
PerceiverIO
One model to rule them all
Jörg Simon, 13.Oct.2021

About me

- PhD on using DeepLearning to detect Human Factors from BioSignals
- Prof. Eduardo Veas and Herbert Danzinger

Outline

PerceiverIO: One model to rule them all

- Perceiving & Multimodality
- Attention
- Perceiver
- Quick what is PerceiverIO
- War of Frameworks: JAX, TF, PyTorch....
- Some small code examples (in PyTorch)

Perceiving & Multimodality

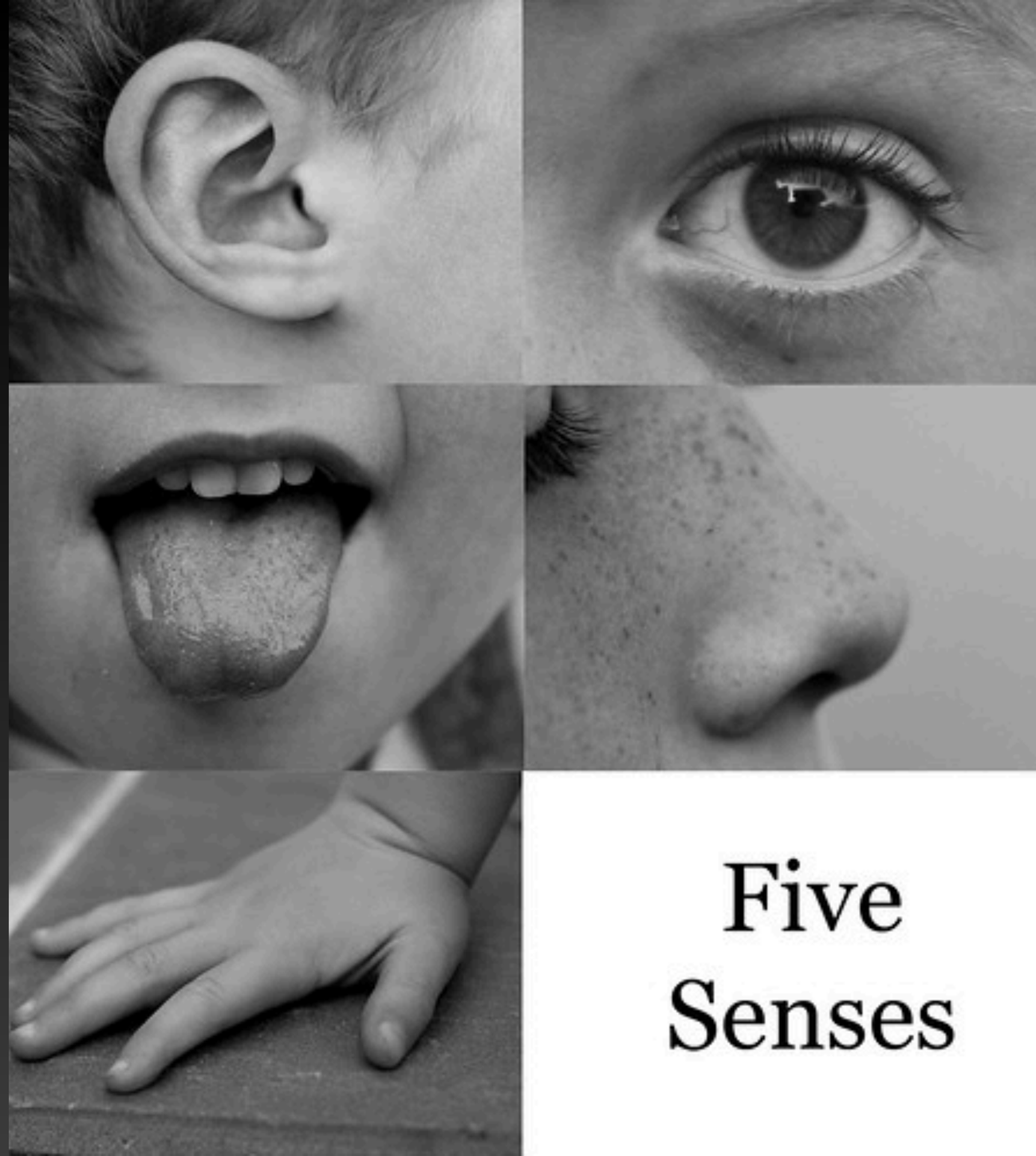


Image from https://quizlet.com/539807260/chapter-4-sensing-and-perceiving-our-world-flash-cards/

Perceiving & Multimodality

- Biological Systems Inherently Multi Modal
- DeepLearning:
  - CNN for image like Data
  - LSTMs for Time Series
  - Hourglas Networks
  - Strong Adaptation of Architecture towards Domain
  - Especially Input and Output very specific

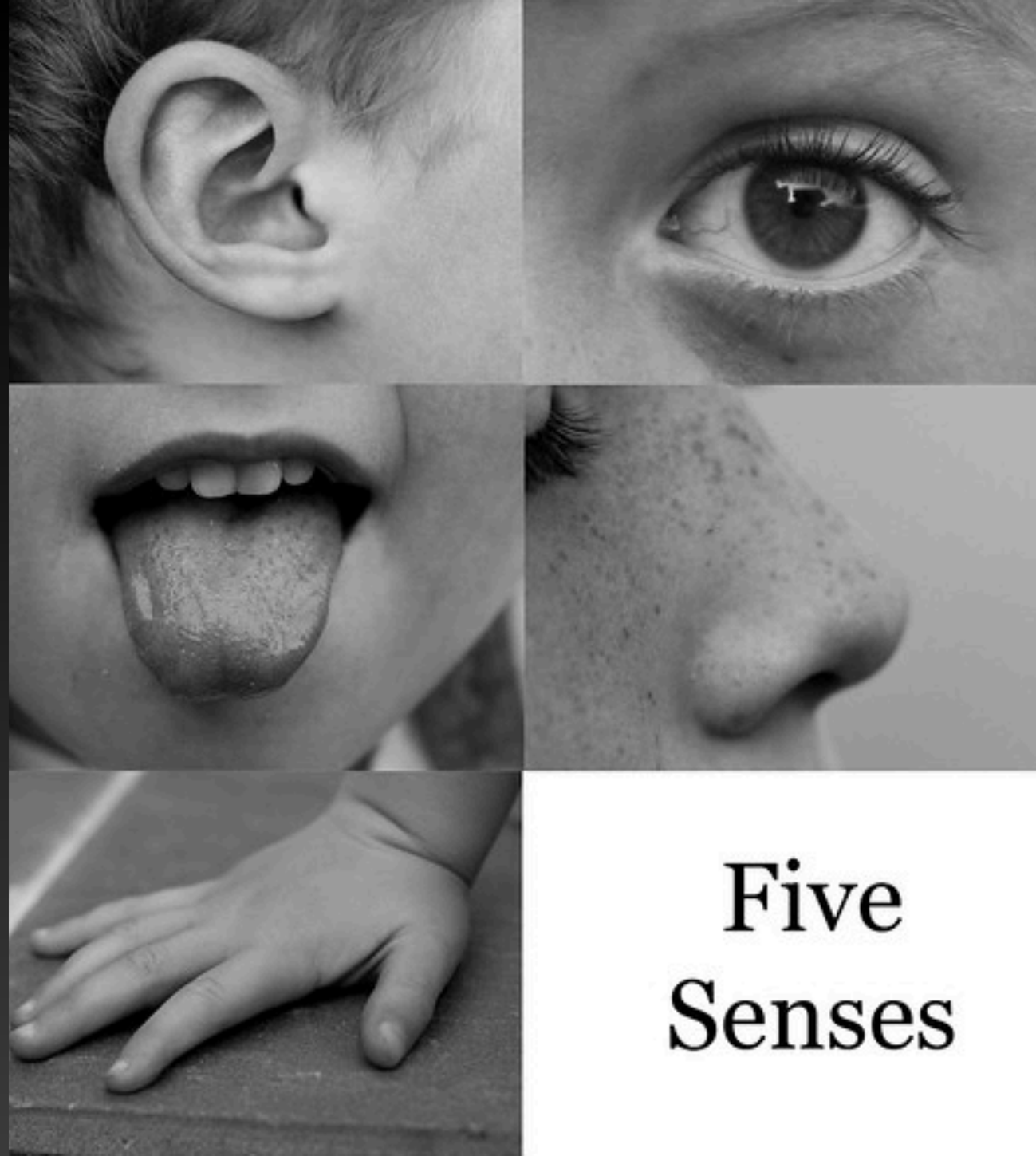


Image from https://quizlet.com/539807260/chapter-4-sensing-and-perceiving-our-world-flash-cards/

Perceiving & Multimodality

- Biological Systems Inherently Multi Modal
- DeepLearning:
  - Modality Specific contradicts the universal algorithm theorem
  - However Brain also has specialised Regions
  - Still it would be awesome to have one good architecture for all modalities and queries

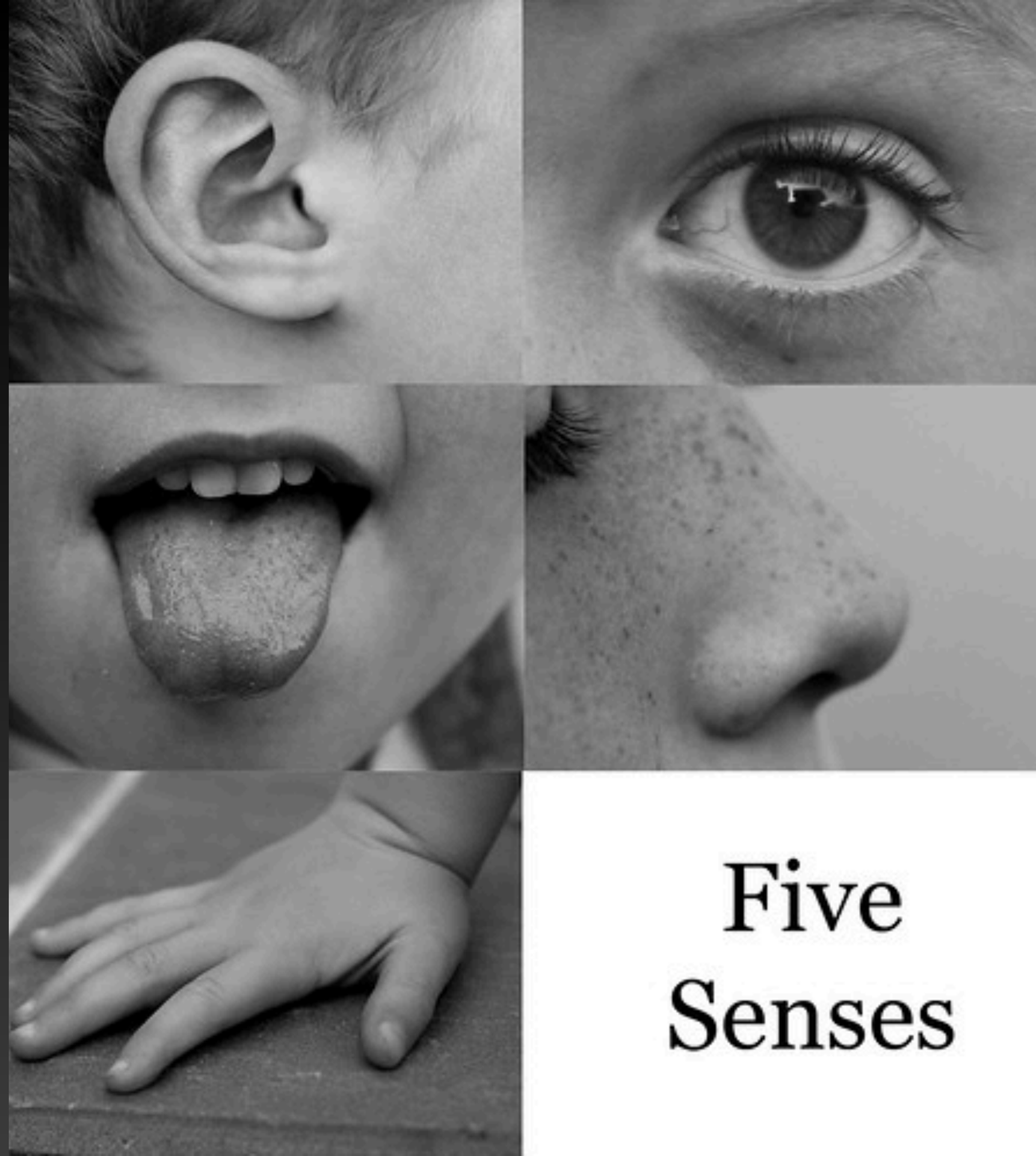
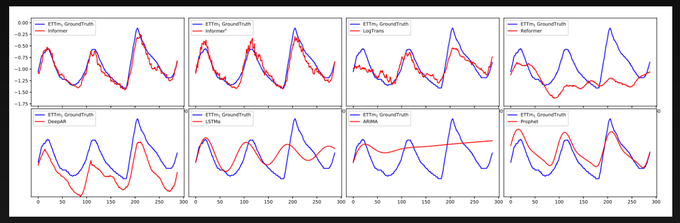
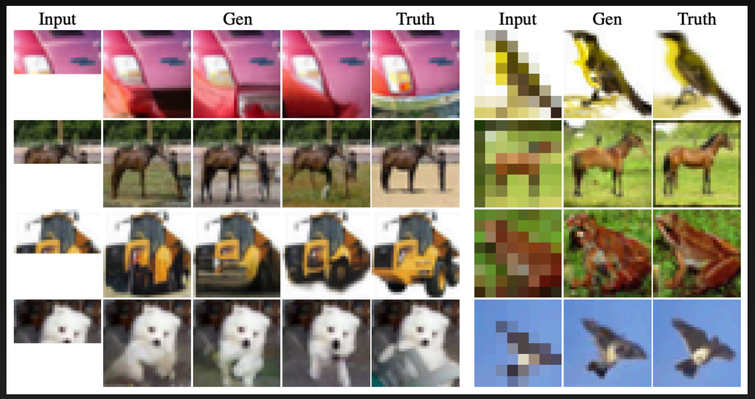


Image from https://quizlet.com/539807260/chapter-4-sensing-and-perceiving-our-world-flash-cards/

