sdu.dk

#sdudk

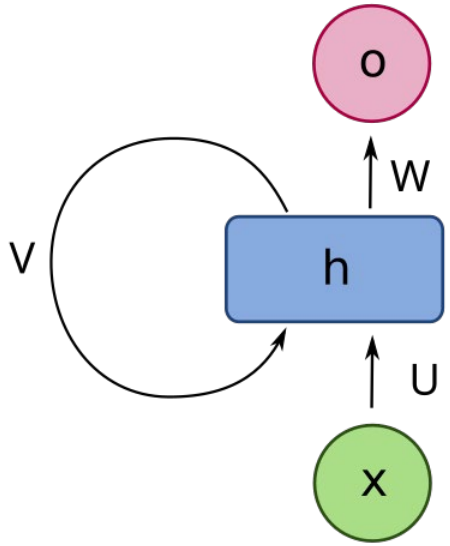
Image-based Representation Learning with Transformers

Joakim Bruslund Haurum
Assistant Professor
Center for Software Technology - SDU Vejle

SDU

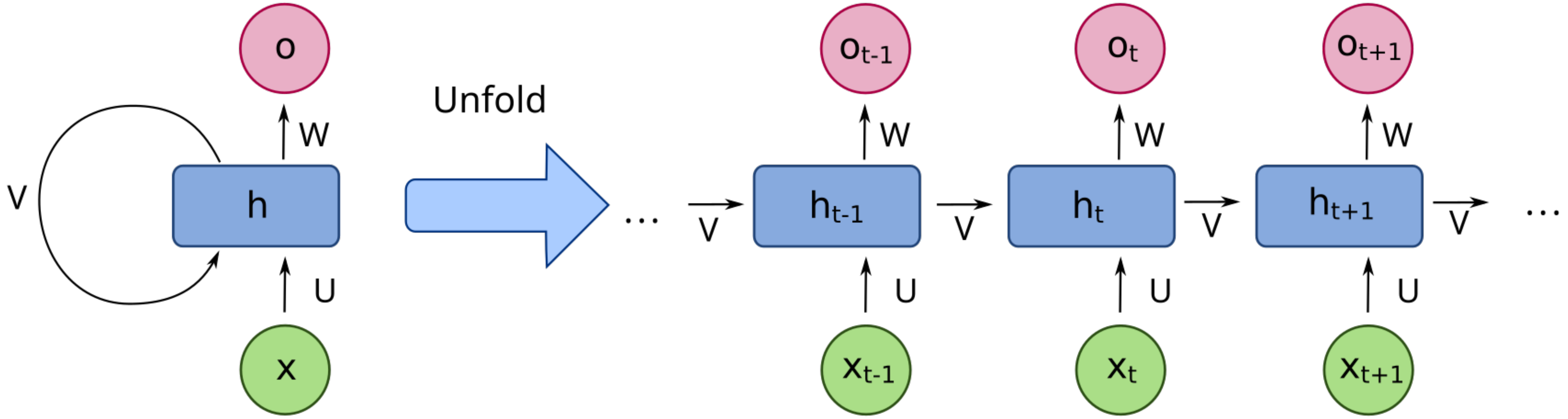
Quick review of Recurrent Neural Networks

Quick review of Recurrent Neural Networks



https://en.wikipedia.org/wiki/Recurrent_neural_network

Quick review of Recurrent Neural Networks



https://en.wikipedia.org/wiki/Recurrent_neural_network

Drawbacks of RNNs

Drawbacks of RNNs

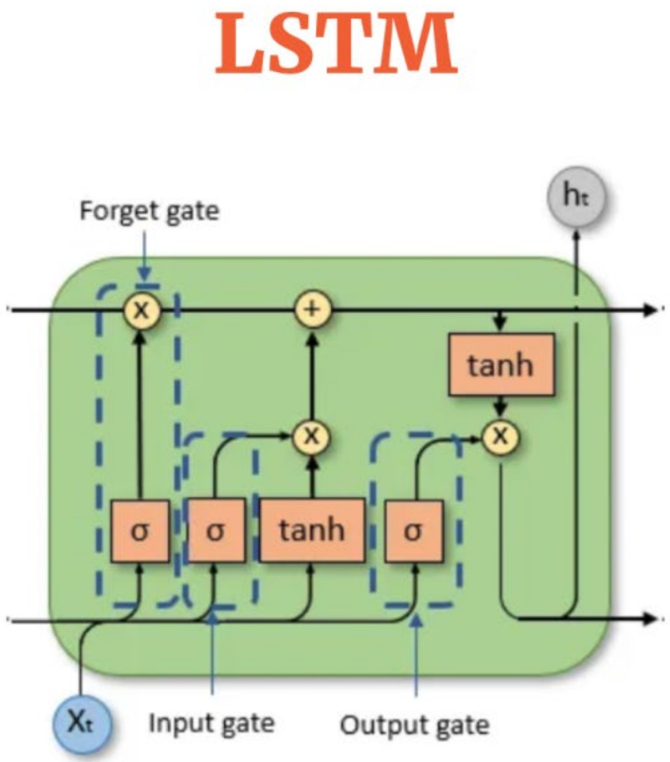
→ Vanishing / Exploding Gradients

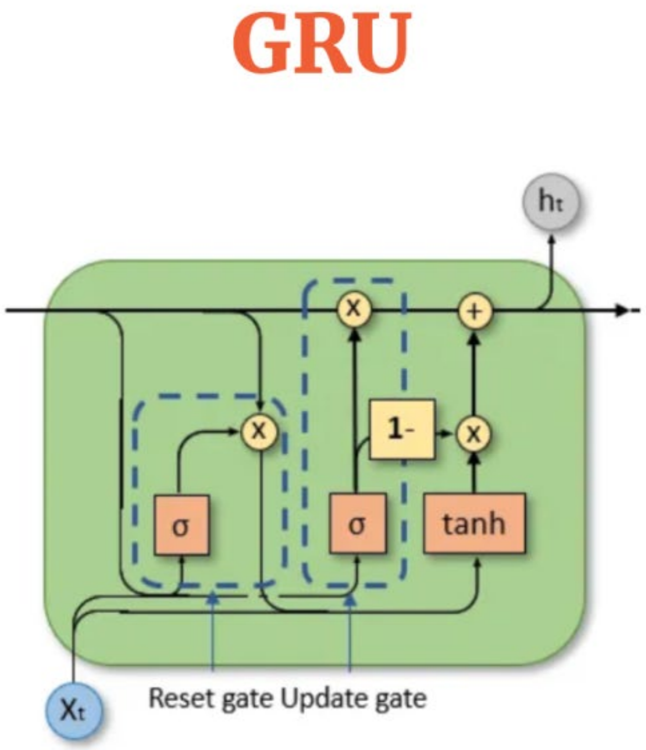
→ Limited Memory / Recency Bias

→ Inefficienct training due to sequential nature

How to handle these drawbacks?

How to handle these drawbacks?





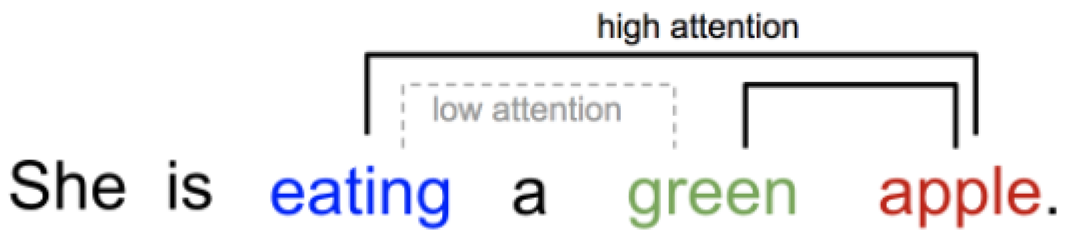
https://aiml.com/compare-the-different-sequence-models-rnn-lstm-gru-and-transformers/

How to handle these drawbacks?

→ What if we used a model which learns what to *attend* to?

How to handle these drawbacks?

→ What if we used a model which learns what to *attend* to?



How to handle these drawbacks?

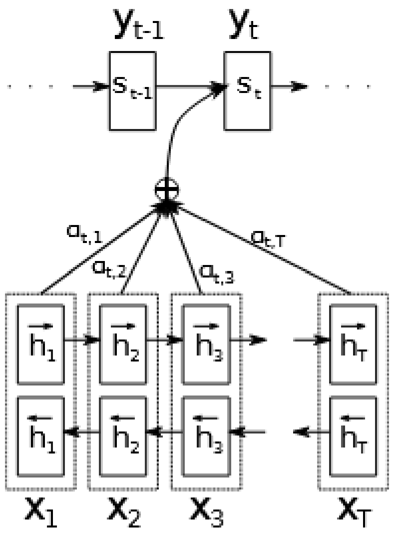
→ What if we used a model which learns what to *attend* to?

1. Collect hiden states for input sequence
2. Aggregate according some data-driven weights
3. Use as context for output prediction

How to handle these drawbacks?

→ What if we used a model which learns what to *attend* to?

1. Collect hiden states for input sequence
2. Aggregate according some data-driven weights
3. Use as context for output prediction

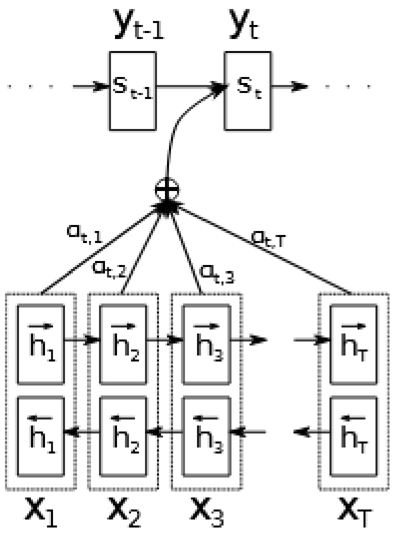


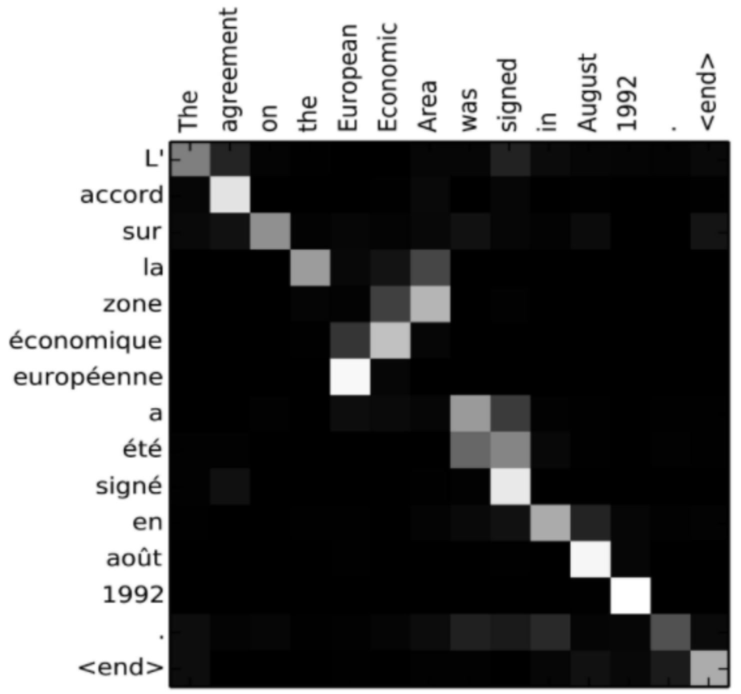
Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau et al, 2015

How to handle these drawbacks?

→ What if we used a model which learns what to *attend* to?

1. Collect hiden states for input sequence
2. Aggregate according some data-driven weights
3. Use as context for output prediction





Is attention all we need?

Is attention all we need?

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Transformer Basics

What is a Transformer?

What is a Transformer?

→ A non-recursive architecture built on Attention and MLPs

What is a Transformer?

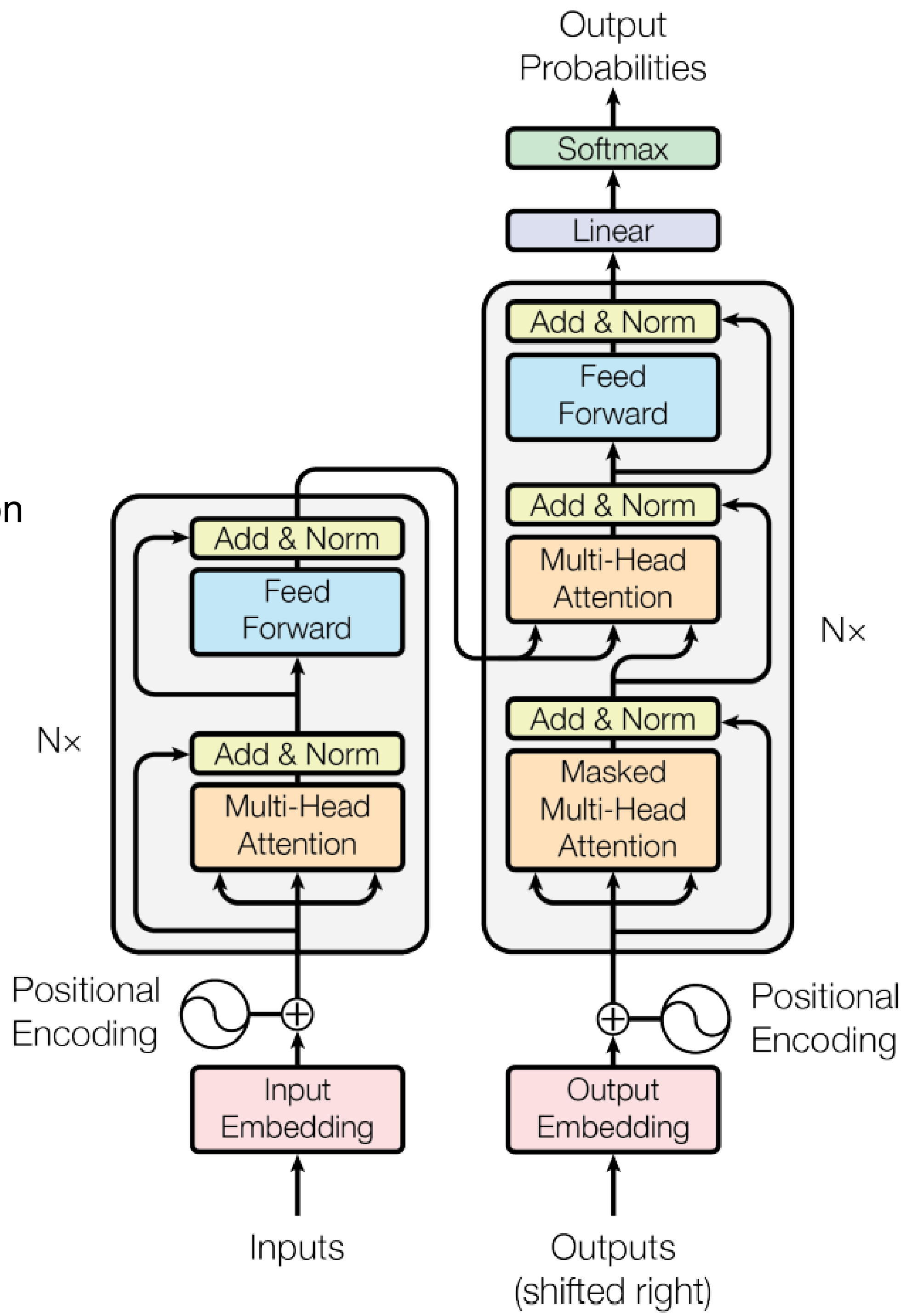
→ A non-recursive architecture built on Attention and MLPs

→ Key insight: Removing reccurrent operations allows for easy parallelization

What is a Transformer?

