CONCH: Inferences

1. Bag level accuracy (Auc, f1)
2. Provide prompts
3. img2text, tissue segmentation, captioning
4. Text2img

Sir aamir:

1. Other tasks rel to conch (code issues and confusions)

Tissue diagnosis report (morphology)

Cell level morphological classification (see kinds of cells in mss and msi dataset) (tissue description)

Tumor description (type, stage, subtype)

Wsi (without breaking tissue) and patch level classification

1. Conch for classification of tissue and tumor classification
2. Json file improvement research (cell types that affect mss), tumor description
3. Else check Fine Tuning, and prepare dataset for it based on paper
4. Conch datasets, check caption generation what is the caption and where it comes from

Model memory size (conch): single run (~1.9 GB)

Number of parameters for conch (1.7 billion)

architectures

Unimodal encoder

Total num of tasks and total num of encoders/decoders

Which encoder to which task

Dataset (details, before training how was data fed into data, prepare and processing of training data)

Input window size for text and images

Limit of prompts(captions)

Output fixing

Make sure no bg in input images

Slide_labels csv to check for msi and mss then choose accordingly

Visualize the slides ur going to use in training

Pick only the ones with cancer slides (darker than the rest of the cell on slide)

10 msi slides and 10 mss slides

Look into the zeroshot_classification func for zeroshot weights as tensor is [1,1] so accuracy will be 0.5

Zero-Shot Classification Setup:

- Construct prompts for each class using contrastively aligned image and text encoders.
- Classify images based on the closest prompt embedding in the shared space

    - *The pretraining process uses both contrastive and captioning objectives. The contrastive objectives align the image and text encoders by maximizing the cosine-similarity scores between paired image and text embeddings while the captioning objective maximizes the likelihood of generating the correct text conditioned on the image and previously generated text*
    - *Ensemble prompts training also used to get better predictive results*
    - *Note that the zero-shot performance of CONCH is (better) highly competitive when compared to few-shot supervised learning.*
    - *retrieval in a zero-shot setting, i.e., retrieving the corresponding text entry based on an image query (image-to-text, abbreviated as "i2t"), or vice versa (text-to-image, abbreviated as "t2i").*
    - *When classifying a WSI using zero-shot transfer, in addition to computing an aggregated, slide-level prediction, we can also create a heatmap to visualize the cosine-similarity score between each tile in the slide and the text prompt corresponding the predicted class label. Regions with high similarity scores are deemed by the model to be close matches with the diagnosis (e.g. invasive*

ductal carcinoma, IDC) while regions with low similarity scores do not match the diagnosis. In an example of breast IDC slide, we find that
regions highlighted in the heatmap closely resemble the tumor regions as delineated by pathologist annotation

- *Since the slide-level prediction score is a simple average of the similarity scores of the top-K tiles for a given class, the heatmap enables human interpretability by directly highlighting regions involved in the model's decision making process, which can be displayed in high resolution to the human user for inspectioN.*
-

.

WSI Processing:

- Divide Whole Slide Images (WSIs) into tiles.
- Compute similarity scores for tiles using prompt-based embeddings.
- Aggregate scores using top-K pooling for slide-level predictions.

Performance Evaluation:

- Evaluate zero-shot performance on subtyping and grading tasks across various datasets.
- Report metrics like Cohen's κ and balanced accuracy.

Supervised Embedding Evaluation:

- Use linear probing for ROI-level tasks and ABMIL for slide-level tasks.
- Report similar metrics as for zero-shot tasks.

Image Interpretation:

- Visualize annotated images and corresponding heatmaps.
- Highlight tiles with high similarity scores indicating specific morphological features.

PRE TRAINING DATASET CURATION:

1) detecting histopathology images (as single images or sub-images)
2) splitting captions that refer to image panels into separate captions into sub-captions
3) aligning sub-images with sub-captions within each image panel.

- *The second challenge is that a significant portion of EDU and most of PMC OA are in the form of figure panels, where the image consists of multiple sub-images arranged in a panel with parts of the caption addressing all or some of the sub-images.*

PubMed Open Access Dataset —> (3 steps) PMC Paths

1. Manually clean Edu (smaller)
2. Yolov5 object detection
    - *We use an object detection model (YOLOv5)88 to generate bounding boxes for extracting detected images.*
    - *To avoid the laborious task of manually labeling ground truth bounding boxes in EDU, we generate **synthetic** data by **randomly** selecting single-panel images and arranging them in an image panel.*
    - *We iteratively refine the detection model by validating on a small subset (< 0.5%) of PMC OA and adding incorrectly labeled samples to the training set.*
3.
4.