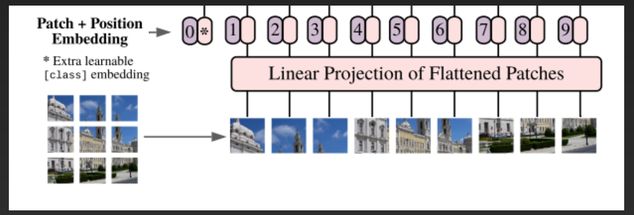
Progress towards Multimodality
in the Transformer World

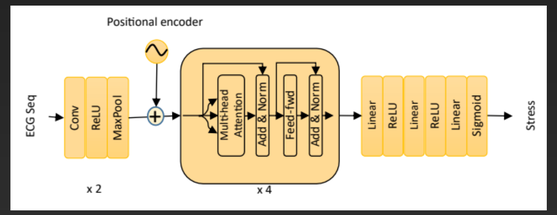
- One Architecture for many things:
  - T5[1]: Language
  - Image Transformer[2] / ViT[3]: Image (&Video)
  - Wave2Vec 2.0[4]: Audio
  - Informer[5]: Long Range Timeseries (Power Grid)
  - ECG/Stress Transformer[6]

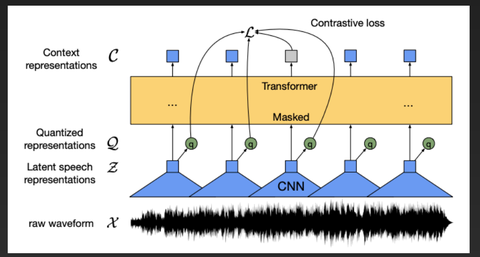




Modality Specific Customisations or Training



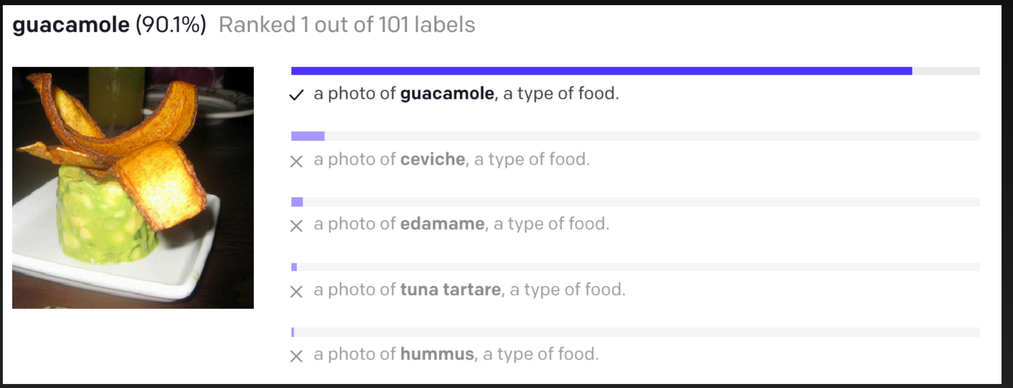




1. Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2020). "*Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*". ArX abs/1910.10683.
2. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N.M., Ku, A., & Tran, D. (2018). "*Image Transformer*". ArXiv, abs/1802.05751.
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". ArXiv, abs/2010.11929.
4. Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). "*wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*". ArXiv, abs/2006.11477.
5. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). "*Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*". AAAI.
6. Behinaein, B., Bhatti, A., Rodenburg, D., Hungler, P.C., & Etemad, A. (2021). "*A Transformer Architecture for Stress Detection from ECG*". 2021 International Symposium on Wearable Computers.

Progress towards Multimodality
in the Transformer World

- DALL-E[1], CLIP[2], UC2[3], MUM (based on T5[4]+MoE[5]), Wu Dao 2.0[6]



1. Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever. (2021) "*Zero-Shot Text-to-Image Generation*", Proceedings of the 38th International Conference on Machine Learning, PMLR 139:8821-8831.
2. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). "*Learning Transferable Visual Models From Natural Language Supervision*". ICML.
3. Mingyang Zhou Luowei Zhou Shuohang Wang Yu Cheng Linjie Li Zhou Yu Jingjing Liu. (2021) "*UC2: Universal Cross-lingual Cross-modal Vision-and-Language Pre-training*" IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2021)
4. Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2020). "*Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*". ArX abs/1910.10683.
5. Fedus, W., Zoph, B., & Shazeer, N.M. (2021). "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity". ArXiv, abs/2101.03961.
6. He, J., Qiu, J., Zeng, A., Yang, Z., Zhai, J., & Tang, J. (2021). "*FastMoE: A Fast Mixture-of-Expert Training System*". ArXiv, abs/2103.13262.

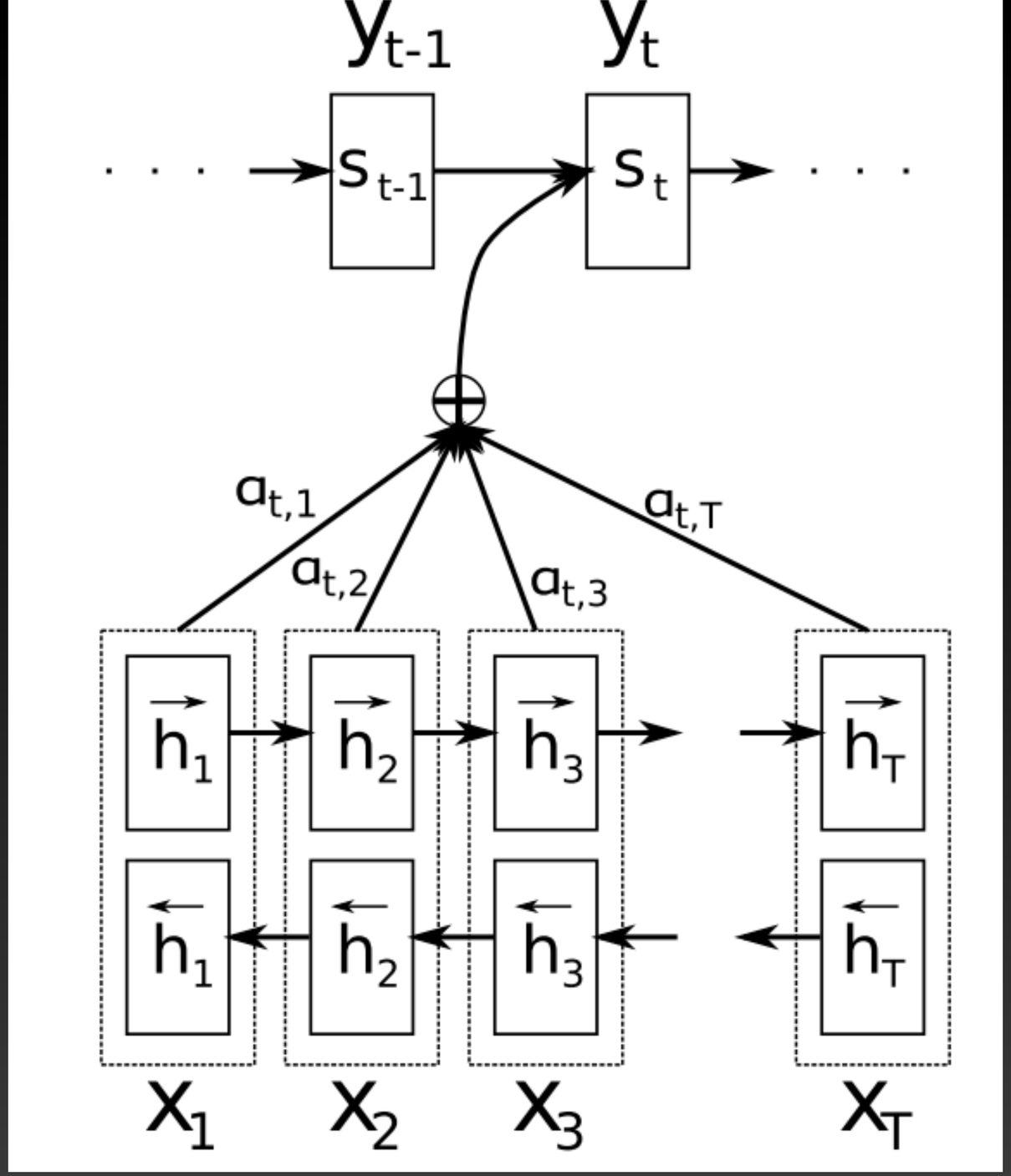
Progress towards Multimodality

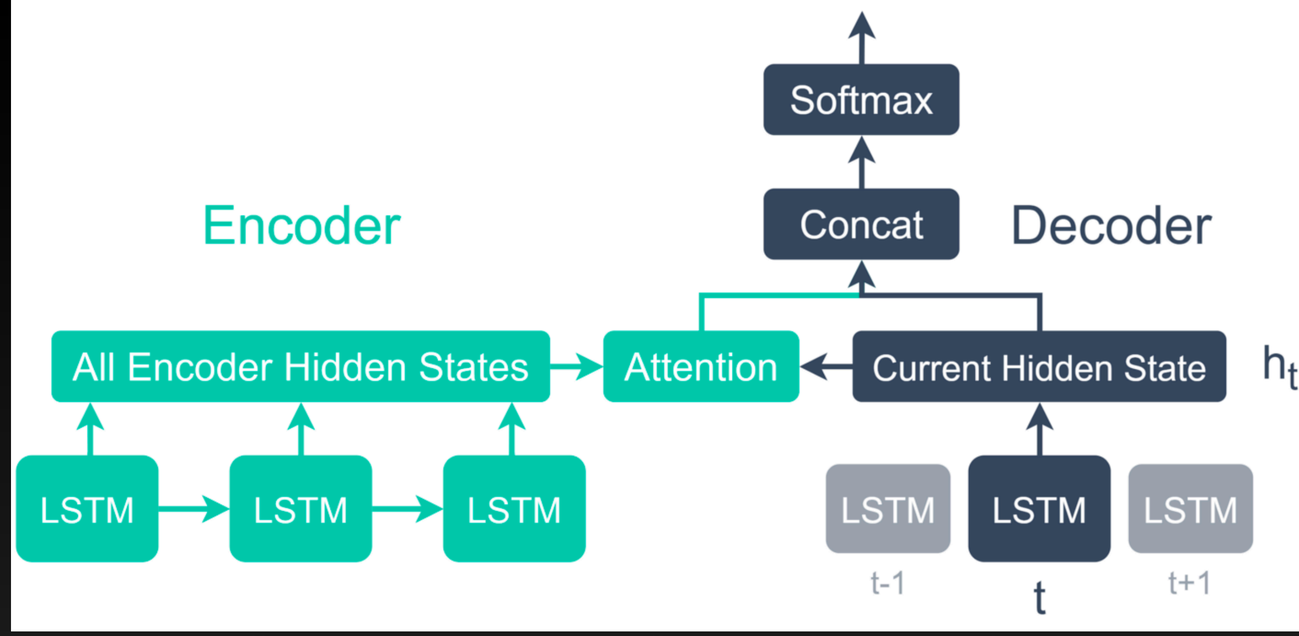
- Other options like CNN per modality and then some top level network or mechanism
- These are all late fusions of Modalities, something the brain most probably does not do
- Majority Voting, Boosting and Bagging and other Ensemble Methods exist

Problem with Transformers for Multimodality

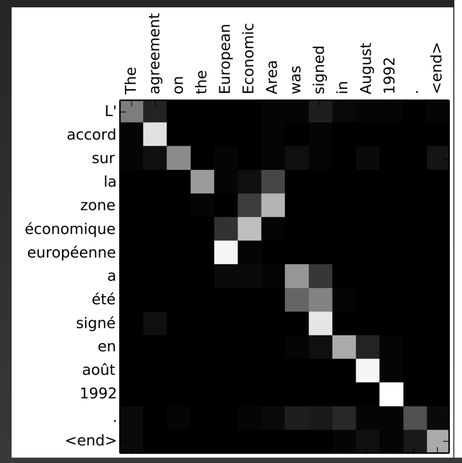
- They either are still tuned towards a modality (like ViT, Wave2Vec...)
- The multimodal examples above have **extreme Hardware** requirements. You basically need an unlimited hardware budget.
- Why: Standart Multi-Head Attention has $O(n^2 \times d)$ time and memory complexity **per layer** with n being the sequence length
- Means that a 512x512 picture (not a video) and a hidden dimension of 512 and 10 layers needs at least **3.52e+14 operations** to classify
- Modern GPUs: 15 Tera Flops per second = **1.5e+13/s**
- You wait about **10s** for **one picture** to be processed on a kick ass GPU

Attention





from: https://towardsdatascience.com/how-transformers-work-6cb4629506df



$$e_{ij} = a(s_{i-1}, h_j) \tag{1}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{2}$$