→ A non-recursive architecture built on Attention and MLPs

→ Key insight: Removing reccurrent operations allows for easy parallelization

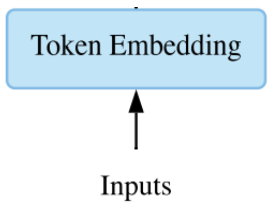


Attention is All You Need, Vaswani, 2017

Key Components

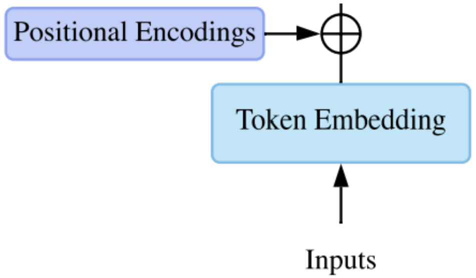
Key Components

→ Tokenizer



Key Components

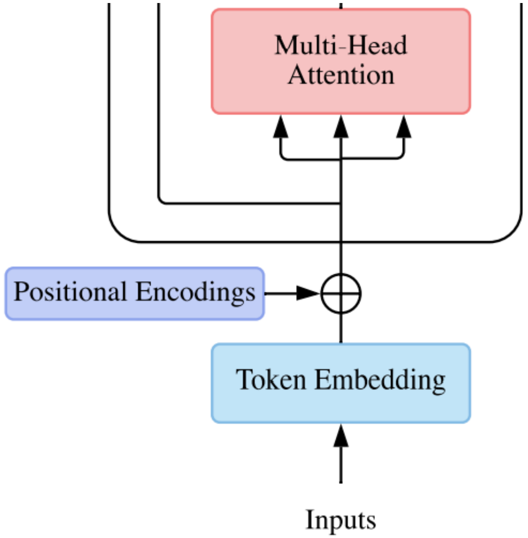
→ Tokenizer
→ Positional Information



A Survey of Transformers, Lin et al, 2021

Key Components

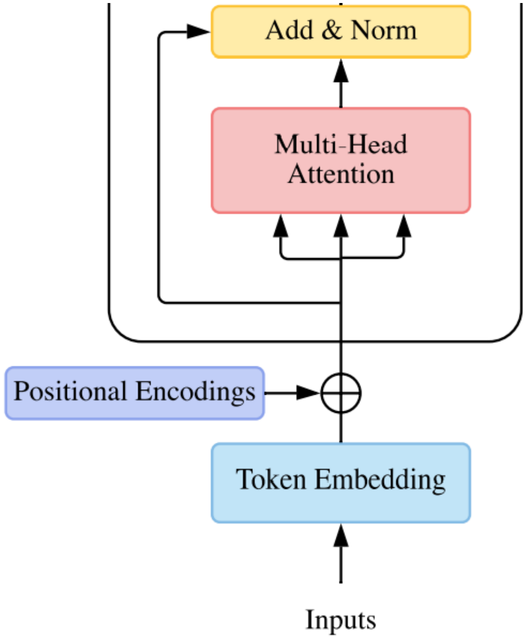
→ Tokenizer
→ Positional Information
→ Token Mixing



A Survey of Transformers, Lin et al, 2021

Key Components

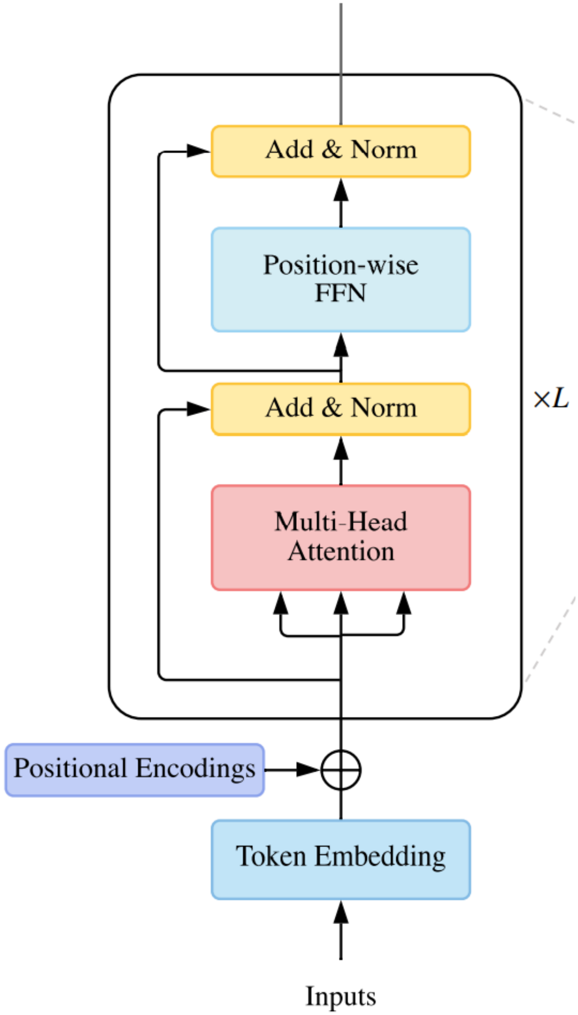
→ Tokenizer
→ Positional Information
→ Token Mixing
→ Residual Connection and Normaliztion Layers



A Survey of Transformers, Lin et al, 2021

Key Components

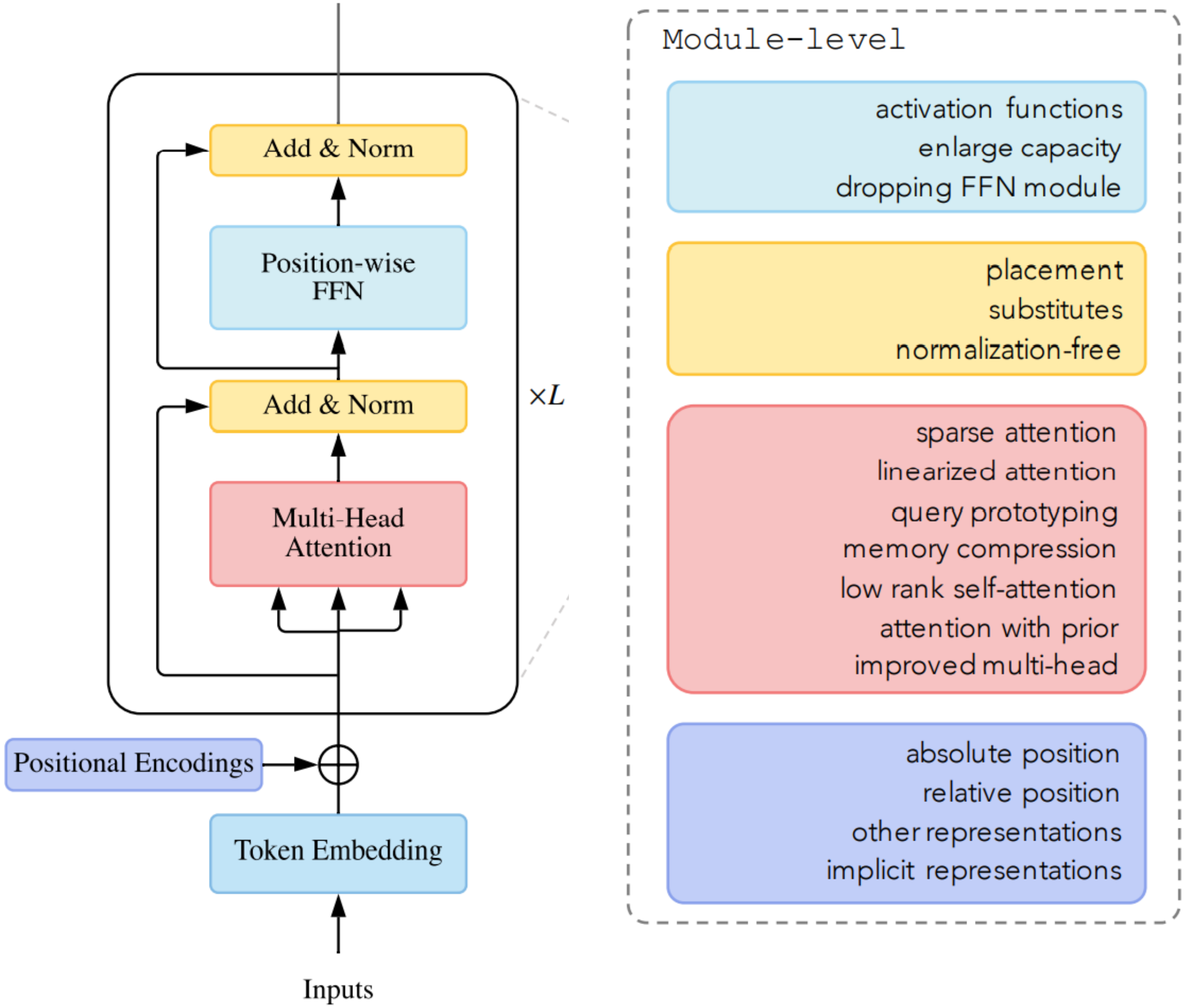
→ Tokenizer
→ Positional Information
→ Token Mixing
→ Residual Connection and Normaliztion Layers
→ Per-token Processing



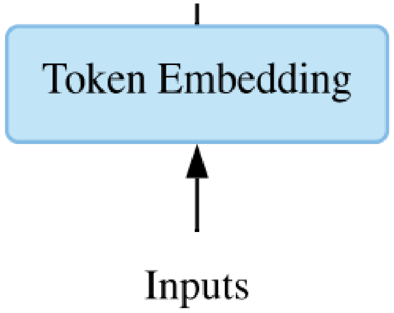
A Survey of Transformers, Lin et al, 2021

Key Components

→ Tokenizer
→ Positional Information
→ Token Mixing
→ Residual Connection and Normaliztion Layers
→ Per-token Processing

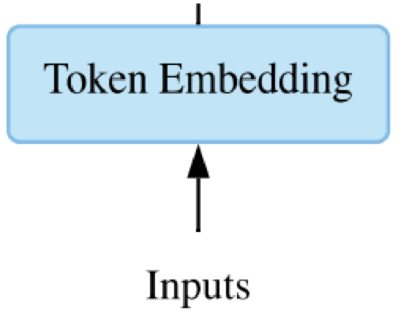


Tokenizer



Tokenizer

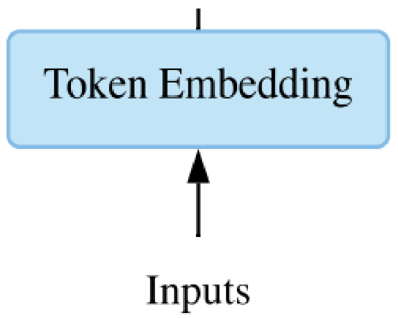
→ Massively important in NLP
  → Used for RNN, LSTMs, GRU, Transformers etc.



Tokenizer

- → Massively important in NLP
  - → Used for RNN, LSTMs, GRU, Transformers etc.

- → Turns sentences into a set of features based on:
  - → Words
  - → Subwords
  - → Byte pair encodings
  - → Etc.



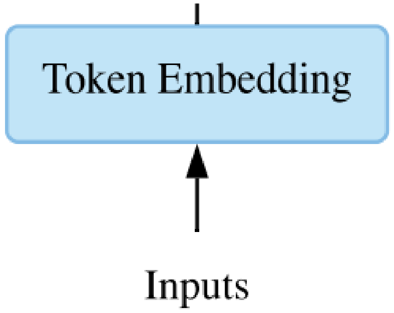
A Survey of Transformers, Lin et al, 2021

Tokenizer

→ Massively important in NLP
  → Used for RNN, LSTMs, GRU, Transformers etc.

→ Turns sentences into a set of features based on:
  → Words
  → Subwords
  → Byte pair encodings
  → Etc.

→ Typically learned independently of the classification model
→ Can be trained on general or domain specific text corpus



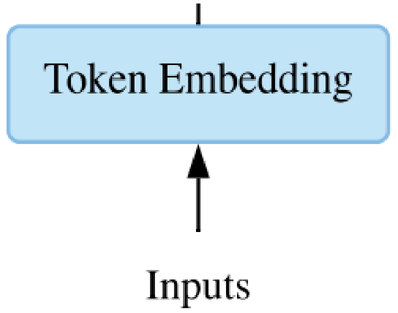
Tokenizer

→ Massively important in NLP
  → Used for RNN, LSTMs, GRU, Transformers etc.

→ Turns sentences into a set of features based on:
  → Words
  → Subwords
  → Byte pair encodings
  → Etc.

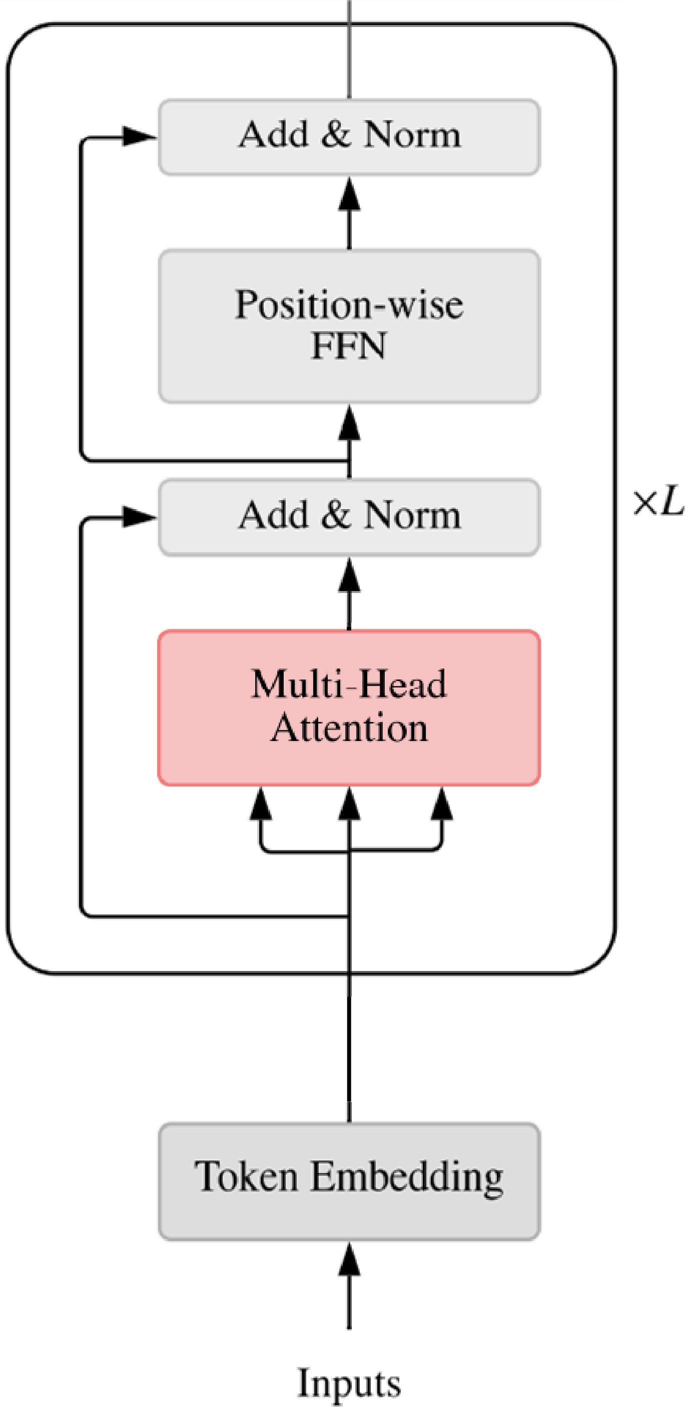
→ Typically learned independently of the classification model
→ Can be trained on general or domain specific text corpus

→ Sometimes includes a special **CLS** token representing the entire input sequence



A Survey of Transformers, Lin et al, 2021

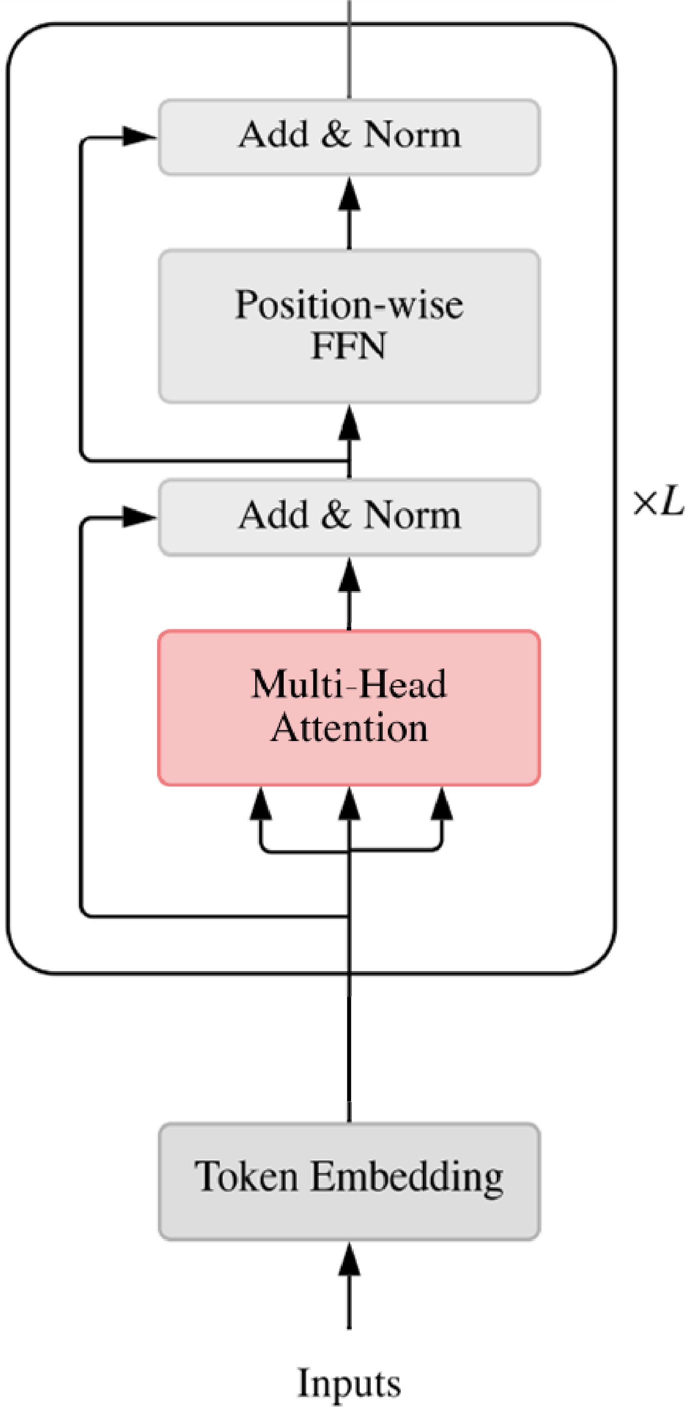
Token Mixing



Token Mixing

→ Where the magic happens!

