Vision Language Model Sensing for pervasive devices

Sasika Amarasinghe

28-05-2025

MENS MANUS AND MACHINA

SMU SINGAPORE MANAGEMENT UNIVERSITY

School of Computing and Information Systems

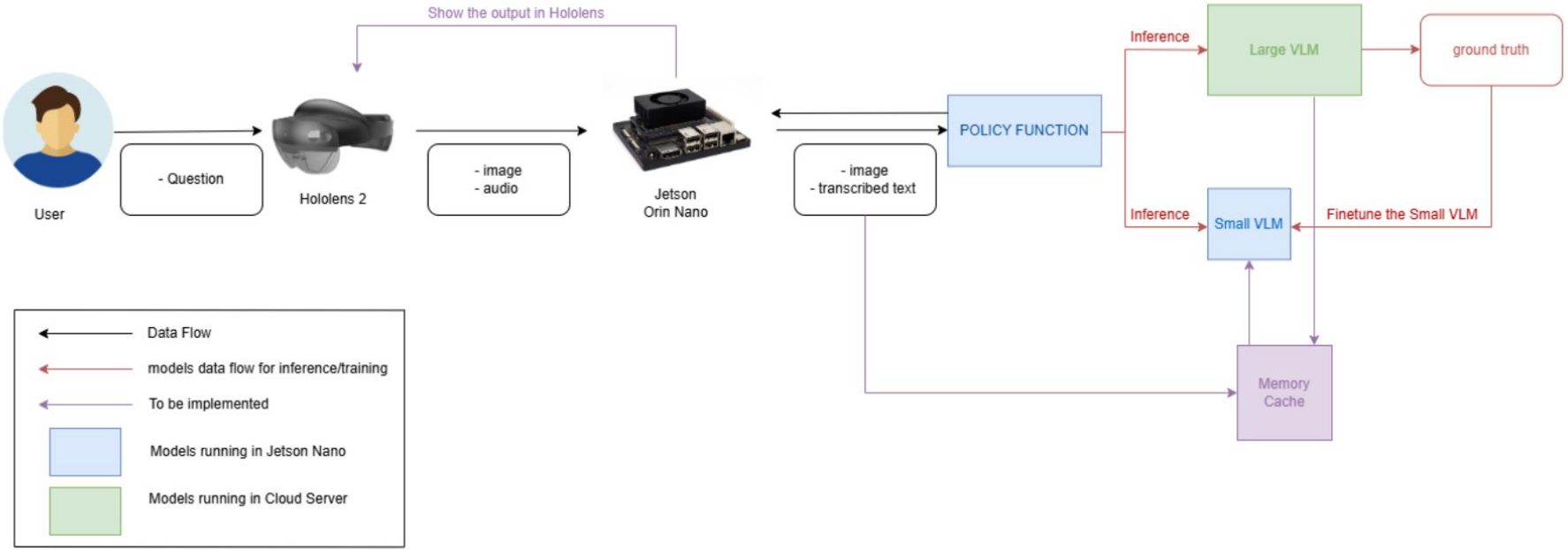
Application Scenario

A user enters a room and asks 20 questions about the environment. These questions combine visual input with natural language queries. The first set of questions is answered by a large vision-language model (VLM) running on the server, acting as the expert model.

Based on these expert answers, a smaller VLM running on the edge device adapts, enabling it to answer subsequent questions from the user.

Research Motivation

Can knowledge be effectively transferred from a large VLM to a smaller, edge-deployed VLM?



