Vision Transformer Architectures with Registers (Thesis outline)

Florian Weidner
Philipps-University Marburg, Germany
Department of Mathematics and Computer Science, Deep Learning Group
February 09, 2024

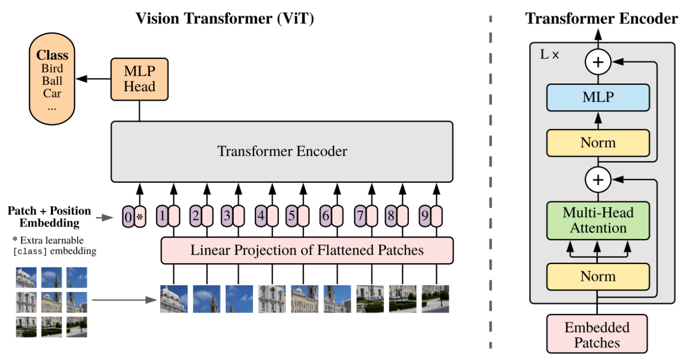


Fig. 1. Overview of a ViT architecture. [4]

***Abstract*—The abstract goes here.**

***Index Terms*—Vision Transformer (ViT),**

I. Introduction

Introduction to the topic...

Explanation of ViTs [**10.1145/3505244**] [4] [6] [**Liu2024-lm**]

II. Vision Transformers

The Transformer architecture is a neural network model architecture, created primarily for sequence-to-sequence tasks in Natural Language Processing (NLP).

> ”The key feature of transformers is the self-attention mechanism, which helps a model learn the global contexts and enables the model to acquire the long-range dependencies.” [6]

It consists of an encoder, which makes the input sequence into a continuos representation and a decoder, which then generates the output sequence. The encoder is built up of n identical layers, containing following components:

- multi-head self-attention mechanism: captures relationships between all tokens in the input, regardless of their distance
- feed-forward network: simple two-layer MLP network with ReLU activation which is applied to each token separately
- add & norm layers using residual connections and layer normalization to stabilize the training

The result outcome of the encoder is a enriched sequence representation, which is then used by the decoder to generate the output sequence. The decoder also consits of n identical layers with:

- masked multi-head self-attention mechanism: ensures a causal generation, by preventing that tokens have impact to future tokens
- encoder-decoder attention mechanism: focuses on the relevant parts of the encoder's output
- feed-forward network: similar to the encoder
- add & norm layer:ssimilar to the encoder

The input text is embedded and combined with a positional encoding to provide token order information. Because several attention layers can run in parallel, the architecture is significantly more parappelizable than Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN) architectures, which makes it very efficient for modern hardware accelerators. That allows the Transformer to scale to very large models and datasets. [11]

Dosovitskiy et al. introduced the idea of using the stated transformer architecture for computer vision. A lot of research tried to combine self-attention mechanisms with CNN architectures, not achieving a effectively scalable method for modern hardware accelerators. [4] proposed to apply a standard Transformer directly to images, that are split into fixed-size patches. Each patch is flattened into a vector and passed through a linear projection layer to form an embedding as input for the Transformer. These embeddings are used as tokens in a NLP scenario. Positional embeddings are added to retain spatial information since they process images as sequences, unlike CNNs which inherently capture spatial hierarchies. For classification tasks, an extra learnable [class] embeding is added in front of the embedded input. At the output of the encoder, the final representation of this token is used for classification. Instead of using encoder and decoder like in NLP tasks, acpvit only uses the encoder since the goal is to find a better representation but an autoregressive prediciton. Additional Layer Normalization is used before the multi-head attention laver. [6] In figure 1 you can see the architecture of a ViT including the split image patches, their embeddings combined with positional embeddings the encoder and the class embedding used for the classificaion prediction. ViTs have much less image-specific inductive bias than CNNs, because other than CNNs, with the global self-attention mechanism spatial relationships needs to learned from scratch, but long-range dependencies across the entire image can be captured. As Transformers, ViTs are normally pre-trained on large datasets and then fine-tuned to more specific tasks. After pre-training, the prediction head is removed and a zero-initialized feedforeward layer, where the size is the number of classes, is added.

