Karlsruher Institut für Technologie

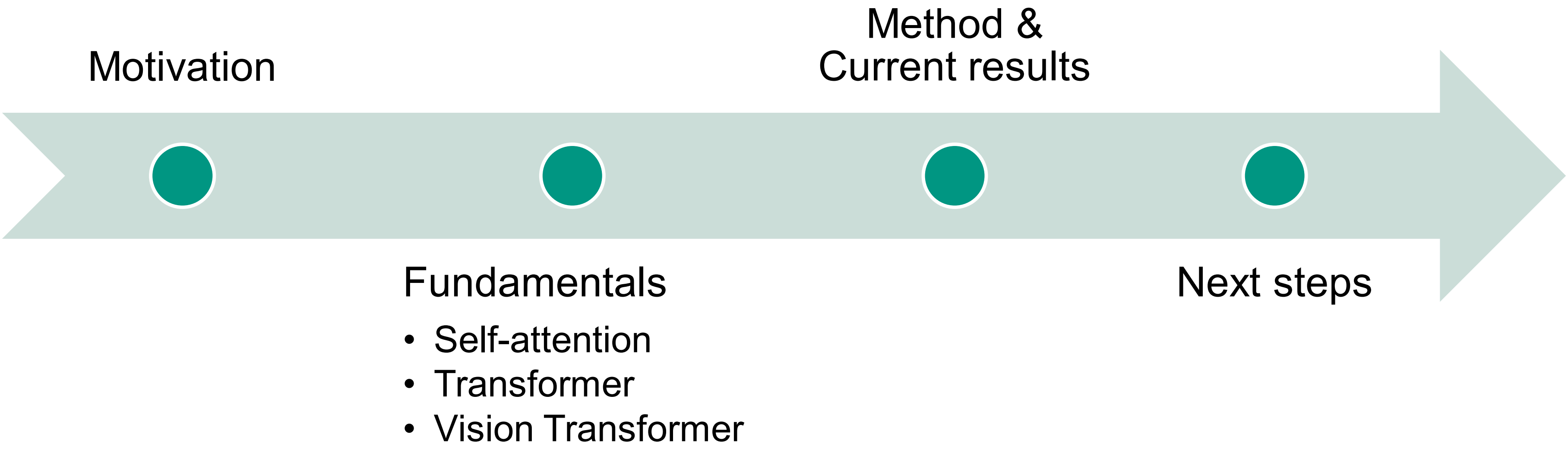
Uncertainty Estimation with Vision Transformers in an Industrial Context

Institute of Photogrammetry and Remote Sensing, Department of Civil Engineering, Geo and Environmental Sciences

Lili Gao

Reviewer: Prof. Dr.-Ing. Markus Ulrich
Advisor: Steven Landgraf und Kira Wursthorn

Agenda



Motivation

Fundamentals

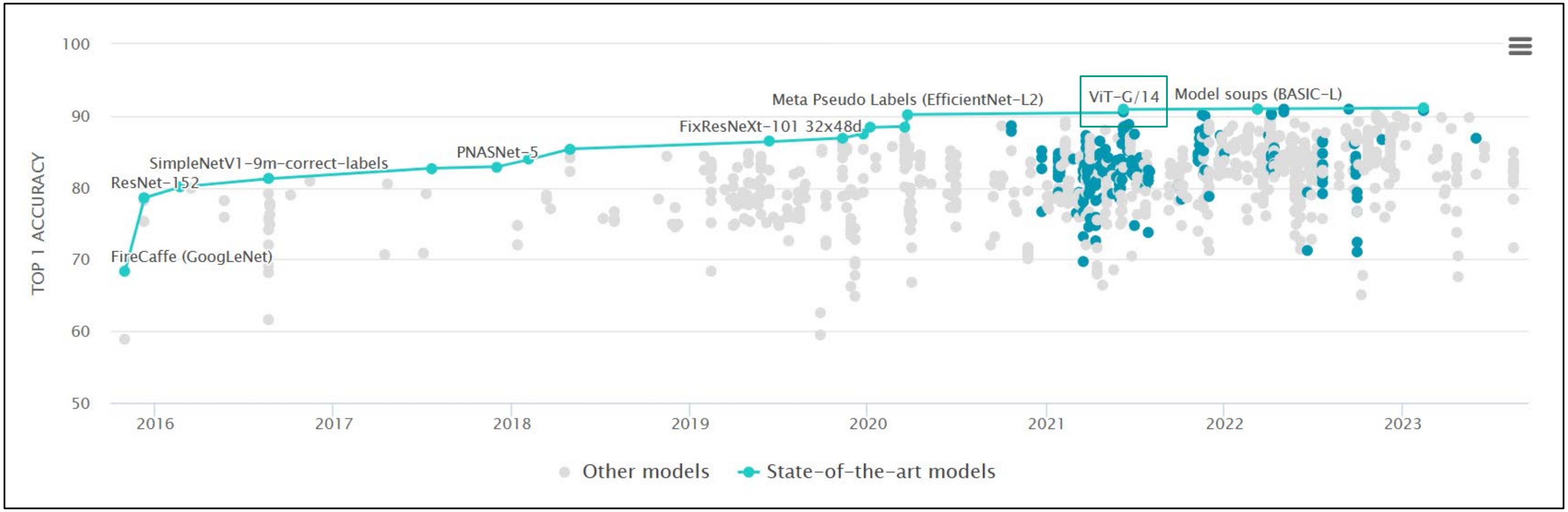
- Self-attention
- Transformer
- Vision Transformer

Method & Current results

Next steps

Motivation

Image Classification on ImageNet

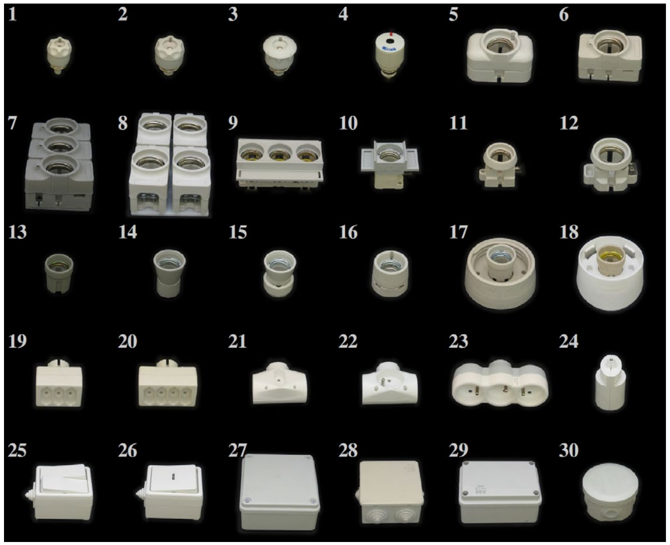


https://paperswithcode.com/sota/image-classification-on-imagenet

→ How is the performance of ViT in detecting & segmenting industrial objects?