- The small VLM (student model), which runs on the edge, has generalized knowledge.
- In contrast, the large VLM (expert model), running on the server, possesses more powerful, expert-level knowledge.
- If the expert model handles every user request, it would result in high latency and increased power consumption.
- Our goal is to **specialize the small model for the user's environment by using the expert model's knowledge with minimal guidance**.

https://www.youtube.com/watch?v=iMDYdlx24jM

Study 1 : VLMs Comparison

Study 2 : ViTs Comparison

Study 3 : Attention Distillation Approach

Study 4 : Reducing the number of features(dims) per token

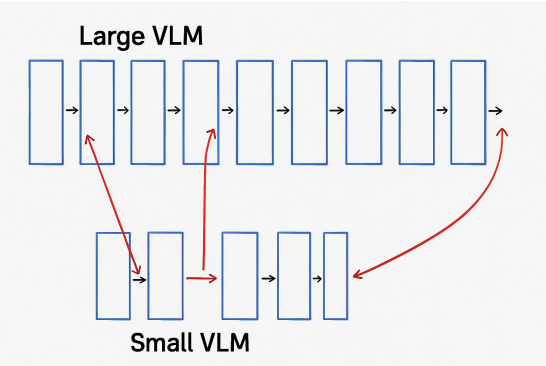
Study 1 : VLMs Comparison

Common tasks VLMs are known for,

- Image Captioning
- **Visual Question Answering**
- Visual Grounding

Hypothesis

Structural similarity of activation tensors for the same input and different output responses ≤ **Structural similarity of activation tensors for the same input and similar output responses**



We selected the VQA task because it goes along with the application scenario

- Method
  - Compare the activations from the layers from beginning, middle and final layers from two models
- Result & Conclusion
  - The MSE value does not support our hypothesis.
  - Since the model architectures are completely different, we began comparing two VLMs which have somewhat similar architectures.

Model	Moondream 2	LLaMA 3.2 Vision
Variants (parameters)	1.8B	11B, 90B
Text encoder	Microsoft Phi 1.5	Meta LLaMA 3
Vision encoder	MobileViT-styled custom ViT	ViT-L/14
Inference Device	Nvidia Jetson Orin Nano (8GB shared RAM/VRAM)	Nvidia RTX 6000 (48GB VRAM)

Study 1 : VLMs Comparison : Florence Model results comparison

Because of the similar architecture (from the same family) I used lightweight VLMs named Florence models to comparison

Feature	Florence-2-large-ft	Florence-2-base-ft
Number of parameters	0.77 B	0.23 B
Number of layers	12 encoder, 12 decoder layers	6 encoder, 6 decoder layers
Number of Heads per Layer	16	12
Number of features per token	1024	768
Vision Encoder	DaViT (Dual Attention Vision Transformer)	DaViT (Dual Attention Vision Transformer)

- Average distance = average MSE between 2-dimension reduced layer embedding (attention map) from the expert model and the student model
- Keeping the Middle layer Embedding from the student model as the reference
- Data cell format (Average distance between **correct** image samples (22) ,Average distance between **Incorrect** image samples (20))

	Vision Encoder	Text Encoder
Middle layer - 2	-	3.9215, 3.3729
Middle layer - 1	3.7855, 4.0453	3.9470, 3.2881
Middle layer	4.1587, 3.8863	4.0245, 3.4320
Middle layer + 1	4.1634, 3.6856	3.9538, 3.5810
Middle layer + 2	4.1092 , 3.7855	4.0095 , 3.5492

Hypothesis : For correct predictions the values from intermediate layers should be less than the values from the incorrect predictions

This table on the right shows a such comparison for the task visual grounding.
Here I have compared two modalities simultaneously. Since it is complicated to keep track of two modalities as once, we started comparing only the vision encoder.

