Transformers for medical image segmentation

THOME Nicolas – Prof. at SORBONNE University
ISIR Lab, MLIA TEAM
Transformers everywhere since 2017

NLP: BERT, GPT-3/4, Chat-GPT, etc

Vision since '21: Vision Image Transformer (ViT)

Explain quantum theory to a child in the style of snoop dogg

Yo, little homie, let me break it down for you
Quantum theory, it's a crazy thing, it's true
See, the world is made up of tiny, tiny particles
That are always moving, and changing, and actin' wild

These particles, they don't always behave like we think
Sometimes they're waves, and sometimes they're points, it's a trip
They can be in two places at once, and they can be in two states
It's a mind-blowing concept, but that's the quantum fate

So don't be scared, little homie, it's all good
Quantum theory, it's just the way the world works, it's understood
It's a wild ride, but it's a cool one too
Quantum theory, it's the way the world does what it do.
Transformer in medical image analysis

Used in various contexts and tasks

- Image classification, detection, *e.g.* COVID, Semantic segmentation
- Image Registration
- Image Generation
- Im-2-im translation

Transformer in medical image analysis

Used in various contexts and tasks

• Image classification, detection, e.g. COVID, Semantic segmentation
• Image Registration
• Image Generation
• Im-2-im translation
Transformer in medical image analysis

Used in various contexts and tasks

• Image classification, detection, e.g. COVID, semantic segmentation
• Image Registration
• Image Generation
• Im-2-im translation

Focus on this talk

• Paper on transformer every day...

• By no means exhaustive literature review
Focus on this talk

1. **Transformers**
2. Vision Image Transformer
3. Transformers for medical image segmentation
4. Current trend & Perspectives

Architecture: main features and processing
- Long-range interactions
- Efficient self-attention
From sequence to set

• A sequence of elements → a set of tokens, no order
  • Token: primitives, elementary elements of data
    • Text: token are e.g. words
    • Image: token are e.g. patches

Text

<table>
<thead>
<tr>
<th>Text</th>
<th>Tokenization</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Hello I love you&quot;</td>
<td>&quot;Hello&quot;, &quot;I&quot;, &quot;love&quot;, &quot;you&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;love&quot;, &quot;Hello&quot;, &quot;I&quot;, &quot;you&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;I&quot;, &quot;love&quot;, &quot;Hello&quot;, &quot;you&quot;</td>
</tr>
</tbody>
</table>
Input embedding

• Token: input vector in $\mathbb{R}^t$
  • Word: $t = |V|$, $V$ vocabulary
  • Image patch: $t = s^2$, where $s$ is the patch size

• Input embedding: linear projection $\mathbb{R}^t \rightarrow \mathbb{R}^d : e_i = x_i W^e$
Positional encoding

• Sequence → set of token:
  • Permutation invariant
  • Loosing structural information from data

• Recovering structure: positional encoding (PE)
  • Mapping token position \( t \) to a vector \( \mathbf{p}_t \in \mathbb{R}^d \)
  • Seminal PE: sinusoidal

\[
\begin{align*}
\mathbf{p}^{(i)} &:= \begin{cases} 
\sin(\omega_k \cdot t), & \text{if } i = 2k \\
\cos(\omega_k \cdot t), & \text{if } i = 2k + 1 
\end{cases} \\
\omega_k &= \frac{1}{10000^{2k/d}}
\end{align*}
\]
Sinusoidal positional encoding

- Unique vector $\mathbf{p}_t$ for each position $t$
- $p_t(i) \in [-1;1]$: natural normalization

$K = PPt$

- Models relative position
- Positional similarity:

$d=128$, max length of token set = 50
Positional encoding

- Other possible encoding, can be learned
- Final embedding:

\[
\begin{align*}
\text{Input sequence} & \quad I \quad am \quad a \quad Robot \\
\text{Word embedding} & \quad v_0 = \text{embedding vector}(I) \quad v_1 = \text{embedding vector}(am) \quad v_2 = \text{embedding vector}(a) \quad v_3 = \text{embedding vector}(Robot) \\
\text{Positional Encoding Matrix} & \quad p_0 = \text{Positional vector}(I) \quad p_1 = \text{Positional vector}(am) \quad p_2 = \text{Positional vector}(a) \quad p_3 = \text{Positional vector}(Robot) \\
\text{Output of positional encoding layer} & \quad y_0 = \text{Positional encoding}(I) \quad y_1 = \text{Positional encoding}(am) \quad y_2 = \text{Positional encoding}(a) \quad y_3 = \text{Positional encoding}(Robot) \\
\end{align*}
\]

