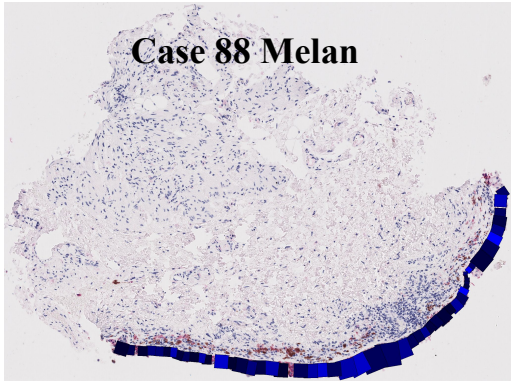


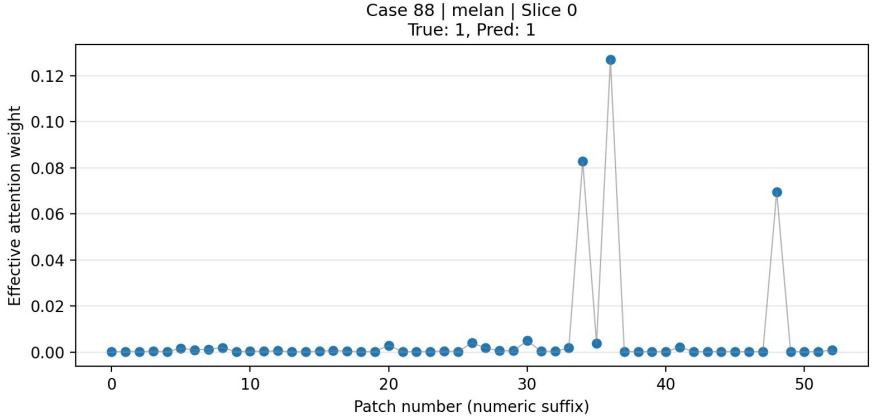


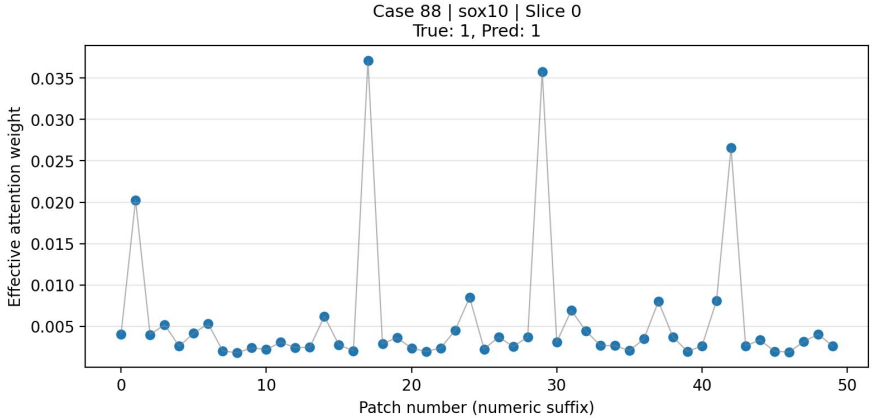


4. Effective Attention for Patches at Slice Level