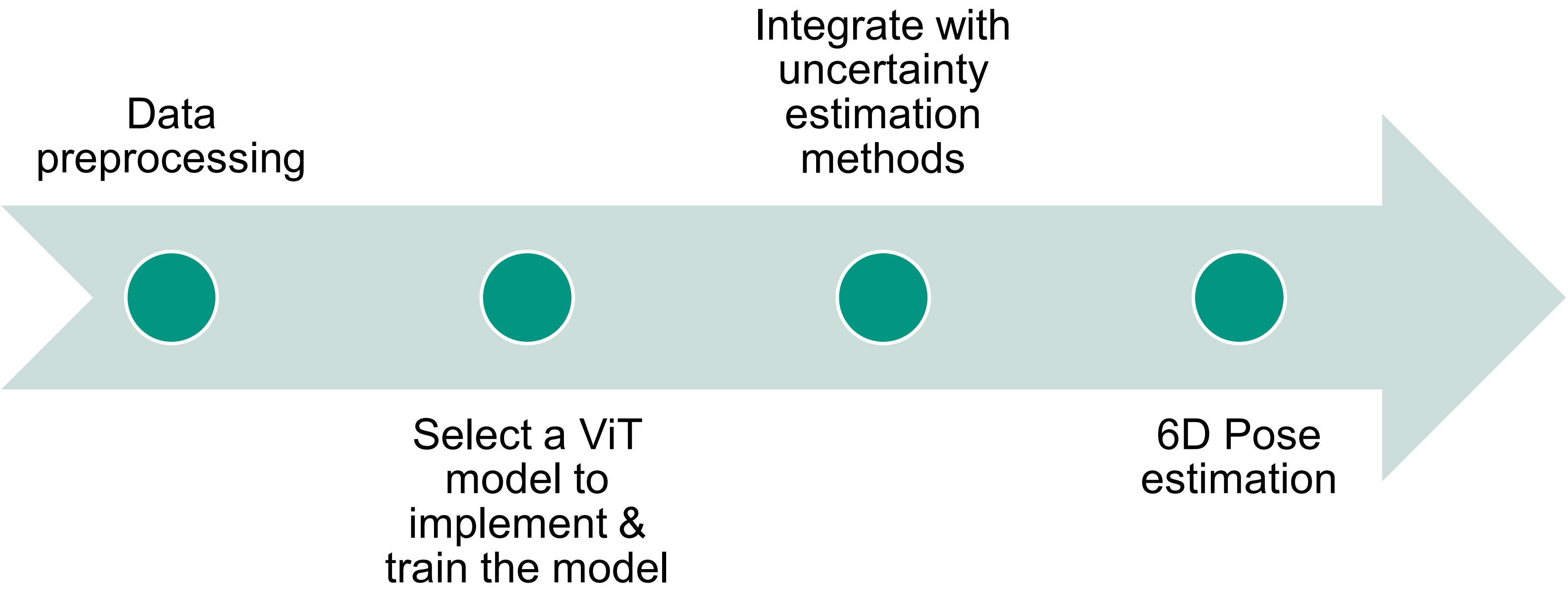
Dataset: T-LESS

- BOP Challenge (bop.felk.cvut.cz): Benchmark for 6D Object Pose Estimation
- A **RGB-D dataset** and **evaluation methodology**
  - 30 industry-relevant objects: texture-less & colorless
  - Three synchronized sensors:
    - Primesense CARMINE 1.09 (a structured-light RGB-D sensor)
    - Microsoft Kinect v2 (a time-of-flight RGB-D sensor)
    - Canon IXUS 950 IS (a high-resolution RGB camera).
  - Training images: 38K from each sensor + **50K synthetic data generated by BlenderProc**
  - Test images: 10K from each sensor



(T. Hodaň et al., 2017)

Motivation



Data preprocessing

Select a ViT model to implement & train the model

Integrate with uncertainty estimation methods

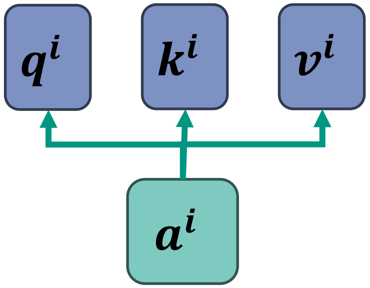
6D Pose estimation

Self-attention

- Establishs relationships between different elements in an input sequence
- Allows model to consider all inputs simultaneously

- Query: $Q = W^q\ I$
- Key: $K = W^k\ I$
- Value: $V = W^v\ I$

→ Parameters to be learned



- Attention matrix: $A' \xleftarrow{\text{softmax}} = A = K^T \cdot Q$
- Weighted sum: $O = V \cdot A'$

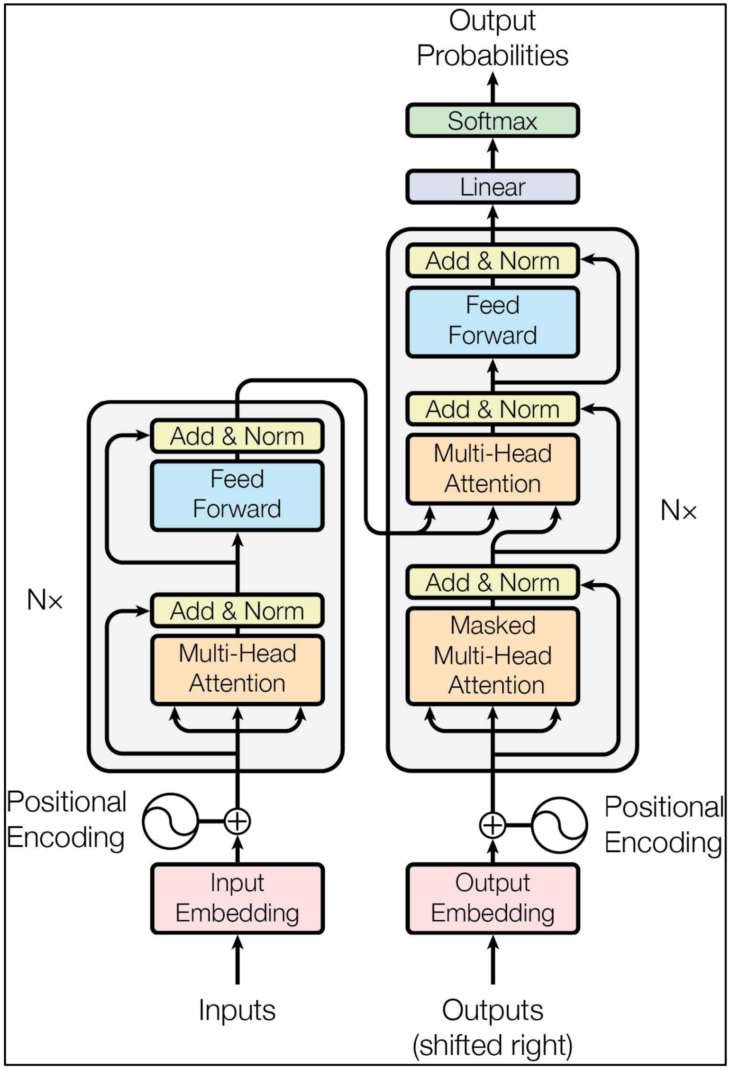
- Multi-head Self-attention: different types of relevance
- Advantage:
  - able to capture long-range dependencies
  - Comprehend global informations