MENS MANUS AND MACHINA

SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

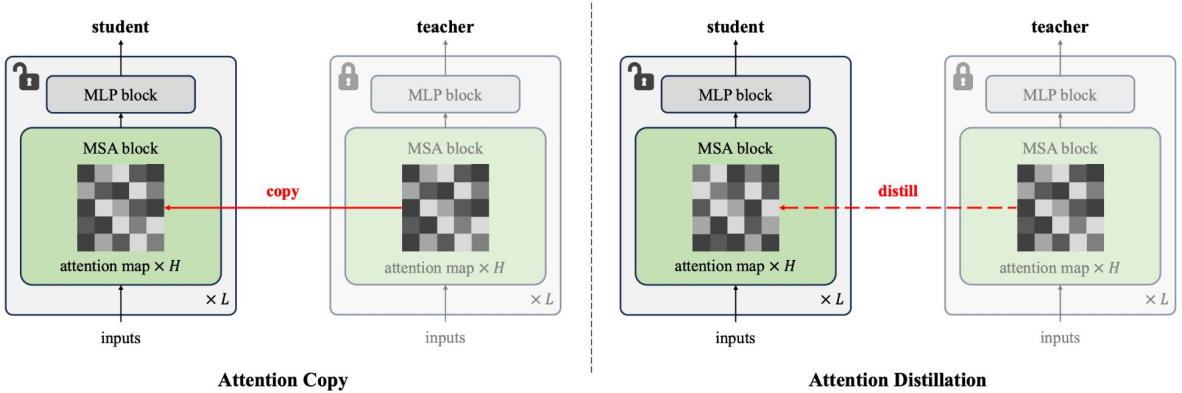
Study 2 : ViTs Comparison

We started studying about ViTs because we want to do a comparison in one modality.

Li, A. C., Tian, Y., Chen, B., Pathak, D., & Chen, X. (2024). *On the Surprising Effectiveness of Attention Transfer for Vision Transformers*. arXiv preprint arXiv:2411.07799.

- In this work, we found out that attention distillation can fully match **fine-tuning accuracy** on ImageNet-1K (85.7% top-1 accuracy).
- However in this paper the expert and student models that they are using have the same architecture. [ViT-L is used]
  In our case the model architectures are different.

$$f_{\text{attn}} = \underbrace{\text{softmax}\left(QK^{\top}\right)}_{\text{attention map}} V,$$



MENS MANUS AND MACHINA

SMU SINGAPORE MANAGEMENT UNIVERSITY

School of Computing and Information Systems

Study 2 : ViTs Comparison

Challenges with knowledge transform from one transformer model to different model

- Differences in the **number of layers** between models
- Variations in the **number of attention heads per layer**
- Differences in the **dimensionality (number of features) per token**
- Use of **different tokenizers** (particularly relevant for text-based transformers)

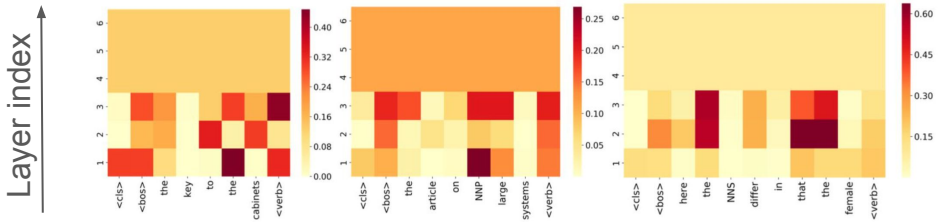
To interpret the results, we plotted the propagation of attention maps across the network.

Model	Number of Layers	Number of heads per layers	Number of features per token	Number of channels for a head to process
ViT-Base/16*	12	12	768	64
ViT-Large/16*	24	16	1024	64
ViT-Huge/14	32	16	1280	80

*These models were analysed in our study

Study 2 : ViTs Comparison

- In self-attention, the information get mixed and combined from different tokens.
- But as you move towards the deeper layers, the information tends to get extremely mixed.



From Fig.2 in Abnar et al. showing attention between the <CLS> token and each input token for each layer in a six layer Transformer encoder. Notice that for the deeper layers the attention signal disappears.

In the higher (final) layers, the <CLS> token tends to attend to all other tokens in a relatively uniform manner.

But if you multiply each attention map from the previous layer with the next layer's attention map, <CLS> token attending to the other tokens in a uniform manner will not happen in the deep layers.