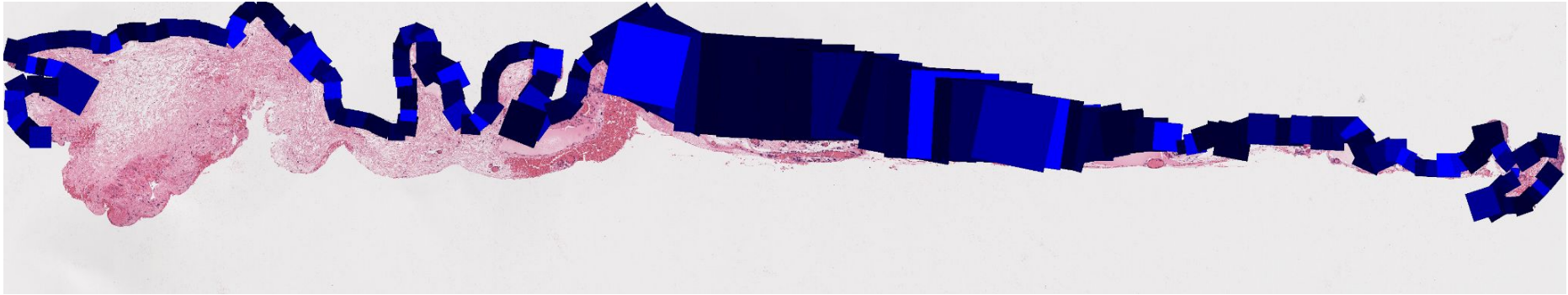


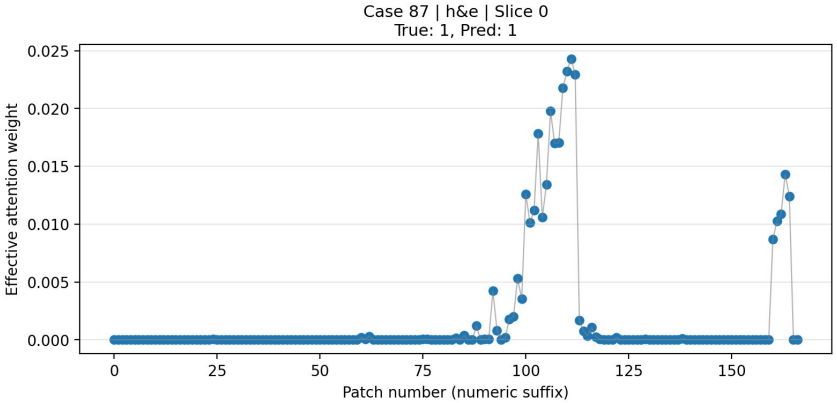




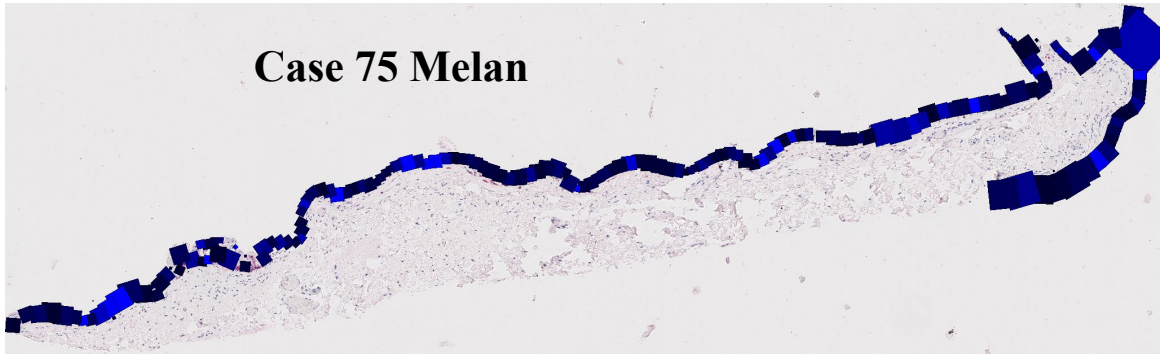
4. Effective Attention for Patches at Slice Level

