# MAP583 - Project Presentation

## 21. Explainability for Vision Transformers

École Polytechnique

Students: N. Lopes, M. Thiaw, Z. Oumzil, F. Reynal

March 18, 2024

# Outline

# Outline

# Motivation

- NLP to Vision - Understand Transformers more in-depth.
- Learn tools to better understand black-box deep learning models.
- Explore Graph-based methods in Machine Learning.

# Outline

# ViT: Vision Transformers

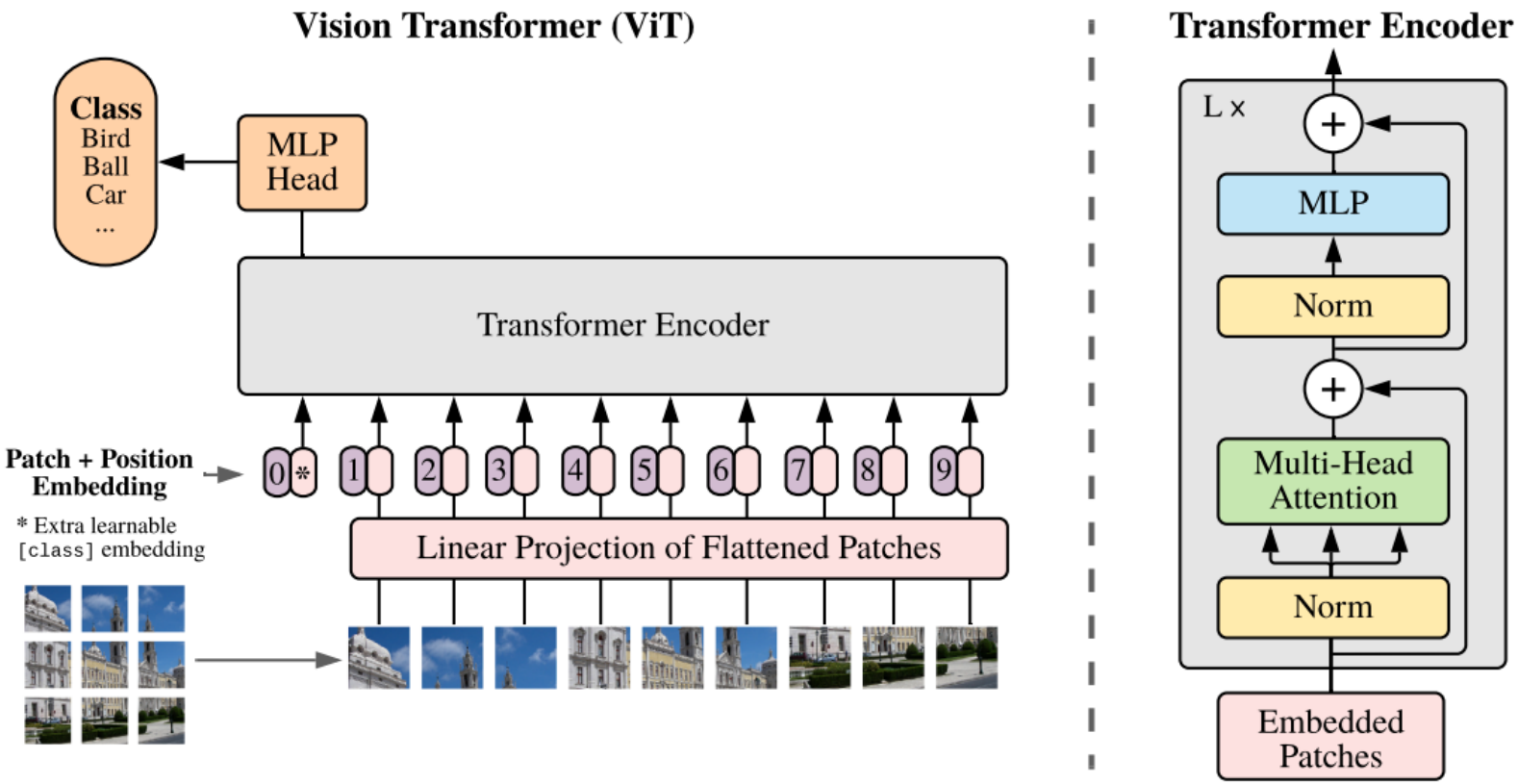



Figure: Vision Transformer Architecture [1]