This forms the basis for Attention Rollout.

$$\tilde{A}(l_i) = \begin{cases} A(l_i)\tilde{A}(l_{i-1}) & \text{if } i > j \\ A(l_i) & \text{if } i = j \end{cases}$$

$\tilde{A}(l_i)$ - Attention rollout

j = 0 (first layer attention map)

Abnar, S., & Zuidema, W. (2020). *Quantifying Attention Flow in Transformers*. arXiv:2005.00928

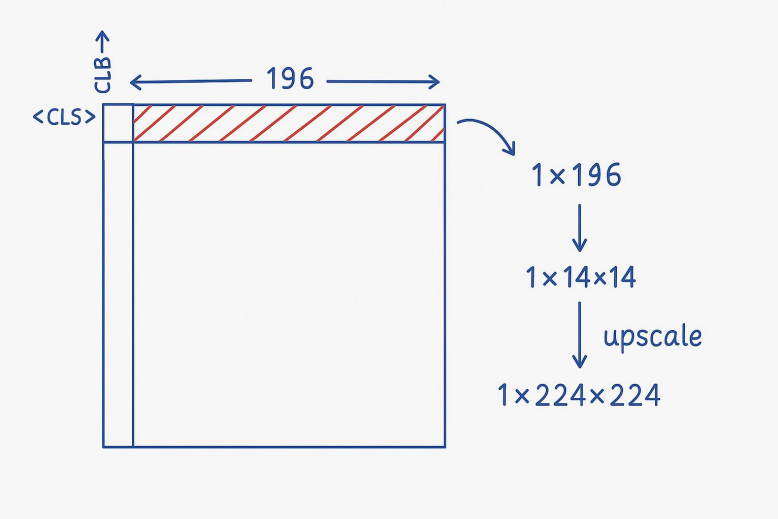
MENS MANUS AND MACHINA

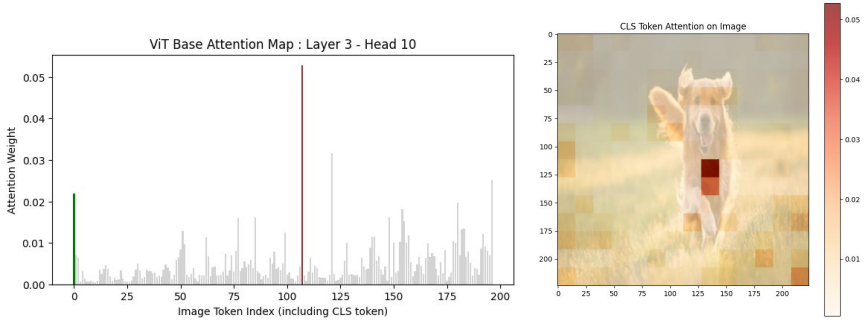
SMU SINGAPORE MANAGEMENT UNIVERSITY

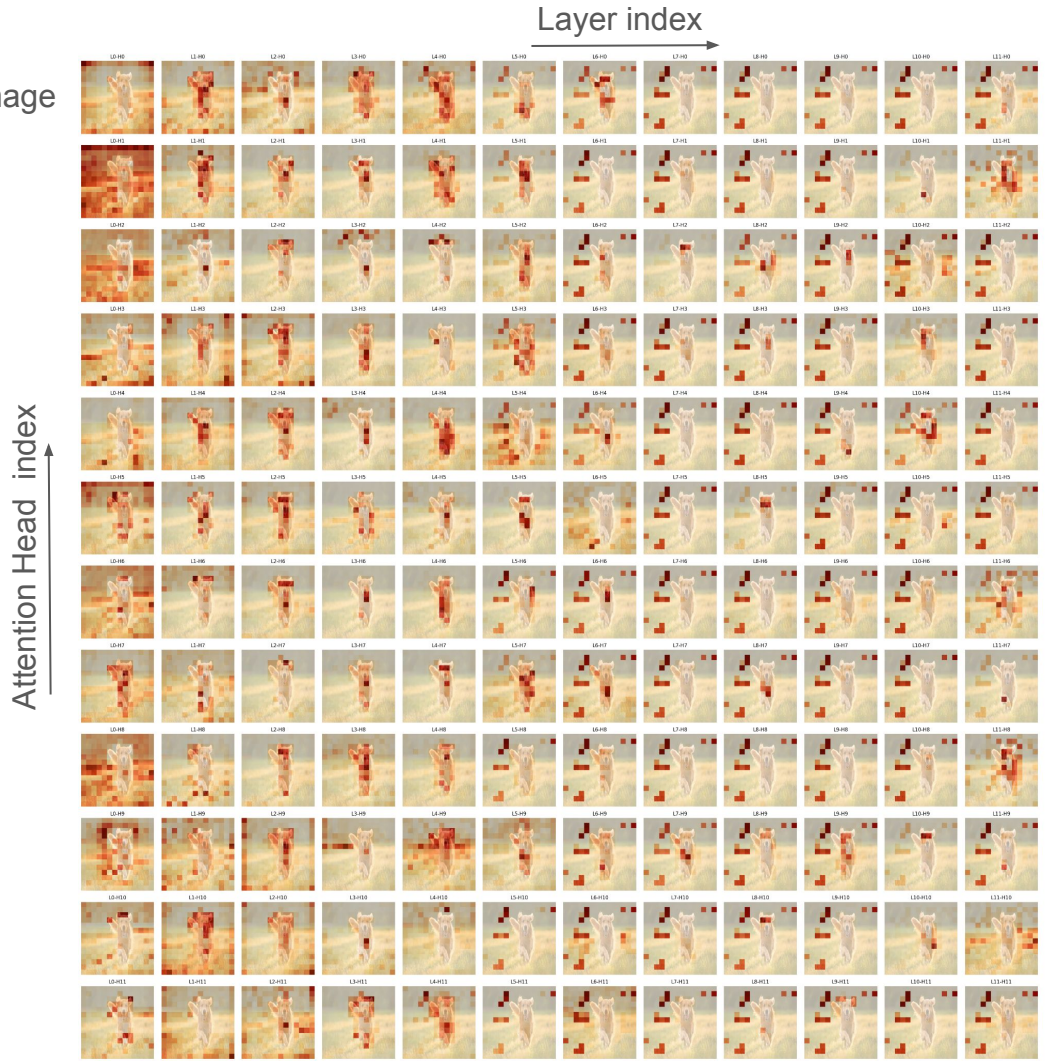
School of Computing and Information Systems

Study 2 : ViTs Comparison

Visualizing how <CLS> token attends to other parts of the image (ViT-B)