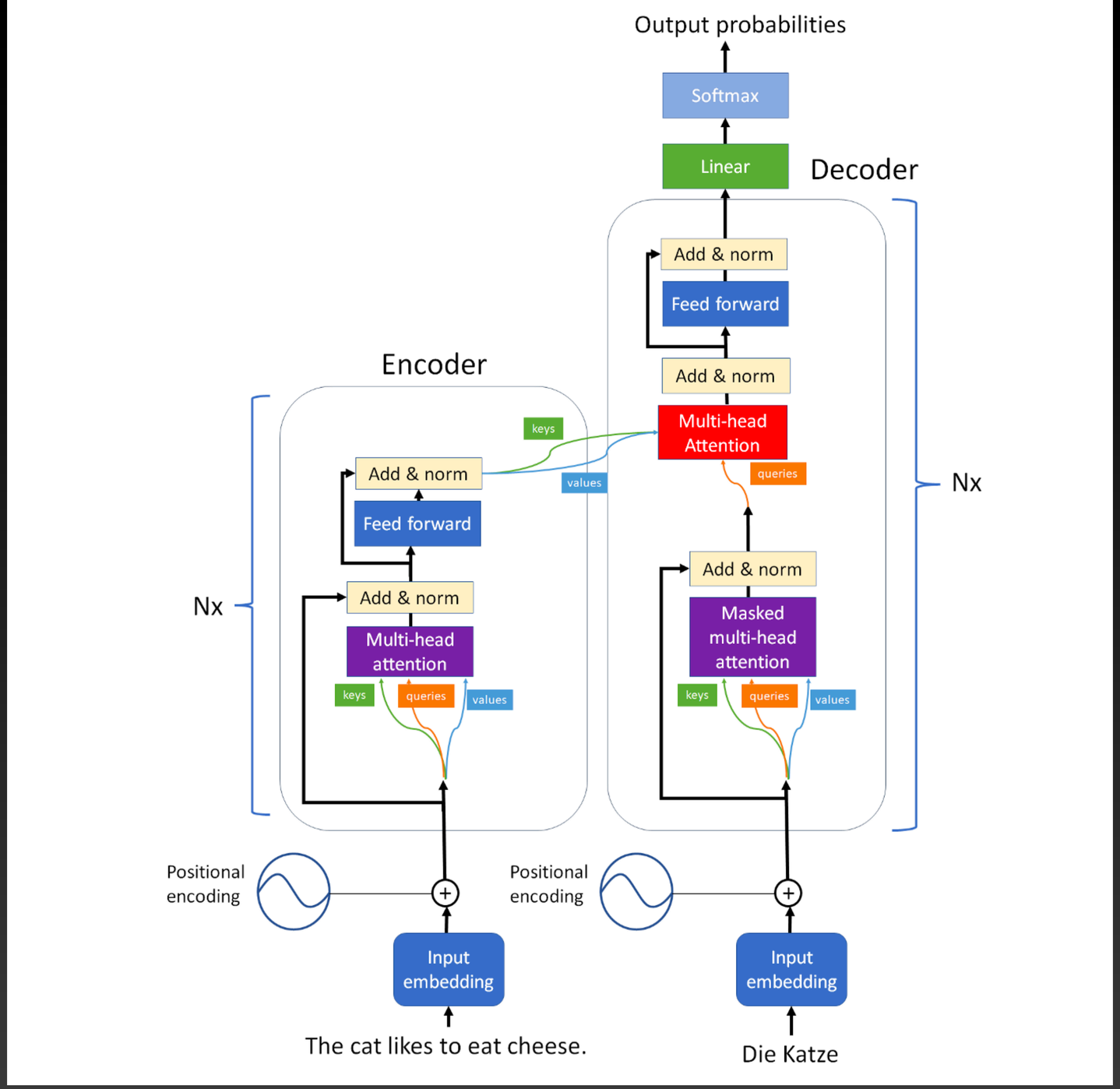
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{3}$$

Bahdanau, D., Cho, K., & Bengio, Y. (2015). "*Neural Machine Translation by Jointly Learning to Align and Translate*". CoRR, abs/1409.0473.

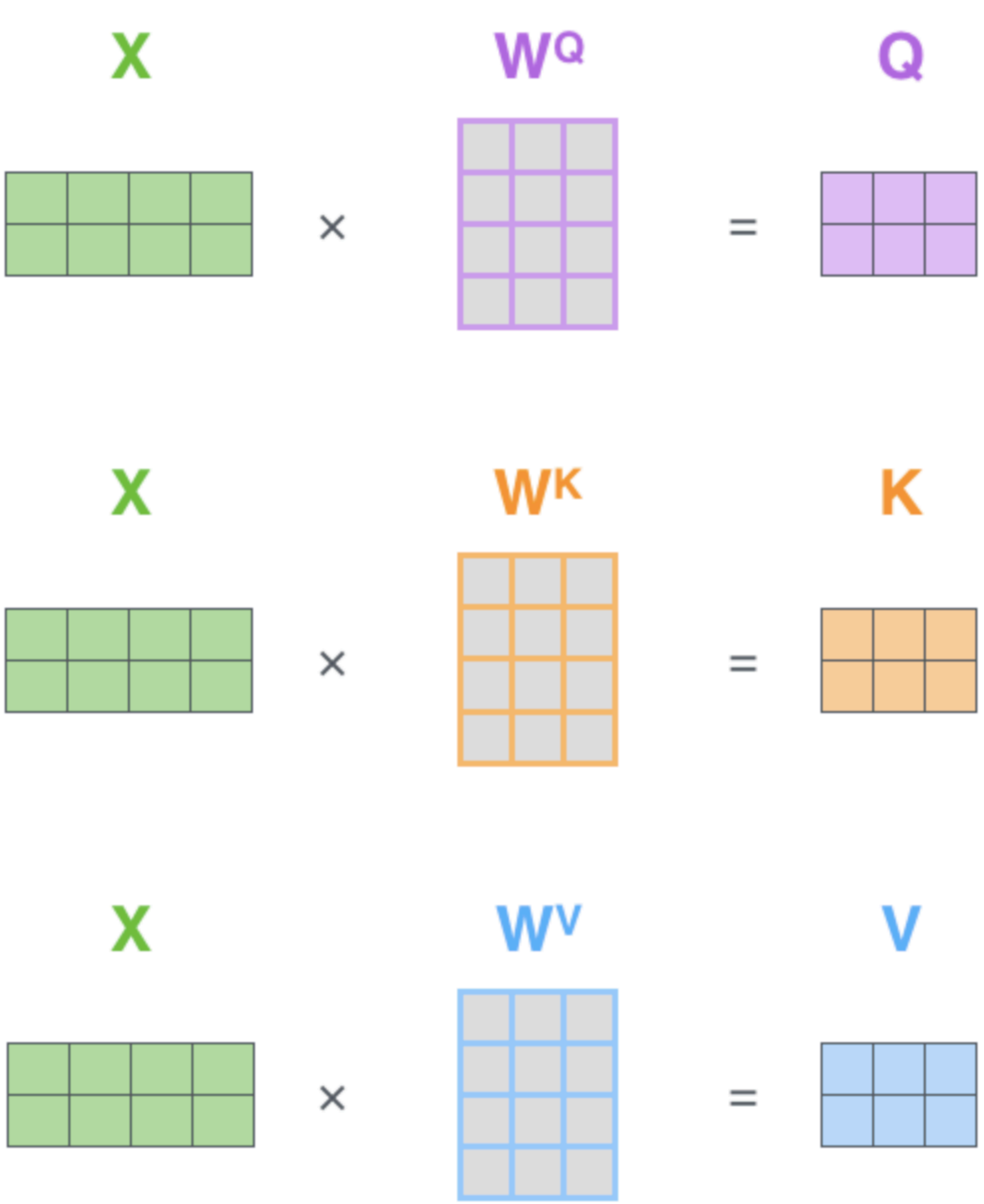
What is the main Problem?

Sequential Training: sequentially encode source (n^2), then compute attention for each output (m)

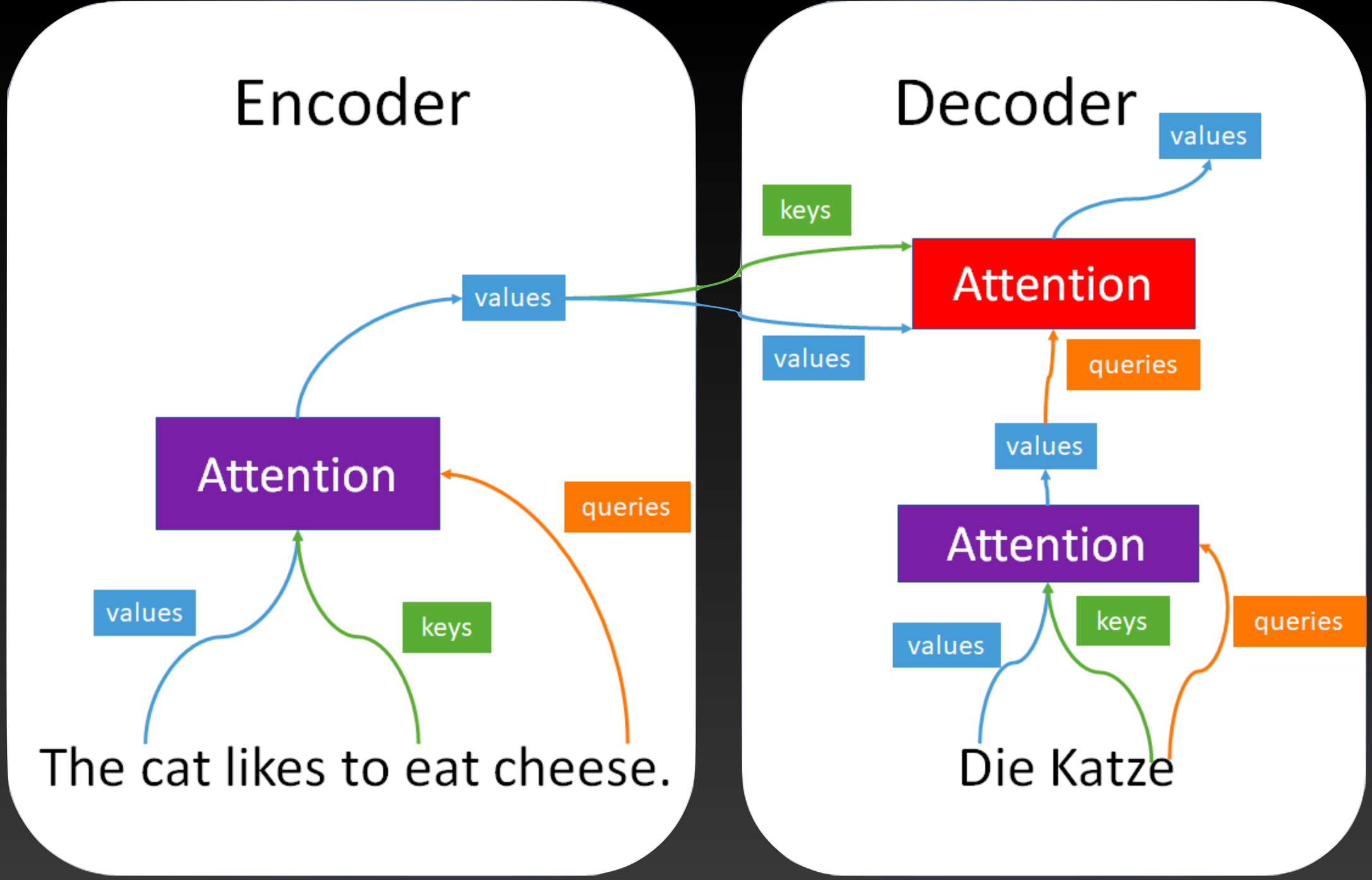
NO PARALLEL TRAINING!



from: https://towardsdatascience.com/attention-please-85bd0abac41



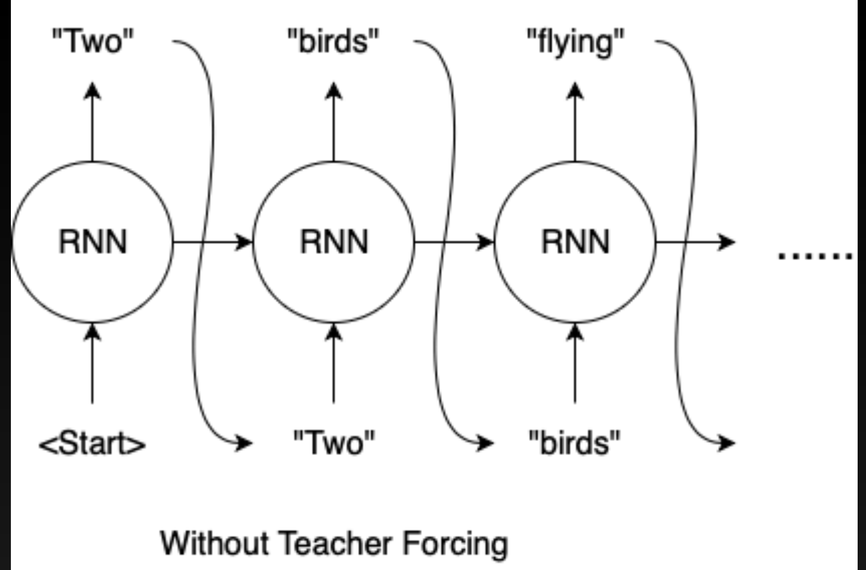
$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$



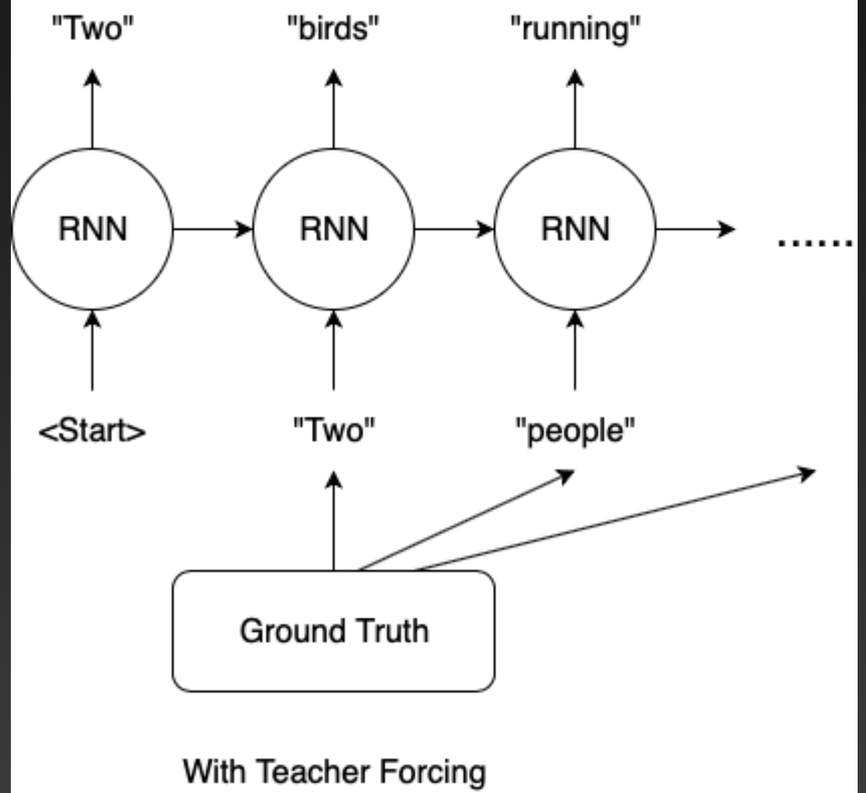
from: https://towardsdatascience.com/attention-please-85bd0abac41

