

✓ Application of Deep Learning on Cancer Images

1000-719bMSB MIM UW, Neo Christopher Chung

In this lab, we make a prediction of lung nodules by a deep learning model and explain that decision making process by attention maps and CDAMs. For the simplicity, we provide a pre-trained model and use an existing medical image dataset. To streamline XAI, we are using a Python library called Obz.

➤ Download the obz package

From your venv, clone: `git clone http://github.com/obzai/obz`

If you have never used a virtual environment for Python, consider using uv.

Alternatively, you can download the repo from the website.

Take a note where you clone the obz repo, and set the working directory to that path. The following codes assume that you are in this repo root directory.

[] ↴ 1 cell hidden

➤ Download the pre-trained ViT model

[] ↴ 2 cells hidden

➤ Configure the ViT classifier based on DINO backbone

We are adding a binary classification head (see how `torch.nn.Linear`) onto a DINO backbone.

[] ↴ 10 cells hidden

➤ Extract features for outlier detection

[] ↴ 2 cells hidden

✓ Explain the AI model

- We first need to choose and set up appropriate XAI tools.

[] ↳ 1 cell hidden

- Then, we can run the XAI method to our data.

We first apply and visualize attention maps

[] ↳ 3 cells hidden

- Generate and visualize CDAM maps

[] ↳ 4 cells hidden

✓ Homework (for 2 weeks)

Train a Vision Transformer model on any of MedMNIST datasets (except NoduleMNIST), followed by prediction and explanation.

Essentially apply the above steps with a different ViT model trained on one of the MedMNIST datasets.

Double-click (or enter) to edit

```
!pip install torch torchvision timm matplotlib opencv-python scikit-learn
```

→ Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: timm in /usr/local/lib/python3.11/dist-packages
Collecting medmnist
 Downloading medmnist-3.0.2-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
 Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (1.2 MB)

```
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl.
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl.
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.w
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.w
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/p
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/pyth
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.11
Requirement already satisfied: safetensors in /usr/local/lib/python3.11/dis
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: scikit-image in /usr/local/lib/python3.11/di
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-pac
Collecting fire (from medmnist)
  Downloading fire-0.7.0.tar.gz (87 kB)
   _____ 87.2/87.2 kB 4.8 MB/s eta 0:0
  Preparing metadata (setup.py) ... done
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/di
```

```
!pip install medmnist
```

```
→ Requirement already satisfied: medmnist in /usr/local/lib/python3.11/dist-p  
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-pack  
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/di  
Requirement already satisfied: scikit-image in /usr/local/lib/python3.11/di  
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packa  
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: fire in /usr/local/lib/python3.11/dist-packa  
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-pack  
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dis  
Requirement already satisfied: termcolor in /usr/local/lib/python3.11/dist-  
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di  
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/  
Requirement already satisfied: scipy>=1.11.4 in /usr/local/lib/python3.11/d  
Requirement already satisfied: networkx>=3.0 in /usr/local/lib/python3.11/d  
Requirement already satisfied: imageio!=2.35.0,>=2.33 in /usr/local/lib/pyt  
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python  
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.11/d  
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.1  
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/d  
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/pytho  
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-p  
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/  
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/loc  
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/l  
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/loc  
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/li  
Requirement already satisfied: nvidia-cUBLAS-cu12==12.4.5.8 in /usr/local/l  
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/li  
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local  
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local  
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/loc  
Requirement already satisfied: nvidia-cusparseL-cu12==0.6.2 in /usr/local/  
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/p  
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib  
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/loca  
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/d  
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/d  
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p  
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11
```

```
!git clone https://github.com/obzai/obzai.git  
Cloning into 'obzai'...  
remote: Enumerating objects: 38, done.  
remote: Counting objects: 100% (38/38), done.  
remote: Compressing objects: 100% (35/35), done.  
remote: Total 38 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (38/38), 2.19 MiB | 12.59 MiB/s, done.  
Resolving deltas: 100% (6/6), done.
```

```
import sys  
sys.path.append("/content/obzai/src")  
from obzai.data_inspector.extractor import FirstOrderExtractor  
from obzai.data_inspector.detector import GMMDetector  
  
import torch  
import torch.nn as nn  
from torch.utils.data import DataLoader  
from torchvision import transforms  
from medmnist import PathMNIST, INFO  
from transformers import ViTModel, ViTConfig  
from tqdm import tqdm  
  
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")  
num_epochs = 1
```

```

from torch.utils.data import Subset
from torchvision import transforms
from medmnist import PathMNIST, INFO

info = INFO['pathmnist']
n_classes = len(info['label'])

transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[.5], std=[.5])
])

train_dataset = PathMNIST(split='train', transform=transform, download=True)
val_dataset = PathMNIST(split='val', transform=transform, download=True)
test_dataset = PathMNIST(split='test', transform=transform, download=True)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=16)
test_loader = DataLoader(test_dataset, batch_size=4, shuffle=False)

class DINO(nn.Module):
    def __init__(self):
        super().__init__()
        config = ViTConfig.from_pretrained('facebook/dino-vits8', attn_implementer='softmax')
        self.backbone = ViTModel(config)
        self.head = torch.nn.Linear(384, 9)

    def forward(self, x: torch.Tensor, output_attentions: bool = False):
        out = self.backbone(x, output_attentions=output_attentions)
        x = out["pooler_output"]
        x = self.head(x)
        if output_attentions:
            att = out["attentions"]
            return x, att
        else:
            return x

model = DINO().to(DEVICE)
optimizer = torch.optim.AdamW(model.parameters(), lr=1e-4)
criterion = nn.CrossEntropyLoss()