# Attention Rollout

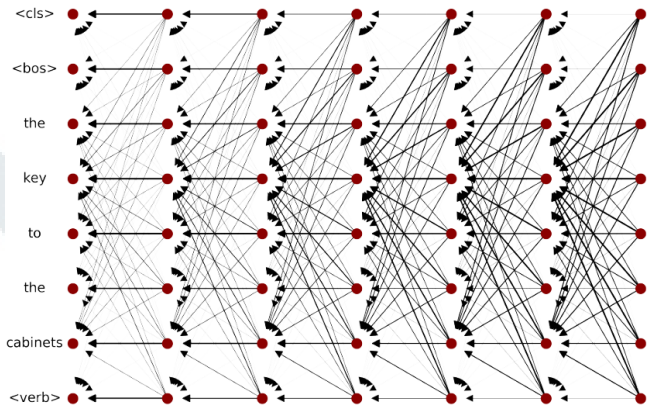



Figure: Attention Graph architecture in NLP [2]

1. Assumption : the if $u$ and $v$ are two nodes of our graph, then the amount of information transported from $u$ to $v$ is given by :

$$\sum_{\substack{P\text{path} \\ \text{from } u \text{ to } v}} \prod_{e \in P} w_e$$

2. Consequence : $\tilde{A} = \tilde{A}_1 \dots \tilde{A}_{L-1} \tilde{A}_L$ where $\tilde{A}_k = \frac{1}{2}(I + A_k)$

# Gradient Attention Rollout

$$\tilde{A} = \prod_{l \text{ layer}} \frac{1}{2}(I + (A_l \odot Grad_l)^+)$$

1. Method based on the gradient of the loss with respect to the input of each layer, as computed through backpropagation.
2. Same idea as the Attention Rollout
3. Difference : Weight all the attentions by the target class gradient.
4. On the implementation : loss = output[category id]

# Attention Flow - Group's Implementation

- Follow original paper from the repository. [2]
- Hidden embedding tokens as sources and input tokens as targets.
- Compute iteratively the **maximum flow** for each graph configuration.

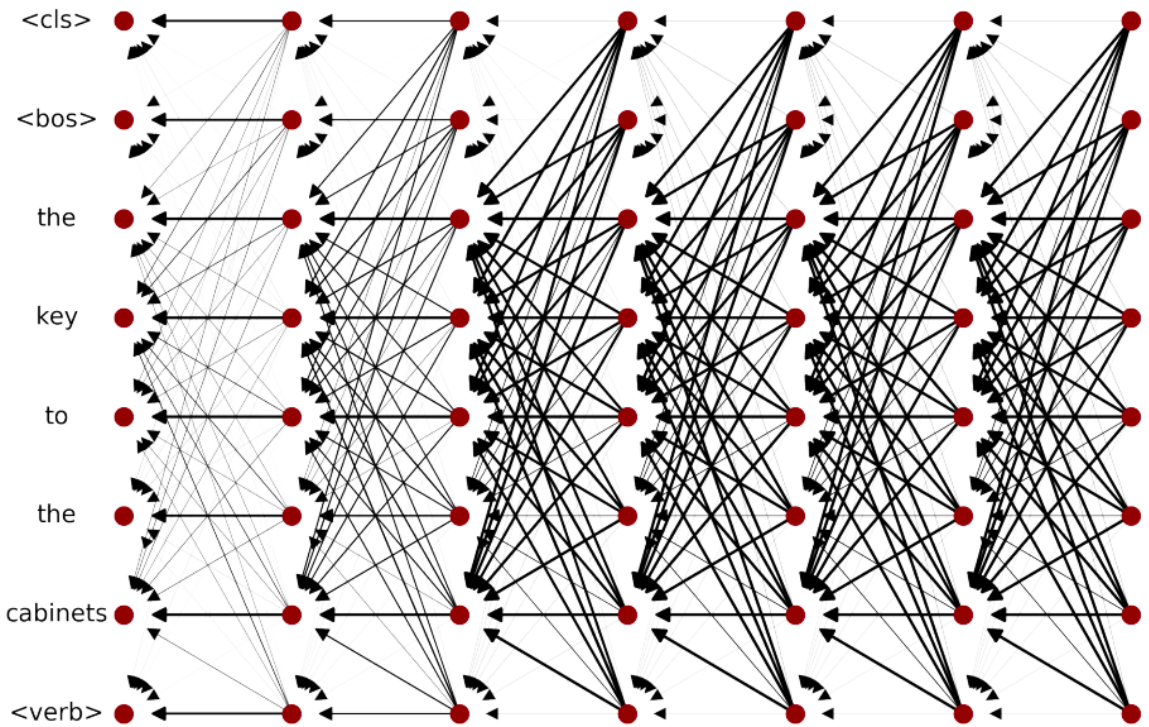



Figure: Attention flow architecture in NLP [2]

# Attention Flow - Other paper ideas

- Attempted more recent interpretability methods in Attention Flow. [3]
- Input tokens as sources, final layer embeddings as targets.
- Compute iteratively the **maximum flow** for each graph configuration.