Teacher Forcing

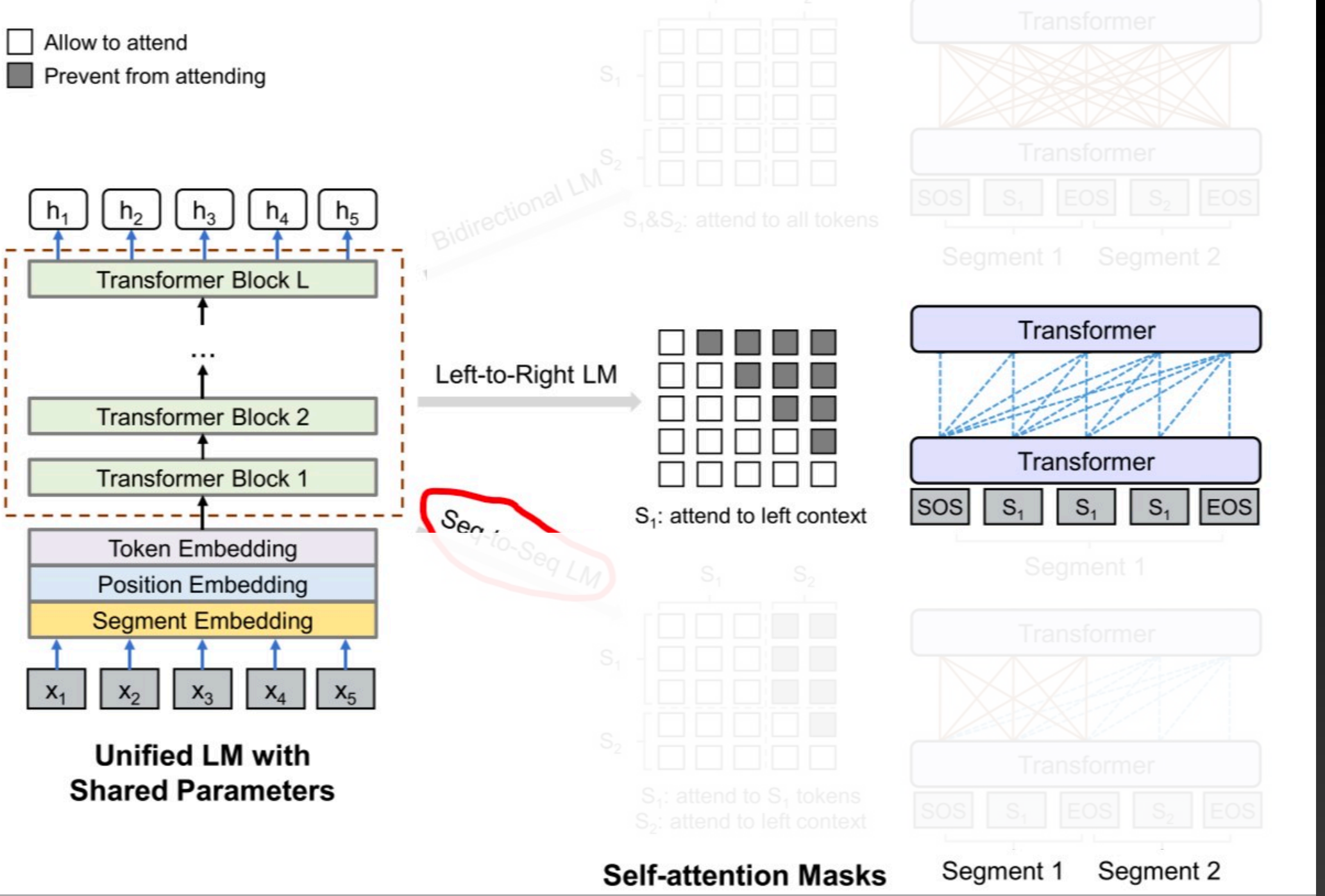
- Originally used to improve training of RNNs to avoid cumulative error
- For Transformer: Also allow to create all training queries at once



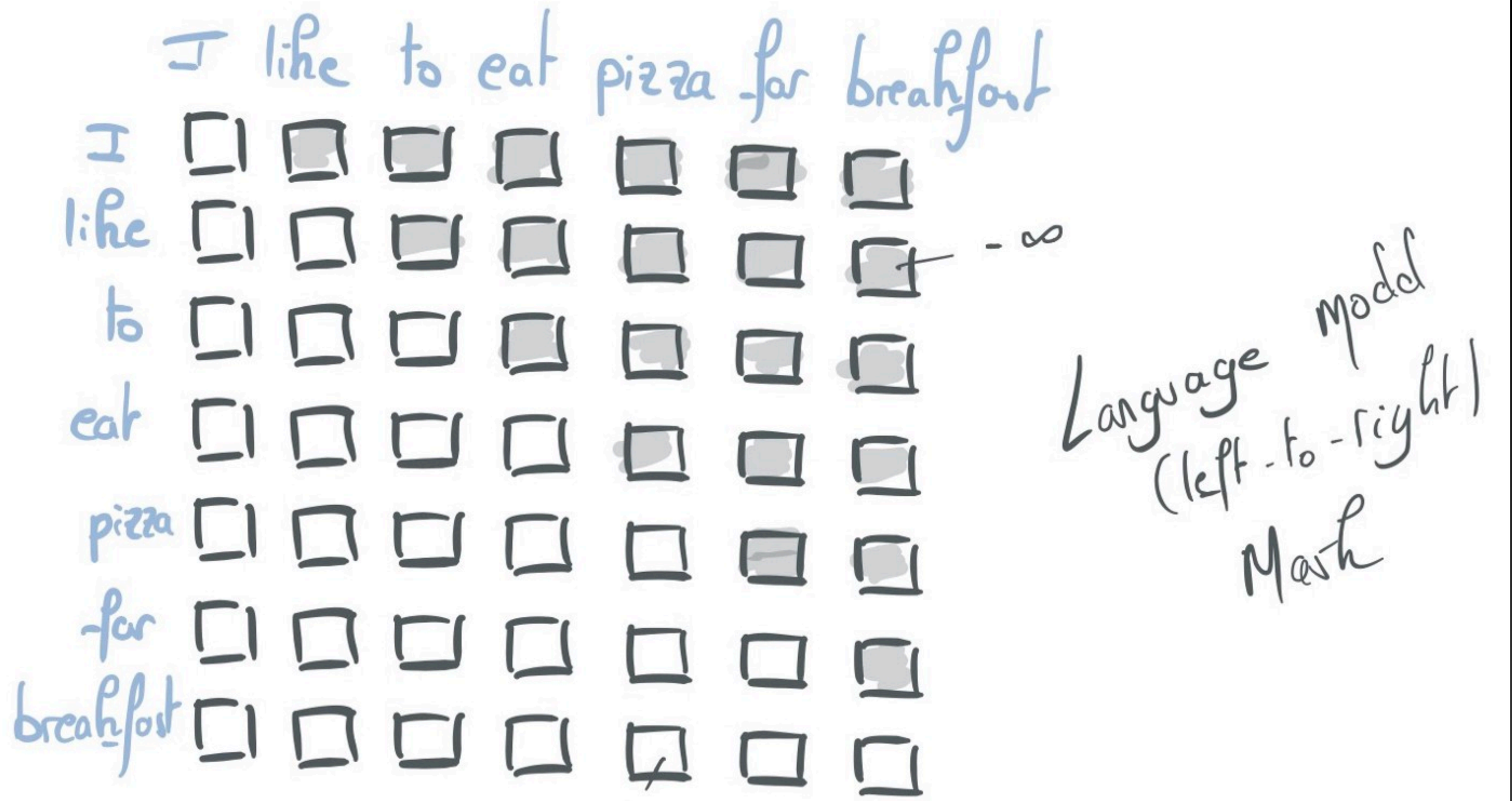
Without Teacher Forcing



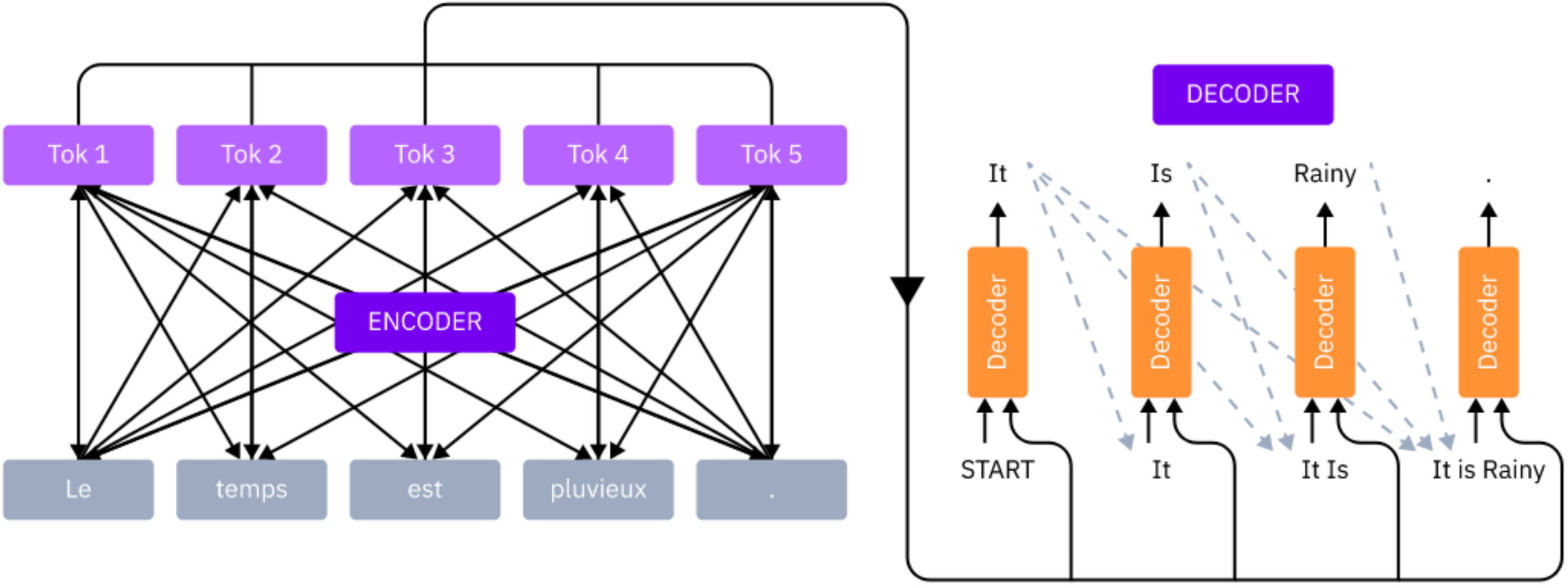
With Teacher Forcing



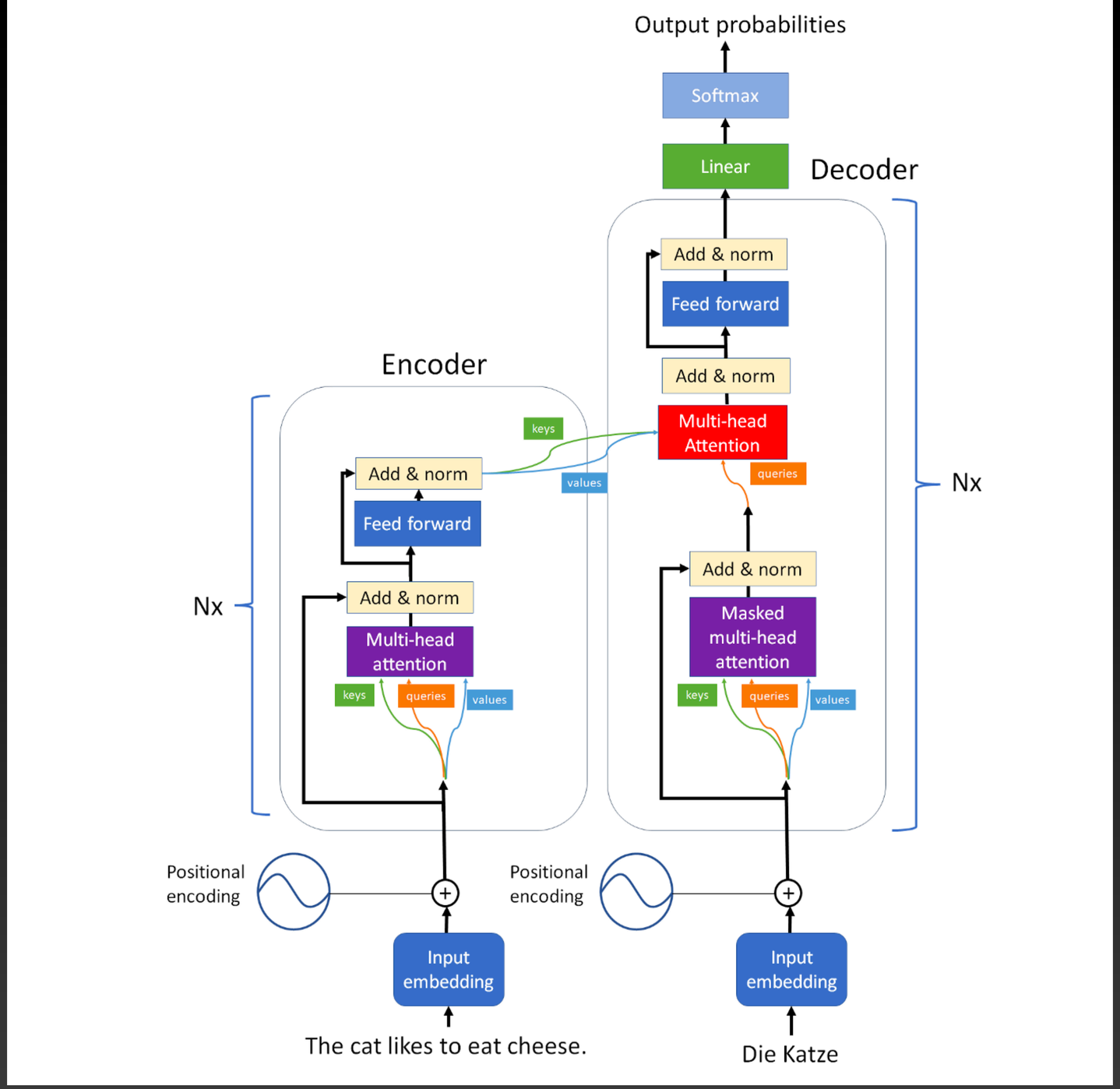
from: https://www.gitmemory.com/issue/huggingface/transformers/9366/754352898



from: https://medium.com/huggingface/encoder-decoders-in-transformers-a-hybrid-pre-trained-architecture-for-seq2seq-af4d7bf14bb8



from: https://scale.com/blog/pytorch-improvements



from: https://towardsdatascience.com/attention-please-85bd0abac41

Is there still a problem?

