Understanding Gaussian Attention Bias of Vision Transformers Using Effective Receptive Fields

BMVC 2023

Bum Jun Kim, Hyeyeon Choi, Hyeonah Jang, Sang Woo Kim

POSTECH EE

Keywords: vision transformer, effective receptive field, self-attention, attention bias, positional embedding

GitHub Repository: https://github.com/kmbmjn/GaussianAttentionBias

Abstract

Vision transformers (ViTs) that model an image as a sequence of partitioned patches have shown notable performance in diverse vision tasks. Because partitioning patches eliminates the image structure, to reflect the order of patches, ViTs utilize an explicit component called positional embedding. However, we claim that the use of positional embedding does not simply guarantee the order-awareness of ViT. To support this claim, we analyze the actual behavior of ViTs using an effective receptive field. We demonstrate that during training, ViT acquires an understanding of patch order from the positional embedding that is trained to be a specific pattern. Based on this observation, we propose explicitly adding a Gaussian attention bias that guides the positional embedding to have the corresponding pattern from the beginning of training. We evaluated the influence of Gaussian attention bias on the performance of ViTs in several image classification, object detection, and semantic segmentation experiments. The results showed that proposed method not only facilitates ViTs to understand images but also boosts their performance on various datasets, including ImageNet, COCO 2017, and ADE20K.

Introduction

*** Positional Embedding**

- Self-attention cannot understand the order of input patches.
- ViT uses separate positional embedding, such as APE or RPE, to reflect the order of patches.

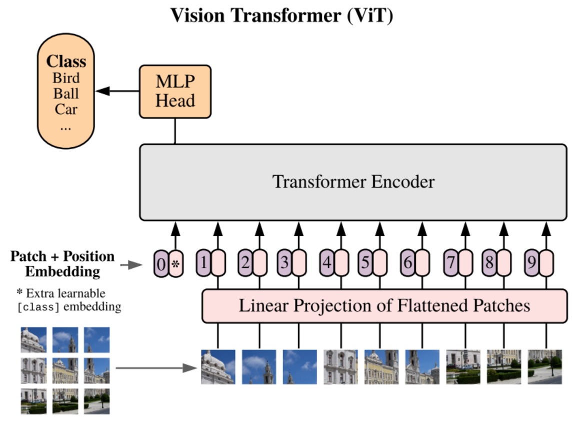


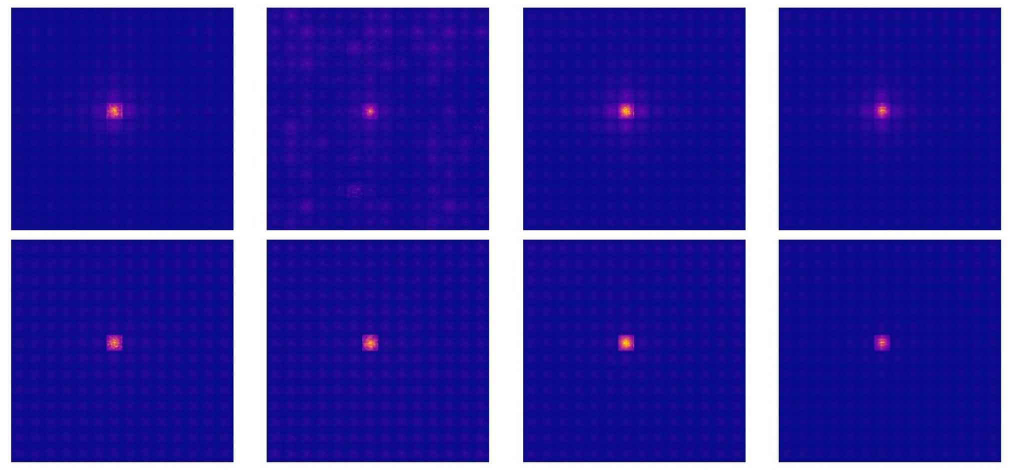
Figure Reference: Dosovitskiy et al. (ICLR 2021)

*** Our Contribution**

- Claim that the use of positional embedding does not simply guarantee the order-awareness of ViT.
- Analyze the actual behavior of ViTs using an effective receptive field.
- Propose explicitly adding a Gaussian attention bias that guides the positional embedding.
- Evaluate the influence of Gaussian attention bias on the performance of ViTs.

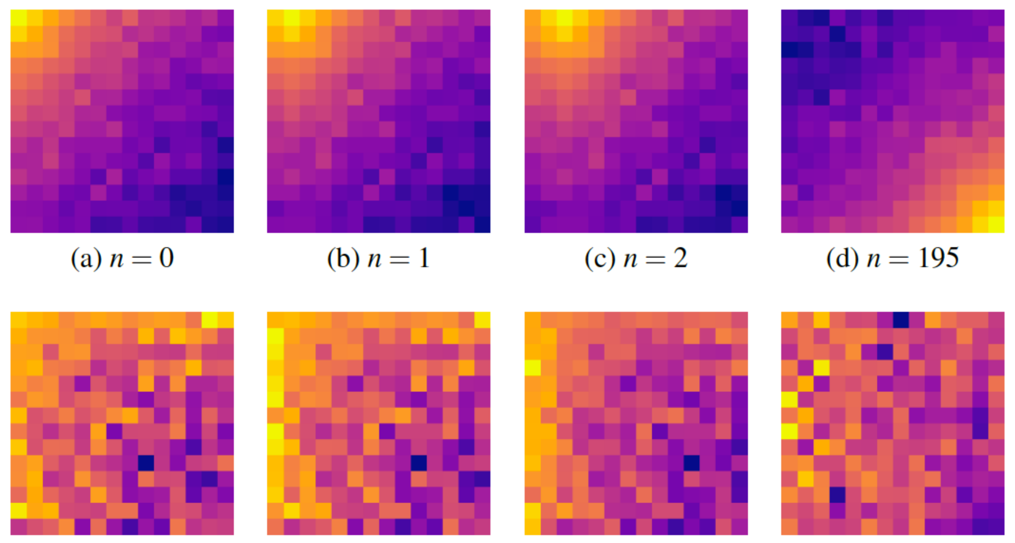
Analysis

We demonstrate that during training, ViT acquires an understanding of patch order from the positional embedding that is trained to be a specific pattern.