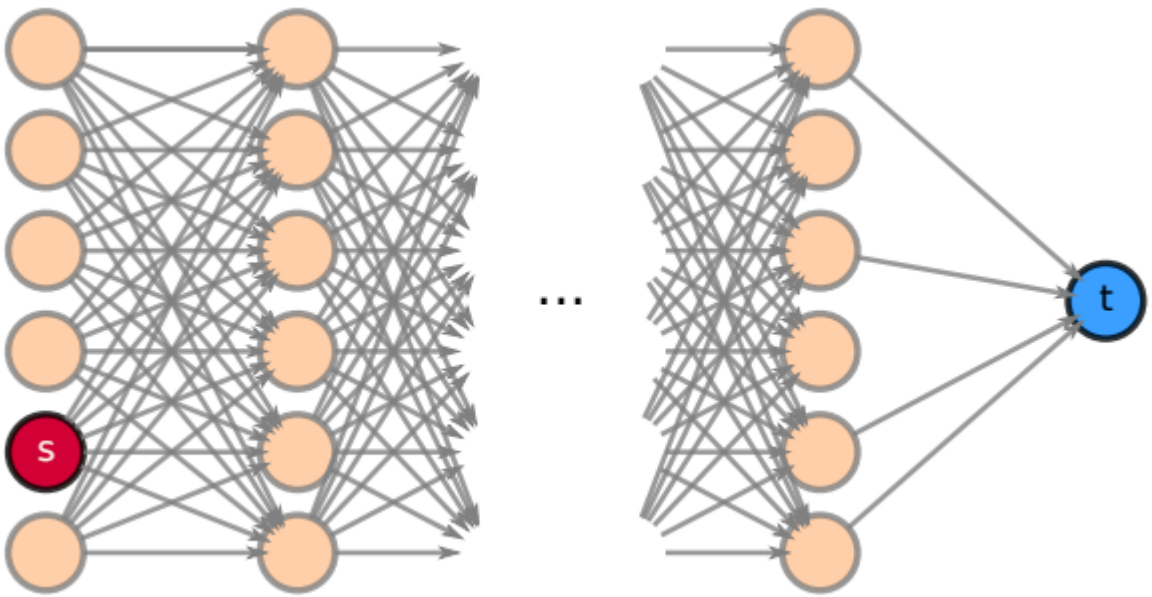



Figure: Attention flow alternative. In classification tasks, only CLS token as target. [3]