Attention Mechanism actually allow one token to see all others especially in encoder: They loose all notion of position! (It is not a RNN)

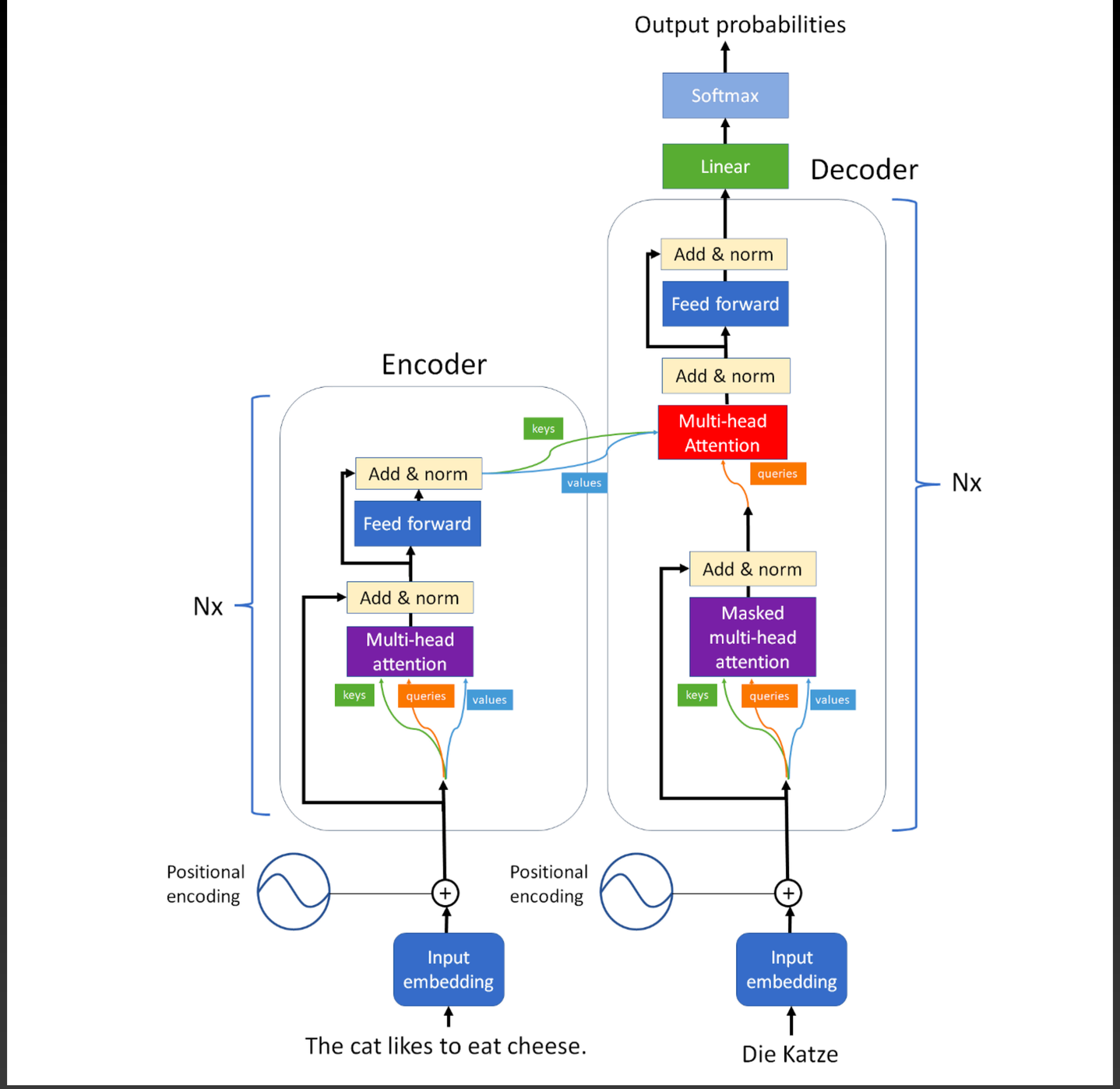
no position encoding

WaveNet

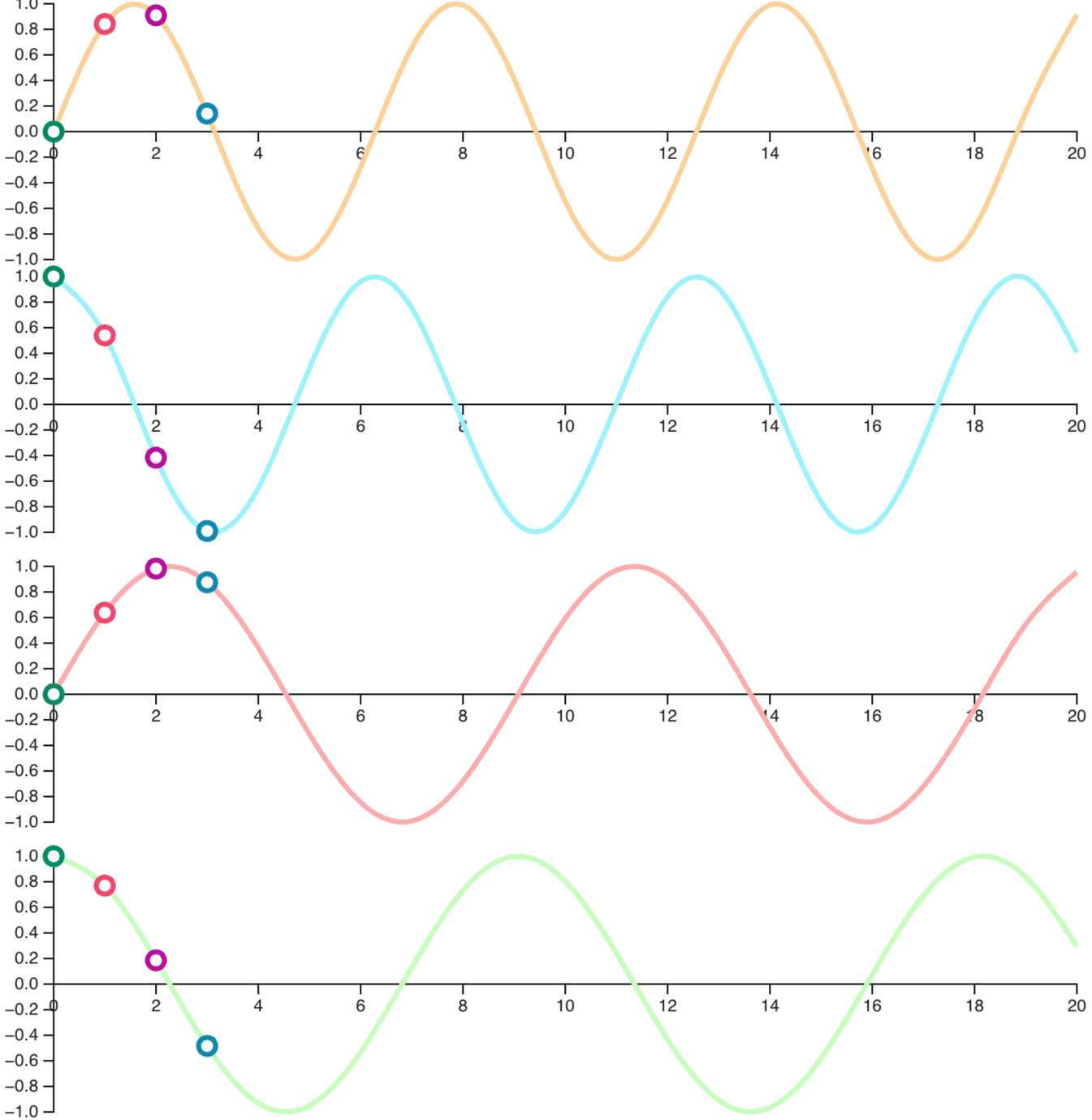
with position encoding

Here simply append (row, colum) num to each pixel value (so add two dimenstions)

Taken from Pieter Abbeel's Unsupervised Deep Learning class (https://www.youtube.com/watch?v=iyEOk8KCRUw)



from: https://towardsdatascience.com/attention-please-85bd0abac41



p0	p1	p2	p3	
0.000	0.841	0.909	0.141	**i=0**
1.000	0.540	-0.416	-0.990	**i=1**
0.000	0.638	0.983	0.875	**i=2**
1.000	0.770	0.186	-0.484	**i=3**

Positional Encoding

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

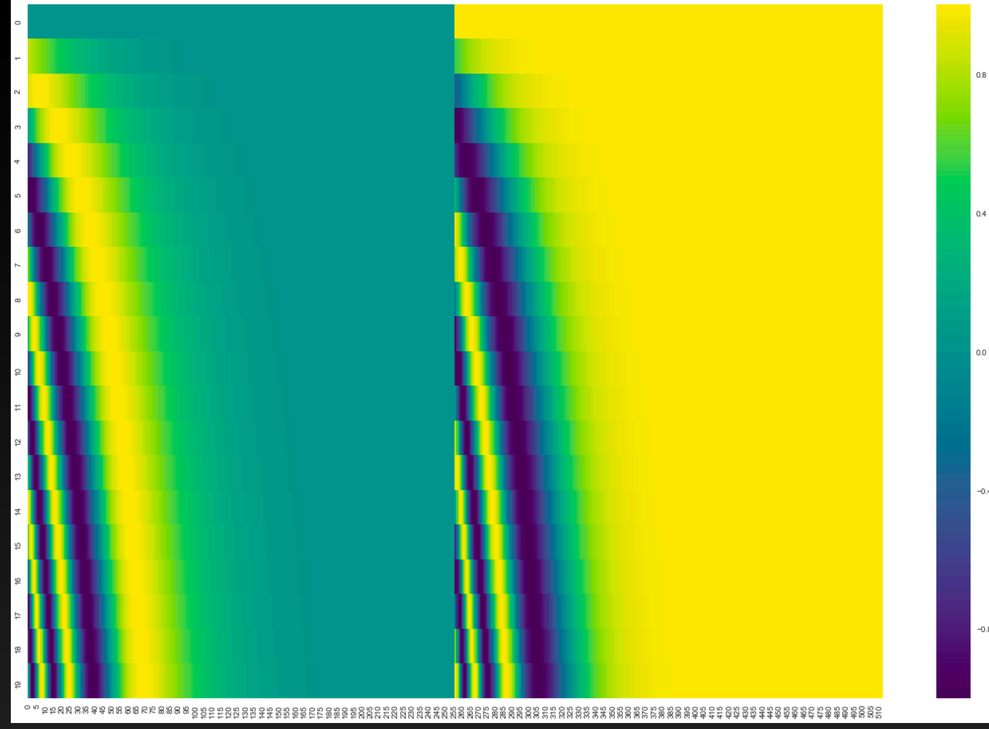
Settings: d = 50

The value of each positional encoding depends on the *position* (*pos*) and *dimension* (*d*). We calculate result for every *index* (*i*) to get the whole vector.