→ Tokens are ”randomly” mixed
  → Ignore spatial and temporal dependencies

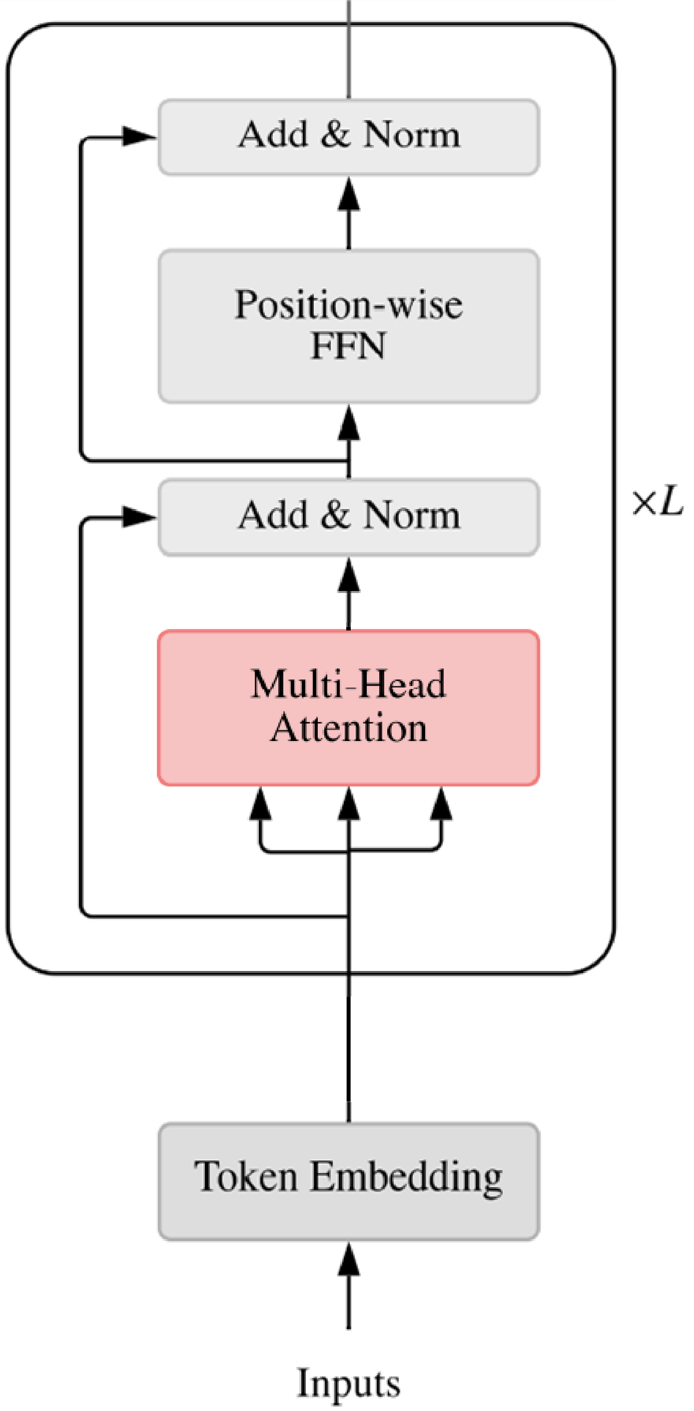


Token Mixing

→ Where the magic happens!

→ Tokens are "randomly" mixed
  → Ignore spatial and temporal dependencies

→ But how?



A Survey of Transformers, Lin et al, 2021

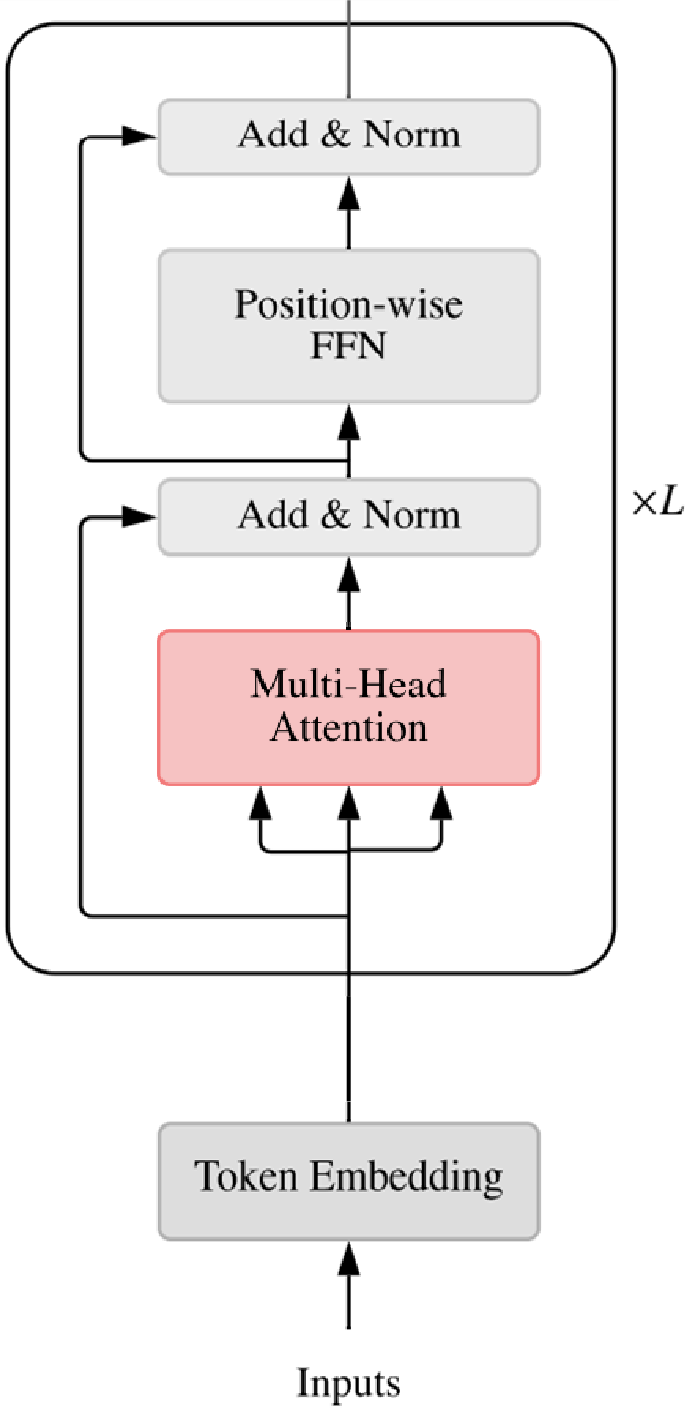
Token Mixing

→ Where the magic happens!

→ Tokens are ”randomly” mixed
  → Ignore spatial and temporal dependencies

→ But how?

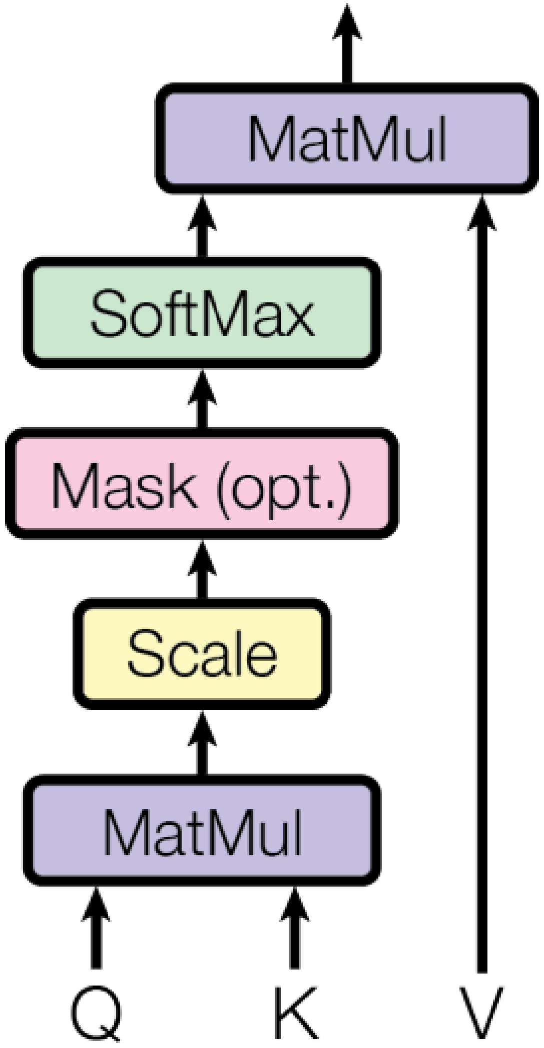
→ **Scaled Dot Product Attention** is the go-to choice



A Survey of Transformers, Lin et al, 2021

Scaled Dot Product Attention

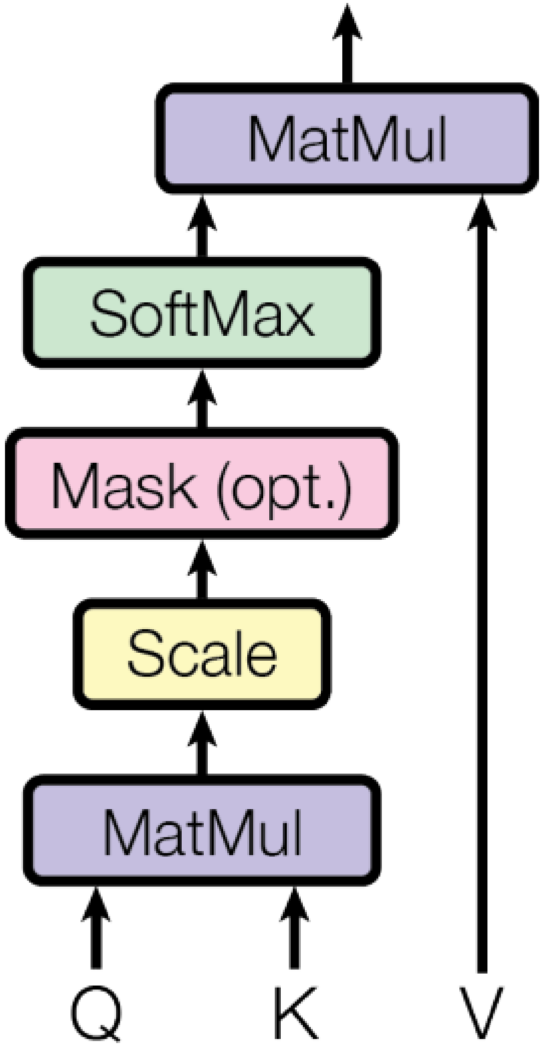
Scaled Dot Product Attention



Scaled Dot Product Attention

→ Input: Sequence, **X**, of M d-dimensional tokens

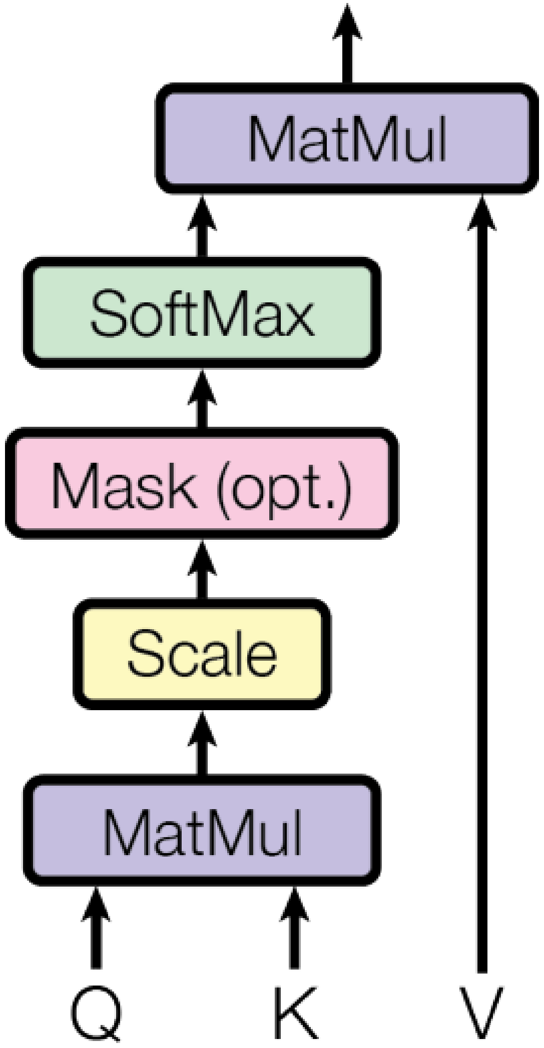
→ Create three transformed sets via independent linear transformations:



Scaled Dot Product Attention

→ Input: Sequence, **X**, of M d-dimensional tokens

→ Create three transformed sets via independent linear transformations:
  - → Queries (Q)
  - → Keys (K)
  - → Values (V)



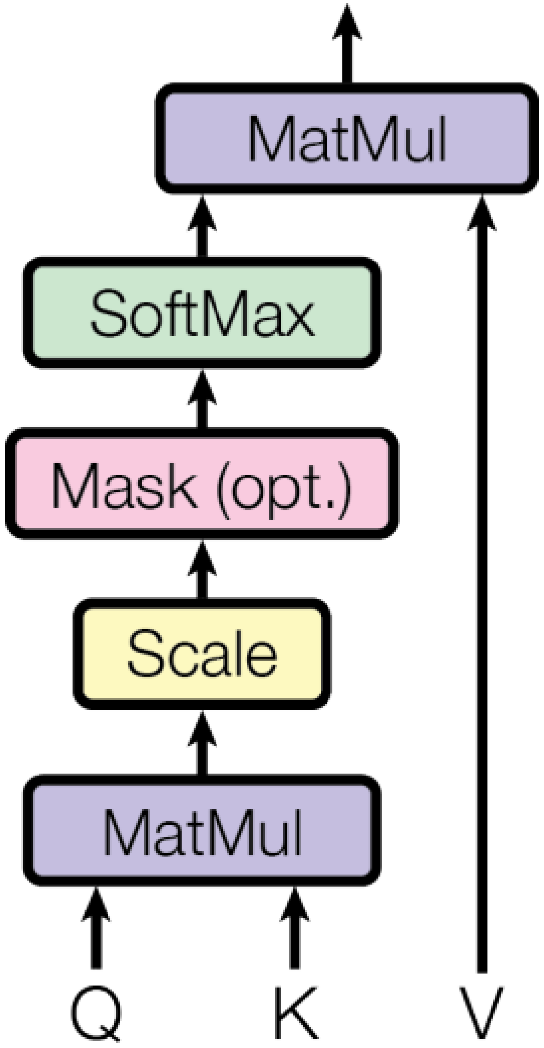
Attention is All You Need, Vaswani, 2017

Scaled Dot Product Attention

→ Input: Sequence, **X**, of M d-dimensional tokens

→ Create three transformed sets via independent linear transformations:
- → Queries (Q)
- → Keys (K)
- → Values (V)

→ If K, V, Q has same input, it is self-attention



Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

$$QK^T$$

Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

→ Scale by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K

$$\frac{QK^T}{\sqrt{d_k}}$$

Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

→ Scale by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K
  → This is done to alleviate vanishing gradients

$$\frac{QK^T}{\sqrt{d_k}}$$

Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

→ Scale by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K
  → This is done to alleviate vanishing gradients

→ Apply Softmax across the **rows**

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

→ Scale by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K
  → This is done to alleviate vanishing gradients

→ Apply Softmax across the **rows**

→ Multiply attention score matrix with Values (V)

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot Product Attention

→ Multiply Queries (Q) and Keys (K)
  → Gives M x M matrix, given M inputs
  → Optionally mask, if some tokens cannot attend to each other

→ Scale by $\frac{1}{\sqrt{d_k}}$, where d_k is the dimension of K
  → This is done to alleviate vanishing gradients

→ Apply Softmax across the **rows**

→ Multiply attention score matrix with Values (V)

→ But what if a token is important to multiple other tokens?