# Illustration of methods

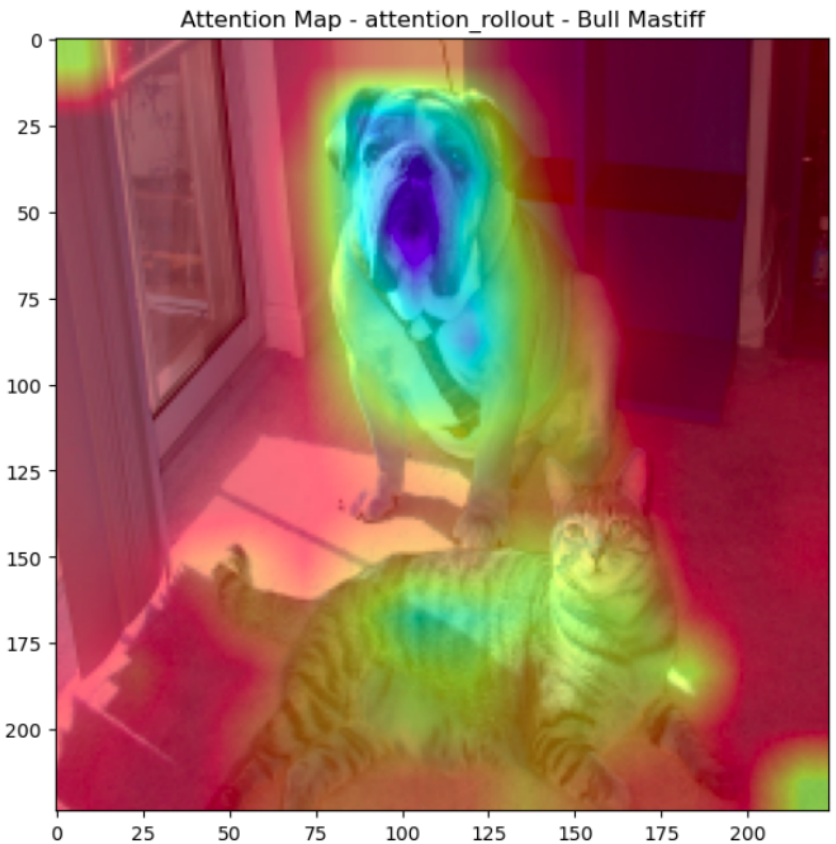



Figure: Attention Rollout

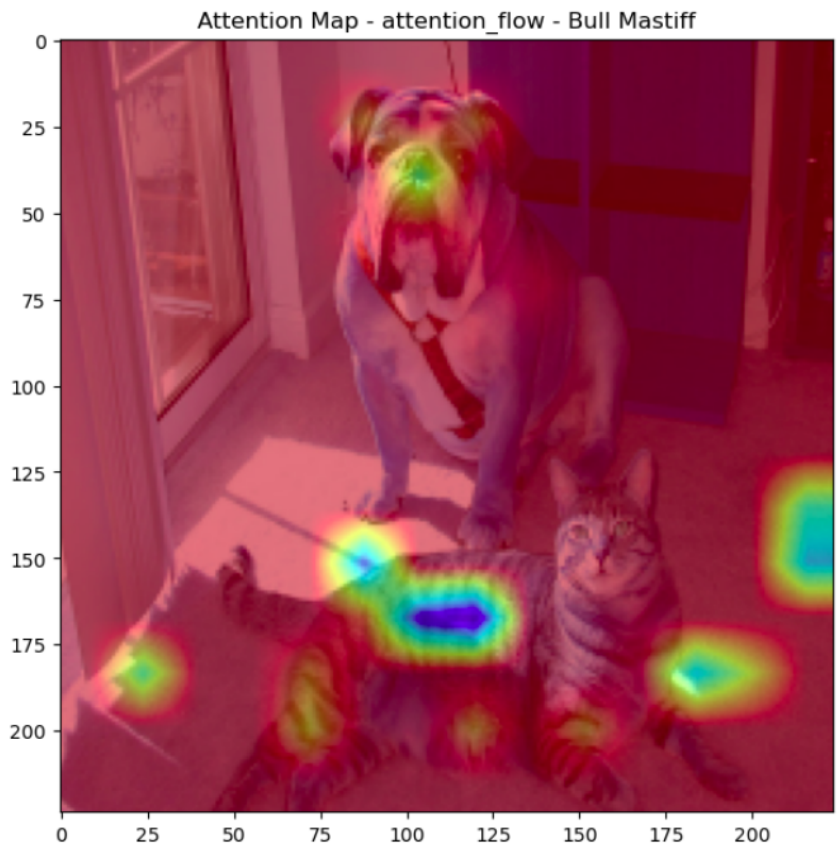



Figure: Attention Flow

# Illustration of methods

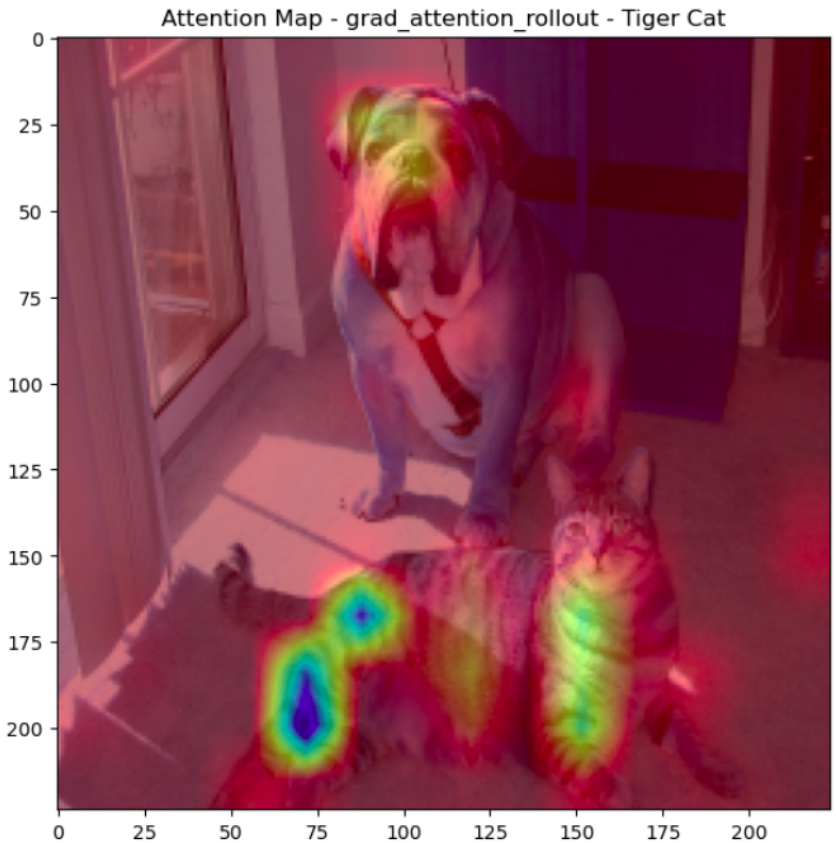



Figure: Gradient Rollout - Cat

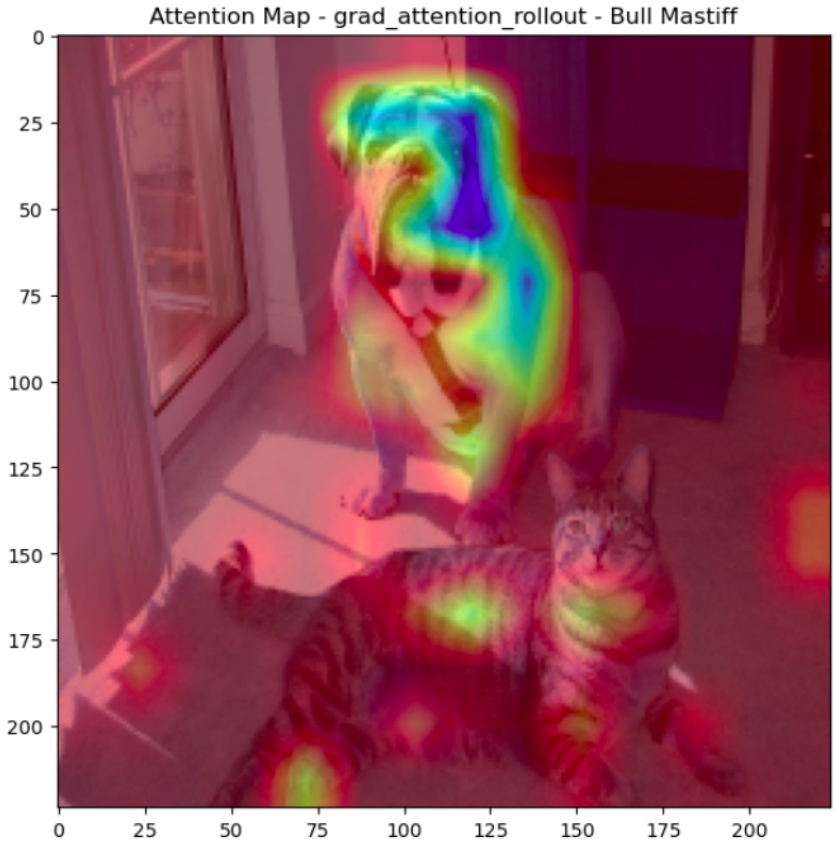



Figure: Gradient Rollout - Dog

# Outline

# Motivation and Methodology

- We follow a similar procedure of **blank-out** as described in Quantifying attention Flow in Transformers for NLP models. [2]
- For each image, our algorithms return the importance of each input token to model's prediction.
- Compare to drop in the probability of predicted class when image patch is replaced by a black square.