```

```
def train(model, loader):
    model.train()
    total_loss, correct, total = 0, 0, 0
    for imgs, labels in tqdm(loader, desc="Training"):
        imgs = imgs.to(DEVICE)
        labels = labels.view(-1).long().to(DEVICE)
        optimizer.zero_grad()
        outputs = model(imgs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
        preds = outputs.argmax(dim=1)
        correct += (preds == labels).sum().item()
        total += labels.size(0)
    return total_loss / len(loader), correct / total

def evaluate(model, loader):
    model.eval()
    total_loss, correct, total = 0, 0, 0
    with torch.no_grad():
        for imgs, labels in tqdm(loader, desc="Evaluating"):
            imgs = imgs.to(DEVICE)
            labels = labels.view(-1).long().to(DEVICE)
            outputs = model(imgs)
            loss = criterion(outputs, labels)
            total_loss += loss.item()
            preds = outputs.argmax(dim=1)
            correct += (preds == labels).sum().item()
            total += labels.size(0)
    return total_loss / len(loader), correct / total
```

```
import torch
torch.cuda.empty_cache()
for epoch in range(num_epochs):
    train_loss, train_acc = train(model, train_loader)
    val_loss, val_acc = evaluate(model, val_loader)
    print(f"[Epoch {epoch+1}] Train Loss: {train_loss:.4f}, Train Acc: {train_ac

test_loss, test_acc = evaluate(model, test_loader)
print(f"Test Accuracy: {test_acc:.4f}")

torch.save(model.state_dict(), "dino_pathmnist.pth")
print("Model saved as dino_pathmnist.pth")

samples, labels = next(iter(test_loader))
samples = samples[:5].to(DEVICE)
labels = labels[:5].to(DEVICE)

→ Training: 100%|██████████| 5625/5625 [1:22:17<00:00, 1.14it/s]
Evaluating: 100%|██████████| 626/626 [03:24<00:00, 3.05it/s]
[Epoch 1] Train Loss: 0.6130, Train Acc: 0.7771 | Val Acc: 0.8400
Evaluating: 100%|██████████| 1795/1795 [02:23<00:00, 12.53it/s]
Test Accuracy: 0.7397
Model saved as dino_pathmnist.pth
```

```
import matplotlib.pyplot as plt
```

```
CLASS_NAMES = {
    0: "background",
    1: "tissue",
    2: "epithelial",
    3: "lymphocyte",
    4: "plasma",
    5: "eosinophil",
    6: "connective",
    7: "muscle",
    8: "nerve"
}
```

```
samples, labels = next(iter(test_loader))
samples = samples[:5].to(DEVICE)
labels = labels[:5].to(DEVICE)

model.eval()
with torch.no_grad():
    logits = model(samples)
    predictions = torch.softmax(logits, dim=1)
    predicted_classes = predictions.argmax(dim=1).cpu().numpy()
```

```
predicted_classes = predictions.argmax(dim=1).cpu().numpy()
```