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Self-Attention (MHSA)

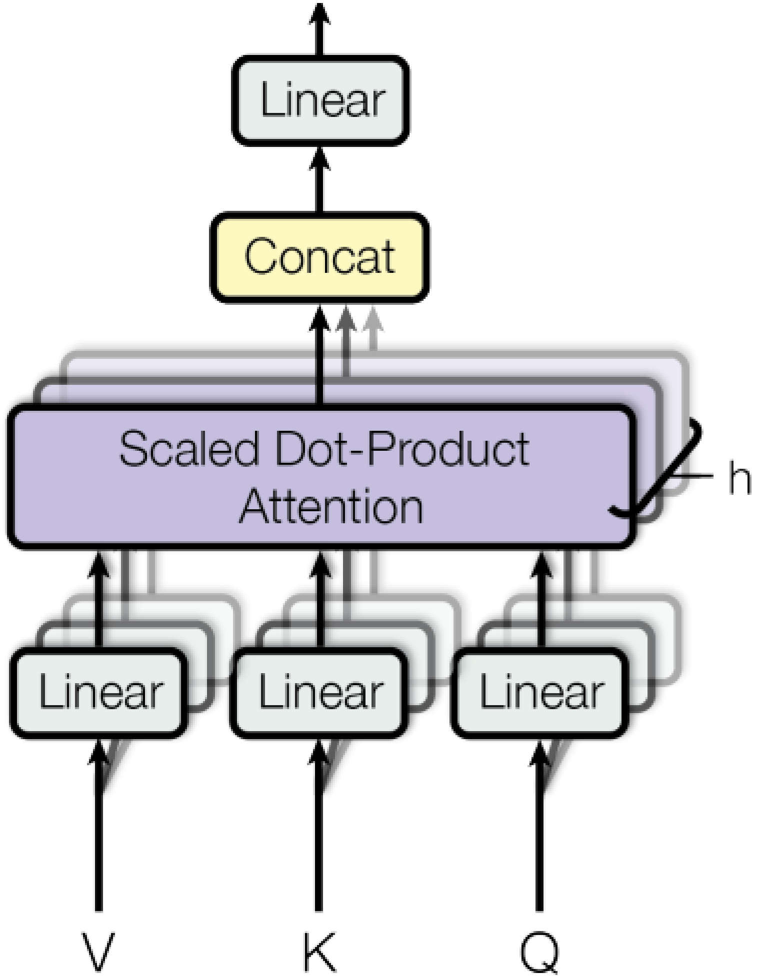
Multi-Head Self-Attention (MHSA)

→ Uses **H** attention heads

Multi-Head Self-Attention (MHSA)

→ Uses **H** attention heads

→ Each head receives a differently linearly transformed V, K, and Q
→ Concat output and apply linear transform



Multi-Head Self-Attention (MHSA)

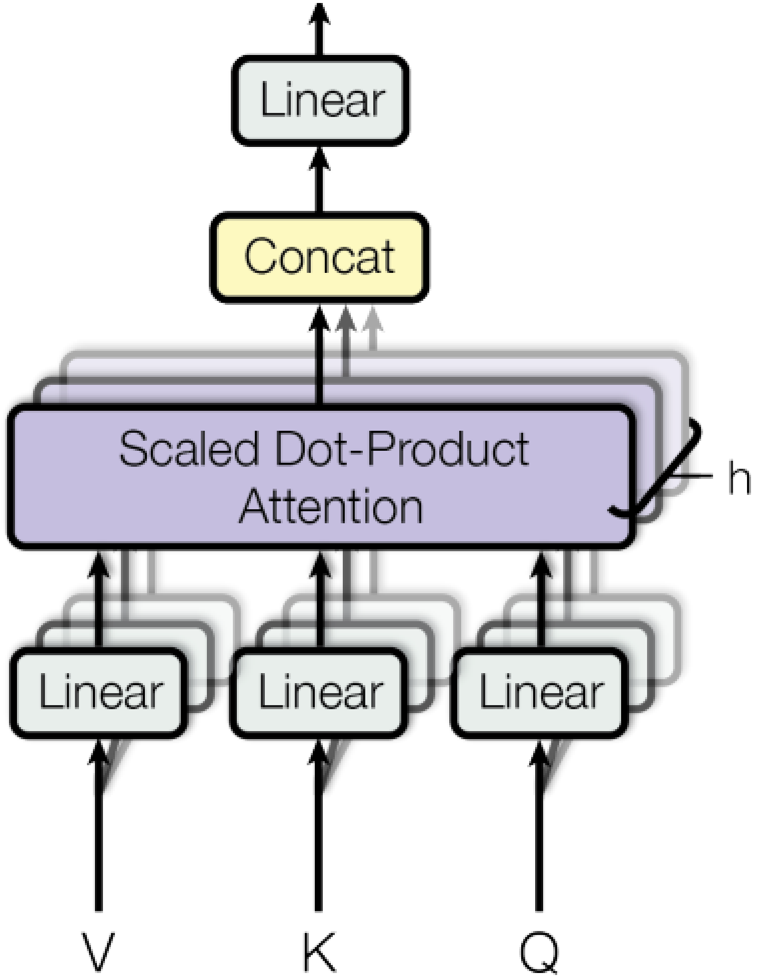
→ Uses **H** attention heads

→ Each head receives a differently linearly transformed V, K, and Q

→ Concat output and apply linear transform

→ To keep computation equal, we reduce dimensionality proportionally

→ $d_k^{new} = \frac{d_k^{old}}{h}$



Attention is All You Need, Vaswani, 2017

Altenative Token Mixing

Altenative Token Mixing

→ MHSA is a good starting point!
→ But its complexity is N^2

Altenative Token Mixing

→ MHSA is a good starting point!
→ But its complexity is N^2

→ Faster approaches can be achieved using
- → Linear attention
- → Sparse attention
- → Fourier transforms
- → …

sparse attention
linearized attention
query prototyping
memory compression
low rank self-attention
attention with prior
improved multi-head

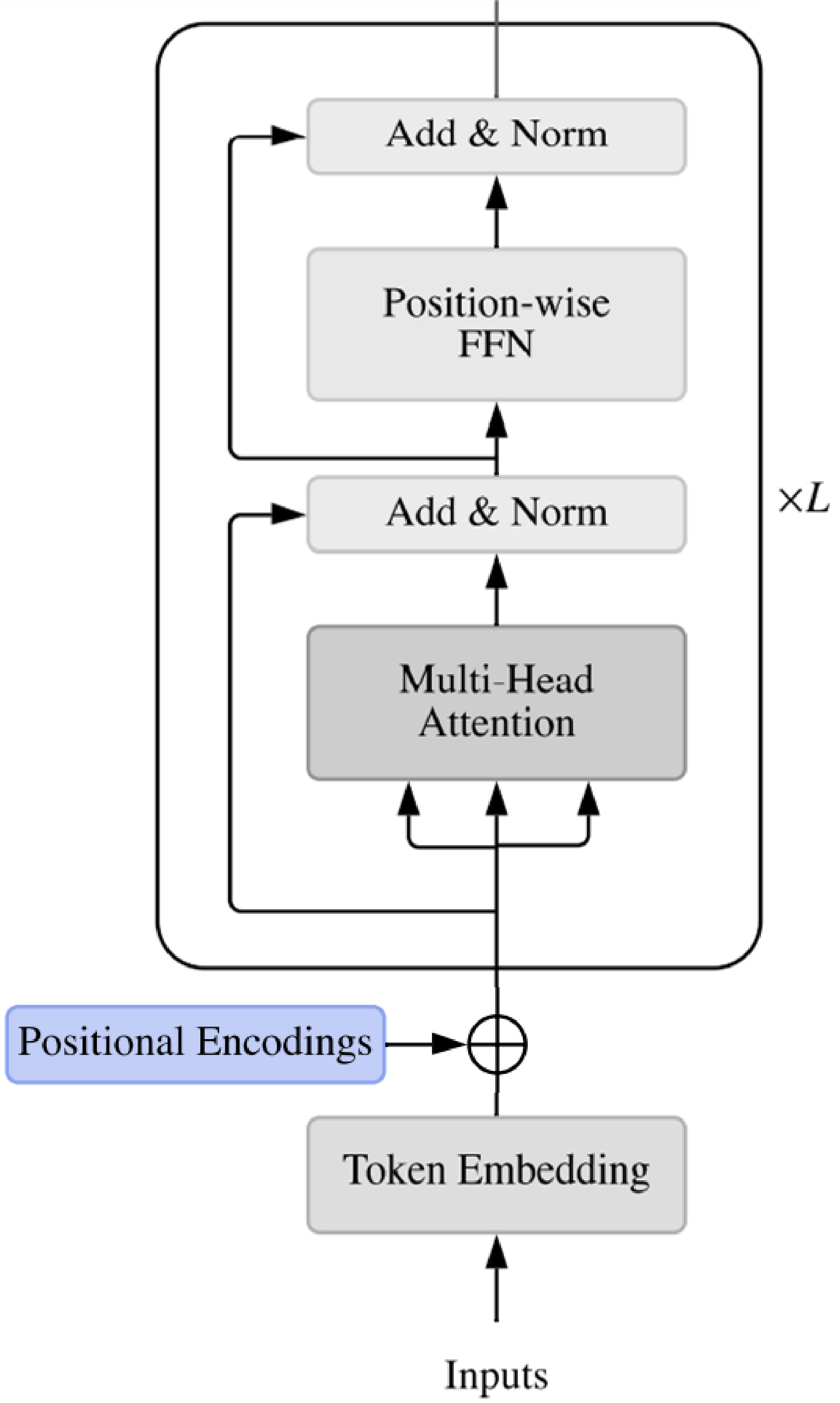
Positional Information

Positional Information

→ Transformers have no inert sense of order
→ Order is however important for most modalities (text, audio, images, ...)

Positional Information

→ Transformers have no inert sense of order
→ Order is however important for most modalities (text, audio, images, ...)

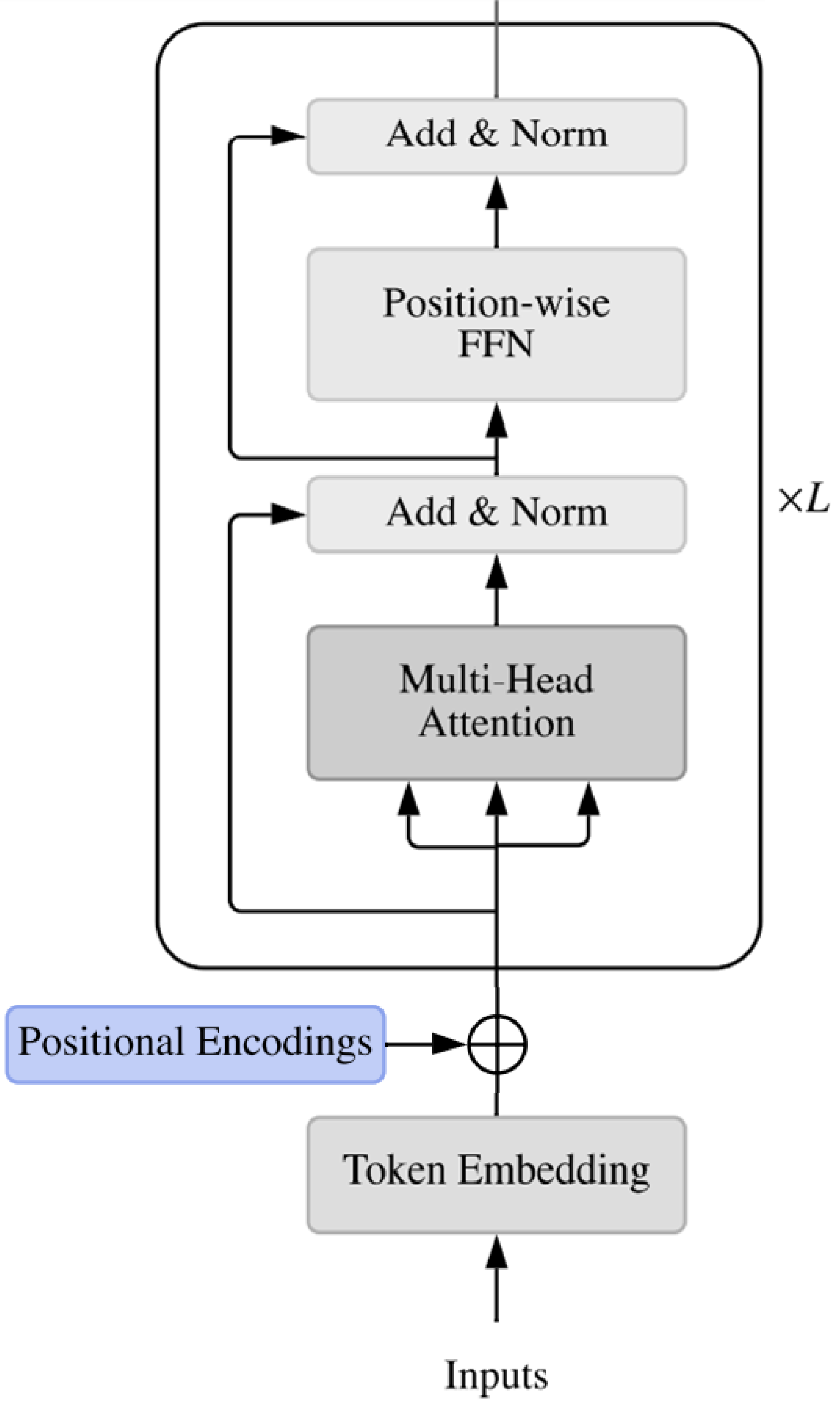


A Survey of Transformers, Lin et al, 2021

Positional Information

→ Transformers have no inert sense of order
→ Order is however important for most modalities (text, audio, images, ...)

→ Positional information can be either:
  - → A priori inserted (Positional Encoding)
  - → Learning using backprop (Positional Embedding)



A Survey of Transformers, Lin et al, 2021

Positional Encoding

Positional Encoding

→ Each token is represented using concatenated sine and cosine functions
→ Forms geometric progession from 2 Pi to 10000 * 2 Pi
→ This is static during training

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Positional Encoding

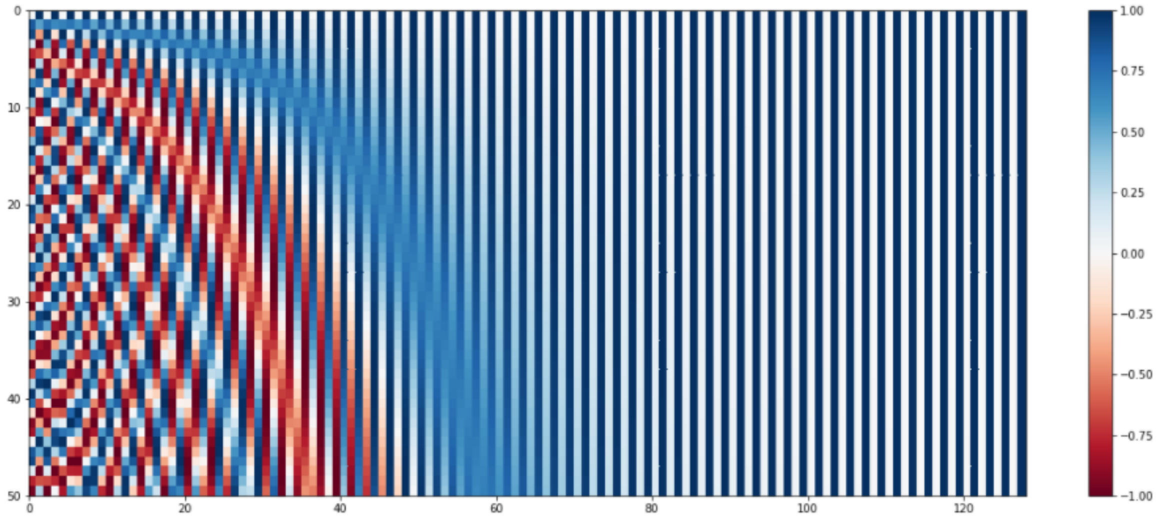
→ Each token is represented using concatenated sine and cosine functions
→ Forms geometric progession from 2 Pi to 10000 * 2 Pi
→ This is static during training

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

Positional Encoding

→ Each token is represented using concatenated sine and cosine functions
→ Forms geometric progession from 2 Pi to 10000 * 2 Pi
→ This is static during training

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



https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

How can this be used for Vision Tasks?

sdu.dk

#sdudk

How can this be used for Vision Tasks?

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.

In computer vision, however, convolutional architectures remain dominant (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention (Wang et al., 2018; Carion et al., 2020), some replacing the convolutions entirely (Ramachandran et al., 2019; Wang et al., 2020a). The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state of the art (Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020).

Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion.

When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases

[1]Fine-tuning code and pre-trained models are available at https://github.com/google-research/vision_transformer

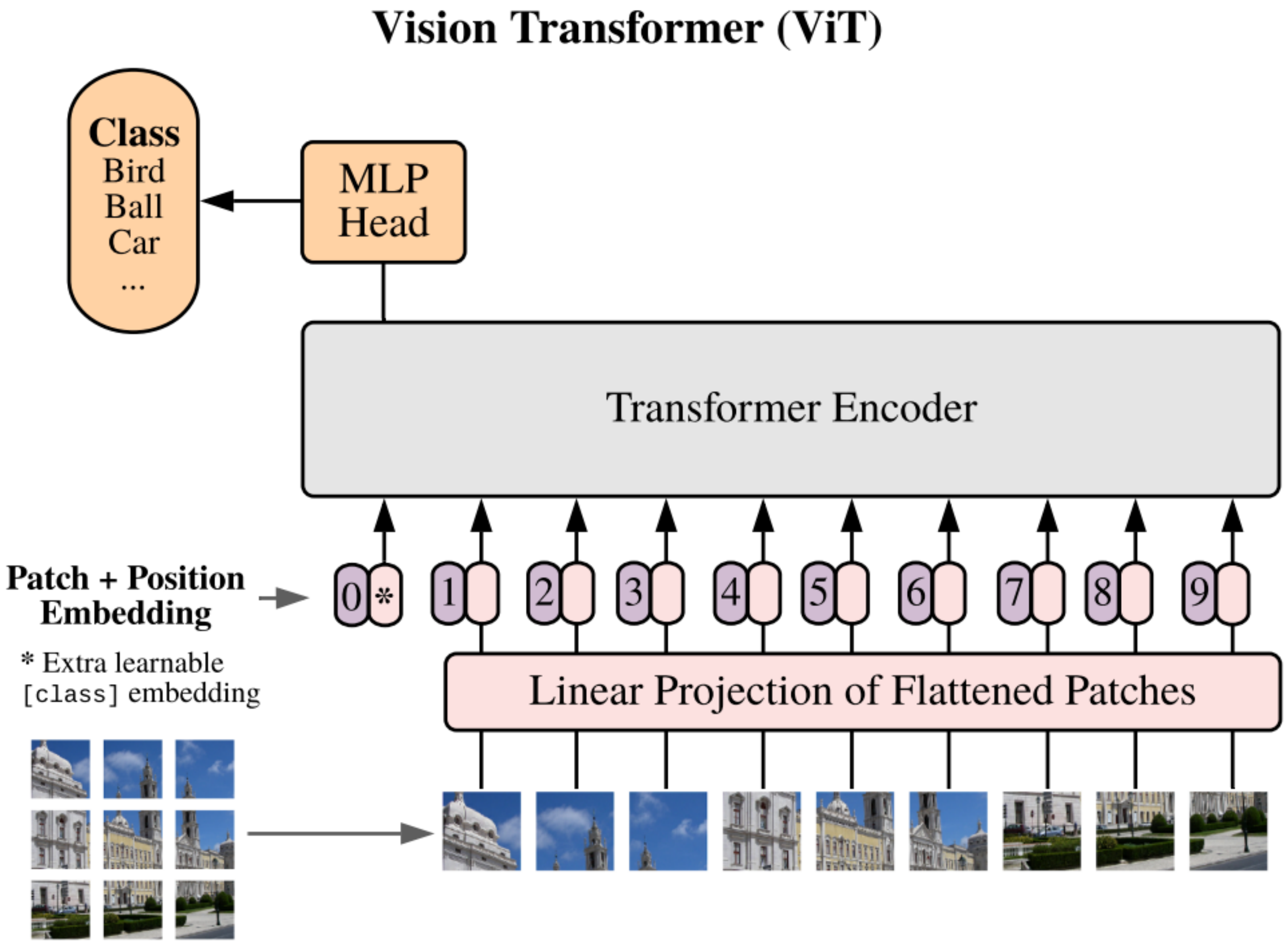
SDU

Vision Transformer (ViT)