Transformer

- Initially developed for natural language processing tasks

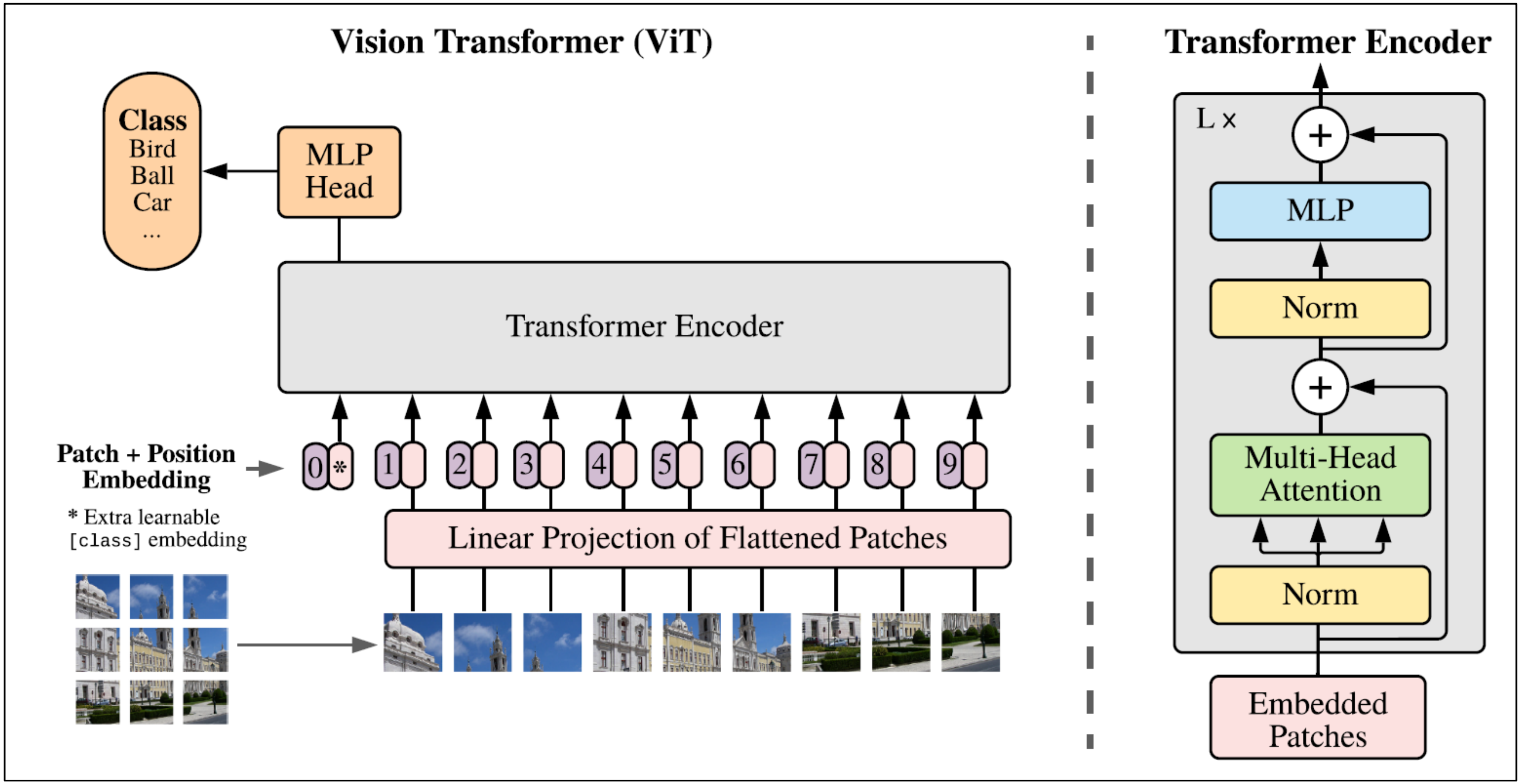
- Employs self-attention mechanisms to process input sequences

- Encoder-Decoder Structure:
  - Encoder processes the input sequence
  - Decoder generates the output sequence



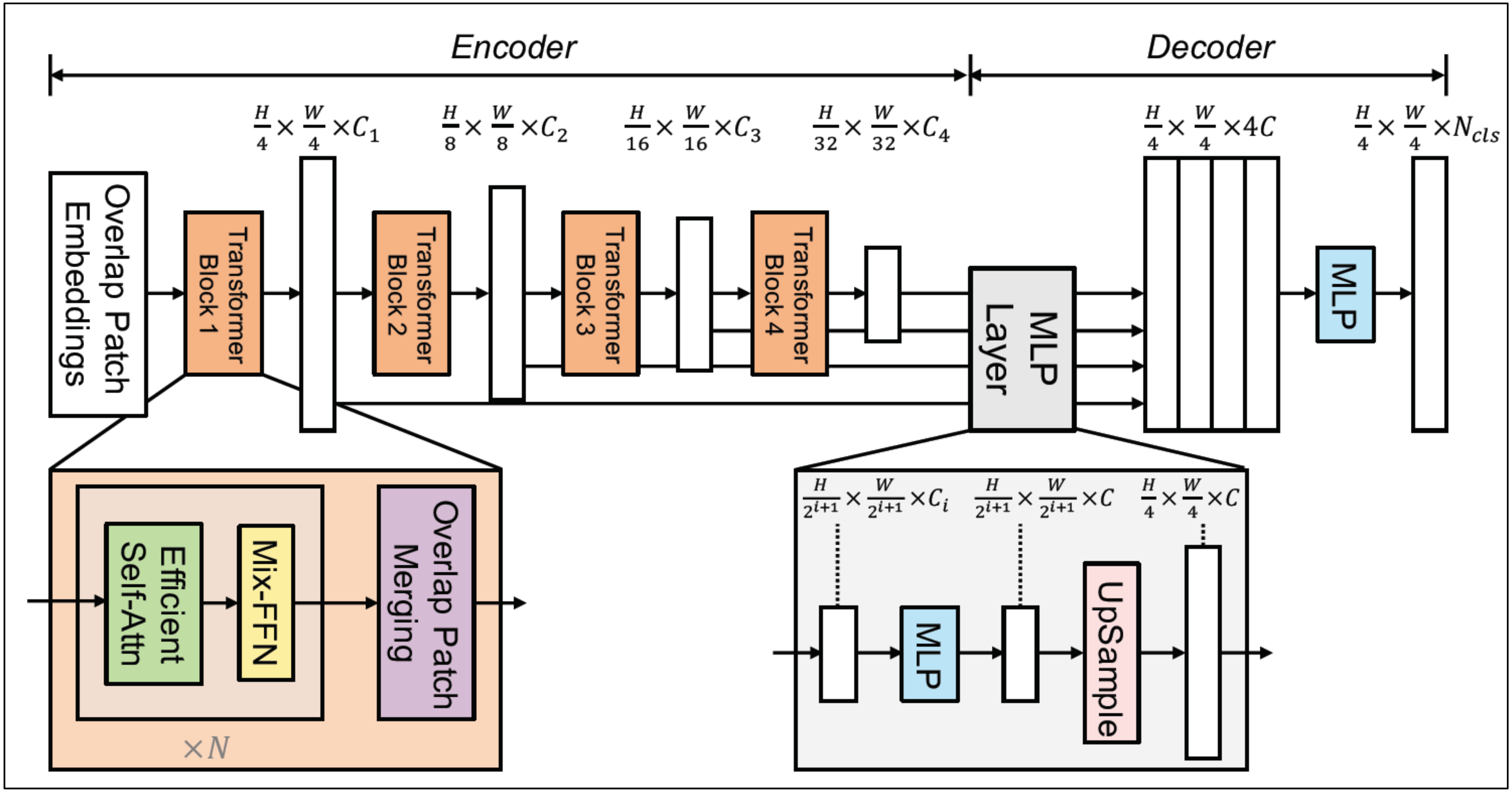
Vaswani, Ashish, et al.