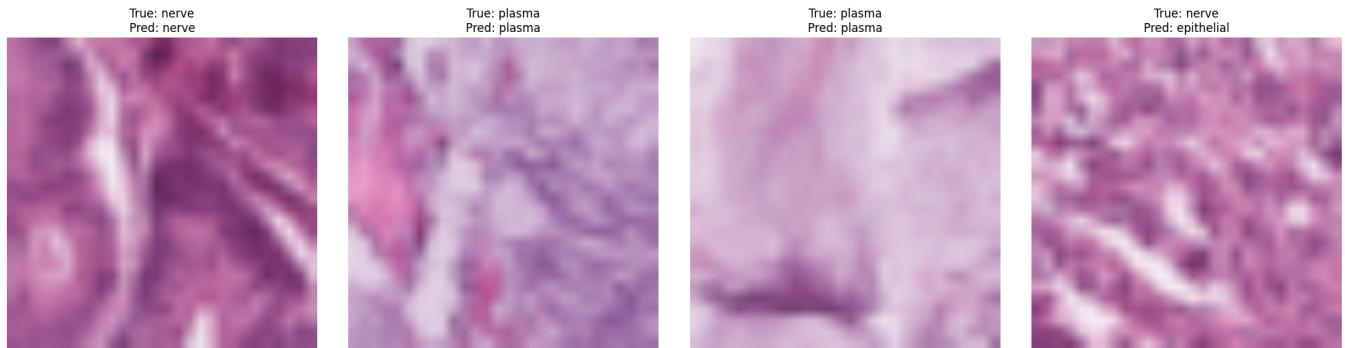
```
predicted_names = [CLASS_NAMES[int(p)] for p in predicted_classes]
true_names = [CLASS_NAMES[int(l)] for l in labels.cpu().numpy()]

fig, axes = plt.subplots(1, samples.size(0), figsize=(20, 5))

for i in range(samples.size(0)):
    image = samples[i].permute(1, 2, 0).cpu().numpy()
    image = (image * 0.5) + 0.5
    axes[i].imshow(image)
    axes[i].set_title(f"True: {true_names[i]}\nPred: {predicted_names[i]}")
    axes[i].axis('off')

plt.tight_layout()
plt.show()
```

```
→ <ipython-input-65-4aea588b1f37>:14: DeprecationWarning: Conversion of an ar
  true_names = [CLASS_NAMES[int(l)] for l in labels.cpu().numpy()]
```



```
model.eval()
with torch.no_grad():
    logits = model(samples)
    probs = torch.softmax(logits, dim=1)

print("Predicted classes:", predicted_names)

→ Predicted classes: ['nerve', 'plasma', 'plasma', 'epithelial']

samples = samples[:5].to(DEVICE)
labels = labels[:5].to(DEVICE)

with torch.no_grad():
    logits = model(samples)
    predictions = torch.softmax(logits, dim=1).argmax(dim=1).cpu().numpy()

predicted_classes = [CLASS_NAMES[int(pred)] for pred in predictions]

for i, pred_class in enumerate(predicted_classes):
    print(f"Prediction for Sample {i + 1}: {pred_class}")
```