Figure: Illustration of blank-out step

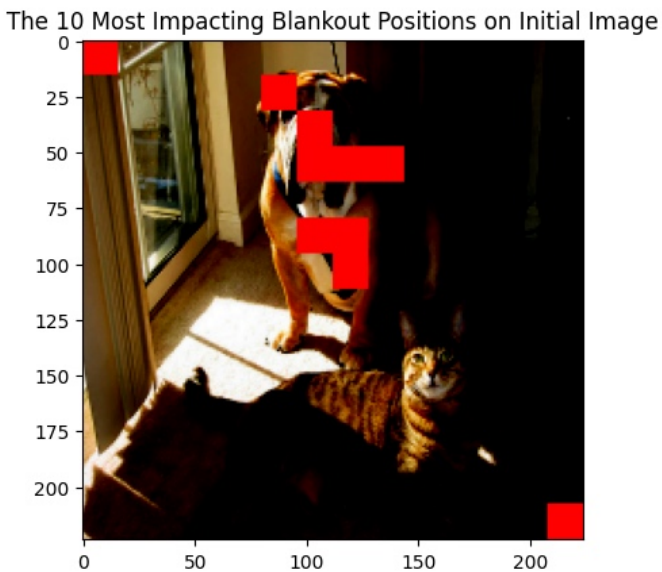



Figure: Illustration of blank-out

# Motivation and Methodology

- Two metrics comparison: Spearman's rank correlation and percentage of shared top 10% features per image.
- We also compare our Attention Flow to other methods.
- Images used: 244 validation images of ImageNet competition.

# Results

- Correlation Rollout and Grad Rollout: $0.47 \pm 0.22$
- Correlation Flow and Rollout: $0.23 \pm 0.17$
- Correlation Flow and Grad Rollout: $0.21 \pm 0.20$

| Rollout | Grad Rollout | Flow |
|---|---|---|
| $0.11 \pm 0.24$ | $0.11 \pm 0.21$ | $-0.01 \pm 0.13$ |

Table: Correlation to blank-out

- Possible limitation: 196 input tokens compared to roughly 10 in NLP.