=> Input of transformer!
Transformer [1] : the encoder

- A stack of $N$ transformer blocks
  - Input: a set of embedded tokens
  - Output: a set of re-embedded tokens

Transformer: self attention

- **The most important and specific module in transformers**
- Project the input set into 3 sets
  - Query: sought info
  - Key: context elements
  - Value: retrieved
Self-attention

\[ X \in \mathbb{R}^{wh \times d}, W_q \in \mathbb{R}^{d \times d}, W_k \in \mathbb{R}^{d \times d}, W_v \in \mathbb{R}^{d \times d} \]
\[ Q = XW_q, K = XW_k, V = XW_v \]
\[ A = \text{Softmax} \left( \frac{QK^T}{\sqrt{d}} \right) \]
\[ Y = AV \]
Self-attention: conclusion

- Each token $y_i$ in $Y$: computed a linear combination of $v_i$
  - Enables to model **global interactions** between $v_i$ tokens: full contextual information
  - ≠ ConvNets in vision, interactions limited by the size of the receptive field
  - ≠ RNNs for sequence processing, interactions limited by vanishing gradients

- **Self attention: $O(N^2)$ complexity**
  - Expensive (or impossible) for large $N$
Multi-headed attention

• High-level idea: multiple self-attention in parallel

  • Each head: attend to different parts

  • Combine the heads’ outputs

[Image: Diagram of multi-headed attention]

[Credit: Anna Goldie]
Multi-headed attention

- Concatenate the heads’ outputs
- Use a linear layer: desired output size
Layer normalization

• Normalization on joint channel and spatial dimensions

• Stabilize training, faster convergence
Layer normalization

- Normalization on joint channel and spatial dimensions

\[
\mu_n = \frac{1}{K} \sum_{k=1}^{K} x_{nk}
\]

\[
\sigma_n^2 = \frac{1}{K} \sum_{k=1}^{K} (x_{nk} - \mu_n)^2
\]

\[
\hat{x}_{nk} = \frac{x_{nk} - \mu_n}{\sqrt{\sigma_n^2 + \epsilon}}, \hat{x}_{nk} \in \mathbb{R}
\]

\[
\text{LN}_{\gamma,\beta}(x_n) = \gamma \hat{x}_n + \beta, x_n \in \mathbb{R}^K
\]

\(\beta, \gamma\) learnable parameters
Layer normalization + residual connections

Residual connections
• Better gradient flow (vanishing gradients)
• Leverage input encoding, e.g. PE
Feed-Forward Network (FFN)

• Position-wise FFN: applied to each token separately and identically

\[
FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2
\]
FNN + residual Layer Norm

\[ \text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \]
Transformer: conclusion

• Importance of attention: global interactions between tokens
• On the other hand relaxes inductive biases
  • e.g. ConvNets translation equivariant
  • vs transformers permutation equivariant
  • More flexibility to learn adequate mapping
  • Needs more data
1. Transformers
2. Vision Image Transformer
3. Transformers for medical image segmentation
4. Current trend & Perspectives
Vision Image Transformer (ViT) [2]

- Direct application of transformer’s encoder for images
- Learned on JFT (300.10⁶ images)
- Extra learnable token: used for class prediction
  - “Learned” pooling wrt visual tokens

Detection Transformer (DETR) [3]