→ Only uses transformer encoder blocks
→ Uses learned postional embeddings
→ Uses CLS token for representing entire image

Vision Transformer (ViT)

→ Only uses transformer encoder blocks
→ Uses learned postional embeddings
→ Uses CLS token for representing entire image

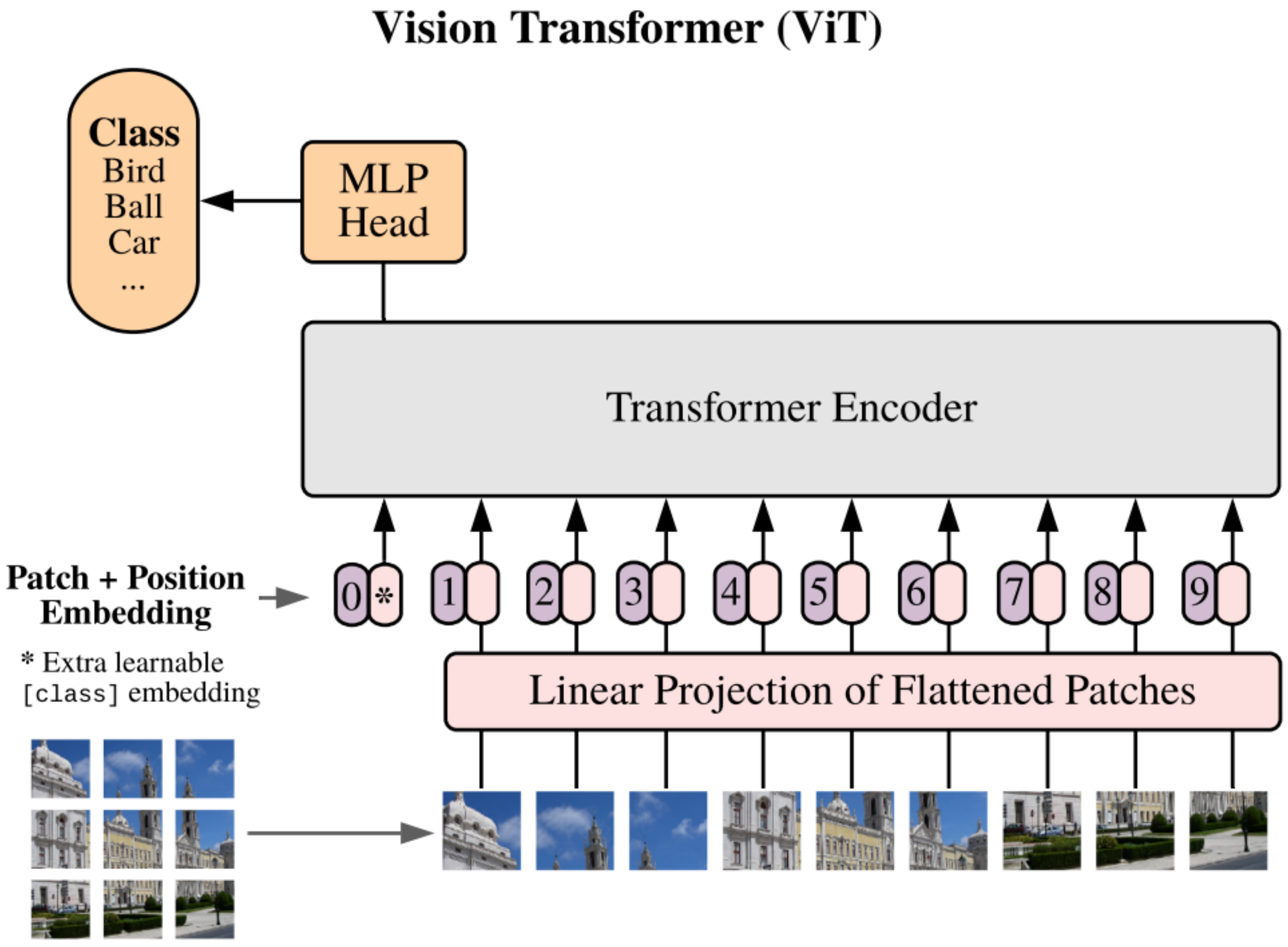


AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE, Dosovitskiy et al, 2021

Vision Transformer (ViT)

→ Only uses transformer encoder blocks
→ Uses learned postional embeddings
→ Uses CLS token for representing entire image

→ Tokenizes the image using non-overlapping convolution
→ Smaller kernel -> more tokens -> longer ”sequence”



AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE, Dosovitskiy et al, 2021

How to train a Vision Transformer

Original Training Approach

Original Training Approach

→ Trained using supervised learning

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE,
Dosovitskiy et al, 2021

Original Training Approach

→ Trained using supervised learning

→ Large amount of data needed due to no inductive bias
- → JFT-300
- → ImageNet-21K

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE, Dosovitskiy et al, 2021

Original Training Approach

→ Trained using supervised learning

→ Large amount of data needed due to no inductive bias
  → JFT-300
  → ImageNet-21K

→ Requires a lot of compute
  → Hidden VRAM requirement due to tokens
  → Smaller conv layer (tokenizer) leads to more tokens

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE,
Dosovitskiy et al, 2021

Original Training Approach

- → Trained using supervised learning
- → Large amount of data needed due to no inductive bias
  - → JFT-300
  - → ImageNet-21K
- → Requires a lot of compute
  - → Hidden VRAM requirement due to tokens
  - → Smaller conv layer (tokenizer) leads to more tokens
- → Achieved SOTA on ImageNet-1K and many other datasets

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE,
Dosovitskiy et al, 2021

Data-efficient image Transformers (DeiT)

Data-efficient image Transformers (DeiT)

→ Proposes learning a ViT by distillation

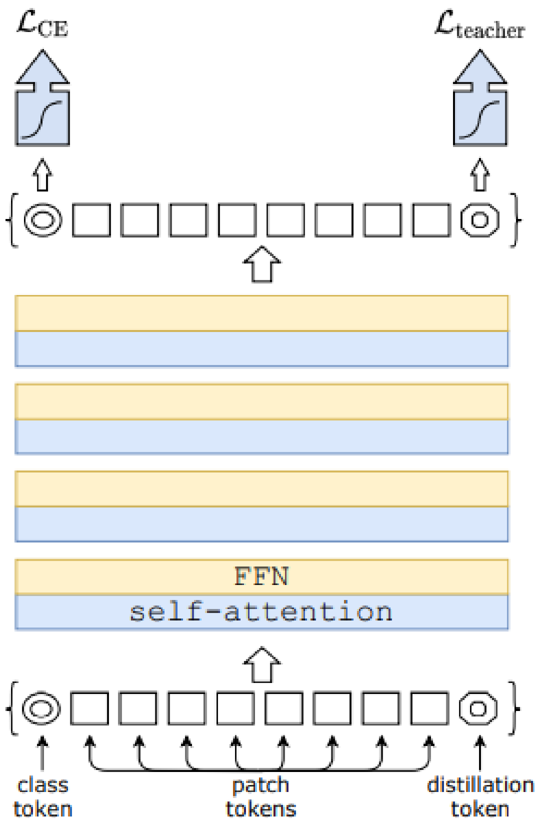
Data-efficient image Transformers (DeiT)

→ Proposes learning a ViT by distillation
- → Distillation = Train model to match predictions of already trained (larger) model

Data-efficient image Transformers (DeiT)

→ Proposes learning a ViT by distillation
  → Distillation = Train model to match predictions of already trained (larger) model

→ Introduces a dedicated distillation token



Training data-efficient image transformers & distillation through attention, Touvron et al, 2021

Data-efficient image Transformers (DeiT)

→ Proposes learning a ViT by distillation

→ Distillation = Train model to match predictions of already trained (larger) model

→ Introduces a dedicated distillation token

→ Can be trained in 3-4 days using 2 GPUs

→ High performance using just ImageNet-1K

Model	ViT model	embedding dimension	#heads	#layers	#params	training resolution	throughput (im/sec)
DeiT-Ti	N/A	192	3	12	5M	224	2536
DeiT-S	N/A	384	6	12	22M	224	940
DeiT-B	ViT-B	768	12	12	86M	224	292

Teacher Models	acc.	Student: DeiT-B ⚗ pretrain	↑384
DeiT-B	81.8	81.9	83.1
RegNetY-4GF	80.0	82.7	83.6
RegNetY-8GF	81.7	82.7	83.8
RegNetY-12GF	82.4	83.1	84.1
RegNetY-16GF	82.9	83.1	84.2

Training data-efficient image transformers & distillation through attention, Touvron et al, 2021

Data-efficient image Transformers (DeiT)

- → Proposes learning a ViT by distillation
  - → Distillation = Train model to match predictions of already trained (larger) model
- → Introduces a dedicated distillation token
- → Can be trained in 3-4 days using 2 GPUs
- → High performance using just ImageNet-1K
- → Outperforms all teachers, as well as ViT-B
- → Was not tested on larger variations of VIT

Model	ViT model	embedding dimension	#heads	#layers	#params	training resolution	throughput (im/sec)
DeiT-Ti	N/A	192	3	12	5M	224	2536
DeiT-S	N/A	384	6	12	22M	224	940
DeiT-B	ViT-B	768	12	12	86M	224	292

Teacher Models	acc.	Student: DeiT-B ⚗ pretrain	↑384
DeiT-B	81.8	81.9	83.1
RegNetY-4GF	80.0	82.7	83.6
RegNetY-8GF	81.7	82.7	83.8
RegNetY-12GF	82.4	83.1	84.1
RegNetY-16GF	82.9	83.1	84.2

Training data-efficient image transformers & distillation through attention, Touvron et al, 2021

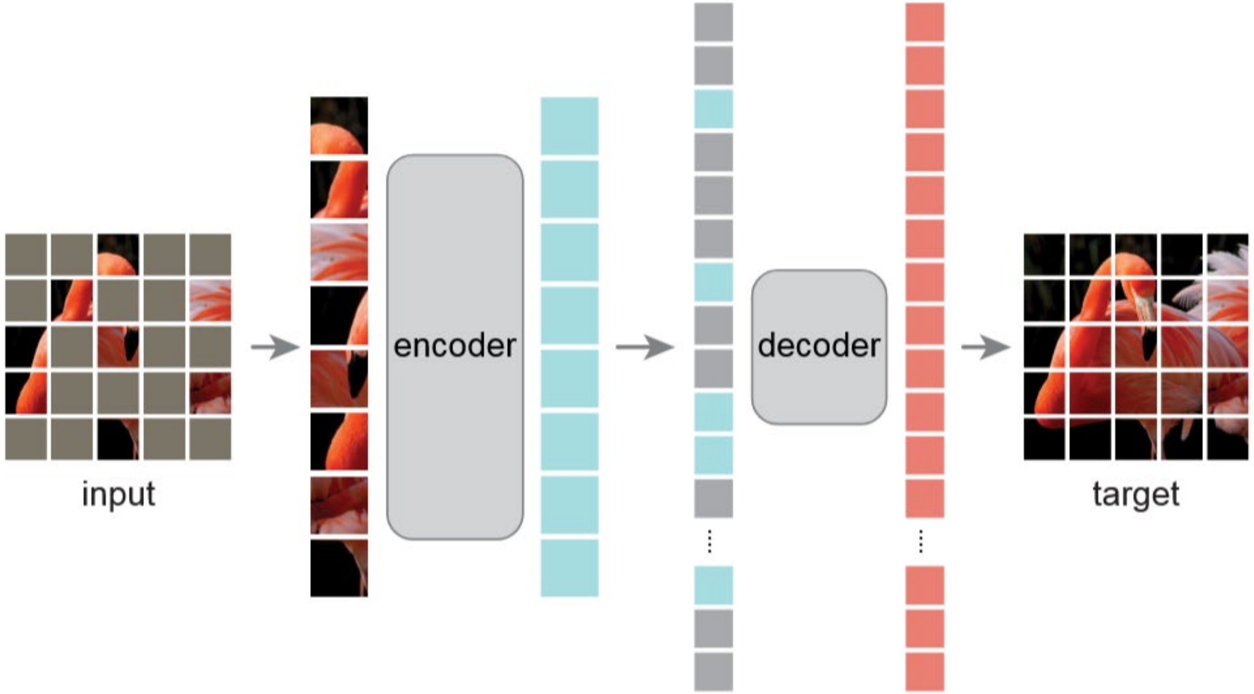
Self-Supervised Pretraining

→ ViTs can be efficiently trained via different SSL approaches

Self-Supervised Pretraining

→ ViTs can be efficiently trained via different SSL approaches

→ Masked Autoencoders



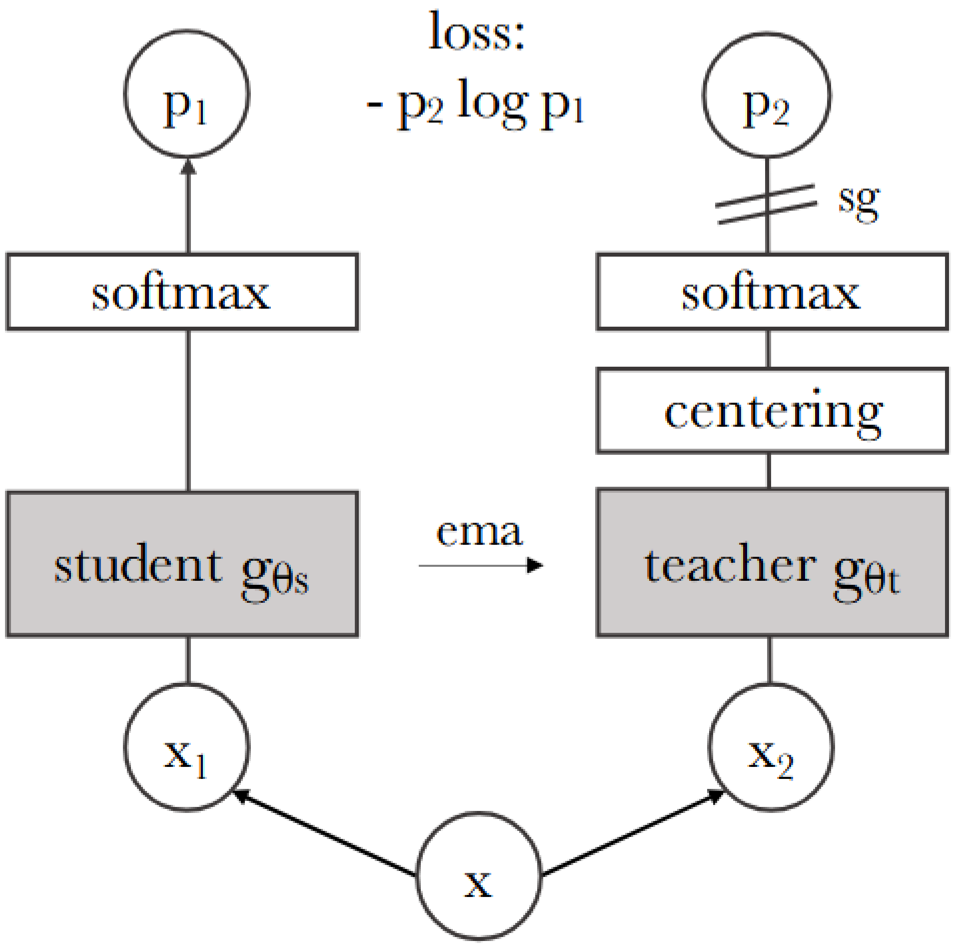
Masked Autoencoders Are Scalable Vision Learner, He et al., 2021