Like Transformers, ViTs are also very parallelizable, which makes them very efficient. But [4] found out that without large-scale pre-training, ViTs often underperform. So ViTs requires sidnificant computational resources. But when pre-trained on large datasets, ViTs outperforms CNNs on image classification tasks. The architecture performs well for transfer learning, where the pre-trained model can be fine-tuned already with limited labeled data [4]. [4] stated that further scaling of ViTs would likely lead to improved performance. Also self-supervised pre-training, where no labeled data is needed, can be improved. They found out that with mimicking the masked language modeling task used in BERT, the model performs still better than CNNs but a bit worse that with supervised pre-training of a ViT. By now different architectures and training-tricks of ViTs have been proposed to further improve ViTs including self-supervised learning. The architectur also got adapted for image recognition, object detection, image segmentation, pose estimation, and 3D reconstruction tasks. [6]

The classical ViT architecture has been adopted and improved by many others. One approach is to include CNN structures, which bring locality through the convolution kernels, into ViTs to improve the data efficiency. DeiT [10] for example uses a CNN as a teacher to train a ViT. It utilizes knowledge distillation of the CNN to add the inductive bias to a vision transformer. It allows to train a ViT without the need of large-scale pre-training the model. [6] Another approach is to diversify the features of ViTs. DeepVit [14] found out that the attention collapses in deeper layers, wich leads to lower performance. By adding a learnable transformation matrix after the attention layer, the model is stimulated to generate new attention maps also in the deeper layer, increasing the performance. [6] Also the heavy computation costs are reseached. Many also try to improve the self-supervised learning, that the pre-training with the need of large datasets can be simplified. One approach is DINO. [1] It uses a teacher-student archtecture, where the student network learns to match the avereaged outputs of the teacher. [6] The following summerized paper, identifies and addresses artifacts in attention maps of supervised and self-supervised ViT networks.

[9] [2]

III. Vision Transformers need registers: A summary

In this chapter we summerize the paper [3]. The paper discoverd artifacts and proposes to use additional register tokens for ViTs to remove these artifacts.

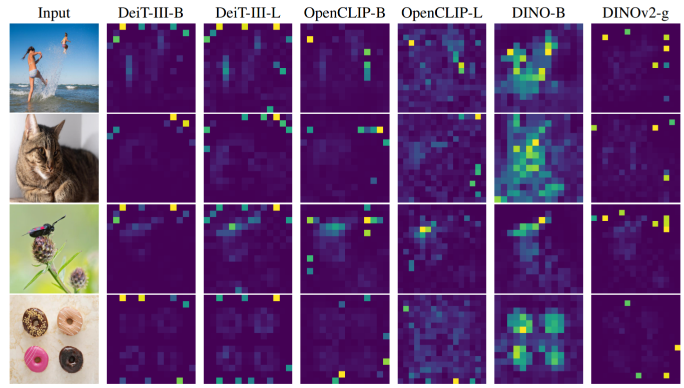


Fig. 2. Illustration of artifacts observed in the attention maps of modern vision transformers. [3]

A. *Artifacts in Vision Transformers*

After introducing to ViTs like we did in this paper, the models they found the artifacts are introduced. The DINO algorithm is a self-supervised learning method, that uses two ViTs. A student network is predicting the output of a teacher network, to learn rich representaions of visual data without the need of manual annotations. [1] DINO is shown to produce models, that contain semantically consistent information in the last attention layer. Object discovery algorihtms like LOST [8], built on top of DINO, are using these attention maps, that often contains semantically interpretable information, used to detect objects without supervision. DINOv2 [5] is a improved followup focusing on dense predition tasks, which are tasks, where detailed outputs are required to provide fine-grained localized informations, like semantic segementation or depth estimation. Despite good performance on these dense tasks, the authors observed that DINOv2 is incompatible with LOST [3]. The different behaviour of DINO and DINOv2 can be observed in the artifacts in the last attention maps. In figure 2 you can see the different models and their artifacts on the last attention layer. While DINO shows no peak outlier values focusing the main object in the image, DINOv2 shows a lot of artifacts on the background of the images. This qualitatively observation can be also made for the label-supervised model DeiT-III and the text-supervised model OpenCLIP. Shown in figure 2, you can observe similar artifacts in the background. To explain why and where the artifacts of ViTs in attention maps appear, the paper focuses on DINOv2.

Artifact patches show higher norm of their token embedding at the output of the model than other patches. In figure 3 you can see the distribution of the local feature norms over a small dataset. While for DINO, the norm stays under 100 for all patches, DINOv2 shows a lot of patches with a norm higher than 150. This cutoff value can vary across different models. They define artifacts as

> ”tokens with norm higher than 150 will be considered as “high-norm” tokens” [3]

The authors found different conditions, when the artifacts appear in the training process of DINOv2. Figure 4 shows the

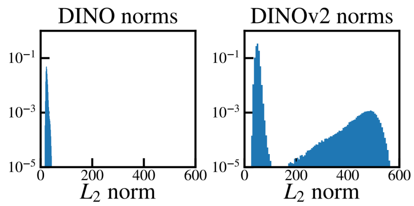
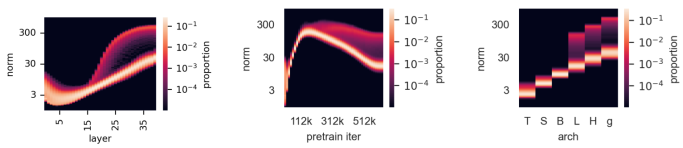


Fig. 3. Comparison of local feature norms for DINO ViT-B/16 and DINOv2 [3]



(a) Norms along layers. (b) Norms along iterations. (c) Norms across model size.