DETR encoder

- Conv Backbone + Standard ViT with PE at each transformer layer
DETR decoder

- Learned object queries (OQ 100)
- Self-attention (can be omitted at 1st decoder layer)
- Cross-attention
  - Query: OQ added
  - Key: encoder output + PE
  - Value: encoder output
- Decoder output: 2 branches
  - FFN for class prediction
    - $\emptyset$ for background
  - FFN for BB prediction

$$[center_x, center_y, height, width]$$
DETR training

- Matching between the set of prediction and set of BB in supervision
- Best match between the sets using the Hungarian algo

\[ \mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[ - \log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \right] \]

\[ \hat{\sigma} = \arg \min_{\sigma \in \mathbb{N}} \sum_{i=1}^{N} -\mathbb{1}_{c_i \neq \emptyset} \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{c_i \neq \emptyset} \mathcal{L}_{\text{box}}(b_i, \hat{b}_i) \]

\[ \mathcal{L}_{\text{box}} = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_i) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\hat{\sigma}(i)}|| \]
DETR: conclusion

• Simple model
• Works well for large objects, less good for small objects
Deformable DETR [4]

- Deformable attention

\[
\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k \in \Omega_k} A_{mqk} \cdot W'_{m} x_k \right]
\]

\[
\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[ \sum_{k=1}^{K} A_{mqk} \cdot W'_{m} x(p_q + \Delta p_{mqk}) \right]
\]

- Query: input vector from tensor.
  - For each head, predict a 3K value
    - 2K elements for the offset for getting K Keys (here K=3)
    - K elements for getting K attention weight
- Value: for each head, weighted average of the K sampled keys
- Complexity: \(O(WH.K)\) vs \(O((WH)^2)\)

Deformable DETR

• Applied in multi-resolution feature maps
• Improve DETR effectiveness for small objects requiring high-resolution feature maps
Transformer in segmentation

- Swin-Transformer [5]
  - Multi-resolution transformer
    - Local attention in lower-layers
      - Shifted windows at layers l/l+1
    - Patch merging => larger receptive field

---

Transformer in segmentation

- SegFormer [6]
  - Efficient attention, at multi-scale

1. Transformers
2. Vision Image Transformer
3. Transformers for medical image segmentation
4. Current trend & Perspectives
Context: 2D organ segmentation example

Organs segmentation illustration

Pancreas automatic segmentation
Segmentation: importance of long-range dependencies

U-Net [A]: unable to represent full context

Segmentation example with U-Net’s receptive field (red square)

Trans U-Net [7], U-Transformer [8]

• Seminal works for using transformers in medical image segmentation
• Adding self-attention on the bottleneck of a U-Net
  • Inspired from non-local networks [9]
U-Transformer [8]

- **U-Transformer**: self and cross attention in medical image segmentation
  - Self-attention in bottleneck
  - Cross attention to improve super-resolution in skip connections
Architecture: Multi-Head Cross-Attention

**MHCA**: Filter high resolution features based on semantically richer features from the encoder.

\[ Y : \text{Semantically richer features from bottleneck} \]
\[ S : \text{High resolution features from skip connections} \]
## Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TCIA</td>
<td>76.13 (± 0.94)</td>
<td>76.82 (± 1.26)</td>
<td>77.71 (± 1.31)</td>
<td>77.84 (± 2.59)</td>
<td><strong>78.50</strong> (± 1.92)</td>
</tr>
<tr>
<td>IMO</td>
<td>86.78 (± 1.72)</td>
<td>86.45 (± 1.69)</td>
<td>87.29 (± 1.34)</td>
<td>87.38 (± 1.53)</td>
<td><strong>88.08</strong> (± 1.37)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pancreas</td>
<td>69.71 (± 3.74)</td>
<td>68.65 (± 2.95)</td>
<td>71.64 (± 3.01)</td>
<td>71.87 (± 2.97)</td>
<td><strong>73.10</strong> (± 2.91)</td>
</tr>
<tr>
<td>Gallbladder</td>
<td>76.98 (± 6.60)</td>
<td>76.14 (± 6.98)</td>
<td>76.48 (± 6.12)</td>
<td>77.36 (± 6.22)</td>
<td><strong>78.32</strong> (± 6.12)</td>
</tr>
<tr>
<td>Stomach</td>
<td>83.51 (± 4.49)</td>
<td>82.73 (± 4.62)</td>
<td>84.83 (± 3.79)</td>
<td>84.42 (± 4.35)</td>
<td><strong>85.73</strong> (± 3.99)</td>
</tr>
<tr>
<td>Kidney(R)</td>
<td>92.36 (± 0.45)</td>
<td>92.88 (± 1.79)</td>
<td>92.91 (± 1.84)</td>
<td>92.98 (± 1.70)</td>
<td><strong>93.32</strong> (± 1.74)</td>
</tr>
<tr>
<td>Kidney(L)</td>
<td>93.06 (± 1.68)</td>
<td>92.89 (± 0.64)</td>
<td>92.95 (± 1.30)</td>
<td>92.82 (± 1.06)</td>
<td><strong>93.31</strong> (± 1.08)</td>
</tr>
<tr>
<td>Spleen</td>
<td>95.43 (± 1.76)</td>
<td>95.46 (± 1.95)</td>
<td>95.43 (± 2.16)</td>
<td>95.41 (± 2.21)</td>
<td><strong>95.74</strong> (± 2.07)</td>
</tr>
<tr>
<td>Liver</td>
<td>96.40 (± 0.72)</td>
<td>96.41 (± 0.52)</td>
<td>96.82 (± 0.34)</td>
<td>96.79 (± 0.29)</td>
<td><strong>97.03</strong> (± 0.31)</td>
</tr>
</tbody>
</table>
Results
Segmentation example with U-Net’s receptive field (red square) and U-Transformer’s attention map.
Results