from https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers

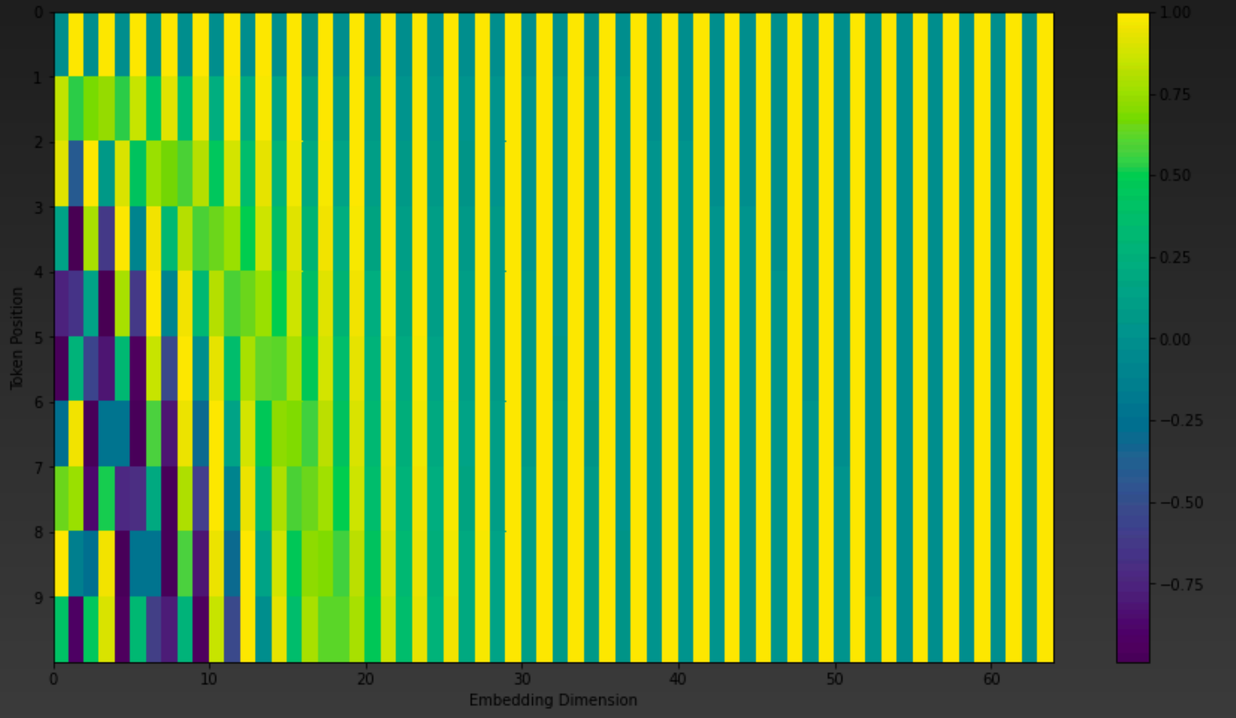
Real Example from Text

sine and cosine separately

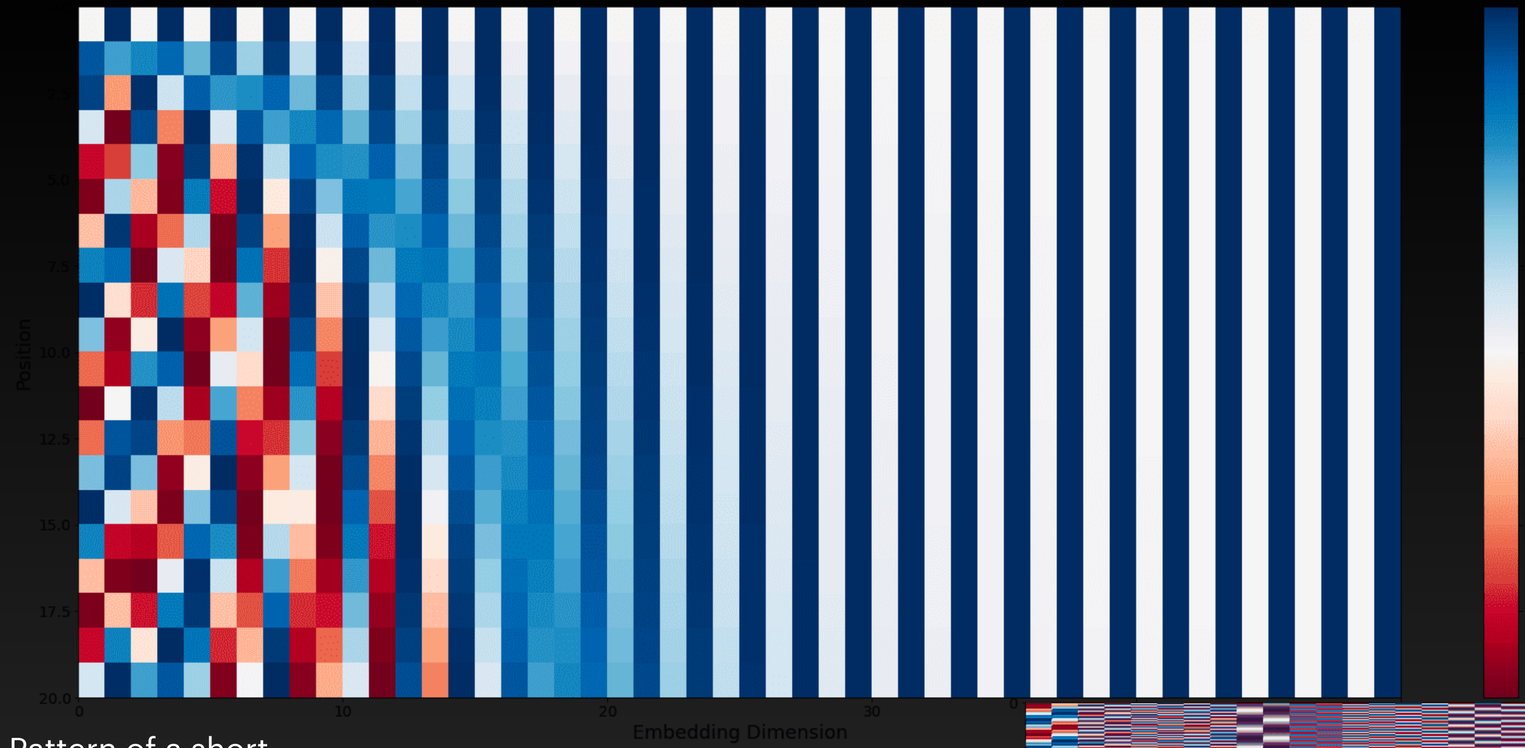


20 words text, 512 embedding
Dimension

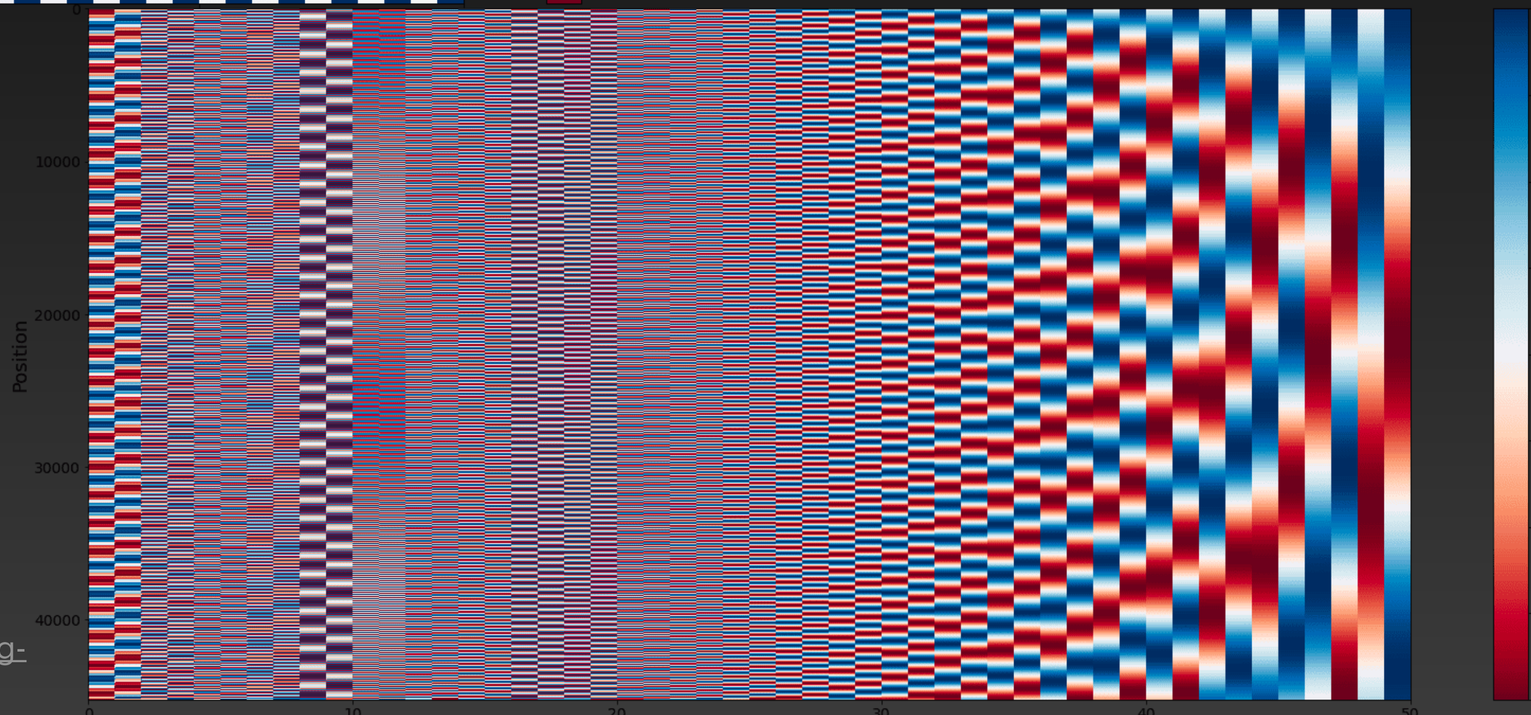
combined position embedding



from: https://jalammar.github.io/illustrated-transformer/

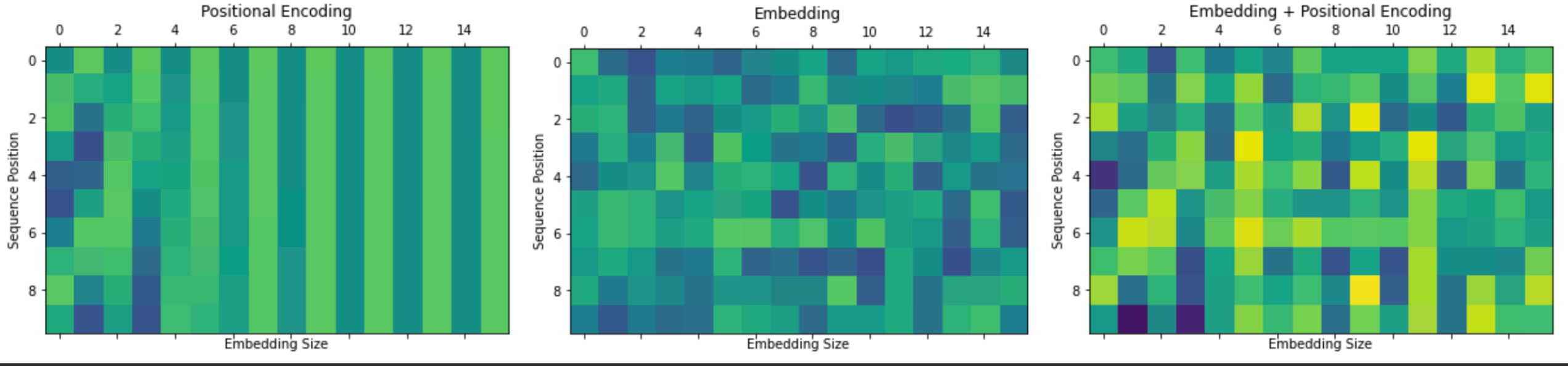


Pattern of a short sequence (20 steps)



Pattern of a long sequence (50k steps)

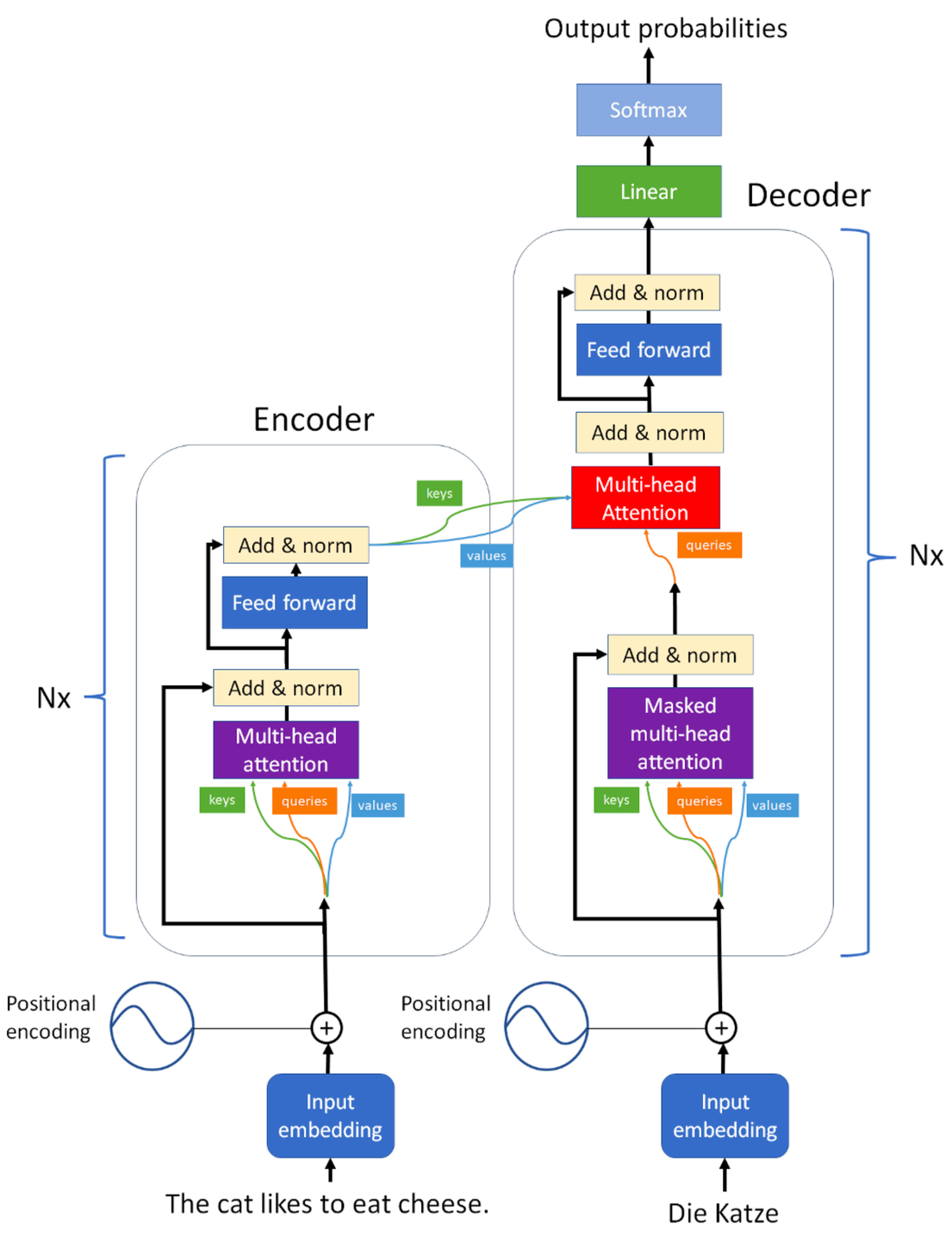
from https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers



from: https://medium.com/swlh/what-exactly-is-happening-inside-the-transformer-b7f713d7aded

Full Picture
Vanilla Transformer

- Data is flattened and embedded into Embeddings
- Position Encodings are created and plainly added to the embeddings
- Key, Query, Value Multi Head Self Attention is encoded
- Causual Self-Attention is used to create all decoder queries at once
- The Encoder and Decoder are combined as keys and values from the encoder and queries from the decoder for the output