Fig. 4. Illustration of several properties of outlier tokens in the 40-layer DINOv2 ViT-g model [3]

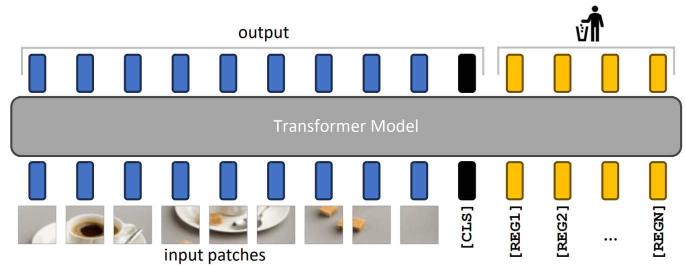


Fig. 5. Illustration of the proposed remediation and resulting model [3]

following conditions:

- artifacts start appearing around layer 15 to 40.
- artifacts start appearing after on thrid of training.
- artifacts only appear in the three largest model versions

Another discovery is that the high-norm tokens appear where patch information is redundant. The authors tested the cosine similarity between high-norm tokens and their four neighbors, directly after the image is emebdded. They observed, that the high norm patches appear where their cosine similarity to the neighbors is high. Compared to the observations, that shows that artifacts appear mostly in the background of images, high-norm pathes seem to have redundant information, that the model can ignore, to achive similar scores at the output.

To further understand the outlier tokens, two linear models were trained, to check the embeddings for different information. Both models were trained on the patch embeddings, the embeddings of the images (see figure 1). The result performance is compared between using high-norm tokens and normal tokens. The first task was position prediction. The model should predict the position of a patch token in the image and measure the accuracy. They observed that high-norm tokens have much lower accuracy than the other tokens and suggested that they contain less information about the position in the image. The second task was pixel reconstruction. The model should predict the pixel value of an image from the patch embeddings and mesaure the accuracy of this model. Also here the high-norm tokens have lower accuracy than the other tokens. The authors concluded that the high-norm tokens contain less information to reconstruct the image than the others. The authors also found out that the high-norm tokens hold more global information by training a logistic regression model. The model predicts the image class by the patch embedding of a random token. I turned out that the high-norm tokens have a much higher accuracy than the other tokens. This suggests that the high-norm tokens contain more global information about the image than the other tokens.

Making these observations the authors make following hypsthesis:

> ”Large, sufficiently trained models learn to recognize redundant tokens, and to use them as places to store, process and retrieve global information.” [3]

B. Registers for Vision Transformers

To address the behaviour, the use of registers is proposed. Since the high-norm patches are overtaking local patch information, even they are mostly not important, it possibly decreases the performance on dense prediction tasks. The called registers are additional tokens after the patch embeddings of the images with a learnable value. They work similar to the [class] token, used for classificaion tasks. They are used durning training and inference and they are discarded afterwards. In figure 5 you can see the register tokens additionally used after the embedding of the image. A complexity analysis show that adding registers increase the FLOPs by up to 6% for 16 registers. With four registers, that are more commonly used, the increase is below 2%.

C. Evaluation of the proposed architecture

In the last part of the paper they validate their architecture by training ViTs with register tokens and compare them quantitativly and qualitativly to the models without token registers. They are evaluating for DEIT-III, OpenCLIP and DINOv2 architectures, therfore including label-supervised, text-supervised and self-supervised learning approaches. In figure 6 you can see three example images including attention maps with and without the use of register tokens. Qualitativly, for all three models, the artifacts in the attention maps are gone. They mesaured quantitativly the effect by calculating the norm of the attention maps at the output of the model. In figure 7 you can see the distribution of the output norms for the three models. For all three models, training it with register tokens removes high-norm tokens, that were present without the token registers. Instead the attention maps of the register tokens have higher norm than the patch and the class tokens. The register tokens are adapting the behaviour of the outlier patches of the model without registers. Visualizations are also showing that the attention maps of the register tokens look similar to the attention maps of the class tokens, all showing a larger support area. The attention maps of the patch tokens are more