Vision Transformer



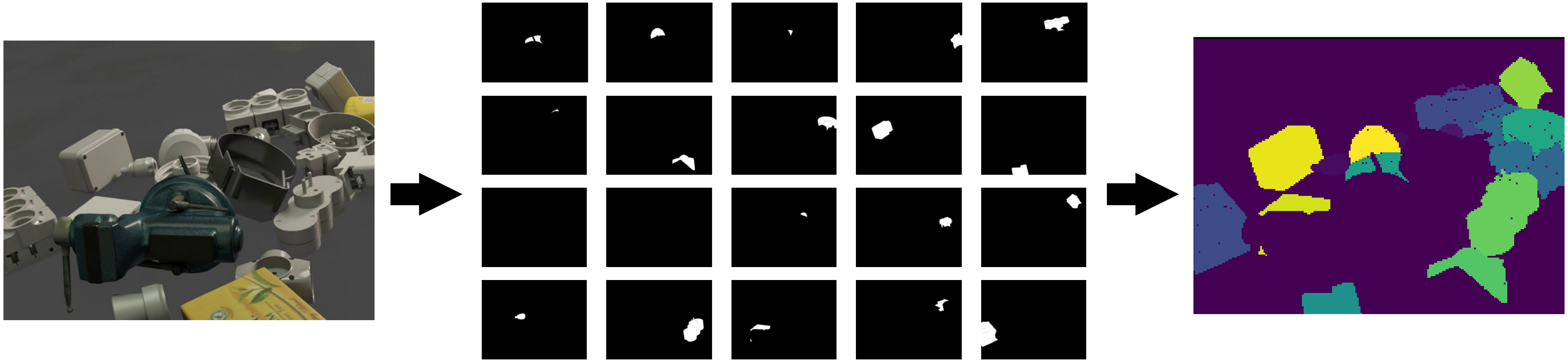
Dosovitskiy, Alexey, et al.

Selected model: SegFormer


Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Xie, Enze, et al.

Data preprocessing

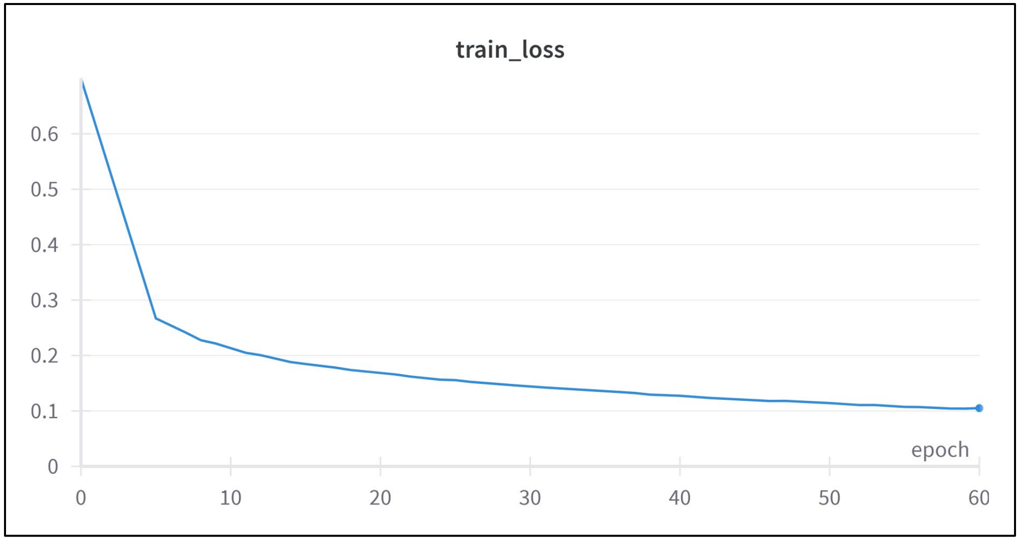


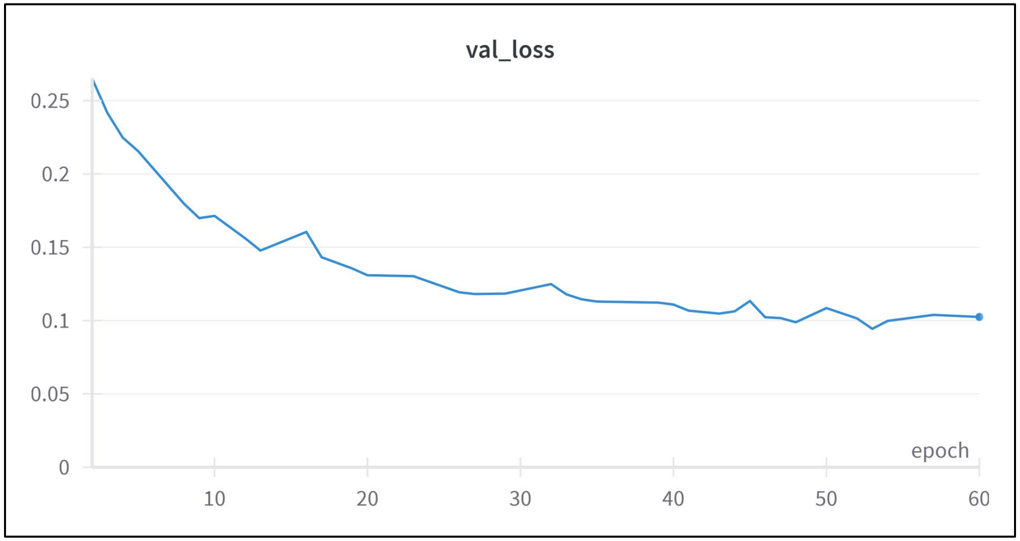
- For training dataset:
  - Random scale with ratio 0.5-2.0
  - Random horizontal flip
  - Random crop to 512x512,