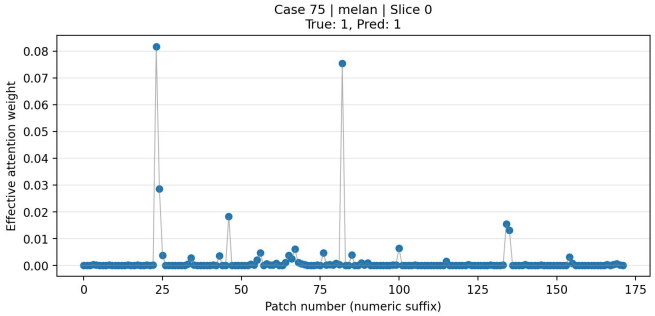
Case 87 H&E

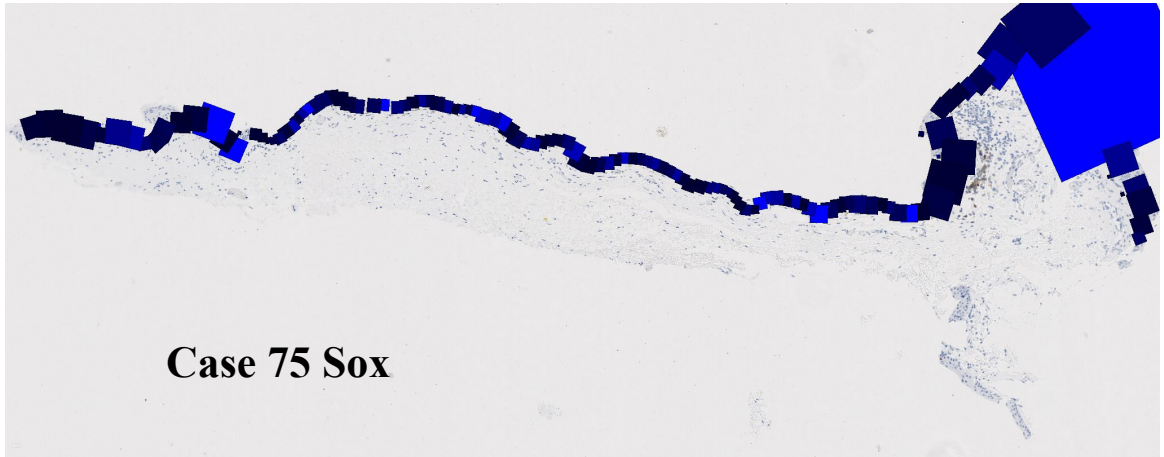


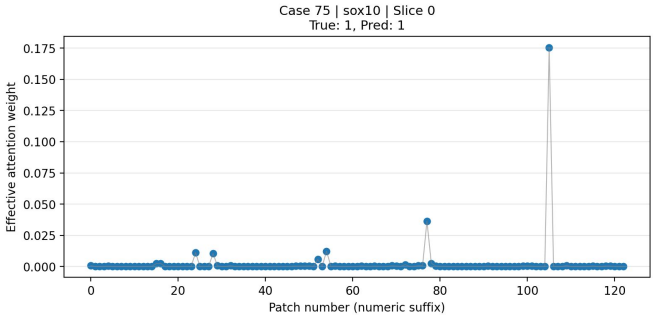


4. Effective Attention for Patches at Slice Level



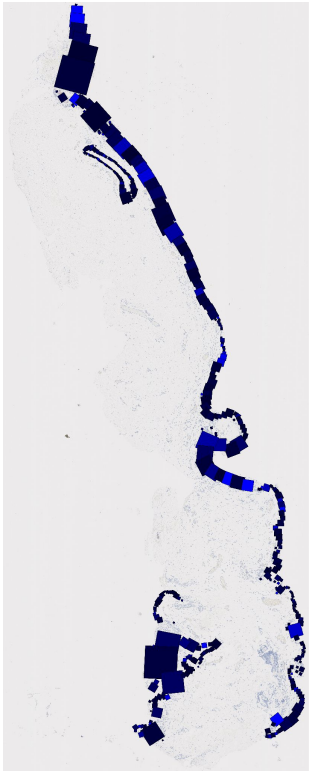




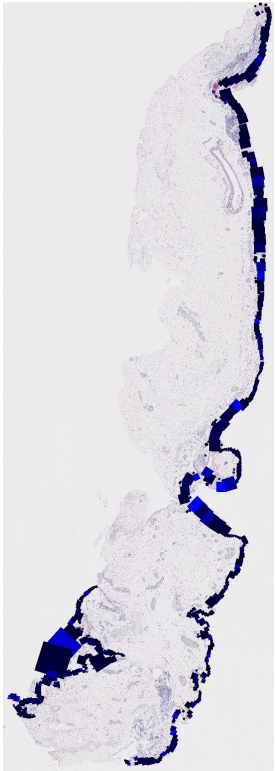


4. Effective Attention for Patches at Slice Level

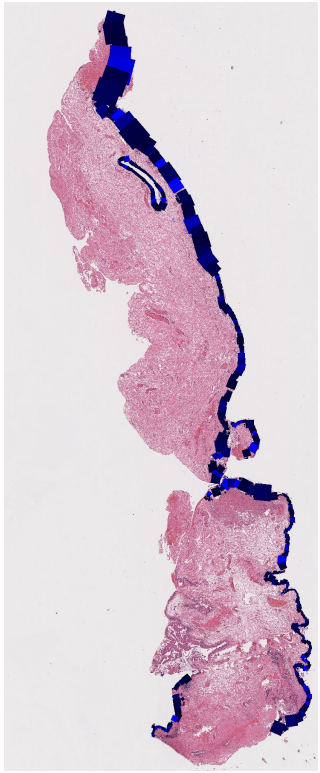