Object Detection Model (YOLOv5):

- Use YOLOv5 to generate bounding boxes for extracting detected images.
- Generate synthetic data by arranging single-panel images into an image panel.
- Refine the detection model iteratively by validating on a small subset of PMC OA and adding incorrectly labeled samples to the training set.

Caption Splitting:

- Collect a dataset of original and split captions to fine-tune a GPT-style model pretrained on PubMed and other medical texts.
- Treat caption splitting as a causal language modeling problem: fine-tune the model to take the original full caption as input and predict sub-captions separated by "Next caption:".

- Use the fine-tuned model to perform caption splitting.

Image-Caption Alignment:

- Train a CLIP model on the cleaned EDU dataset and PMC OA single figures.
- Compute image embeddings and text embeddings in the aligned latent space.
- For each image embedding, compute the cosine-similarity score with each text embedding.
- Pair images and captions based on the highest cosine-similarity scores.

Creating PMC-Path Dataset:

- Apply the above steps to PMC OA to create PMC-Path, a pathology-specific image-caption dataset.
- Combine PMC-Path with EDU to form an unfiltered pre training dataset of 1,786,362 image-caption pairs.

Filtering the Dataset:

- Parse captions to exclude samples referencing non-human animals, forming a dataset of 1,170,647 human pairs.
- Train a classifier to identify H&E stains and further filter the dataset, creating a dataset of 457,372 pairs.

Performance Evaluation:

- Assess the performance of CONCH pretrained on the human-only dataset for downstream tasks.
- Determine that the human-only dataset performed best on downstream tasks.

VISUAL-LANGUAGE PRE TRAINING:

1. Captions → tokens → Text encoder —> text embeddings (caption splitting)

2. Image → Image tokens → Image encoder (transformer block) → attention poolings → object detector (image embeddings)
3. Multi-modal decoder (contrastive alignment of image and text embeddings (fusion multimodal decoder) pooled imgs + encoded text
4. Multimodal decoder: transformer block → cross attention block → transformer block ....
    - *We set the image size to 448 × 448*

TASKS:

- classification of image tiles and gigapixel WSIs, cross-modal image-to-text and text-to-image retrieval, image segmentation, and image captioning in a total of thirteen diverse benchmarks

ACCURACY and EVALUATION:

- On the slide-level tasks CONCH achieves a balanced accuracy score of 84.7%, 94.2%
- Evaluate on four slide-level classification tasks:
    - TCGA BRCA (invasive breast carcinoma subtyping)
    - TCGA NSCLC (non-small cell lung cancer subtyping)
    - TCGA RCC (renal cell carcinoma subtyping)
    - DHMC LUAD (lung adenocarcinoma histologic pattern classification)
- And 3 rois

Zero-shot Performance on Slide-level Benchmarks

Task	Model	Zero-shot Accuracy/Score	Improvement Over Next Best Model	p-value
NSCLC Subtyping	CONCH	90.0%	+11.3% (PLIP)	< 0.01
RCC Subtyping	CONCH	89.3%	+8.9% (PLIP)	< 0.01
BRCA Subtyping	CONCH	84.0%	+28.7% (PLIP)	< 0.01
LUAD Pattern Classification	CONCH	κ = 0.236	+0.16 (PLIP)	ρ = 0.014

Zero-shot Performance on ROI-level Benchmarks

Task	Model	Zero-shot Accuracy/Score	Improvement Over Next Best Model	p-value
SICAP	CONCH	κ = 0.711	+0.158 (BiomedCLIP)	< 0.01
CRC100k	CONCH	79.1%	+11.7% (PLIP)	< 0.01
WSSS4LUAD	CONCH	71.9%	+9.5% (PLIP)	< 0.01

Average Performance

Task Type	Model	Average Accuracy/Score	Compared Models
Slide-level	CONCH	90.5%	PLIP (86.6%), BiomedCLIP (87.9%)
ROI-level	CONCH	-	CTransPath (similar)