Set up for training

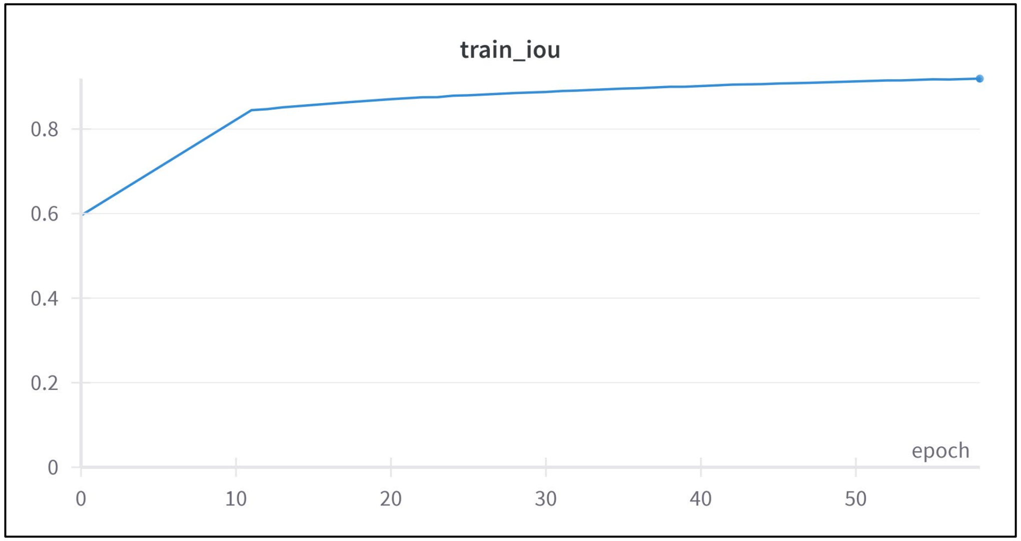
- Use 50K synthetic data generated by BlenderProc for training & validation → split the dataset in 8:2
- Pretrained model: MiT-B2
- Batch size for train = 8, for validation = 4
- using AdamW optimizer
- The learning rate was set to an initial value of 0.00006 and then used a “poly” LR schedule with factor 1.0 by default
- segmentation performance: **mean Intersection over Union (mIoU)**
- Loss: Cross Entropy

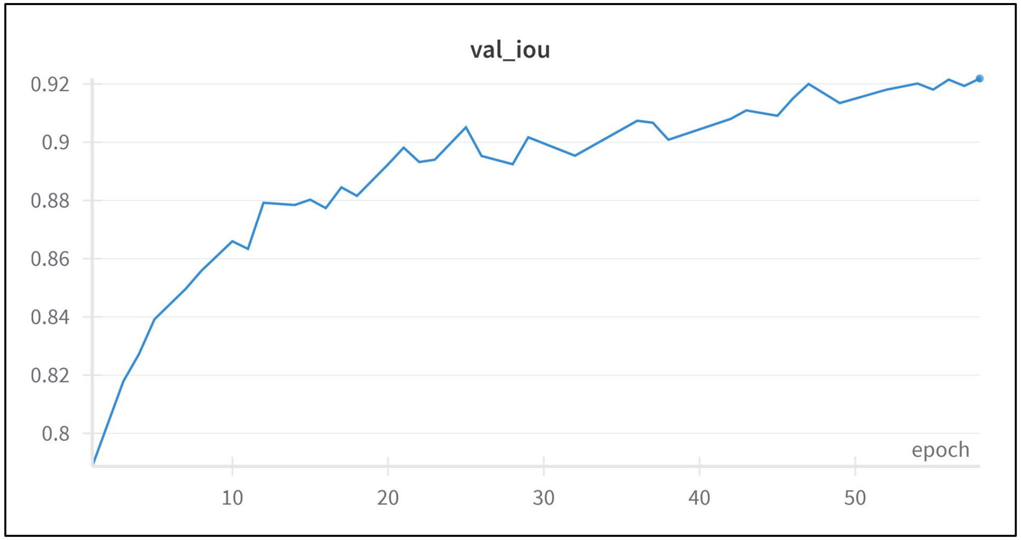
Current results:





Current results





Next steps

- Use the test images captured by sensors to test the model
- Try different combinations of datasets for training & different hyperparameters

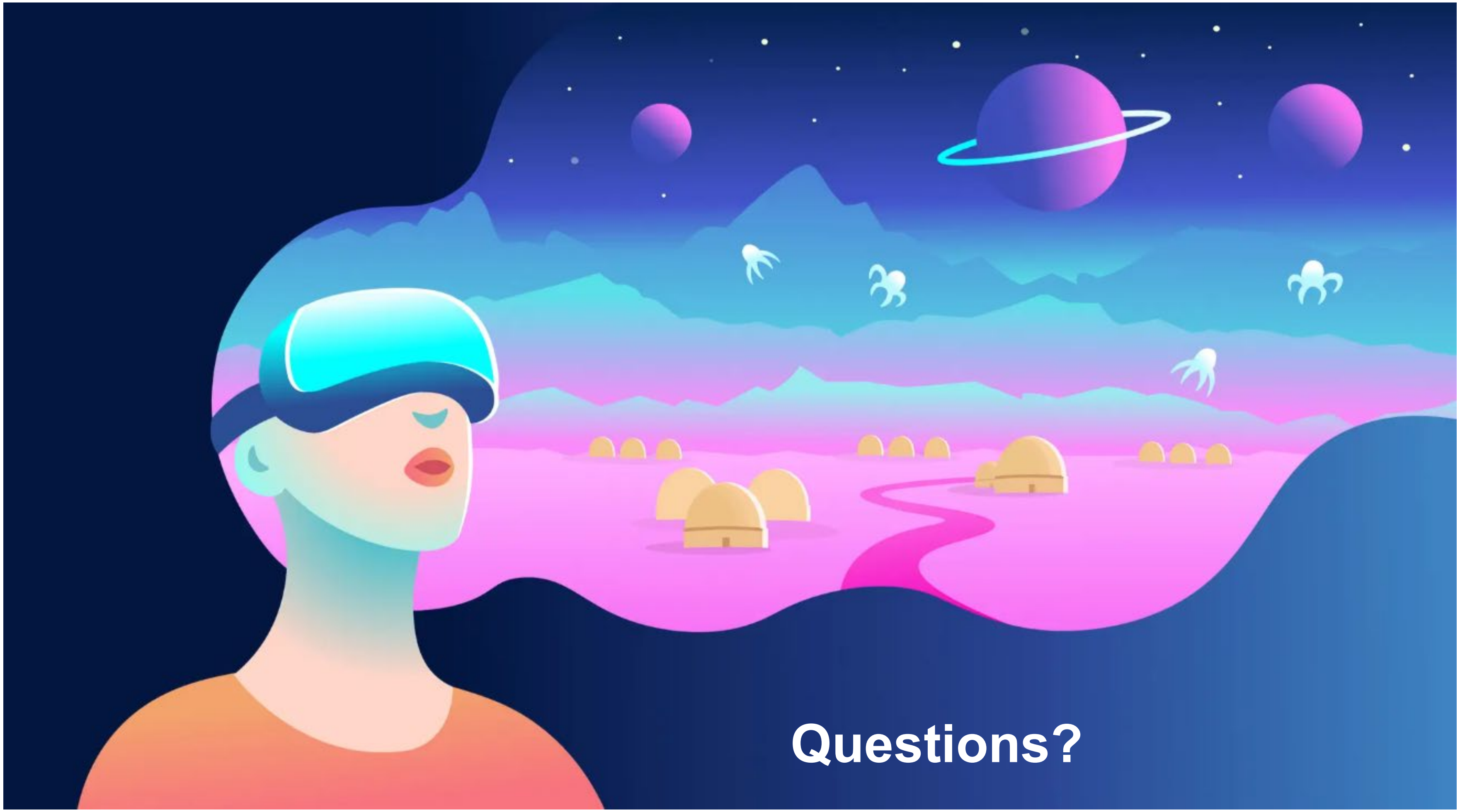
  → train the model without overfitting
- Integrate model with uncertainty estimation methods
  - MCDropout
  - Ensembling
  - Deep Deterministic Uncertainty
- Integrating model into existing methods for 6D pose estimation