→ Prediction for Sample 1: nerve
Prediction for Sample 2: plasma
Prediction for Sample 3: plasma
Prediction for Sample 4: epithelial

```
!pip install captum
```

```
→ Requirement already satisfied: captum in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-
Requirement already satisfied: numpy<2.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-
Requirement already satisfied: torch>=1.10 in /usr/local/lib/python3.11/dis
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/loc
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/l
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/loc
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/li
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/l
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/li
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/l
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/loc
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/p
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/loca
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/di
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/pytho
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11
```

```
from obzai.xai.xai_tool import CDAM, AttentionMap

cdam_tool = CDAM(model=model,
                  mode='vanilla',
                  gradient_type="from_logits",
                  gradient_reduction="average",
                  activation_type="sigmoid")
cdam_tool.create_hooks(layer_name="backbone.encoder.layer.11.layernorm_before")

attention_tool = AttentionMap(model=model,
                               attention_layer_id=-1,
                               head=None)
```

```
attention_maps = attention_tool.explain(samples)

fig, axes = plt.subplots(2, samples.size(0), figsize=(20, 5))

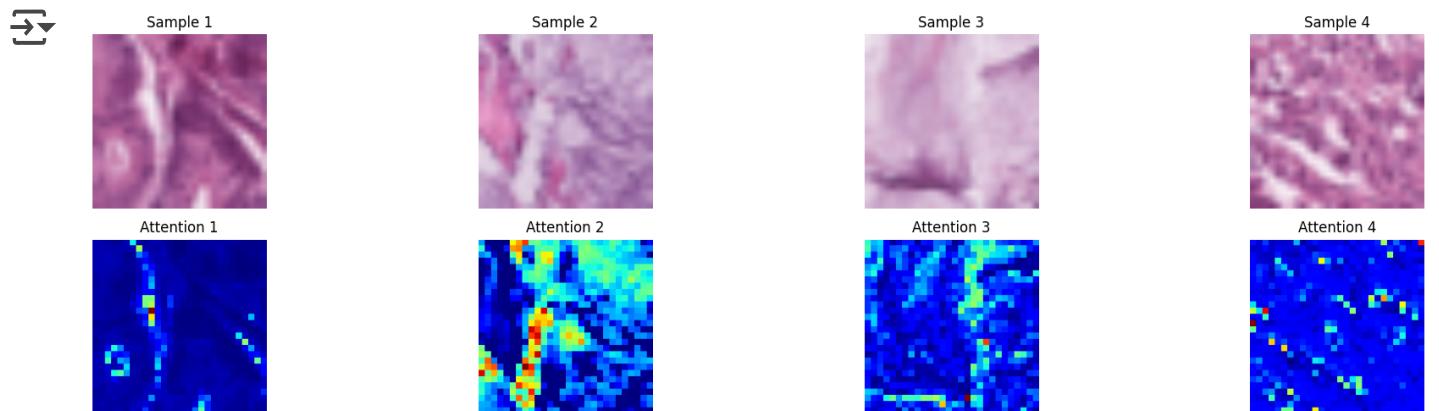
for i in range(samples.size(0)):
    image = samples[i].permute(1, 2, 0).cpu().numpy()
    image = (image * 0.5) + 0.5

    attention_map = attention_maps[i].cpu().numpy()

    axes[0, i].imshow(image)
    axes[0, i].set_title(f"Sample {i + 1}")
    axes[0, i].axis('off')

    axes[1, i].imshow(attention_map, cmap='jet')
    axes[1, i].set_title(f"Attention {i + 1}")
    axes[1, i].axis('off')

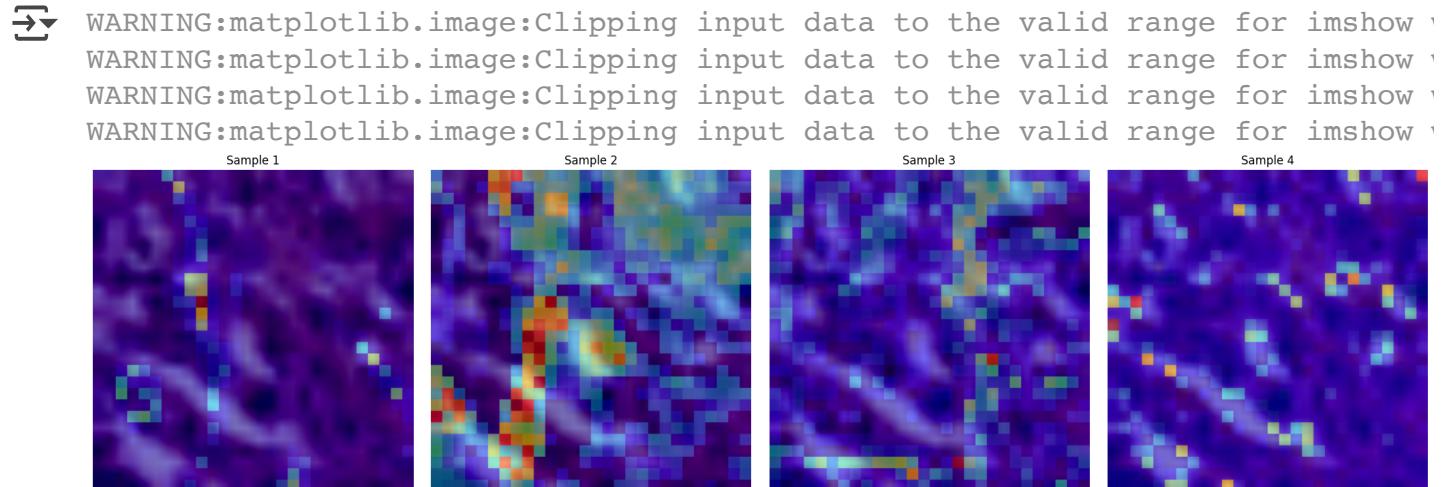
plt.tight_layout()
plt.show()
```



```

fig, axes = plt.subplots(1, samples.size(0), figsize=(20, 5))
for i in range(samples.size(0)):
    image = samples[i].permute(1, 2, 0).cpu().numpy()
    attention_map = attention_maps[i].cpu().numpy()
    axes[i].imshow(original_image, cmap='gray')
    axes[i].imshow(attention_map, cmap='jet', alpha=0.5)
    axes[i].set_title(f"Sample {i + 1}")
    axes[i].axis('off')
plt.tight_layout()
plt.show()

