Attention

Transformer

- All the sentence processed at the same time
- What is an Attention Map?
  - a grid that tells how much **focus** a model places on different parts of the input text when processing a specific token
- Example: "flag" → "American"
  - **T**he brightest square (highlighted) shows it's paying extremely high attention to the token **“American”**
  - **Attention allows a model to dynamically create relationships between tokens**

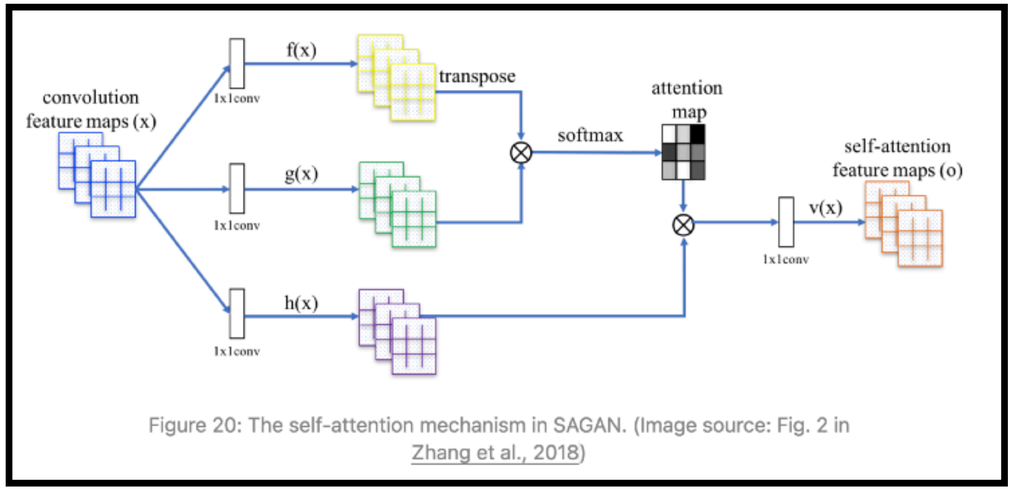
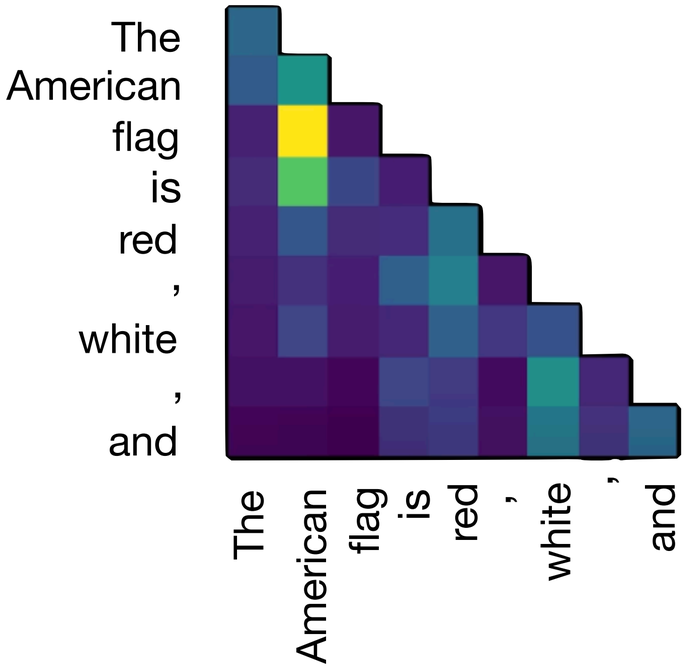


Figure 20: The self-attention mechanism in SAGAN. (Image source: Fig. 2 in Zhang et al., 2018)



Bright squares = Strong connection.
Dark squares = Weak connection.

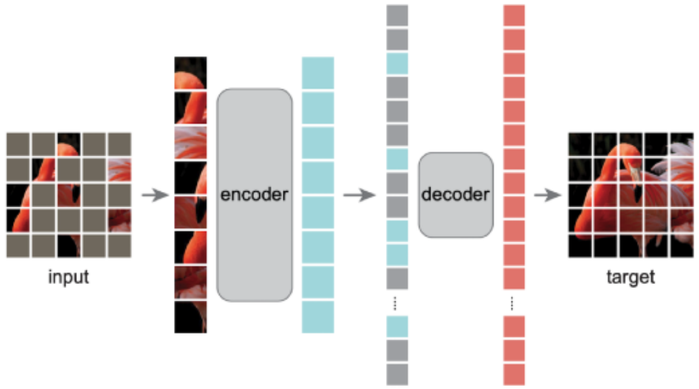
Design: *built upon cutting-edge ML methods such as MAE & Vision Transformer*

Masked Auto-encoders

Learning with fill in the blanks

The "How":

- *Mask:* A large portion of the input (e.g., 75% of patches)
- *Encode:* A deep Encoder processes *only the visible patches*.
- Reconstruct: A Decoder guess the missing patches.



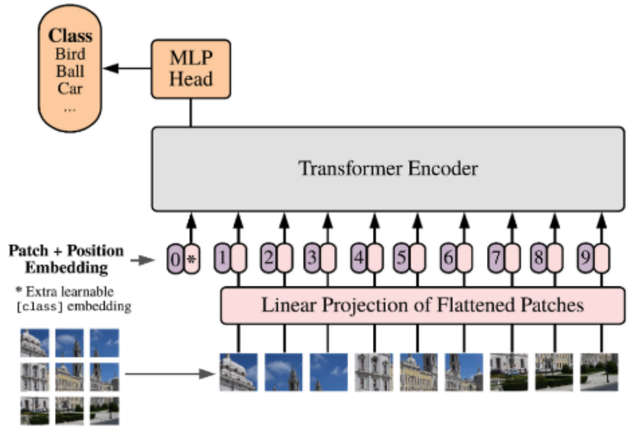
- **The Goal:** force Encoder to learn a rich representation of the data, not just surface-level details.

Vision Transformer

Self-attention to access global features

The "How":

- *Patchify*: An image is broken down into a sequence of patches.
- *Embed*: Each patch is converted to feature vector + positional info
- *Transformer Encoder*: self-attention to model the token relation



- **The Goal:** capture long-range dependencies and global context across the entire input.