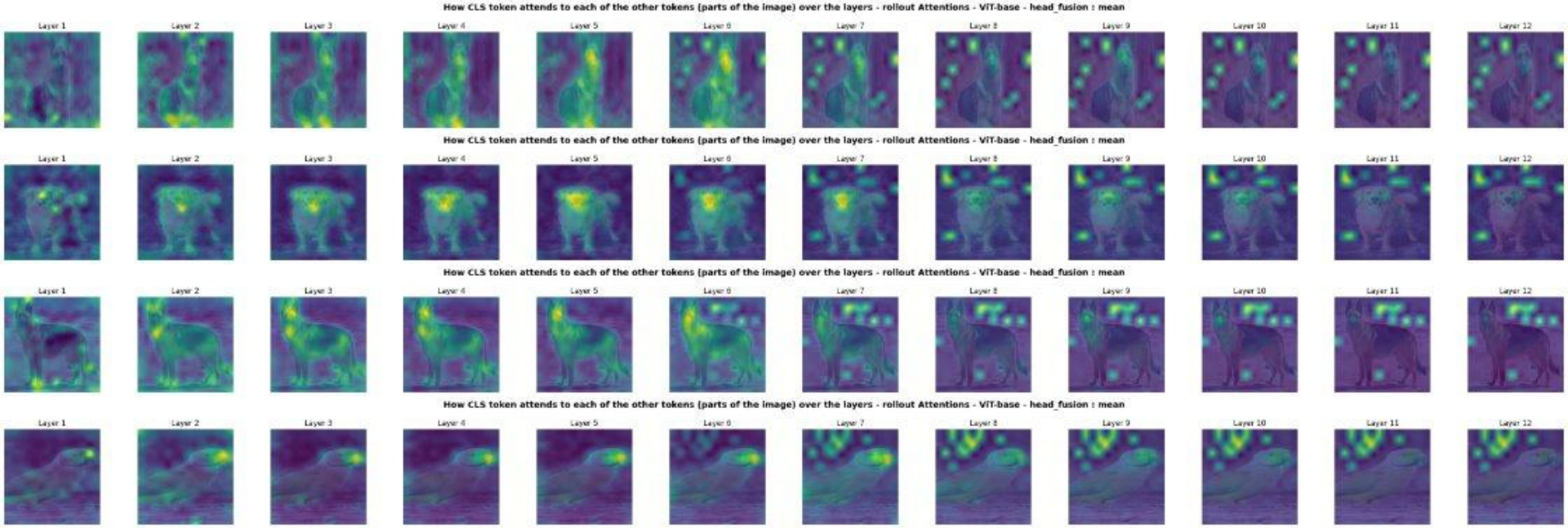
Study 2 : ViTs Comparison (ViT-Base)

German Shepherd 1

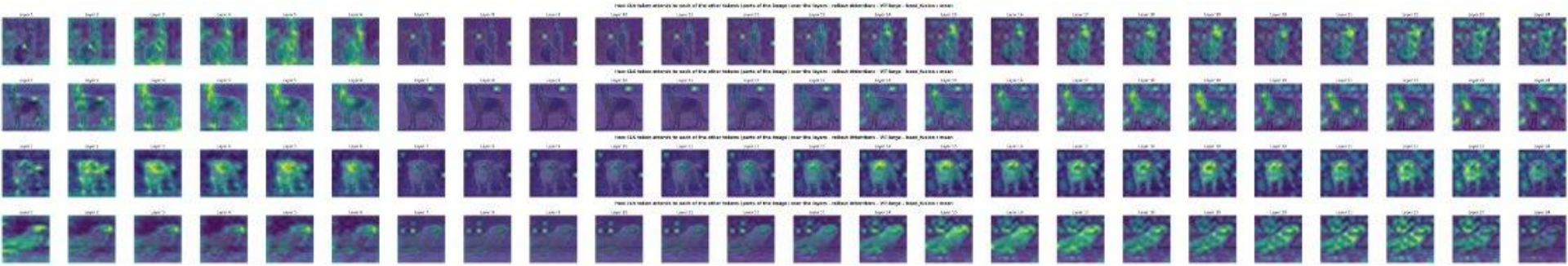
German Shepherd 2

Golden retriever

Macaw



Study 2 : ViTs Comparison (ViT-Large)



The key reasons why the `<CLS>` token attends to more than just the dog's body (including the background) are:

Unlike CNNs, ViTs rely on global context for classification — including the background. This is because the background can provide complementary information (e.g., surroundings or scene type) that supports classification. While CNNs mainly focus on local features, ViTs aggregate information from all over the image.