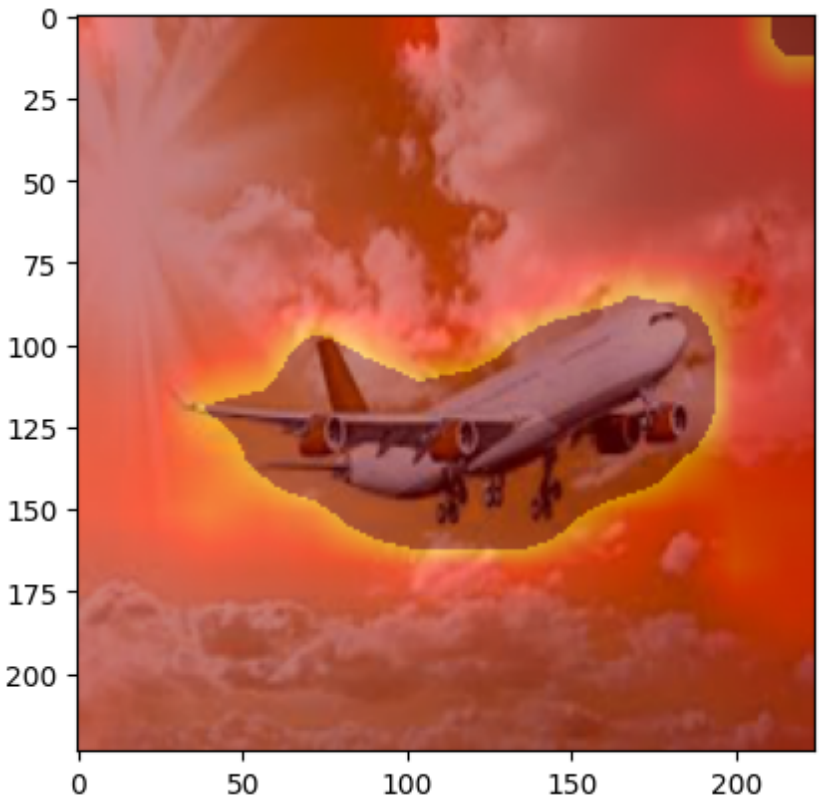
| Rollout | Grad Rollout | Flow |
|---|---|---|
| $0.21 \pm 0.15$ | $0.24 \pm 0.14$ | $0.12 \pm 0.08$ |

Table: Percentage of shared most important tokens

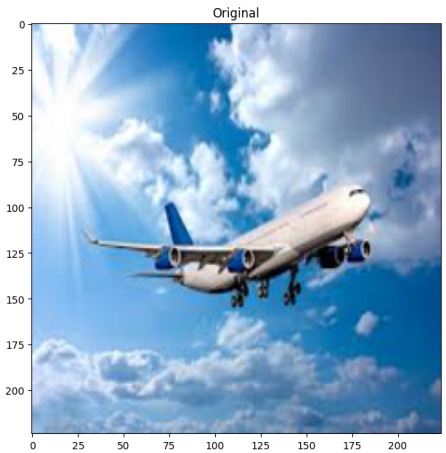
# Outline

# Attacks - Methodology

- Determine the class of the image and choose a target class.
- Get a mask of given the attention area, depending on the strategy (attention rollout, gradient attention rollout, attention flow).
- Attack based on Fast Gradient Sign Method (**FGSM**) to attack inside or outside the attention area.



# Attacks - Result



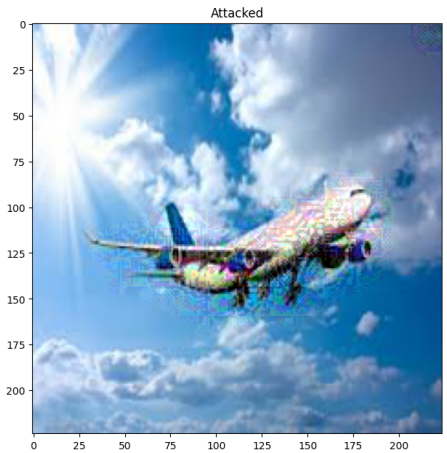
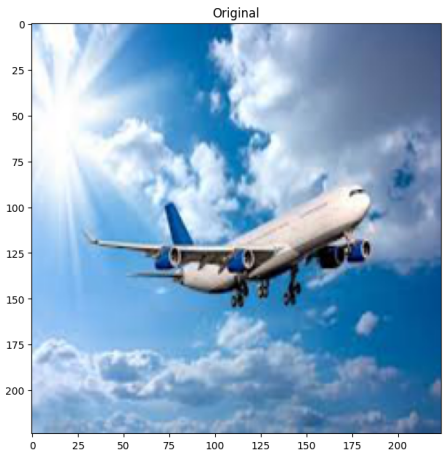