- See details of the implementation:

https://github.com/lilligao/pytorch-masterArbeit/tree/gpu-version

Literatur

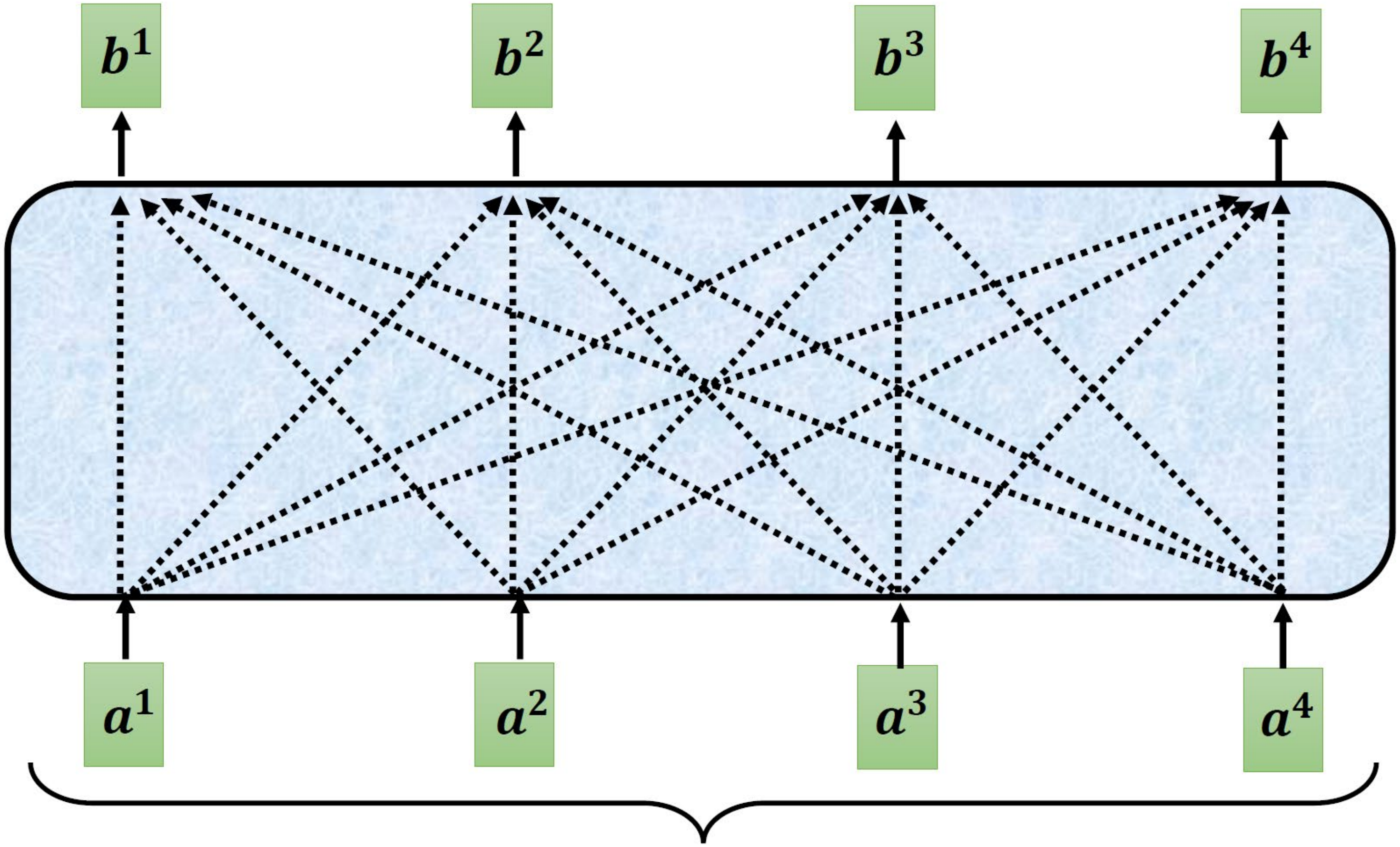
- [1] https://paperswithcode.com/sota/image-classification-on-imagenet (Letzter Zugriff: 27.11.2023)
- [2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- [3] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, X. Zabulis, T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects, IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, Santa Rosa,
- [4] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
- [5] Xie, Enze, et al. "SegFormer: Simple and efficient design for semantic segmentation with transformers." Advances in Neural Information Processing Systems 34 (2021): 12077-12090.


Questions?

Tasks

- Select a ViT model and implement the method
- Evaluate datasets from the BOP Challenge (bop.felk.cvut.cz)
- Assessing the performance in detecting and segmenting industrial objects
  → explore the integration of ViT with uncertainty estimation methods
  - MCDropout
  - Ensembling
  - Deep Deterministic Uncertainty
- Integrating ViT into existing methods for 6D pose estimation