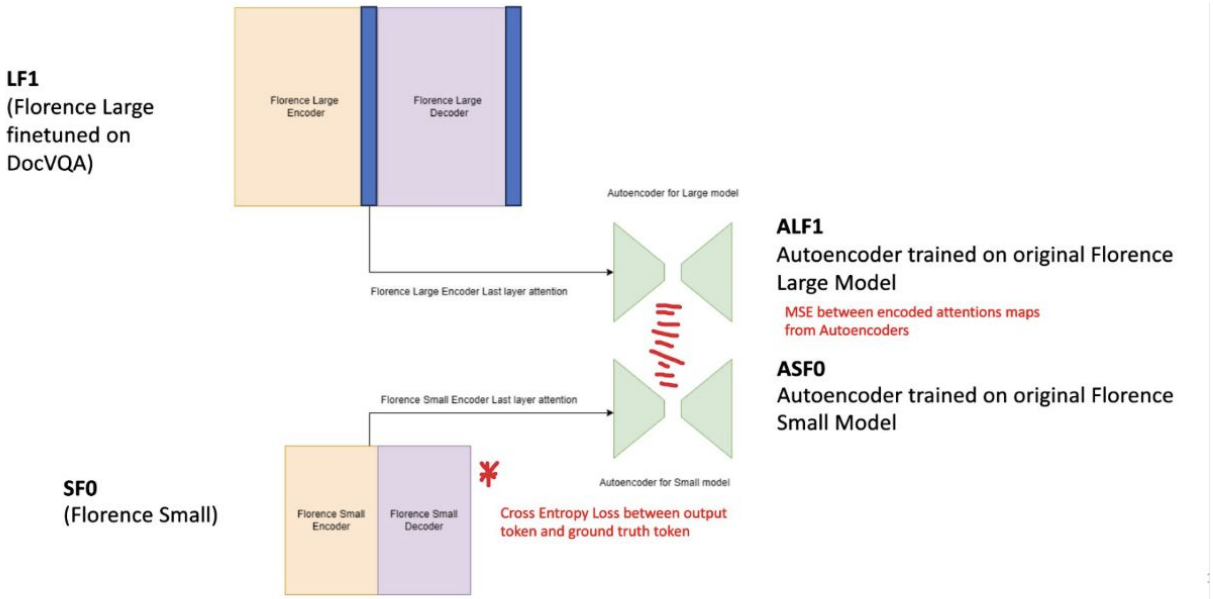
In the early layers: Attention is more diffuse and broad. At this stage, the transformer hasn't yet learned to focus tightly on the object of interest. It attends to many patches as it starts to build hierarchical representations.

In the later layers (e.g., layers 10–12), attention becomes more focused on key parts of the dog, but some dispersion remains. This is due to head fusion and the classifier's need for rich contextual information.

If we compare attention propagation, in two models it is apparent that there's some similarity between patterns.Therefore our hypothesis still holds.

Study 3 : Attention Distillation Approach

Getting inspiration from Attention Transfer paper, we created this pipeline to do distill attention from different architectures for VLMs.