Ground Truth  Cross-attn level 1  Cross-attn level 2  Cross-attn level 3
3D medical image segmentation

Challenges:
- Size of the input
- Large memory requirements
- 180Gb for U-Net with image size 512x512x256

Common strategies to reduce the memory footprint:
- Downsampling ⇒ Drop in quality
- Limited model size
- Train on 2D slices
- Train on patches ⇒ No full contextual information
Approaches based on patches

To keep the **full resolution**, work on patches, e.g.:
- Original image size: **512x512x256**
- Cropped patch size: **128x128x64**

- Full context lost
- Even on patch: full context challenging!
Swin-UNet [10]

- Window attention (~Swin) in a 2D multi-resolution transformer
- Patch merging: pooling

nn-Former [11]

- Global self-attention in bottleneck
- Local self-attention in higher-resolution feature maps
  - ~ 3D Swin-UNet

Multi-resolution transformers: limitations

- **Windowed transformers** designed to reduce the complexity, e.g. Swin
  - **BUT:** no more long-range attention for high resolution feature maps

*Windowed input at different hierarchy levels*
CoTR: Convolutional NN and Transformer [12]

- **CoTr**: Conv encoder => flattened multi-scale feature
  - Deformable transformer encoder (DeTrans) in multi-res input
  - Several DeTrans layers, sent to conv decoder

---

CoTR: Convolutional NN and Transformer [12]

- Good performances on several datasets
- Deformable attention => reasonable to train

Table 1. Dice scores of our CoTr and several competing methods on the BCV test set. CoTr* and CoTr† are two variants of CoTr with small CNN-encoders