Case 47 Sox

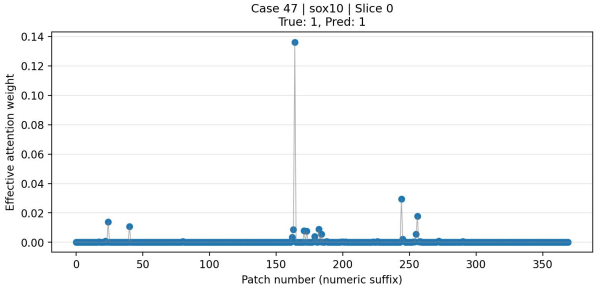


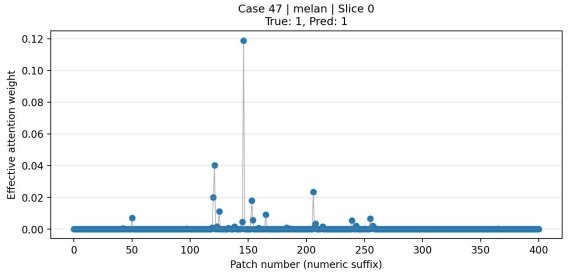
Case 47 Melan

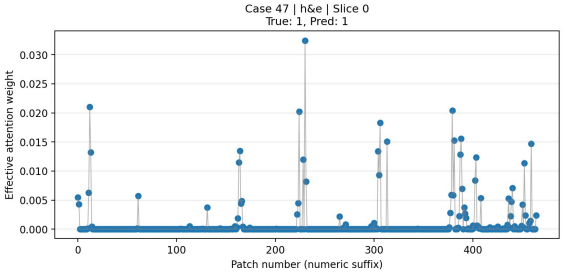


Case 47 H&E



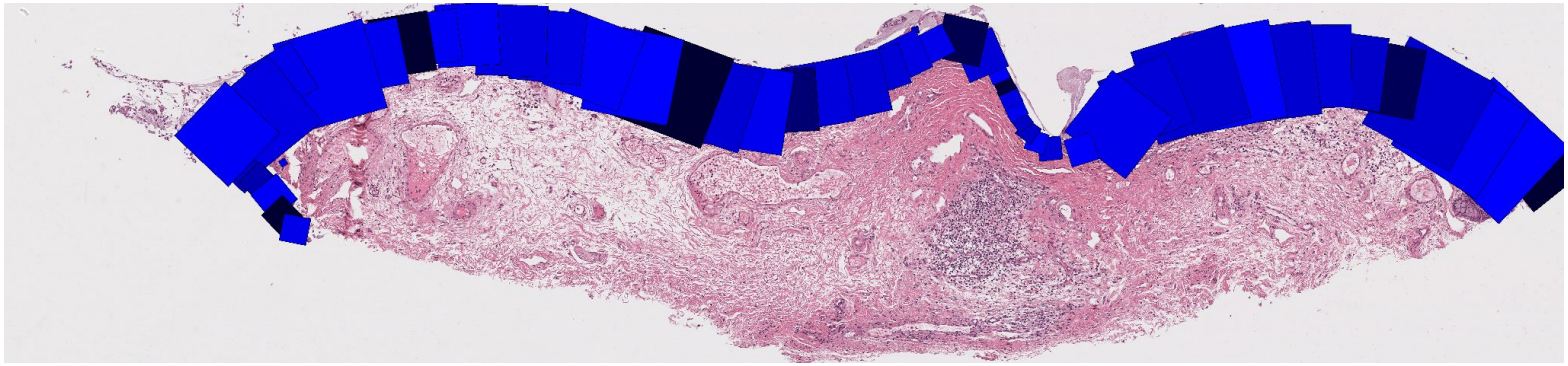


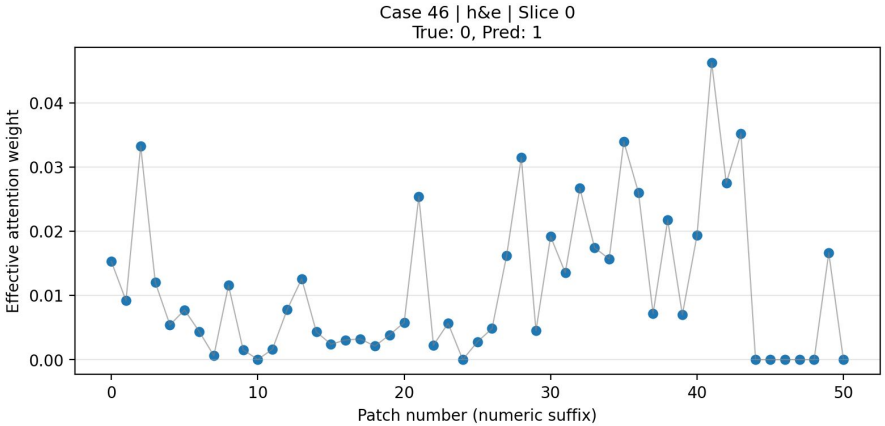




4. Effective Attention for Patches at Slice Level

Case 46 H&E





References

GenAI tools