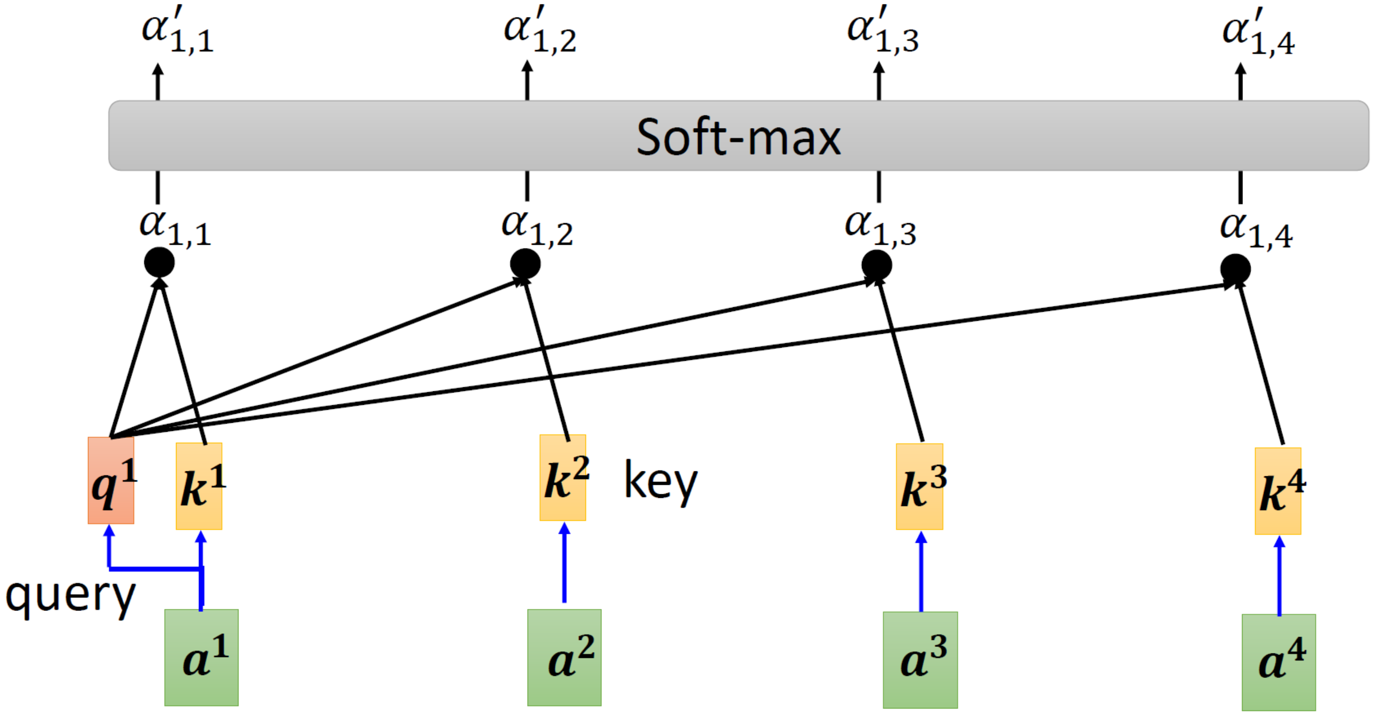
Self-attention



Can be either **input** or **a hidden layer**

Self-attention

$$\alpha'_{1,i} = exp(\alpha_{1,i})/\sum_j exp(\alpha_{1,j})$$



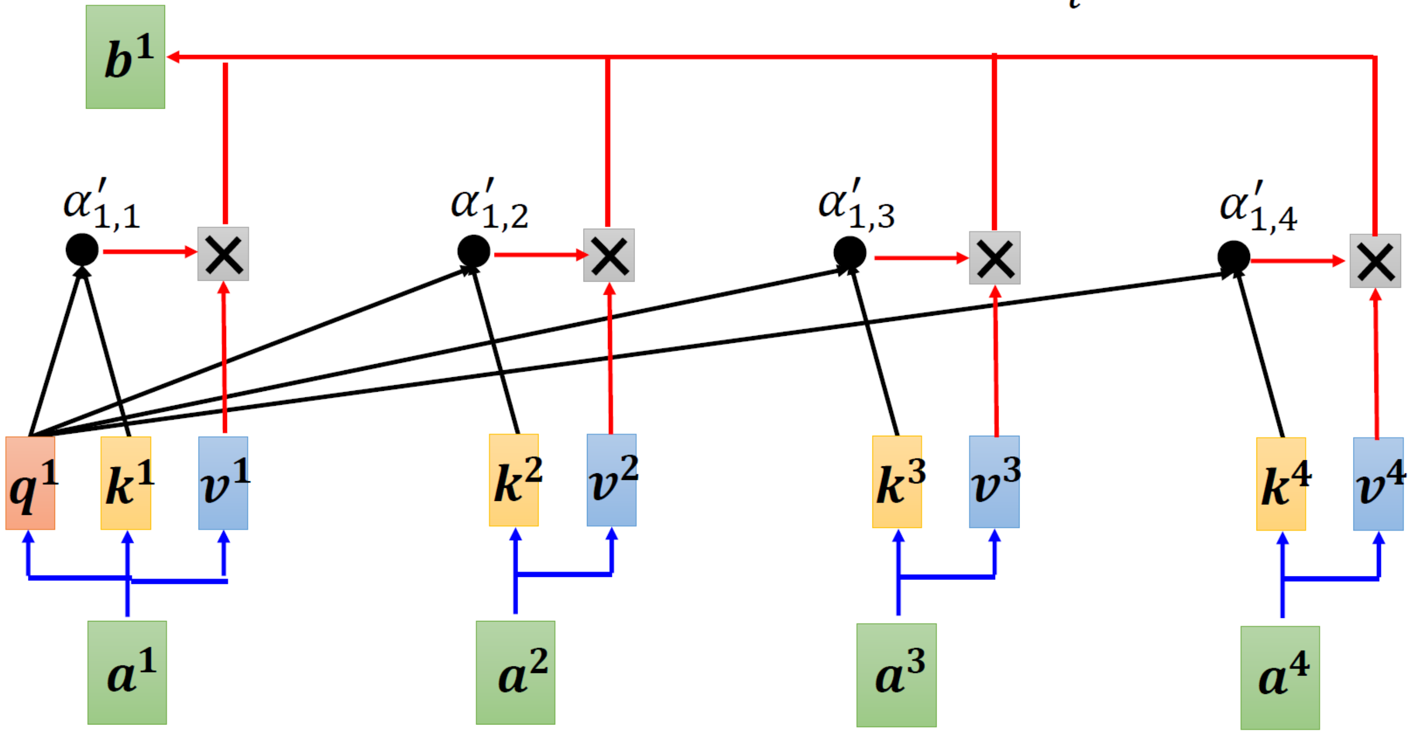
$q^1 = W^q a^1$ $\quad k^2 = W^k a^2$ $\quad k^3 = W^k a^3$ $\quad k^4 = W^k a^4$