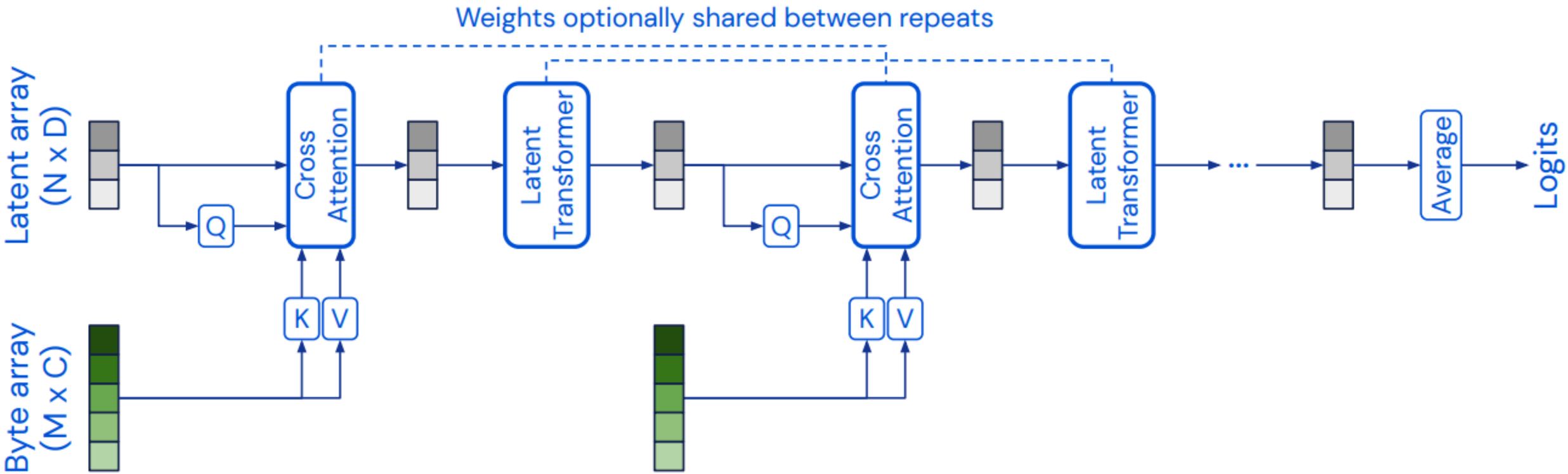


Some other aspect

We don’t have time to do that in detail too

- The classic Transformer has an encoder and decoder part
  - Depending on the use case you only need one
  - GPT: Decoder only / focus on generation
  - BERT: Encoder only / focus on classification
  - T5: All of it
  - Physiological Transformers often classify: Often only Transformer
- Training Types matter:
  - Mainly Masked Self-Supervised Pre Training. Especially in the encoder in both directions
    - BERT
    - Wave2Vec 2.0

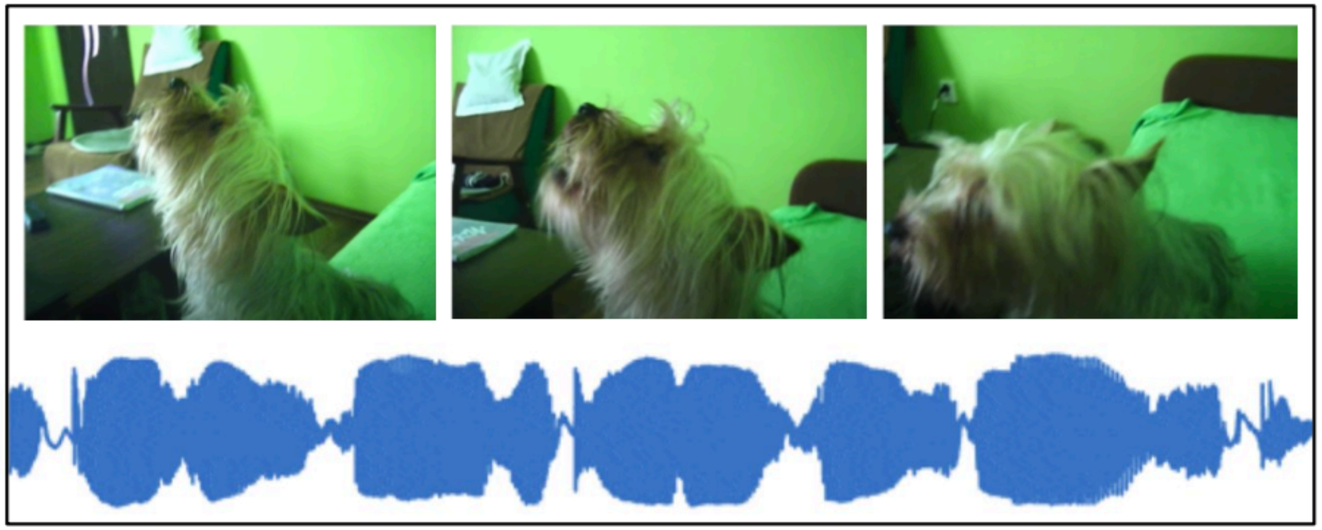
Perceiver

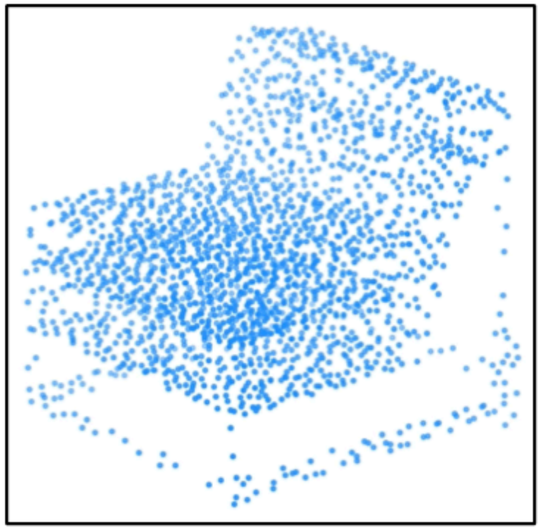


Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). "*Perceiver: General Perception with Iterative Attention*". ICML.

Pervceiver
General Idea

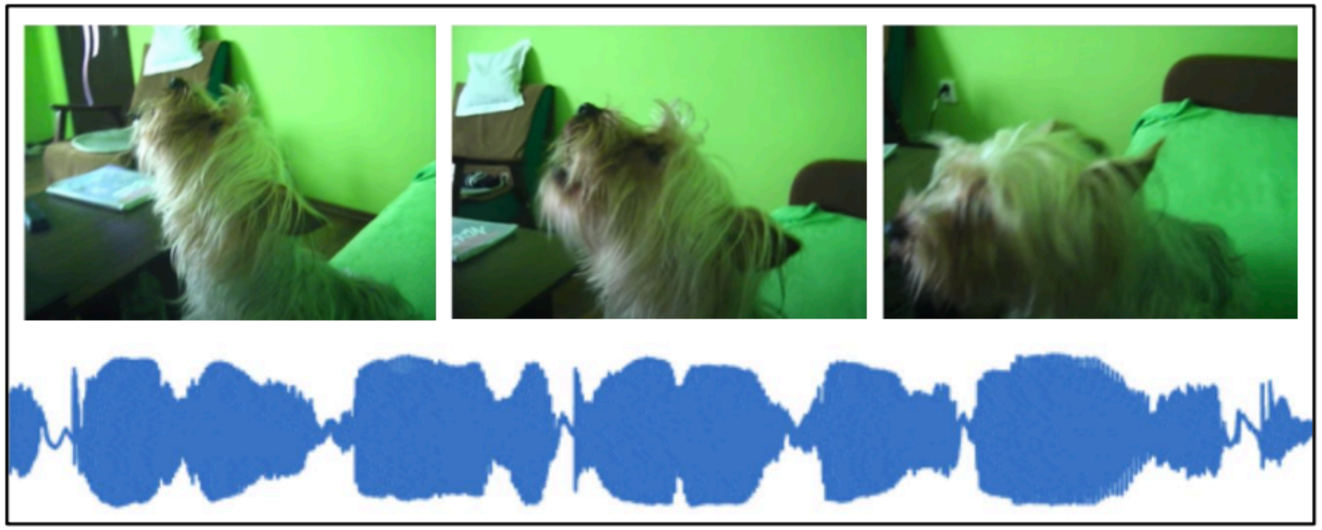
- We have less timepoints than the original input in the latent dimension for most layers (Latent Transformer)
- We occasionally attend to the whole input again (Cross Attention)
- Since the latent is much smaller than the sequence length, we gain speedups
- We only allow classification for now / only encoder of a Transformer
- Instead of Modality specific Architectures we always keep the Architecture the same and:
  - Always just flatten the input into one vector (or in patches: matrix)
  - Use modality specific position encodings
- Some other tricks like weight sharing exist, but are again modality specific



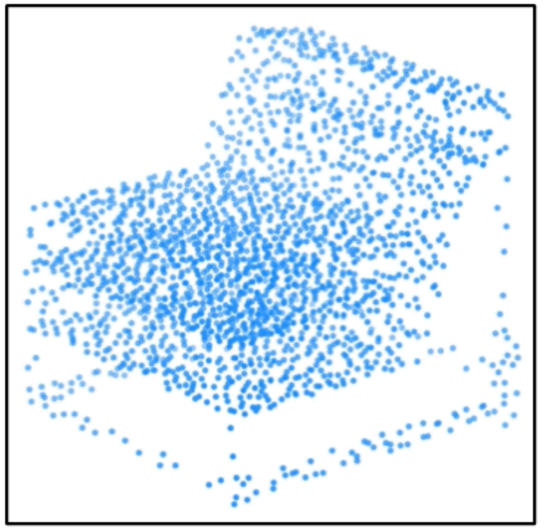


Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). "*Perceiver: General Perception with Iterative Attention*". ICML.

One ImageNet Model



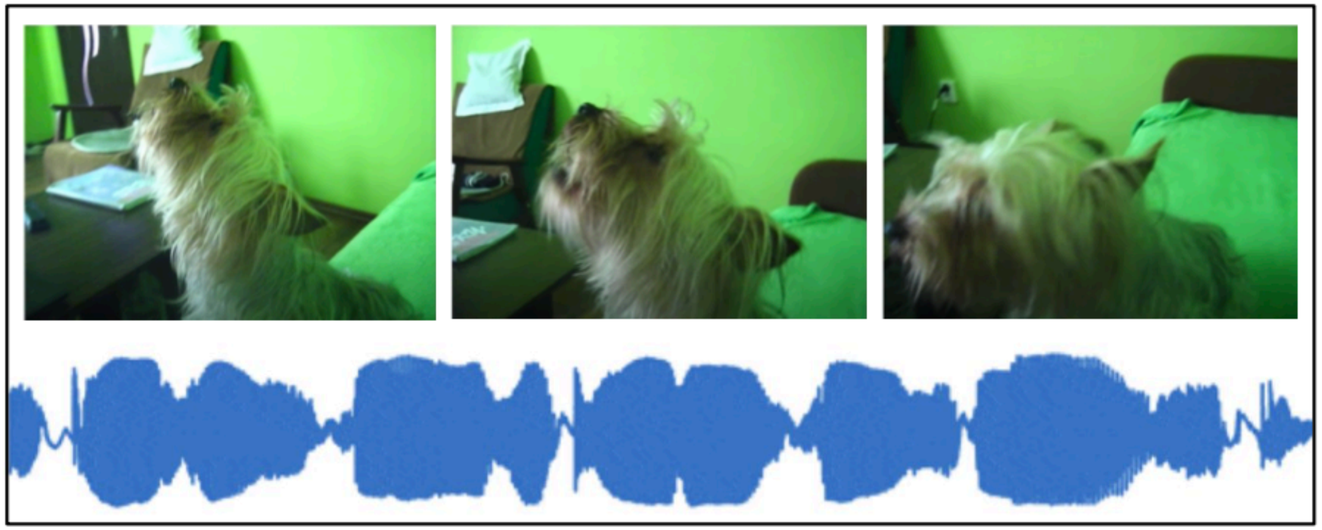
One AudioSet Model



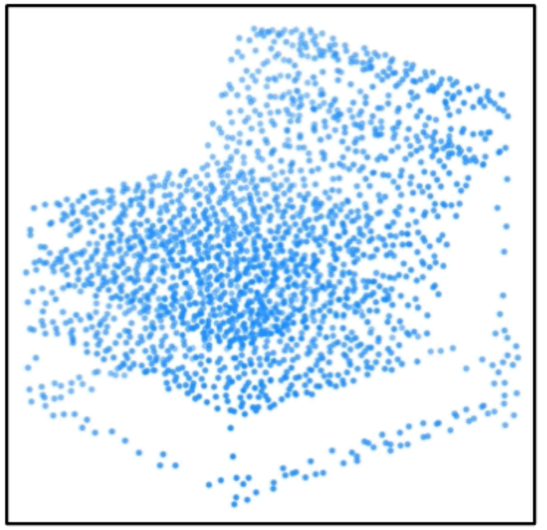
One ModelNet40 Model

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). "*Perceiver: General Perception with Iterative Attention*". ICML.