Self-Supervised Pretraining

→ ViTs can be efficiently trained via different SSL approaches

→ Masked Autoencoders

→ DINO



Emerging Properties in Self-Supervised Vision Transformers, Caron et al., 2021

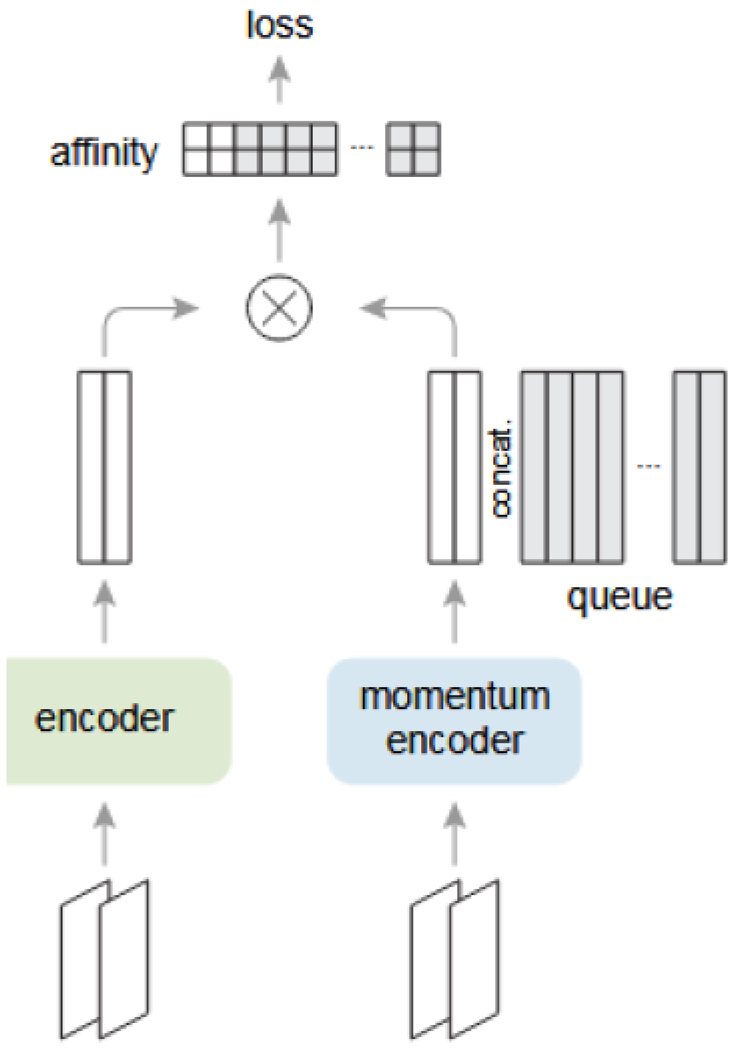
Self-Supervised Pretraining

→ ViTs can be efficiently trained via different SSL approaches

→ Masked Autoencoders

→ DINO

→ Momentum Contrast



Improved Baselines with Momentum Contrastive Learning, Chen et al, 2020

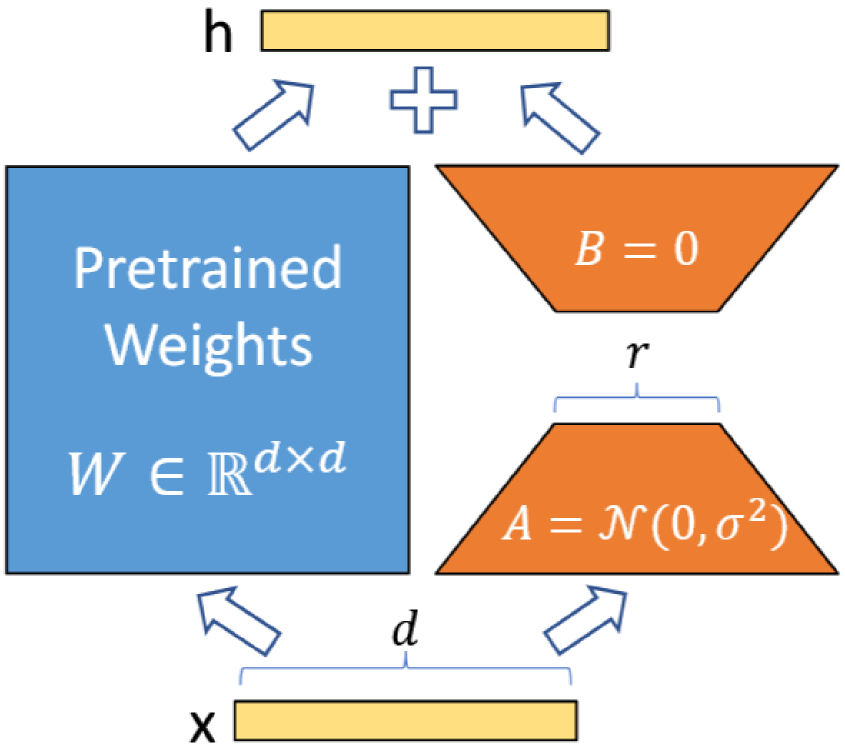
Parameter Efficient Fine-Tuning

→ ViTs can also be efficiently fine-tuned by intelligently utilizing the Transformer components

Parameter Efficient Fine-Tuning

→ ViTs can also be efficiently fine-tuned by intelligently utilizing the Transformer components

→ Low-Rank Adapters (LoRA)



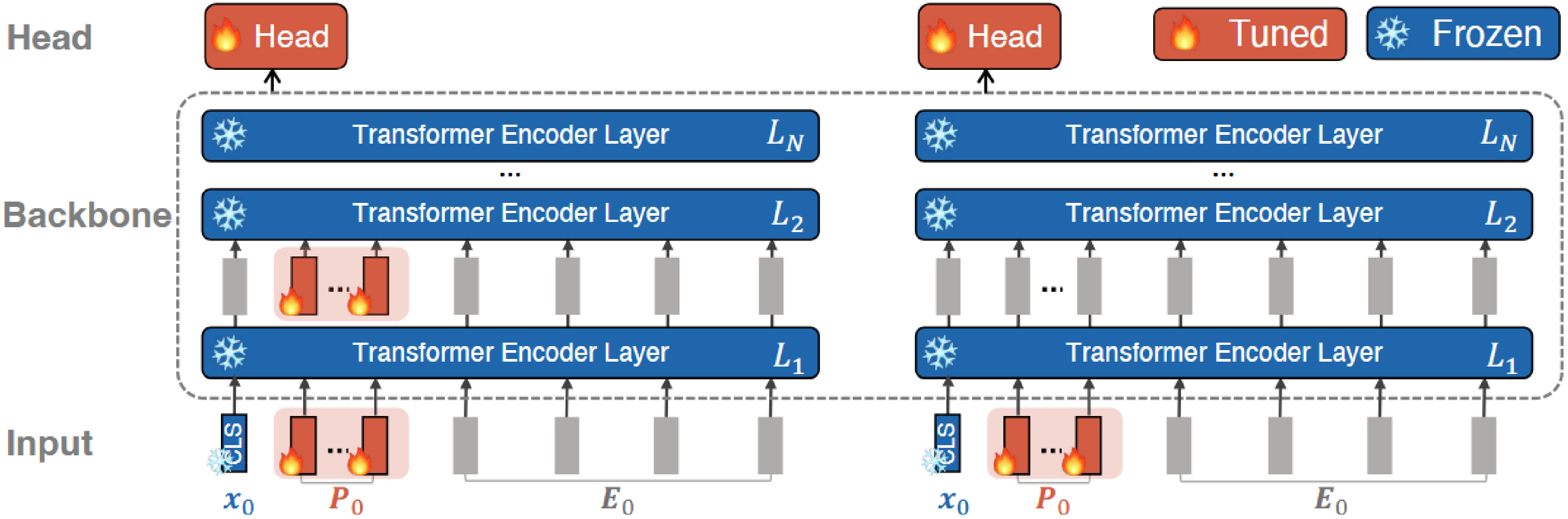
LoRA: Low-Rank Adaptation of Large Language Models, Hu et al., 2021

Parameter Efficient Fine-Tuning

→ ViTs can also be efficiently fine-tuned by intelligently utilizing the Transformer components

→ Low-Rank Adapters (LoRA)

→ Visual Prompt Tuning



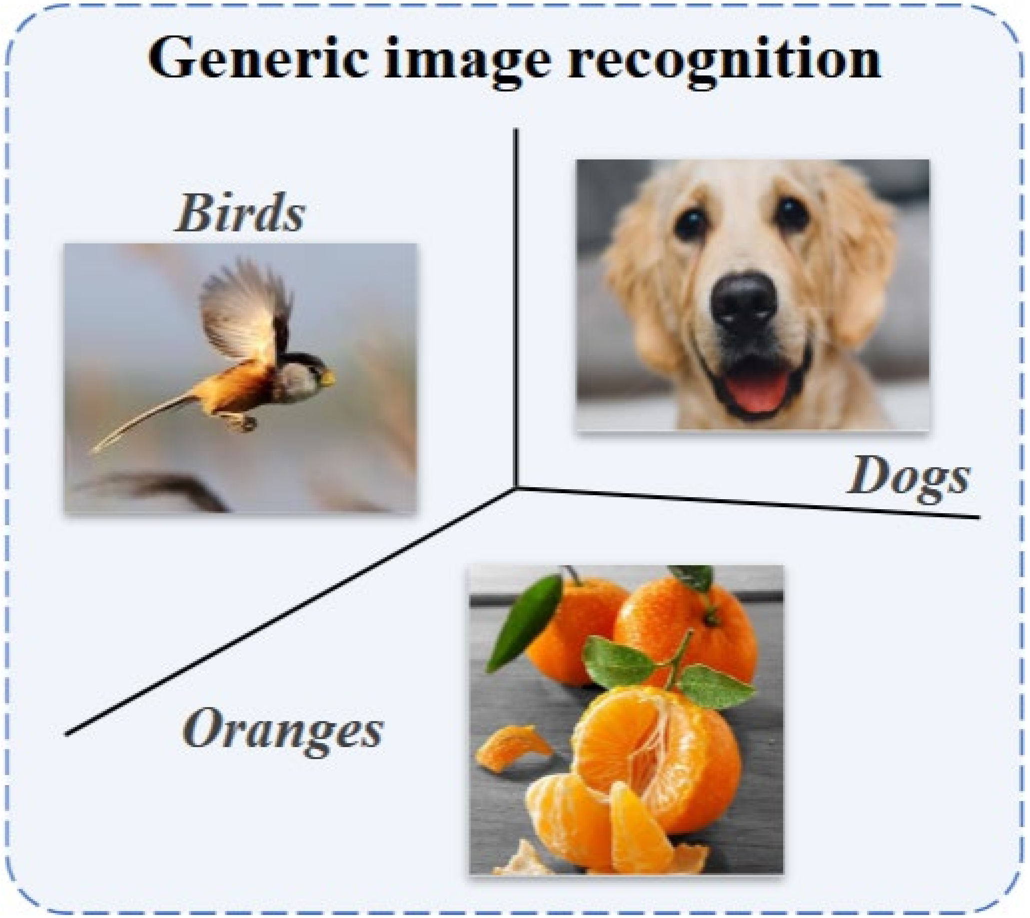
(a) Visual-Prompt Tuning: Deep (b) Visual-Prompt Tuning: Shallow

Visual Prompt Tuning, Jia et al., 2022

ViTs for Fine-Grained Tasks

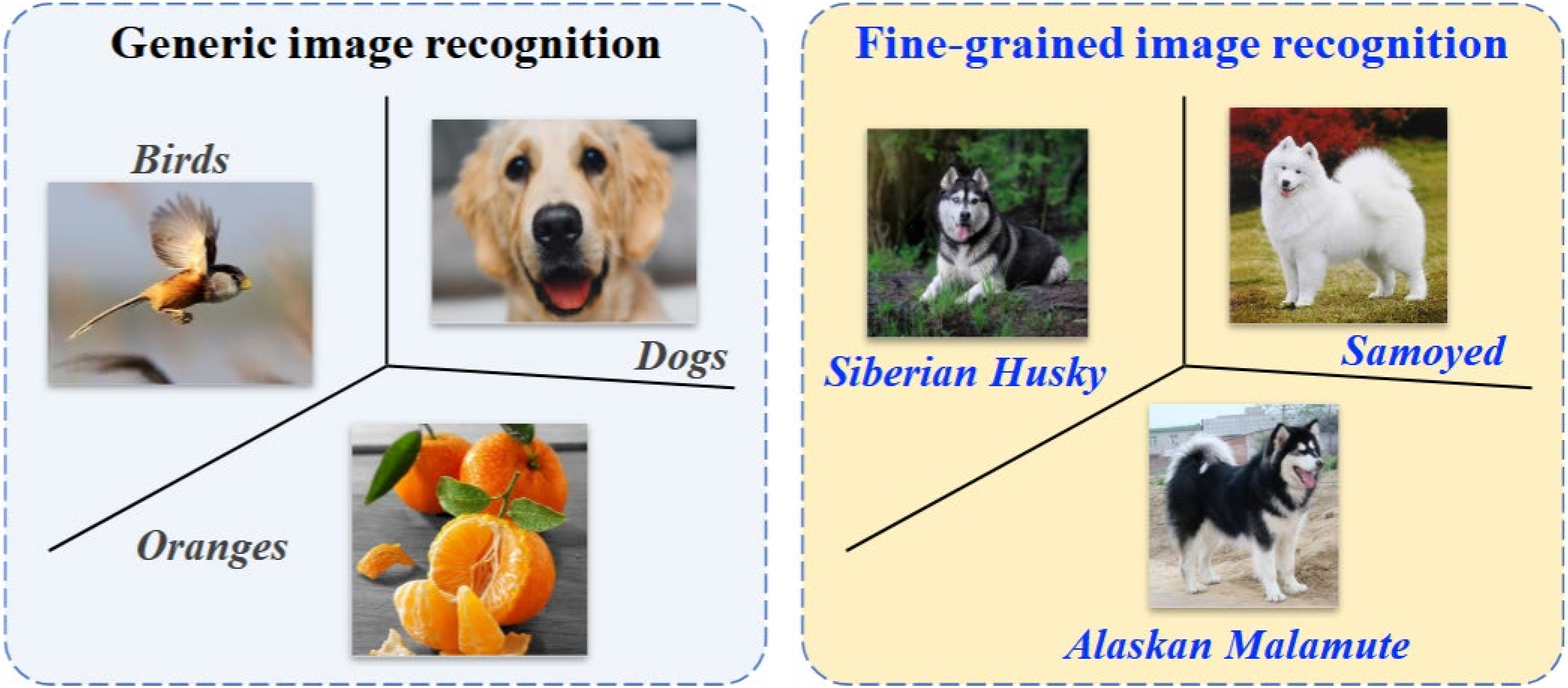
Continuum

Continuum



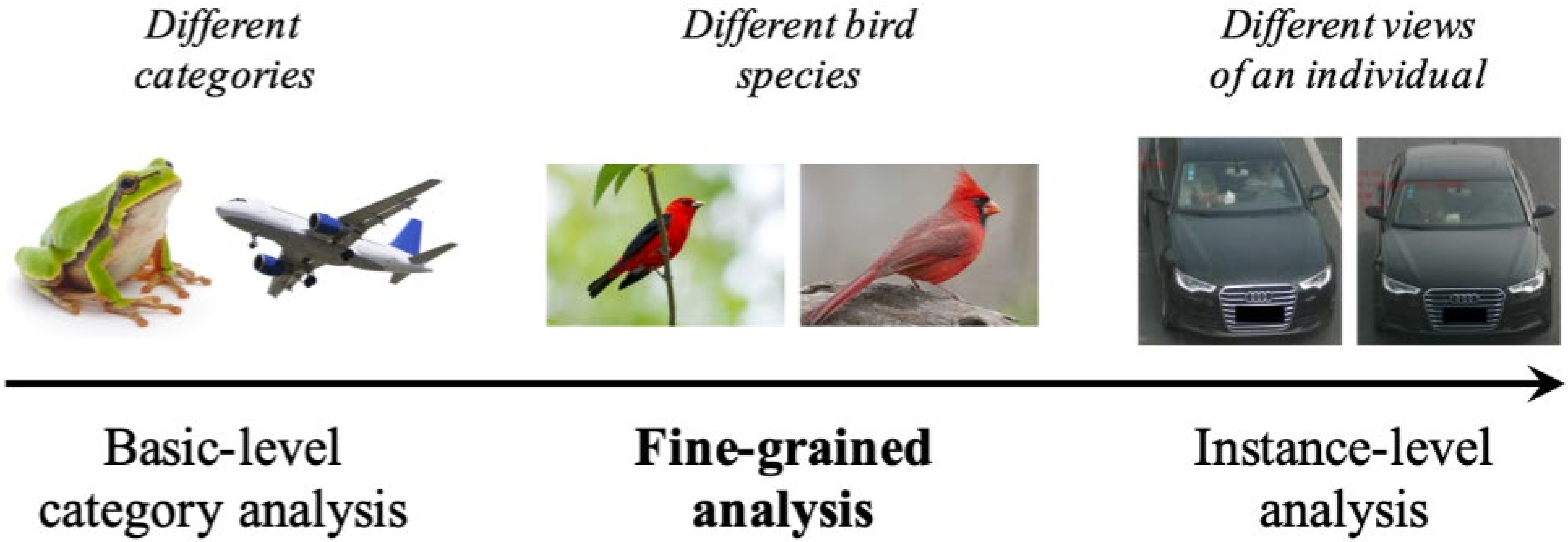
Fine-Grained Image Analysis with Deep Learning: A Survey, Wei et al., 2021

Continuum



Fine-Grained Image Analysis with Deep Learning: A Survey, Wei et al., 2021

Continuum



Fine-Grained Image Analysis with Deep Learning: A Survey, Wei et al., 2021

TransFG: A ViT for Fine-Grained Recognition

TransFG: A Transformer Architecture for Fine-Grained Recognition

Ju He[1] Jie-Neng Chen[1] Shuai Liu[2]
Adam Kortylewski[1] Cheng Yang[2] Yutong Bai[1] Changhu Wang[2]
[1]Johns Hopkins University [2]ByteDance Inc.