```



```

cdam_maps = cdam_tool.explain(samples, target_idx=predictions.tolist())

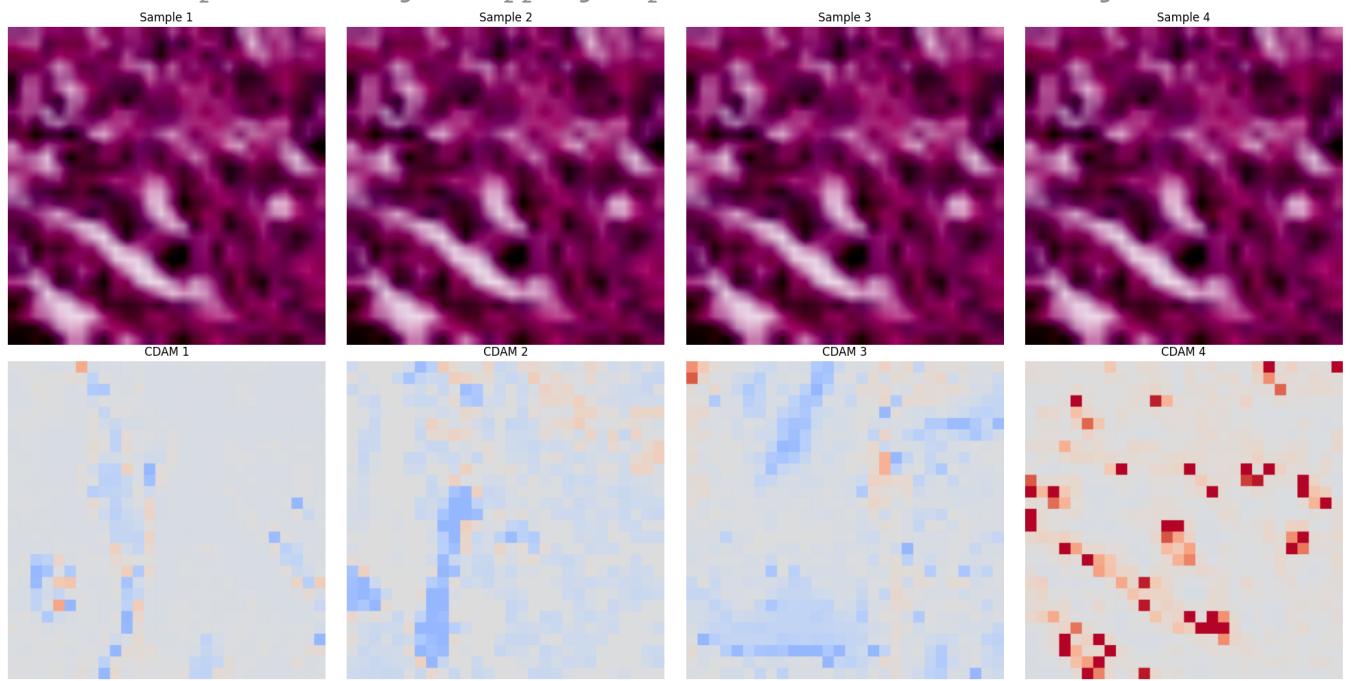
fig, axes = plt.subplots(2, samples.size(0), figsize=(20, 10))
for i in range(samples.size(0)):
    image = samples[i].permute(1, 2, 0).cpu().numpy()
    axes[0, i].imshow(original_image, cmap='gray')
    axes[0, i].set_title(f"Sample {i + 1}")
    axes[0, i].axis('off')

    cdam_map = cdam_maps[i].squeeze().cpu().numpy()
    axes[1, i].imshow(cdam_map, cmap='coolwarm', vmin=-cdam_maps.abs().max(), vmax=cdam_maps.abs().max())
    axes[1, i].set_title(f"CDAM {i + 1}")
    axes[1, i].axis('off')
plt.tight_layout()