<table>
<thead>
<tr>
<th>Methods</th>
<th>Param (M)</th>
<th>Sp</th>
<th>Ki</th>
<th>Gb</th>
<th>Es</th>
<th>Li</th>
<th>St</th>
<th>Ao</th>
<th>IVC</th>
<th>PSV</th>
<th>Pa</th>
<th>AG</th>
<th>Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETR (VIT-B/16-rand)</td>
<td>100.5</td>
<td>95.2</td>
<td>92.3</td>
<td>55.6</td>
<td>71.3</td>
<td>96.2</td>
<td>80.2</td>
<td>89.7</td>
<td>83.9</td>
<td>68.9</td>
<td>68.7</td>
<td>60.5</td>
<td>78.4</td>
</tr>
<tr>
<td>SETR (VIT-B/16-pre)</td>
<td>100.5</td>
<td>94.8</td>
<td>91.7</td>
<td>55.2</td>
<td>70.9</td>
<td>96.2</td>
<td>76.9</td>
<td>89.3</td>
<td>82.4</td>
<td>69.6</td>
<td>70.7</td>
<td>58.7</td>
<td>77.8</td>
</tr>
<tr>
<td>CoTr w/o CNN-encoder</td>
<td>21.9</td>
<td>95.2</td>
<td>92.8</td>
<td>59.2</td>
<td>72.2</td>
<td>96.3</td>
<td>81.2</td>
<td>89.9</td>
<td>85.1</td>
<td>71.9</td>
<td>73.3</td>
<td>61.0</td>
<td>79.8</td>
</tr>
<tr>
<td>CoTr w/o DeTrans</td>
<td>32.6</td>
<td>96.0</td>
<td>92.6</td>
<td>63.8</td>
<td>77.9</td>
<td>97.0</td>
<td>83.6</td>
<td>90.8</td>
<td>87.8</td>
<td>76.7</td>
<td>81.2</td>
<td>72.6</td>
<td>83.6</td>
</tr>
<tr>
<td>APSS [5]</td>
<td>45.5</td>
<td>96.5</td>
<td>93.8</td>
<td>65.6</td>
<td>78.1</td>
<td>97.1</td>
<td>84.0</td>
<td>91.1</td>
<td>87.9</td>
<td>77.0</td>
<td>82.6</td>
<td>73.9</td>
<td>84.3</td>
</tr>
<tr>
<td>PP [26]</td>
<td>33.9</td>
<td>96.1</td>
<td>93.1</td>
<td>64.3</td>
<td>77.4</td>
<td>97.0</td>
<td>85.3</td>
<td>90.8</td>
<td>87.4</td>
<td>77.2</td>
<td>81.9</td>
<td>72.8</td>
<td>83.9</td>
</tr>
<tr>
<td>Non-local [20]</td>
<td>32.8</td>
<td>96.3</td>
<td>93.7</td>
<td>64.6</td>
<td>77.9</td>
<td>97.1</td>
<td>84.1</td>
<td>90.8</td>
<td>87.7</td>
<td>77.2</td>
<td>82.1</td>
<td>73.3</td>
<td>84.1</td>
</tr>
<tr>
<td>TransUnet [4]</td>
<td>43.5</td>
<td>95.9</td>
<td>93.7</td>
<td>63.1</td>
<td>77.8</td>
<td>97.0</td>
<td>86.2</td>
<td>91.0</td>
<td>87.8</td>
<td>77.8</td>
<td>81.6</td>
<td>73.9</td>
<td>84.2</td>
</tr>
<tr>
<td>CoTr*</td>
<td>27.9</td>
<td>96.4</td>
<td>94.0</td>
<td>66.2</td>
<td>76.4</td>
<td>97.0</td>
<td>84.2</td>
<td>90.3</td>
<td>87.6</td>
<td>76.3</td>
<td>80.8</td>
<td>72.9</td>
<td>83.8</td>
</tr>
<tr>
<td>CoTr†</td>
<td>36.9</td>
<td>96.2</td>
<td>93.8</td>
<td>66.5</td>
<td>76.8</td>
<td>97.1</td>
<td>86.9</td>
<td>90.8</td>
<td>87.8</td>
<td>77.7</td>
<td>82.8</td>
<td>73.2</td>
<td>84.7</td>
</tr>
<tr>
<td>CoTr</td>
<td>41.9</td>
<td>96.3</td>
<td>93.9</td>
<td>66.6</td>
<td>78.0</td>
<td>97.1</td>
<td>88.2</td>
<td>91.2</td>
<td>88.0</td>
<td>78.1</td>
<td>83.1</td>
<td>74.1</td>
<td>85.0</td>
</tr>
</tbody>
</table>
Global attention in multi-resolution transformers (GLAM) [13]

- Architecture based on hierarchical transformer (e.g. Swin, nn-Former)
  - Can also be included in any multi-resolution model (e.g. Conv)
  - GLAM Motivation: Full attention even in high-resolution features

GLAM block

- Define learnable global tokens in each window, cf CLS in VIT
  - Window self-attention (W-MSA): attention between visual and global tokens
  - Global attention (G-MSA) between global token
- G-MSA: indirection between all visual tokens
  - Break computational complexity of full attention between visual token
  - But enables full indirect interaction between them
FINE : Full resolution memory transformer module [14]

- Extends GLAM for full context modelling in 3D segmentation
- Reminder: state-of-the-art methods based on 3D crops

- Original image size: 512x512x256
- Cropped patch size: 128x128x64

**Goal:** learning a global representation of the full volume from batch training with crops

FINE architecture

• 2 levels of global tokens:
  • Window tokens (red)
  • Volume tokens (green)