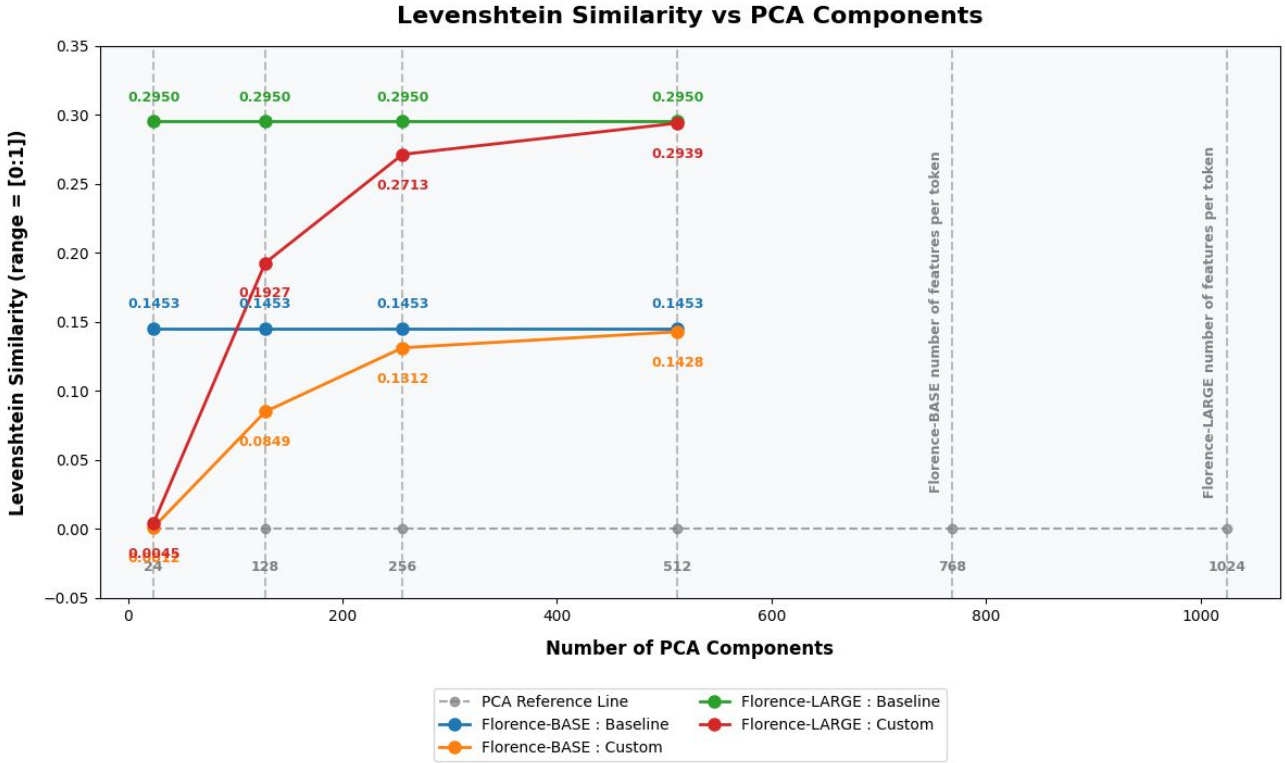
- $Loss = loss_{original} + \alpha * MSE(EncodedAttention_{large}, EncodedAttention_{small})$

Study 3 : Attention Distillation Approach

Model Name	Description	Levenshtein Similarity Score
SF0	Original Florence2 Base model (0.22 Million parameters)	0.1642
SF1	Original Florence2 Base model is fintetuned on DocVQA model; **Loss = Original Loss** The loss is Cross Entropy Loss is between Logits of the predicted output and the vectorised label	0.3727 (after 10 epochs)
SF2	**Loss = Alpha* Autoencoder Distillation + Original Loss ; Alpha= 1** There are two autoencoders trained to reconstruct attention maps from the finetuned Florence Large Model and not finetuned Florence Small Model. MSE Loss is taken from the Encoded attention maps from Autoencoders for Small and Large Florence Models	0.3633 (after 10 epochs)
SF3	Autoencoder Loss (SF3) Here Only the autoencoder loss is used.	0.1618 (after 10 epochs)
SF4	Traditional Distillation - Attention Distillation (SF4) ; Using the final outputs of the Florence large model and the Florence small model.	Not checked
SF5	Attention Rollout; To train the Florence models with autoencoders we use Attention Rollout instead of Raw attentions **Loss = Alpha* Autoencoder + Original Loss ; Alpha= 0.1**	0.3710 (after 10 epochs)
SF6	**Loss = Alpha* Autoencoder + Original Loss ; Alpha= 0.1**	0.3703 (after 9 epochs)
SF7	Here uses a Variational Autoencoder	Not checked

However, according to the results above, (SF1) fine-tuning the small model directly on DocVQA resulted in the highest Levenshtein similarity score.

Study 4 : Reducing the number of features(dims) per token



MENS MANUS AND MACHINA

SMU SINGAPORE MANAGEMENT UNIVERSITY

School of Computing and Information Systems

Thank you!