One ImageNet Model

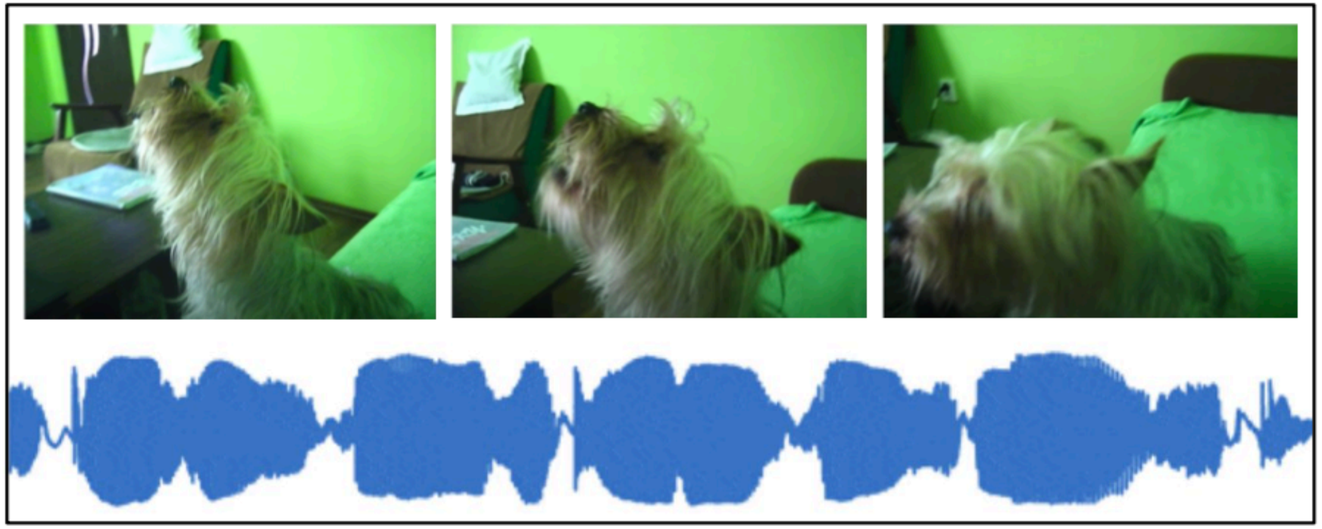


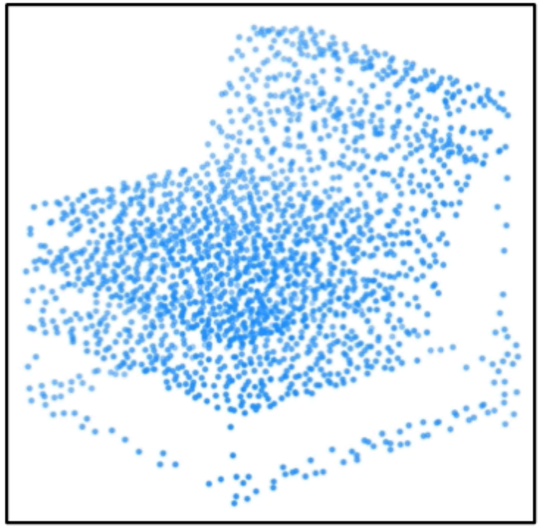
One AudioSet Model



One ModelNet40 Model

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). "*Perceiver: General Perception with Iterative Attention*". ICML.





	Raw	Perm.	Input RF
ResNet-50 (FF)	73.5	39.4	49
ViT-B-16 (FF)	76.7	61.7	256
Transformer (64x64) (FF)	57.0	57.0	4,096
Perceiver:			
(FF)	78.0	78.0	50,176
(Learned pos.)	70.9	70.9	50,176

Model / Inputs	Audio	Video	A+V
Benchmark (Gemmeke et al., 2017)	31.4	-	-
Attention (Kong et al., 2018)	32.7	-	-
Multi-level Attention (Yu et al., 2018)	36.0	-	-
ResNet-50 (Ford et al., 2019)	38.0	-	-
CNN-14 (Kong et al., 2020)	43.1	-	-
CNN-14 (no balancing & no mixup) (Kong et al., 2020)	37.5	-	-
G-blend (Wang et al., 2020c)	32.4	18.8	41.8
Attention AV-fusion (Fayek & Kumar, 2020)	38.4	25.7	46.2
Perceiver (raw audio)	38.3	25.8	43.5
Perceiver (mel spectrogram)	38.4	25.8	43.2
Perceiver (mel spectrogram - tuned)	-	-	44.2

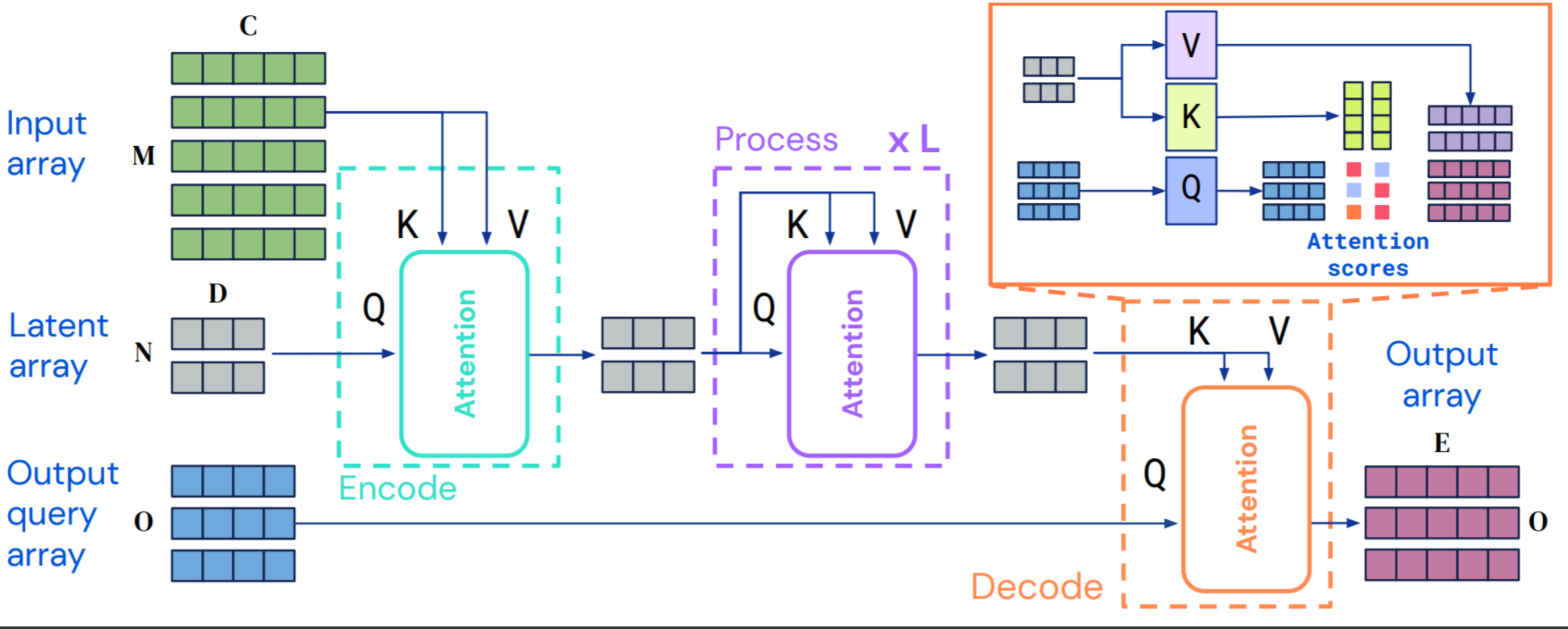
	Accuracy
PointNet++ (Qi et al., 2017)	**91.9**
ResNet-50 (FF)	66.3
ViT-B-2 (FF)	78.9
ViT-B-4 (FF)	73.4
ViT-B-8 (FF)	65.3
ViT-B-16 (FF)	59.6
Transformer (44x44)	82.1
Perceiver	**85.7**

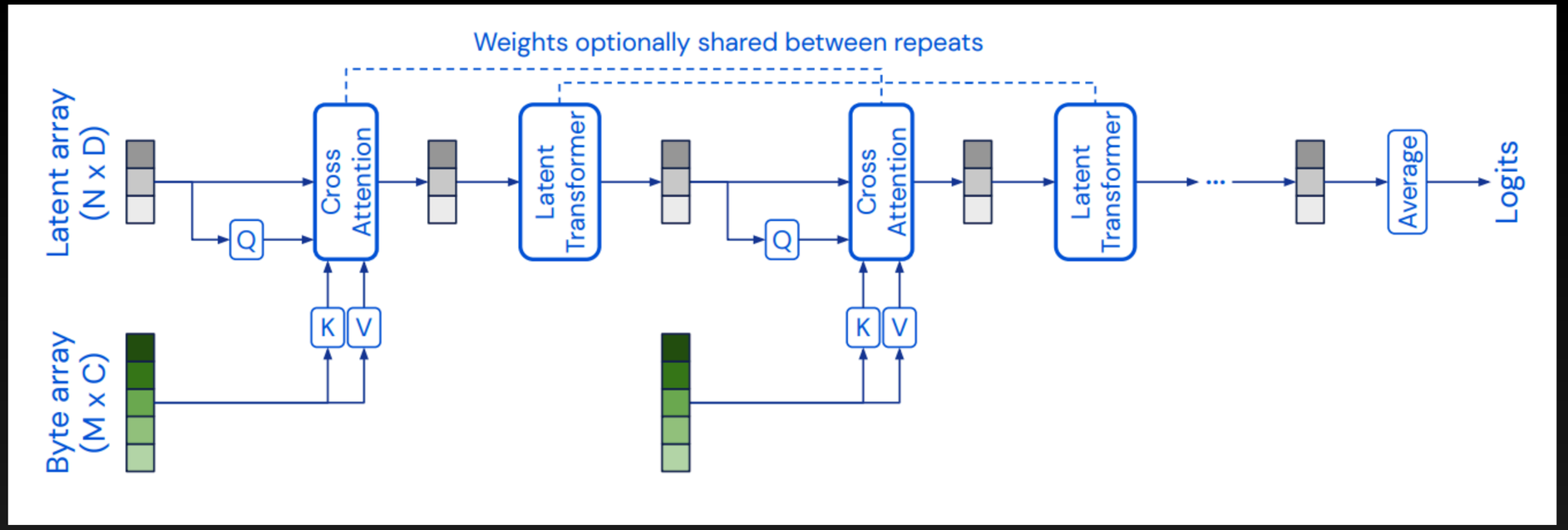
Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). "*Perceiver: General Perception with Iterative Attention*". ICML.

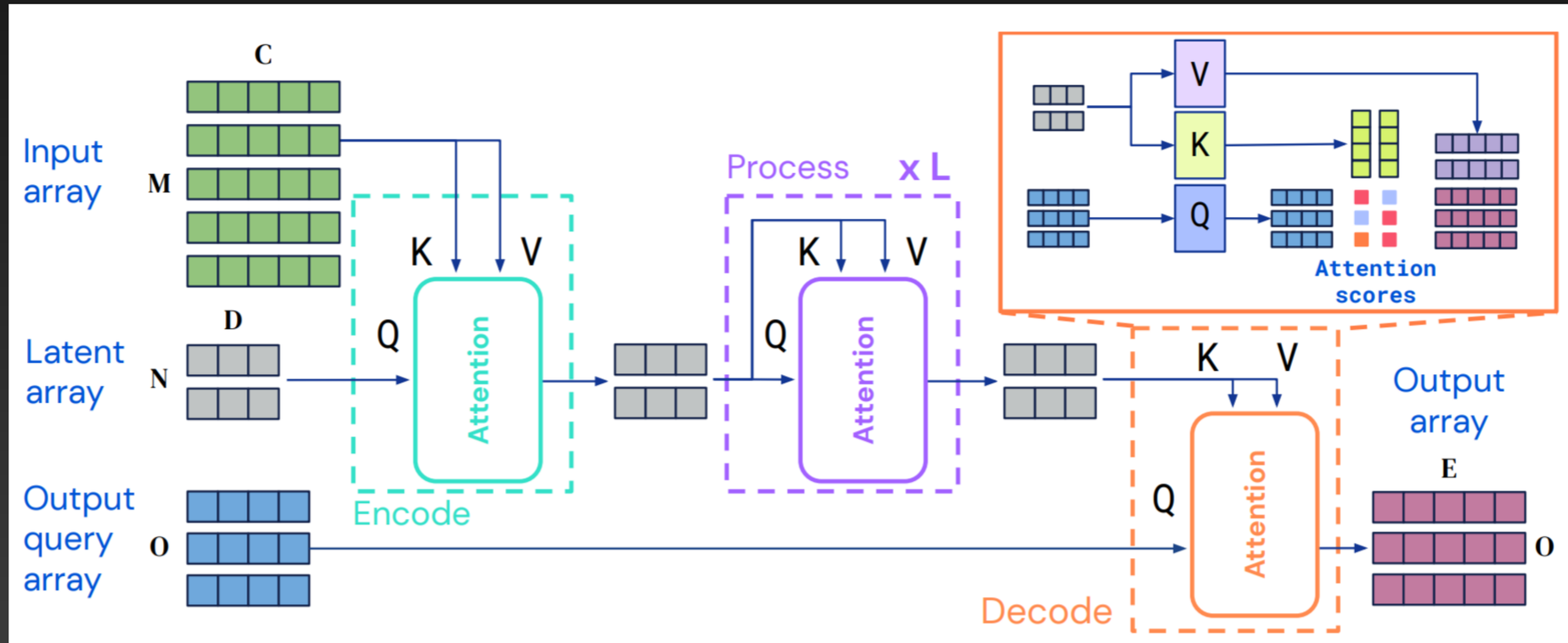
PerceiverIO

PerceiverIO

- The original Perveiver only can do classification
- But what if you want do do all kind of tasks?
  - Decoding? Generation of Data? Segmentation?
- Key for a Transformer: Getting the queries right
- Key Idea of Perveiver: One output format to rule them all!
  - Everything is a query of position encodings or learned queries


C
Input array
M
D
Latent array
N
Output query array
O
K
V
Q
Attention
Encode
Process
x L
Decode
Attention scores
Output array
E


Weights optionally shared between repeats
Latent array (N x D)
Byte array (M x C)
Q
K
V
Cross Attention
Latent Transformer
Cross Attention
Latent Transformer
...
Average
Logits


C
Input array
M
D
Latent array
N
Output query array
O
K
V
Q
Attention
Encode
Process
x L
K
V
Q
Attention
V
K
Q
Attention scores
K
V
Q
Attention
Decode
Output array
E
O

War of Frameworks

The AI Community can not decide

- Current state of Frameworks:
  - PyTorch most popular in research
  - Tensorflow is the main legacy
  - But there is JAX
    - And there is JAX (Flask)
    - TRAX
    - JAX as part of TF
    - JAX from OpenAI

Where to get Code

- PyTorch:
  - https://github.com/krasserm/perceiver-io (PyTorch Lightening)
  - https://github.com/esceptico/perceiver-io/
  - https://github.com/lucidrains/perceiver-pytorch (LucidRains is the PyTorch Transformer provider beside Huggingface [which is NLP focused])
    - https://github.com/fac2003/perceiver-multi-modality-pytorch (Multimodal Extension)
    - https://github.com/frenkiboy/perceiver-pytorch (Radiology Data)
- Tensorflow:
  - https://github.com/Rishit-dagli/Perceiver
- JAX:
  - https://github.com/deepmind/deepmind-research/tree/master/perceiver

Code examples