Abstract

Fine-grained visual classification (FGVC) which aims at recognizing objects from subcategories is a very challenging task due to the inherently subtle inter-class differences. Most existing works mainly tackle this problem by reusing the backbone network to extract features of detected discriminative regions. However, this strategy inevitably complicates the pipeline and pushes the proposed regions to contain most parts of the objects thus fails to locate the really important parts. Recently, vision transformer (ViT) shows its strong performance in the traditional classification task. The self-attention mechanism of the transformer links every patch token to the classification token. In this work, we first evaluate the effectiveness of the ViT framework in the fine-grained recognition setting. Then motivated by the strength of the attention link can be intuitively considered as an indicator of the importance of tokens, we further propose a novel Part Selection Module that can be applied to most of the transformer architectures where we integrate all raw attention weights of the transformer into an attention map for guiding the network to effectively and accurately select discriminative image patches and compute their relations. A contrastive loss is applied to enlarge the distance between feature representations of confusing classes. We name the augmented transformer-based model TransFG and demonstrate the value of it by conducting experiments on five popular fine-grained benchmarks where we achieve state-of-the-art performance. Qualitative results are presented for better understanding of our model.

Introduction

Fine-grained visual classification aims at classifying subclasses of a given object category, e.g., subcategories of birds (Wah et al. 2011; Van Horn et al. 2015), cars (Krause et al. 2013), aircrafts (Maji et al. 2013). It has long been considered as a very challenging task due to the small interclass variations and large intra-class variations along with the deficiency of annotated data, especially for the long-tailed classes. Benefiting from the progress of deep neural networks (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; He et al. 2016), the performance of FGVC has obtained a steady progress in recent years. To avoid labor-intensive part annotation, the community currently focuses on weakly-supervised FGVC with

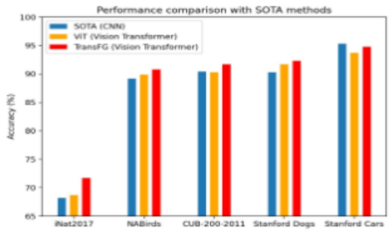


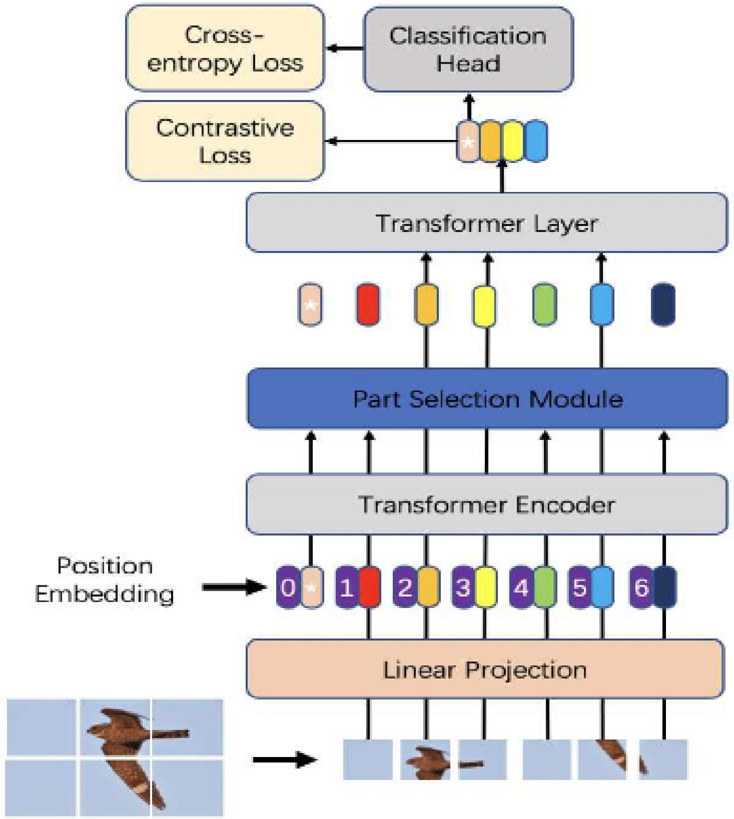
Figure 1: An overview of performance comparison of ViT and TransFG with state-of-the-art methods CNNs on five datasets. We achieve state-of-the-art performance on most datasets while performing a little bit worse on Stanford Cars possibly due to the more regular and simpler car shapes.

only image-level labels. Methods now can be roughly classified into two categories, i.e., localization methods and feature-encoding methods. Compared to feature-encoding methods, the localization methods have the advantage that they explicitly capture the subtle differences among subclasses which is more interpretable and yields better results.

Early works in localization methods rely on the annotations of parts to locate discriminative regions while recent works (Ge, Lin, and Yu 2019a; Liu et al. 2020; Ding et al. 2019) mainly adopt region proposal networks (RPN) to propose bounding boxes which contain the discriminative regions. After obtaining the selected image regions, they are resized into a predefined size and forwarded through the backbone network again to acquire informative local features. A typical strategy is to use these local features for classification individually and adopt a rank loss (Chen et al. 2009) to maintain consistency between the quality of bounding boxes and their final probability output. However, this mechanism ignores the relation between selected regions and thus inevitably encourages the RPN to propose large bounding boxes that contain most parts of the objects which fails to locate the really important regions. Sometimes these bounding boxes can even contain large areas of background

TransFG: A ViT for Fine-Grained Recognition

→ Adapt the ViT to perform part selection to detect discriminative parts



TransFG: A Transformer Architecture for Fine-grained Recognition, He et al., 2022

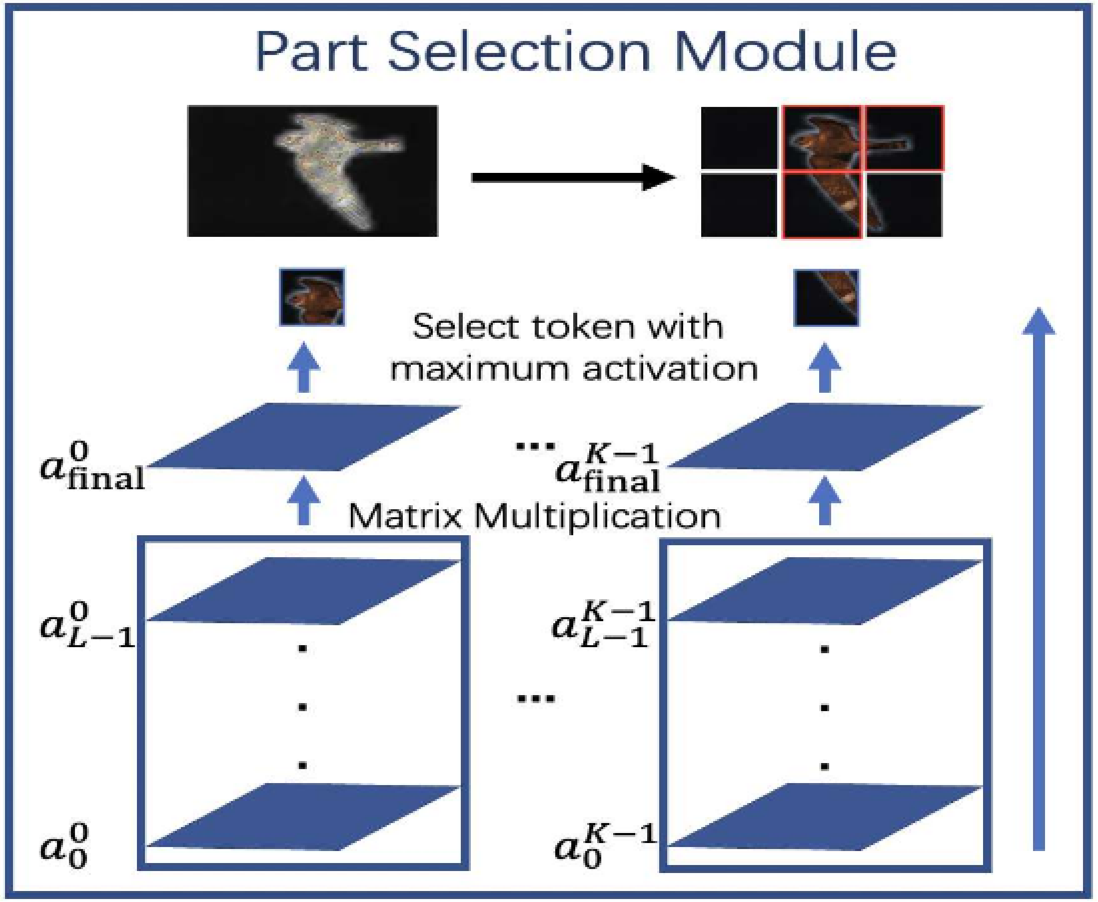
TransFG: A ViT for Fine-Grained Recognition

→ Adapt the ViT to perform part selection to detect discriminative parts

→ Aggregate attention maps of the first L-1 layers

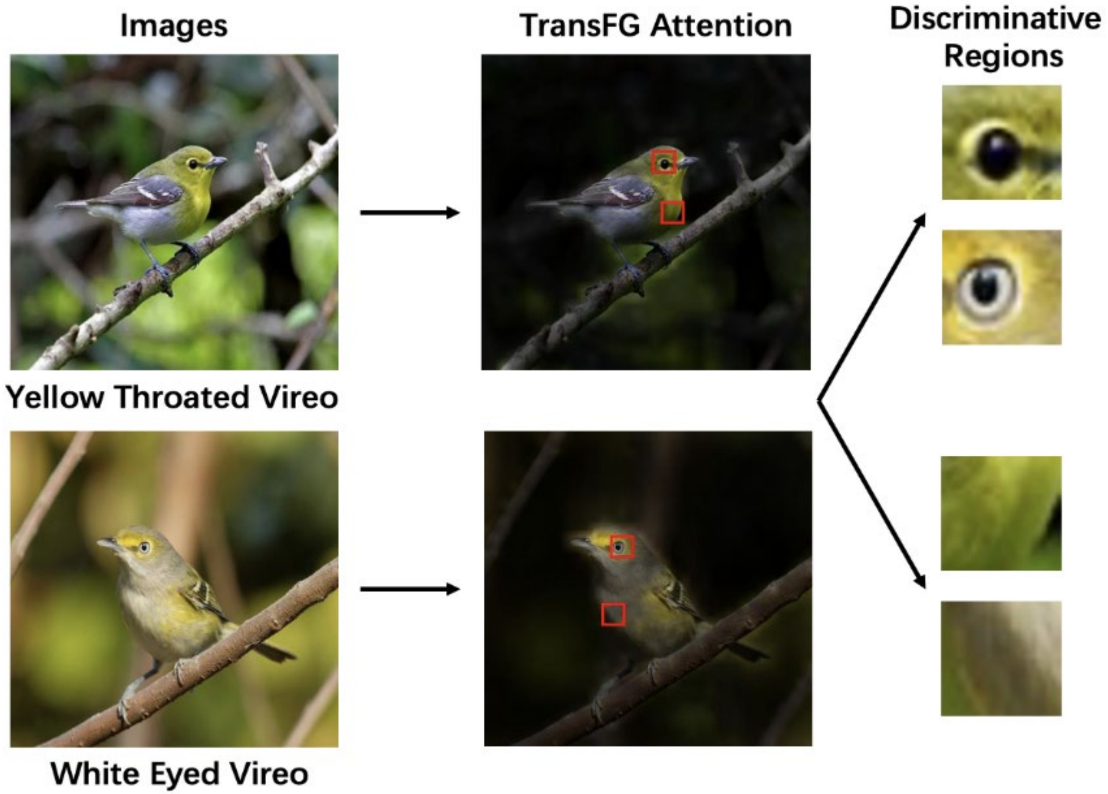
→ Only keep the most attended to token per attention head

→ Pass these to the final Transformer layer



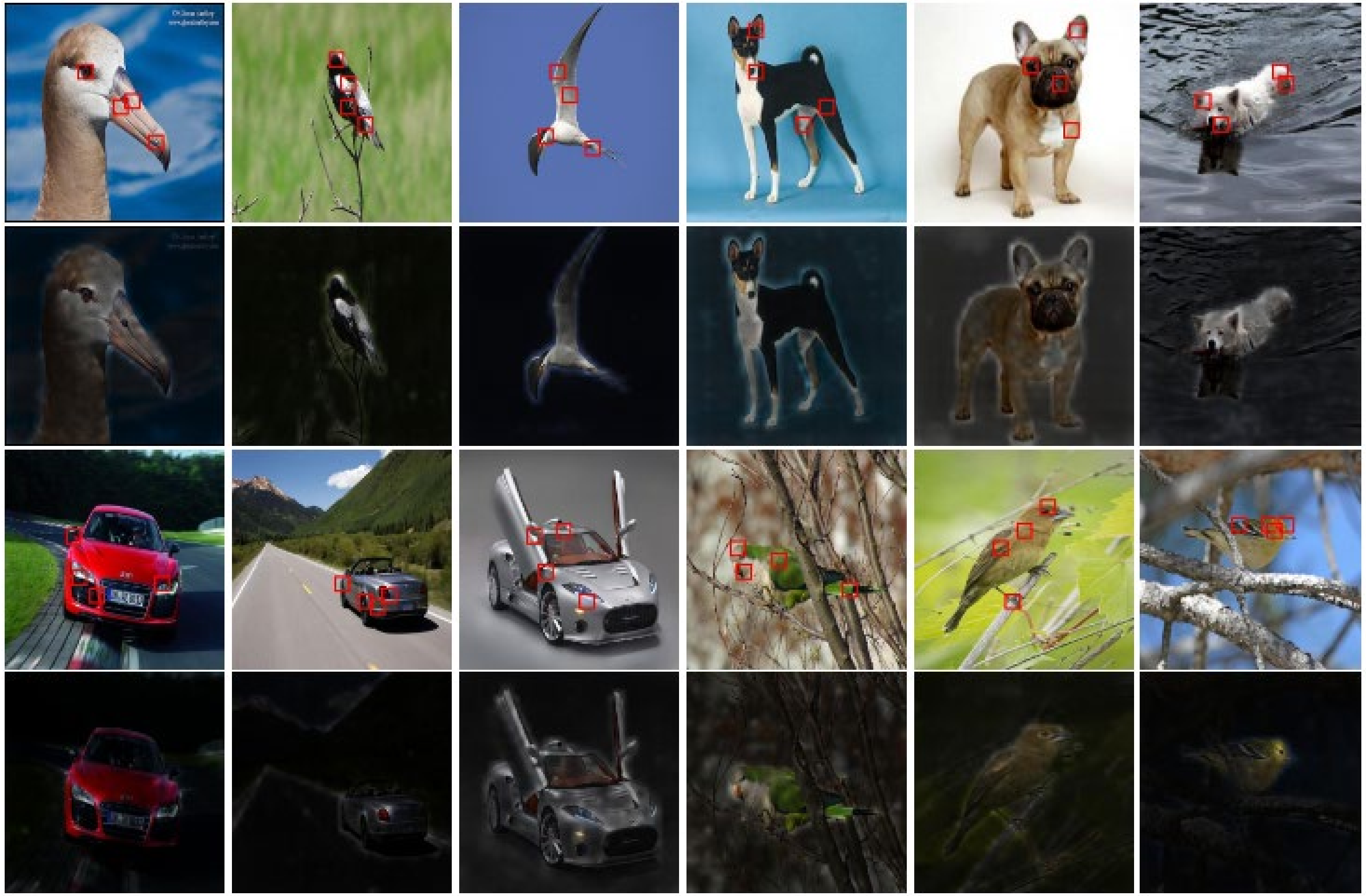
TransFG: A Transformer Architecture for Fine-grained Recognition, He et al., 2022

TransFG: A ViT for Fine-Grained Recognition



TransFG: A Transformer Architecture for Fine-grained Recognition, He et al., 2022

TransFG: A ViT for Fine-Grained Recognition



TransFG: A Transformer Architecture for Fine-grained Recognition, He et al., 2022

Grafit: Learning Coarse-to-Fine training

→ Proposes a setup for training fine-grained classifiers from coarse labels

→ Not specific to ViTs, but of great interest for taxonomic classification

Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

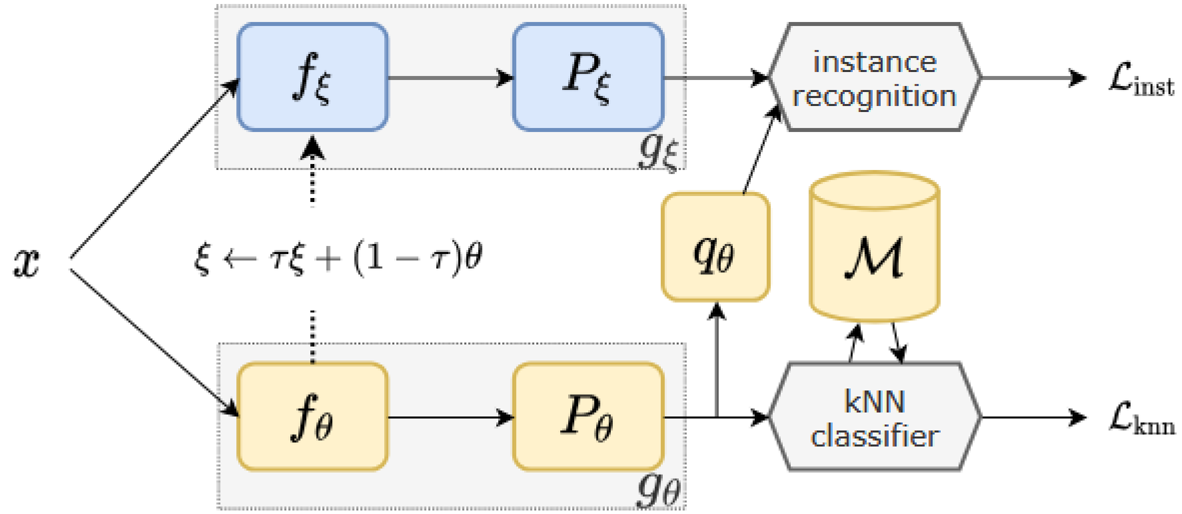
Grafit: Learning Coarse-to-Fine training