$k^1 = W^k a^1$

Self-attention

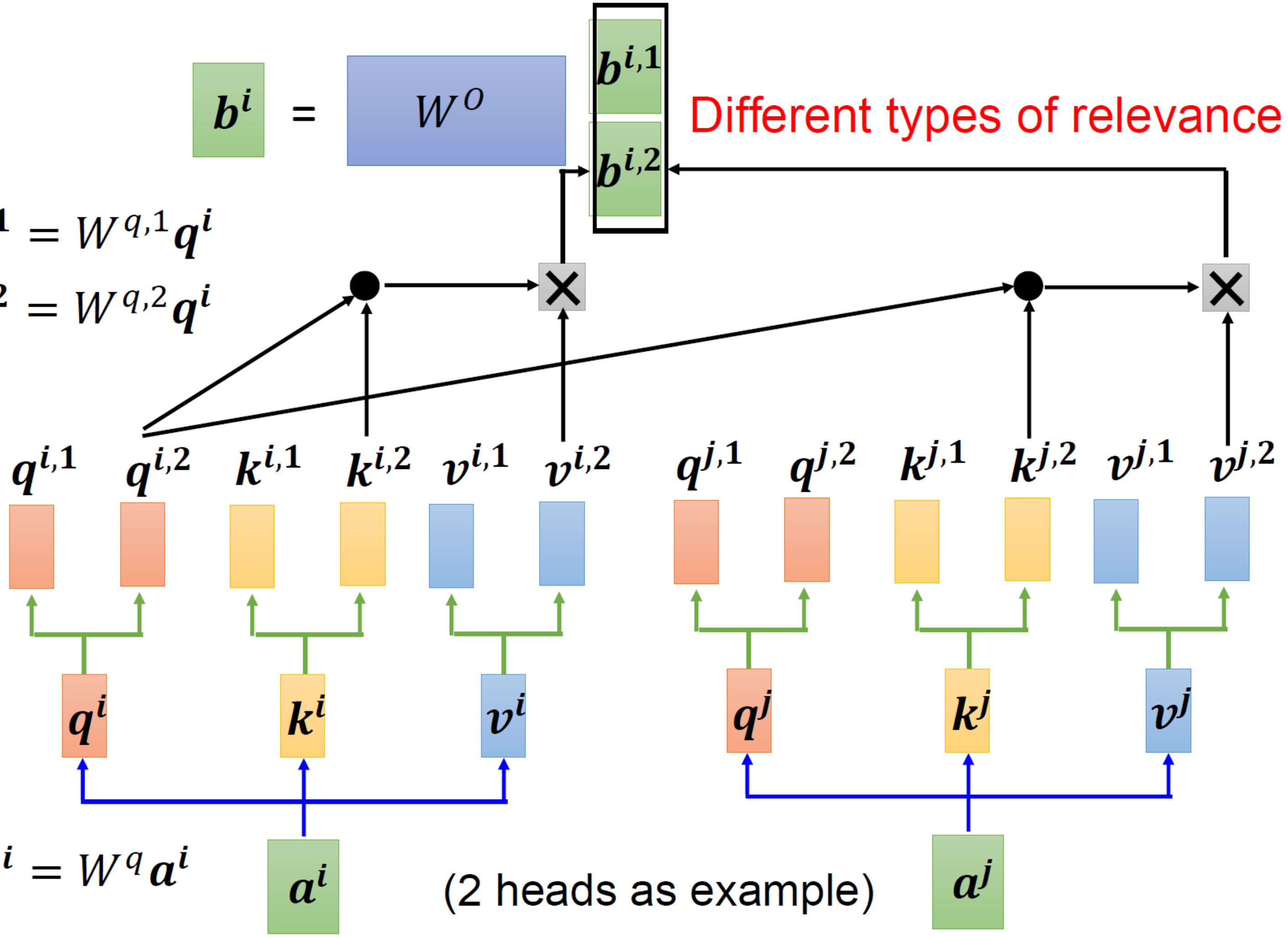
Extract information based on attention scores

$$b^1 = \sum_i \alpha'_{1,i} v^i$$



$v^1 = W^v a^1$ $v^2 = W^v a^2$ $v^3 = W^v a^3$ $v^4 = W^v a^4$

Multi-head Self-attention



Different types of relevance

$$q^{i,1} = W^{q,1} q^i$$

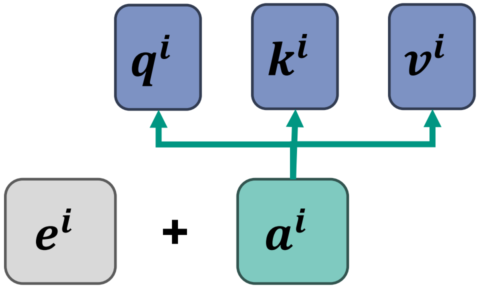
$$q^{i,2} = W^{q,2} q^i$$

$$q^i = W^q a^i$$

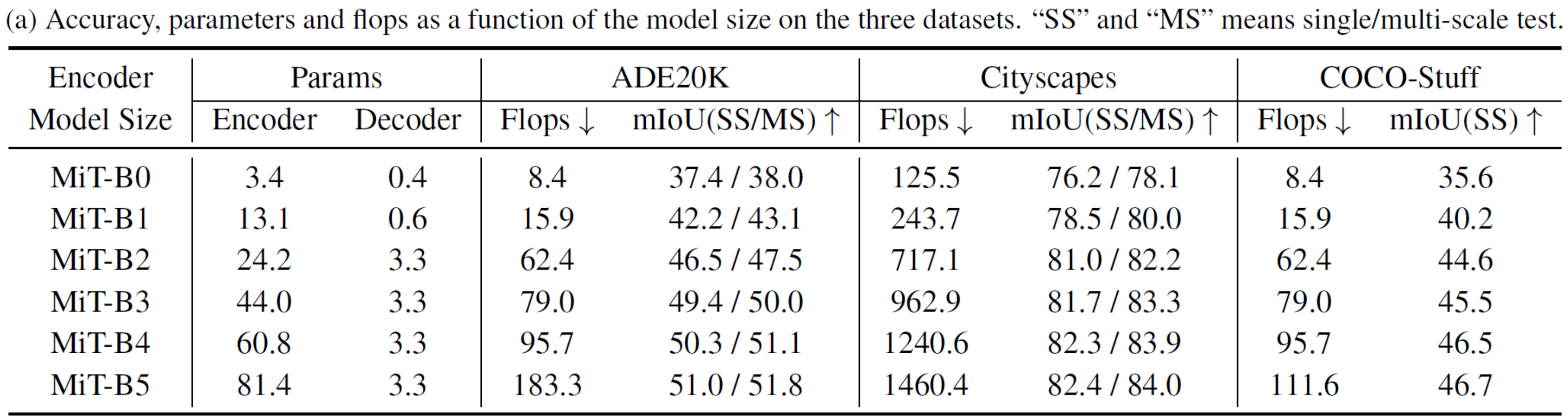
(2 heads as example)

Self-attention: Positional Encoding

- No position information in self-attention
- Each position has a unique positional vector e^i
  - **Handcrafted**
  - **Learned from data**



SegFormer: model size



(a) Accuracy, parameters and flops as a function of the model size on the three datasets. “SS” and “MS” means single/multi-scale test.

Encoder	Params		ADE20K		Cityscapes		COCO-Stuff	
Model Size	Encoder	Decoder	Flops ↓	mIoU(SS/MS) ↑	Flops ↓	mIoU(SS/MS) ↑	Flops ↓	mIoU(SS) ↑
MiT-B0	3.4	0.4	8.4	37.4 / 38.0	125.5	76.2 / 78.1	8.4	35.6
MiT-B1	13.1	0.6	15.9	42.2 / 43.1	243.7	78.5 / 80.0	15.9	40.2
MiT-B2	24.2	3.3	62.4	46.5 / 47.5	717.1	81.0 / 82.2	62.4	44.6
MiT-B3	44.0	3.3	79.0	49.4 / 50.0	962.9	81.7 / 83.3	79.0	45.5
MiT-B4	60.8	3.3	95.7	50.3 / 51.1	1240.6	82.3 / 83.9	95.7	46.5
MiT-B5	81.4	3.3	183.3	51.0 / 51.8	1460.4	82.4 / 84.0	111.6	46.7