```

```
plt.show()
```

→ WARNING:matplotlib.image:Clipping input data to the valid range for imshow
WARNING:matplotlib.image:Clipping input data to the valid range for imshow
WARNING:matplotlib.image:Clipping input data to the valid range for imshow
WARNING:matplotlib.image:Clipping input data to the valid range for imshow



```
fig, axes = plt.subplots(2, samples.size(0), figsize=(20, 10), gridspec_kw={'he
for i in range(samples.size(0)):
    original_image = samples[i].permute(1, 2, 0).cpu().numpy()
    cdam_map = cdam_maps[i].squeeze().cpu().numpy()
```

```

axes[0, i].imshow(original_image, cmap='gray')
axes[0, i].imshow(cdam_map, cmap='coolwarm', alpha=0.5, vmin=-cdam_maps.abs
axes[0, i].set_title(f"Sample {i + 1}")
axes[0, i].axis('off')

axes[1, i].hist(cdam_map.ravel(), bins=30, color='blue', alpha=0.7)
axes[1, i].set_title(f"Histogram {i + 1}")
axes[1, i].set_xlabel('Value')
axes[1, i].set_ylabel('Frequency')
plt.tight_layout()
plt.show()

```

→ WARNING:matplotlib.image:Clipping input data to the valid range for imshow
 WARNING:matplotlib.image:Clipping input data to the valid range for imshow
 WARNING:matplotlib.image:Clipping input data to the valid range for imshow
 WARNING:matplotlib.image:Clipping input data to the valid range for imshow