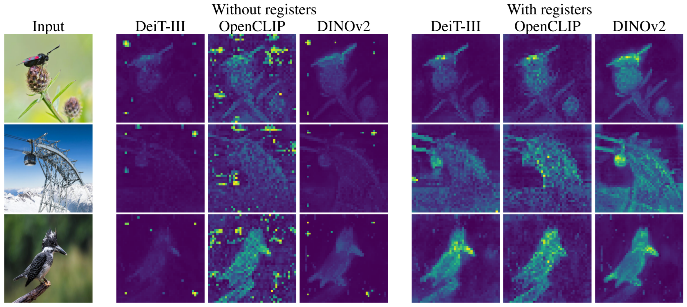


Fig. 6. Three examples of attention maps with and without register tokens [3]

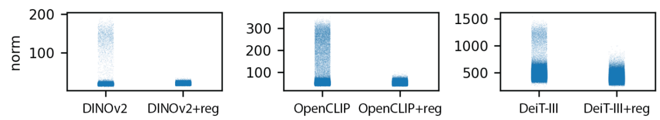


Fig. 7. : Effect of register tokens on the distribution of output norms [3]

localized. Since the class token carries global informations, it suggests that the register tokens are also used to store global information. Comparing the performance of the models with and without register tokens, linear probing on ImageNet classification, ADE20k Segmentation, and NYUd monocular depth estimation datasets was used. The results show no lose in performance, when additoinally using register tokens. Also for zero-shot classification on ImageNet with OpenCLIP, the performance is not affected by using register tokens. They also found out that one register is enough to remove the high-norm tokens in the attention maps. For DINOv2 and DeiT-III, adding register tokens significantly improves the discovery performance and for OpenCLIP, the performance is slighty worse with registers. They concluded that their proposal improves the performance in dense prediction and object discovery.

IV. Comparison to other papers with performance improvements of ViTs

A. [12] applies the idea of registers to a n State Space Model (SSM)

B. [13] uses a token learner to improve the performance of ViTs

C. [7] also uses a token learner to improve the performance of ViTs

V. Conclusion

The conclusion goes here.

References

[1] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV]. URL: https://arxiv.org/abs/2104.14294.

[2] Mehdi Cherti et al. "Reproducible Scaling Laws for Contrastive Language-Image Learning". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023, 2818–2829. DOI: 10.1109/cvpr52729.2023.00276. URL: http://dx.doi.org/10.1109/CVPR52729.2023.00276.

[3] Timothée Darcet et al. *Vision Transformers Need Registers*. 2024. arXiv: 2309.16588 [cs.CV]. URL: https://arxiv.org/abs/2309.16588.

[4] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[5] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV]. URL: https://arxiv.org/abs/2304.07193.

[6] Bo-Kai Ruan, Hong-Han Shuai, and Wen-Huang Cheng. *Vision Transformers: State of the Art and Research Challenges*. 2022. arXiv: 2207.03041 [cs.CV]. URL: https://arxiv.org/abs/2207.03041.

[7] Michael S. Ryoo et al. *TokenLearner: What Can 8 Learned Tokens Do for Images and Videos?* 2022. arXiv: 2106.11297 [cs.CV]. URL: https://arxiv.org/abs/2106.11297.

[8] Oriane Siméoni et al. *Localizing Objects with Self-Supervised Transformers and no Labels*. 2021. arXiv: 2109.14279 [cs.CV]. URL: https://arxiv.org/abs/2109.14279.

[9] Hugo Touvron, Matthieu Cord, and Hervé Jégou. "DeiT III: Revenge of the ViT". In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 516–533. ISBN: 978-3-031-20053-3.

[10] Hugo Touvron et al. *Training data-efficient image transformers & distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV]. URL: https://arxiv.org/abs/2012.12877.

[11] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[12] Feng Wang et al. *Mamba-R: Vision Mamba ALSO Needs Registers*. 2024. arXiv: 2405.14858 [cs.CV]. URL: https://arxiv.org/abs/2405.14858.

[13] Yuxin Wen et al. *Efficient Vision-Language Models by Summarizing Visual Tokens into Compact Registers*. 2024. arXiv: 2410.14072 [cs.CV]. URL: https://arxiv.org/abs/2410.14072.

[14] Daquan Zhou et al. *DeepViT: Towards Deeper Vision Transformer*. 2021. arXiv: 2103.11886 [cs.CV]. URL: https://arxiv.org/abs/2103.11886.

that's all folks