Figure: Inside after 1 attack



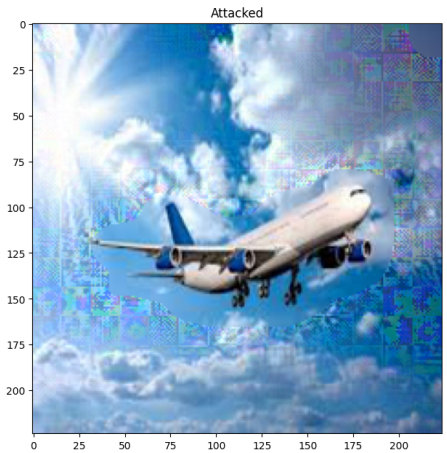




Figure: Outside after 6 attacks

# Attacks - Result

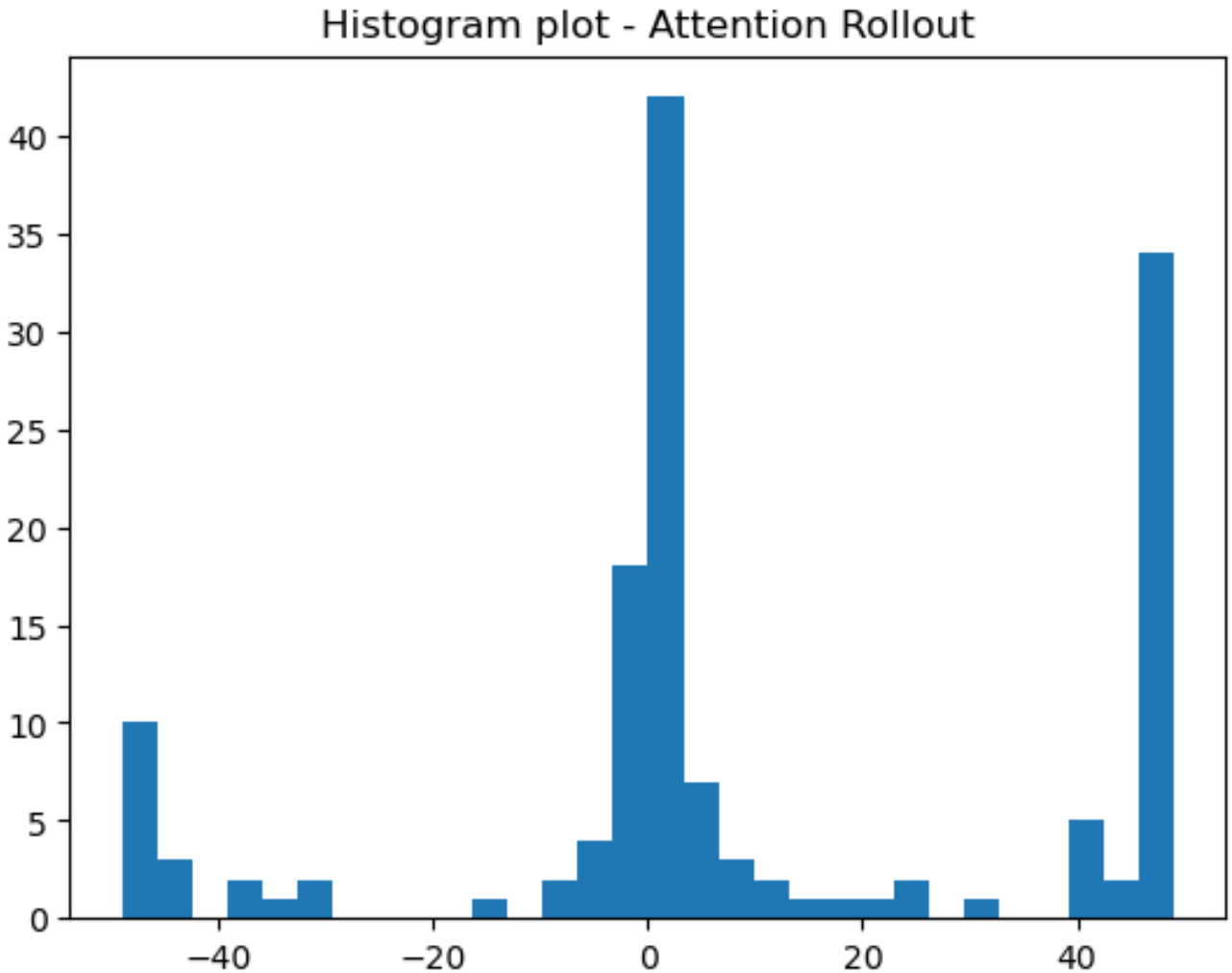



Figure: Difference of number of attacks

# Outline

# Conclusion

- Understood three new methods for Transformers architectures and successfully implemented Attention Flow from first paper.
- Methods in ViT seem to under-perform compared to NLP model in the paper, possibly due to size of input token layer.
- Attacks: Promising results, attacks show that methods manage to capture regions looked by the models.