→ Utilizes two losses: Instance and kNN classification loss

Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

Grafit: Learning Coarse-to-Fine training

→ Utilizes two losses: Instance and kNN classification loss



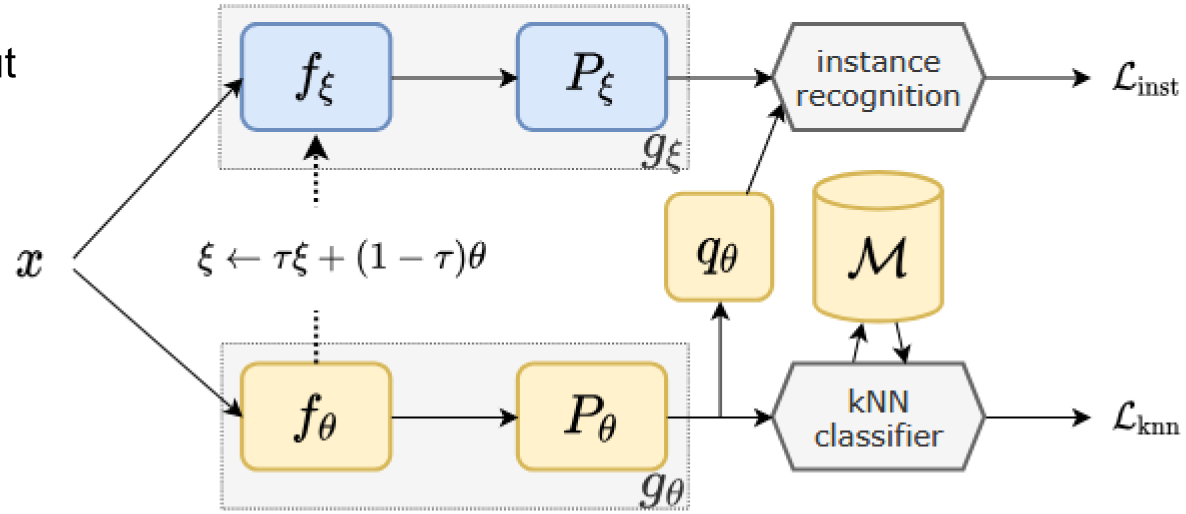
Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

Grafit: Learning Coarse-to-Fine training

→ Utilizes two losses: Instance and kNN classification loss

→ Instance Loss: Learn how to align different views of input

$$\mathcal{L}_{\text{inst}}(x) = -\sum_{1 \le i \ne j \le T} \frac{\cos\left(q_\theta \circ g_\theta(t_i(x)).g_\xi(t_j(x))\right)}{T(T-1)}$$



Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

Grafit: Learning Coarse-to-Fine training

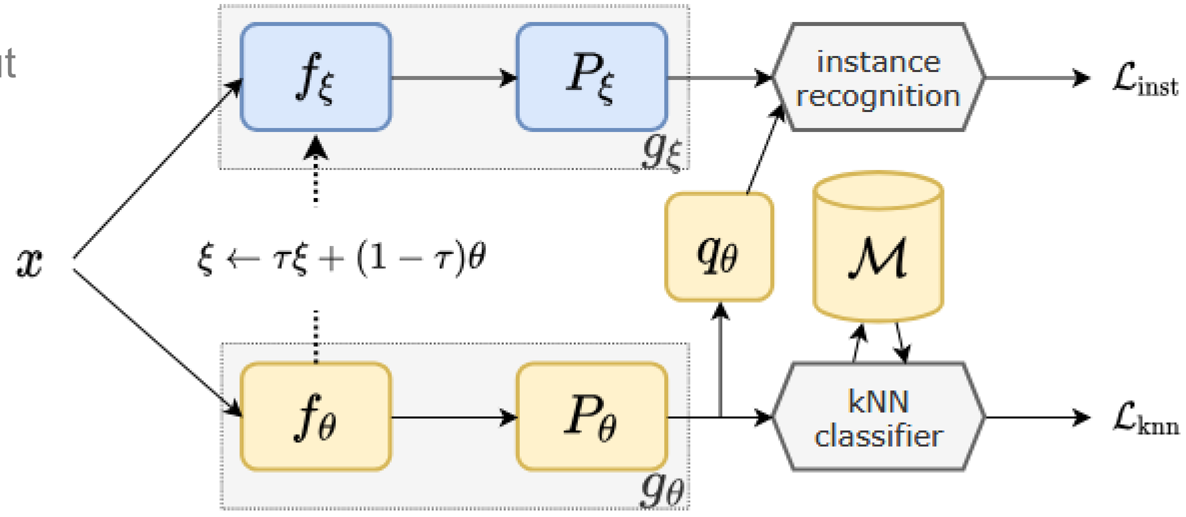
→ Utilizes two losses: Instance and kNN classification loss

→ Instance Loss: Learn how to align different views of input

→ kNN classifier: Learn how to perform kNN classification
Trains for generalization of new classes

$$\mathcal{L}_{\text{knn}}(x_i, y_i) = -\log \sum_{j, y_j = y_i, j \neq i} p_{i,j}.$$

$$p_{i,j} \propto \exp\left(\cos(g_\theta(x_i), g_\theta(x_j))/\sigma\right)$$



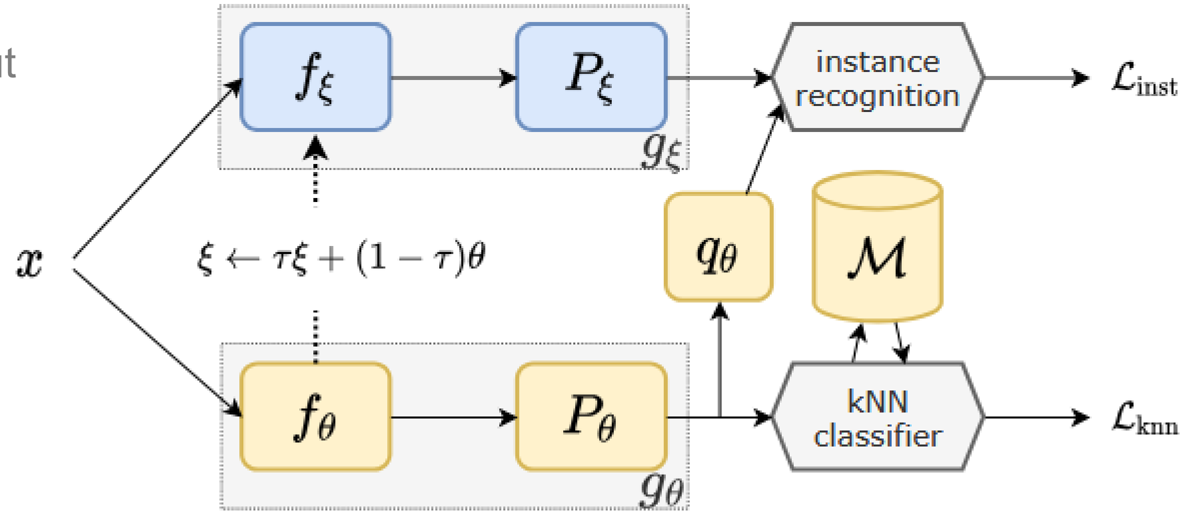
Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

Grafit: Learning Coarse-to-Fine training

→ Utilizes two losses: Instance and kNN classification loss

→ Instance Loss: Learn how to align different views of input

→ kNN classifier: Learn how to perform kNN classification
Trains for generalization of new classes



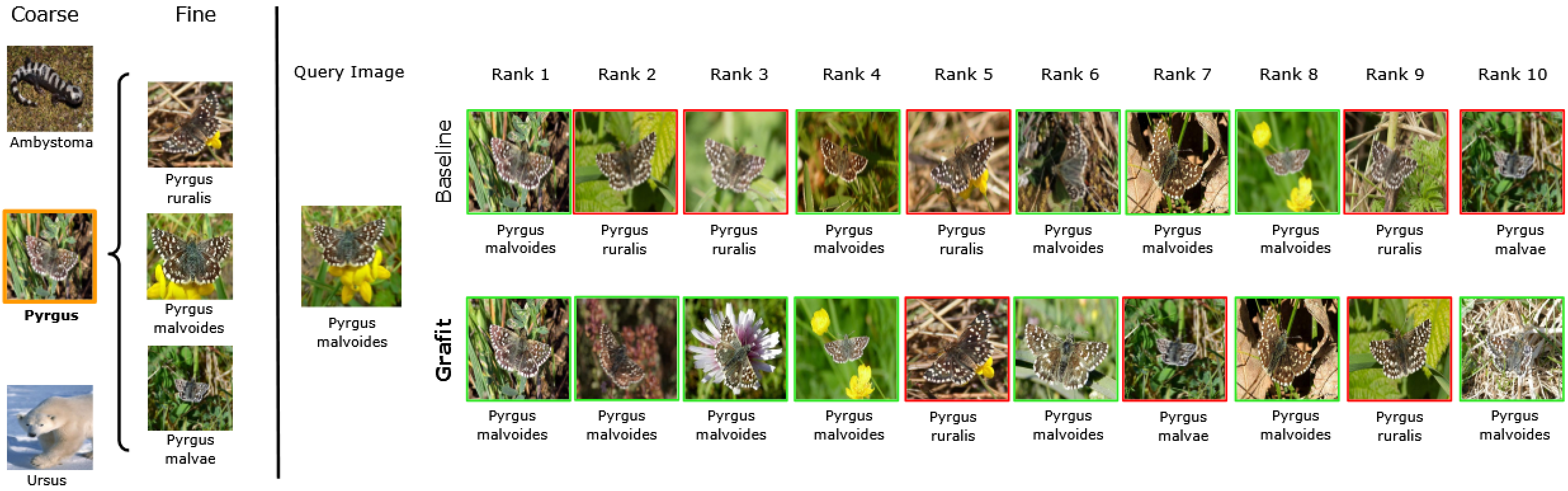
$$\mathcal{L}_{\text{knn}}(x_i, y_i) = -\log \sum_{j, y_j = y_i, j \neq i} p_{i,j}.$$

$$p_{i,j} \propto \exp\left(\cos(g_\theta(x_i), g_\theta(x_j))/\sigma\right)$$

Selection probability of each neighbour

SDU

Grafit: Learning Coarse-to-Fine training



Grafit: Learning fine-grained image representations with coarse labels, Touvron et al., 2020

Evaluation Metrics

Standard Classification Metrics

Standard Classification Metrics

→ Fine-grained tasks are typically evaluated with standard metrics:

→ Accuracy, F1-score, mean Average Precision, ...

Standard Classification Metrics

→ Fine-grained tasks are typically evaluated with standard metrics:

→ Accuracy, F1-score, mean Average Precision, ...

→ However, it is important to consider how you average your metrics!

Macro vs Micro metrics

Macro vs Micro metrics

→ Metrics are typically averaged in two ways: Globally (micro) or per-class (macro)

Macro vs Micro metrics

→ Metrics are typically averaged in two ways: Globally (micro) or per-class (macro)

Class	True Positive	False Positive
A	1	1
B	1	9
C	10	90

Macro vs Micro metrics

→ Metrics are typically averaged in two ways: Globally (micro) or per-class (macro)

→ Micro-Precision = $\frac{1+1+10}{1+9+90} = \frac{12}{100} = 0.12$

Class	True Positive	False Positive
A	1	1
B	1	9
C	10	90

Macro vs Micro metrics

→ Metrics are typically averaged in two ways: Globally (micro) or per-class (macro)

→ Micro-Precision = $\frac{1+1+10}{1+9+90} = \frac{12}{100} = 0.12$

→ Macro−Precision = $\frac{0.5+0.1+0.1}{3} = \frac{0.7}{3} = 0.23$

Class	True Positive	False Positive
A	1	1
B	1	9
C	10	90

Macro vs Micro metrics

→ Metrics are typically averaged in two ways: Globally (micro) or per-class (macro)

Class	True Positive	False Positive
A	1	1
B	1	9
C	10	90

→ Micro-Precision = $\frac{1+1+10}{1+9+90} = \frac{12}{100} = 0.12$

→ Macro−Precision = $\frac{0.5+0.1+0.1}{3} = \frac{0.7}{3} = 0.23$

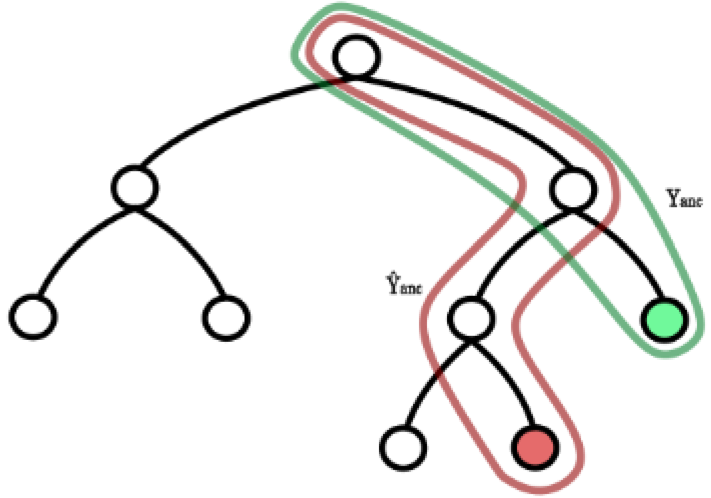
→ Macro metrics tend to overemphasize majority classes
→ Micro metrics assigns equal weights to each class

Hierarichal Evaluation

→ In taxonomic classification we have a known hierarchy

→ This can be used during evaluation to quantify the ”degree of correctness”

Hierarichal Evaluation

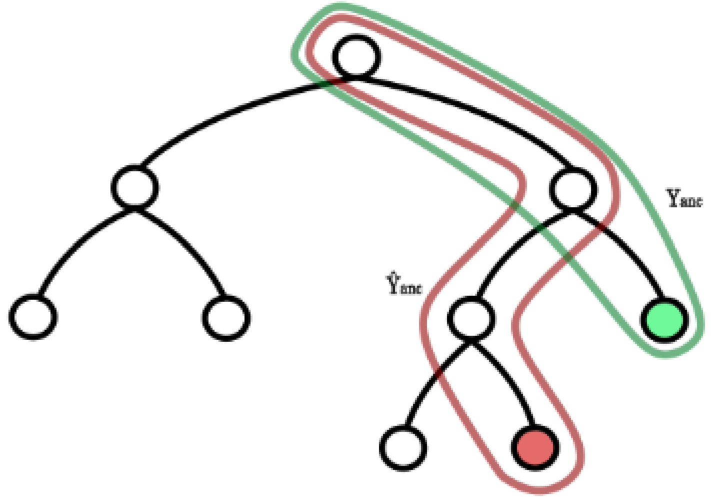


Hierarichal Evaluation

→ Precision and Recall

$$P_H = \frac{|\hat{Y}_{anc} \cap Y_{anc}|}{|\hat{Y}_{anc}|}$$

$$R_H = \frac{|\hat{Y}_{anc} \cap Y_{anc}|}{|Y_{anc}|}$$

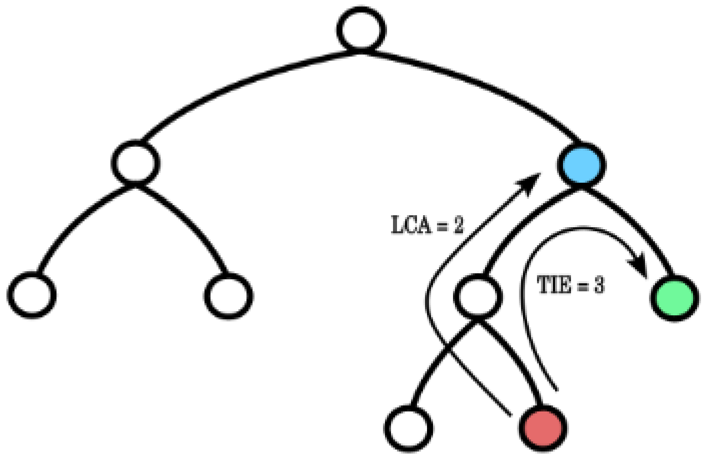


Hierarichal Evaluation

→ Precision and Recall

→ Least Common Ancestor (LCA)

→ Tree Induced Error



D3A Elevator Pitch

→ If you are:

→ A Student / PhD / PostDoc an

→ Working on Computer Vision

→ Registered for D3A

→ Please consider submitting an Elevator Pitch!

Deep Dive workshop

FROM PIXELS TO PRODUCTS: CV IN DANISH RESEARCH & INDUSTRY

Wednesday 27 August 9.00

Organizer: Joakim Bruslund Haurum, University of Southern Denmark

Computer Vision (CV) is a well-established research field with diverse applications that are vital to many industries. Fueled by the recent AI boom, access to advanced methods has increased, leading to a surge in use-cases and in-depth research across Denmark. This rapidly evolving field presents exciting opportunities but can be difficult to navigate. Our goal is to foster awareness and establish an overview and connections by showcasing the diverse computer vision landscape in Denmark and spark connections across disciplines and sectors.

In this session we highlight select topics from the wide spectrum of computer vision work in Denmark, both in academia and industry. Experts will present state-of-the-art research from Danish universities, as well as innovative, yet often under-the-radar, applications developed by Danish companies. During a panel discussion, speakers will also reflect on the future direction of CV in Denmark.

Program

0-10: Intro and quick overview of Danish CV landscape

10-25: Talk: Nico Lang, Assistant Professor, DIKU / Global Wetland Center

25-40: Talk: Kåre L. Jensen, Senior Software Developer and Computer Vision Specialist, iMotions

https://forms.gle/3Cmdno93L7GRKwf58