• \(W\)-transformer in 3D crops

• \(G\)-transformer between window and volume token

=> (indirect) **full interaction between all voxels!**

![Diagram](image-url)
Results

**Synapse BCV [17]**: CT scans Abdominal multi-organs segmentation

7 classes

30 volumes

**Metrics**:  
- Dice score in % (DSC)  
- 95% Hausdorff distance in mm (HD95)
Results
Results

Input image  FINE attention

- Ribs
- Spine
- Aorta
1. Transformers
2. Vision Image Transformer
3. Transformers for medical image segmentation
4. Current trend & Perspectives
Transformer in medical image analysis

• Key feature: self-attention
  • Long-range dependencies, global context
  • Potential in image segmentation: best of both words between accurate info and full context
  • Challenge: full attention computation

• Transformer used in several medical image analysis tasks: Image Registration, Image Generation, Im-2-im translation

• Discussion and perspectives
  • Self-supervised learning
  • Multi-task learning, Multi-modal learning
  • Foundation models

61
Self-supervised learning and transformers

• The way transformers have been trained in NLP: pretext task
  • Predict masked word (BERT), next word (GPT), etc

• Several pretext tasks in vision
  • Pretext tasks (RotNet, MAE), contrastive methods (BYOL, MoCO)

• In medical image analysis: pre-train on generalist or medical images [15]


• Generally leads to better OOD robustness
Multi-task learning

• Usual to combine tasks
  • e.g. classification and segmentation in medical images [16]

Multi-task learning & foundation models

Current trend: Train huge transformers, e.g. GPT-3/GPT-4 in NLP

- General-purpose AI, can be fine-tuned on several tasks => foundation model
- Trained on diverse datasets, predict next word
- Prompted ("in-context learning") with emerging properties
  - Can beat even model fine-tuned for the target task (e.g. translating to English)
  - Not fully understood
Multi-modal learning & foundation models

Transformers naturally handle multi-modal data: **token homogeneity**

- **Different goals depending on the task** [17]
  - **Fusion**: complementarity between models
  - **Alignment**: making modalities closer
- **Multi-modal**: can also be used as a self-supervised signal

Multi-modal learning & foundation models

Multi-modal foundation models, e.g. NLP and images:

- Contrastive Language-Image Pre-training (CLIP): image/ text encoder, alignment

Multi-modal learning & foundation models

Multi-modal foundation models, e.g. NLP and images:

DALL-E [19]: image decoder

Flamingo [18]: text decoder

Figure 4: GATED XATTN-DENSE layers. To condition the LM on visual inputs, we insert new cross-attention layers between existing pretrained and frozen LM layers. The keys and values in these layers are obtained from the vision features while the queries are derived from the language inputs. They are followed by dense feed-forward layers. These layers are gated so that the LM is kept intact at initialization for improved stability and performance.


Foundation models

Combination of multi-modal and multi-task learning

• Unified-IO [20]
• Segment Anything Model (SAM) [21]


Foundation model in healthcare

- CheXzero [22]: POC of CLIP-based model

Foundation models: towards generalist medical AI? [23]

• Solve more diverse and challenging tasks than current medical AI models
• Relaxing the need for labels in specific tasks.
• Potential of foundation models:
  • Flexible and dynamic interactions
  • Multi-modal inputs and outputs
  • Medical domain knowledge, more elusive?

Foundation models: towards generalist medical AI? [22]

- Important potential applications

- In-context learning for effective adaptability?

For example, a clinician might say, “Check these chest X-rays for Omicron pneumonia. Compared to the Delta variant, consider infiltrates surrounding the bronchi and blood vessels as indicative signs”40.
Foundation models: risks and challenges

• Access to huge-scale datasets
  • Diverse, anonymized data
  • Pre-training on generalist data?

• Robustness and certification: uncertainty, OOD detection, stability, etc
  • A general issue in deep learning, exacerbated with general-purpose AI systems
  • Crucial and especially sensible in healthcare

• Explainability, interpretability: harder or easier?

• Ethical considerations
  • Biases and fairness/discriminability
  • Privacy, informed consent, transparency
Thank you for your attention!

Questions?