# References I

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An image is worth 16x16 words: transformers for image recognition at scale", arXiv preprint arXiv:2010.11929 (2020).

[2] N. Metzger et al., "Quantifying attention flow in transformers", arXiv preprint arXiv:2005.00928 (2020).

[3] N. Metzger, C. Hahn, J. Siber, F. Schmitt, and B. Finkbeiner, "Attention flows for general transformers", arXiv preprint arXiv:2205.15389 (2022).

# Appendix: Raw Attention


Figure: Raw Attention Visualization

# Appendix: Raw Attention

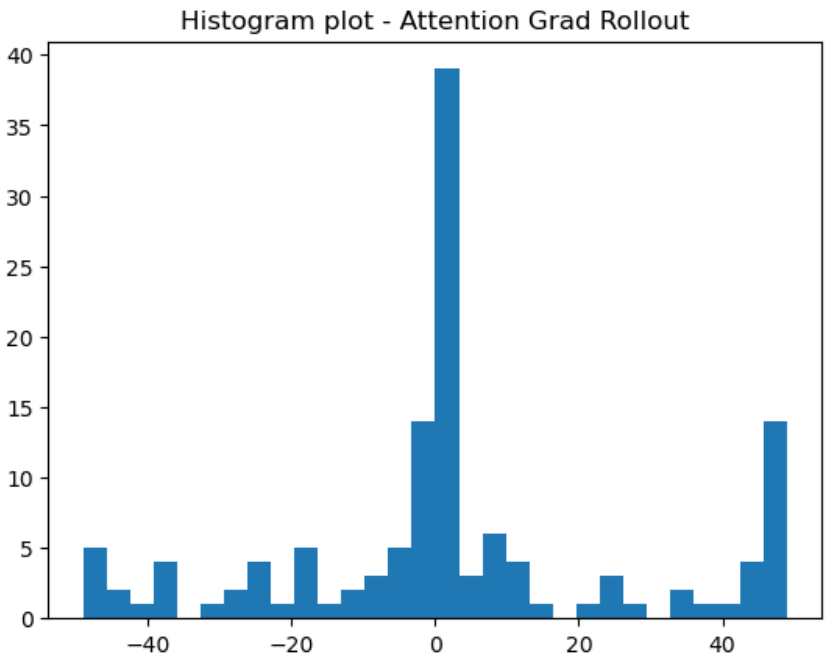



Figure: Gradient Attention Rollout

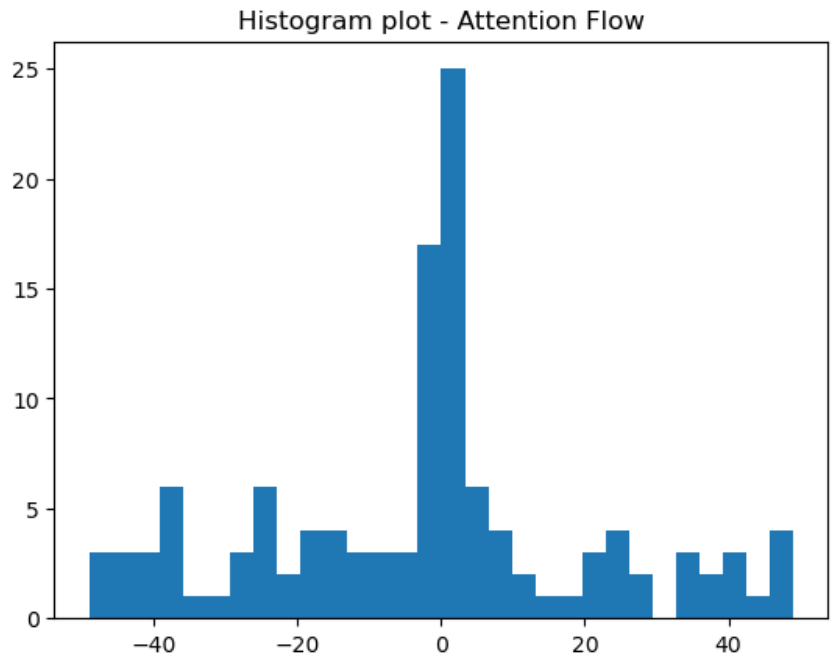



Figure: Attention Flow