Cross-modal Retrieval Performance

Dataset	Task	Model	Mean Recall	Improvement Over Next Best Model	p-value

Source A	Text-to-image	CONCH	68.8%	+31.5% (BiomedCLIP)	< 0.01
Source B	Text-to-image	CONCH	39.0%	+15.1% (BiomedCLIP)	< 0.01
TCGA LUAD	Text-to-image	CONCH	24.0%	+5.3% (BiomedCLIP)	p = 0.22
Source A	Image-to-text	CONCH	-	-	-
Source B	Image-to-text	CONCH	-	-	-
TCGA LUAD	Image-to-text	CONCH	-	-	-

Zero-shot Segmentation Performance

Task	Model	Average Dice Score	Average Recall Score	Average Precision Score	p-value (Dice)	p-value (Recall)	p-value (Precision)
SICAP	CONCH	0.601	0.751	0.672	p = 0.08 (PLIP), < 0.01 (BiomedCLIP)	< 0.01 (Both)	p = 0.024 (PLIP), < 0.01 (BiomedCLIP)
DigestPath	CONCH	0.569	0.684	0.644	< 0.01 (Both)	< 0.01 (Both)	p = 0.024 (PLIP), < 0.01 (BiomedCLIP)

Captioning Performance

Metric	Model	Score	Improvement Over Next Best Model	p-value

METEOR	CONCH	0.193	+0.071 (GIT-base), +0.068 (GIT-large)	< 0.01
ROUGE	CONCH	0.215	+0.08 (GIT-base), +0.062 (GIT-large)	< 0.01

Model Details

1. Memory Size and Parameters:
    - The CONCH model has 2 billion parameters.
2. Architectures:
    - The model uses a multimodal architecture.
    - It includes both unimodal and multimodal encoders and decoders.
3. Unimodal Encoder:
    - It employs a Vision Transformer (ViT) for image encoding.
    - A BERT-based transformer is used for text encoding.
4. Tasks and Encoders/Decoders:
    - The model is evaluated on 6 tasks using 3 encoders and 3 decoders.
    - Specific tasks include histopathology image classification and captioning.

Dataset and Training

1. Dataset Details:
    - The training dataset consists of 1,786,362 image-caption pairs.
    - It includes both human histopathology and animal histopathology images.
    - For human-only data, 1,170,647 pairs are used.
    - Further filtered to 457,372 pairs for H&E stained human histopathology.
2. Data Preparation and Processing:

- Data preparation involved filtering out non-human and non-H&E stain samples.
- Synthetic data was generated to avoid manual labeling of ground truth bounding boxes.

3. Input Window Size:
    - Text input window size:
    - Image input size:
4. Limit of Prompts (Captions length (128 chars):
    - There is no explicit limit mentioned for the number of captions (prompts).

Additional Information

- The model aligns image and text embeddings using a CLIP-based approach.
- Cosine similarity is used to match image embeddings with text embeddings.

Hyperparameters used in visual-language pretraining: 8 × 80GB NVIDIA A100 GPUs were used for training. Effective batch size used for optimization is batch size × gradient accumulation steps. The maximum sequence length for captions is set to 128.
Hyperparameters used in pretraining the vision model: 4 × 80GB NVIDIA A100 GPUs were used for training. Batch size refers to the total batch size across GPUs (64)
Hyperparameters used in pretraining the language model: In-house pathology reports were first de-identified using regex pattern matching before tokenization. 4 × 80GB NVIDIA A100 GPUs were used for training. Batch size refers to the total batch size across GPUs. Effective batch size used for optimization is batch size × gradient accumulation steps. The sequence length of training examples was set to the maximum sequence length supported by the model (i.e. 512).

Hyperparameter	Value
Batch size	1
Weight decay	1e-5
AdamW β	(0.9, 0.999)

Peak learning rate 1e-4

Learning rate schedule Cosine

Epochs 20

Hyperparameters used in slide-level supervised classification: A single 24GB NVIDIA GeForce RTX 3090 GPU was used for each ABMIL model using weakly-supervised learning and slide-level labels.

Hyperparameter Value

Batch size 16

Weight decay 0.2

AdamW β (0.9, 0.999)

Learning rate 1e-4

Warmup steps 10

Early stopping patience 10

Epochs 40

Hyperparameters used in caption fine-tuning: A single 24GB NVIDIA GeForce RTX 3090 GPU was used for training. The maximum sequence length for captions is set to 128. Top-K sampling with K = 50 was used as decoding strategy at generation time.

Tissue segmentation (wsi)

Classification (patches + wsi)

Msi / mss

Msih and non msih