(a) ViT-S/16 (b) ViT-B/16 (c) ViT-M/16 (R) (d) ViT-B/16 (R)

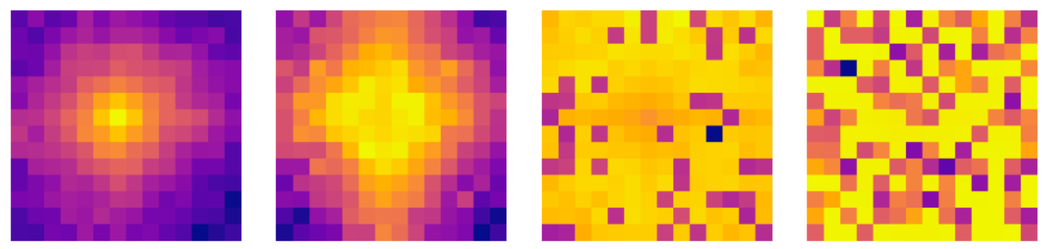
Figure 4: ERFs of ViTs, where (R) indicates the model with RPE. The second row illustrates ERFs when the APE or RPE is re-initialized to random parameters. Note that the ✚-shape is lost in the second row.



(a) $n=0$ (b) $n=1$ (c) $n=2$ (d) $n=195$

(e) $n=0$ (f) $n=1$ (g) $n=2$ (h) $n=195$

Figure 5: RPE of ViT-B/16 (R) for each patch index. The first row is obtained from the pretrained model, whereas the second row is obtained from the untrained model.



(a) $l=1$ (b) $l=10$ (c) $l=11$ (d) $l=12$

Figure 6: RPE corresponding to the center was extracted for each layer of ViT-B/16 (R).

	ViT-S/16, 224^2 (R)			ViT-M/16, 224^2 (R)			ViT-B/16, 224^2 (R)		
l	R^2	$\hat{\sigma}_X$	$\hat{\sigma}_Y$	R^2	$\hat{\sigma}_X$	$\hat{\sigma}_Y$	R^2	$\hat{\sigma}_X$	$\hat{\sigma}_Y$
1	0.731	6.837	7.063	0.893	4.219	4.113	0.914	4.553	4.394
2	0.798	4.704	4.538	0.728	6.257	5.679	0.573	6.672	6.719
3	0.831	6.185	6.392	0.824	4.715	5.039	0.870	4.649	4.849
4	0.867	4.757	5.020	0.838	5.250	5.355	0.813	4.901	5.404
5	0.753	6.798	5.310	0.795	5.597	4.920	0.853	5.055	4.807
6	0.730	5.624	4.631	0.694	8.054	5.540	0.817	5.421	4.276
7	0.796	5.872	4.848	0.844	5.509	4.660	0.877	6.895	5.020
8	0.805	4.865	5.473	0.798	5.715	5.010	0.825	5.640	4.006
9	0.771	5.668	5.681	0.729	5.472	6.538	0.873	5.328	4.914
10	0.786	5.111	6.125	0.878	4.430	5.348	0.896	5.342	6.132
11	0.231	8.709	272.743	0.359	5.824	298.676	0.012	21.137	702.646
12	0.019	690.530	181.928	0.002	396.639	415.174	0.004	579.639	332.651

Table 1: Results of fitting RPEs to a 2D Gaussian.

Proposed Method

* In light of the observation that learned RPE fits suitably with a 2D Gaussian, we propose injecting Gaussian attention bias into RPE:

$$\mathrm{Attention}_l(\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l) = \mathrm{softmax}\left(\frac{\mathbf{Q}_l\mathbf{K}_l^T}{\sqrt{D}} + \mathbf{B}_{\mathrm{rel},l} + \mathbf{B}_{\mathrm{Gaussian},l}\right)\mathbf{V}_l.$$

* Build Gaussian attention bias by reversing the process of extracting RPE.

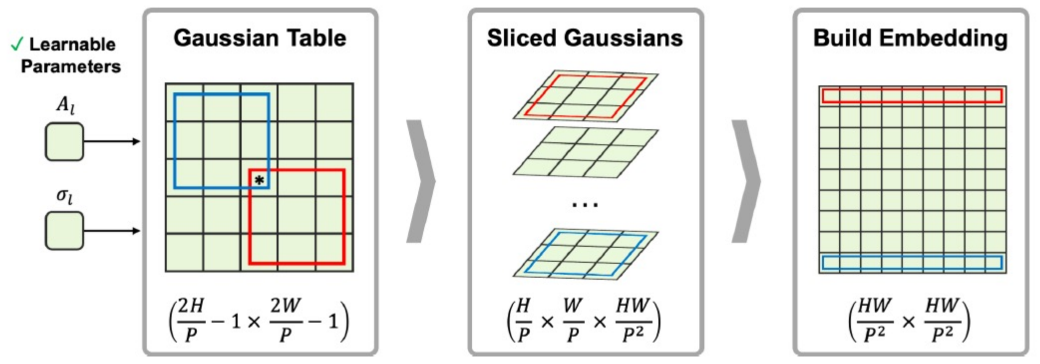


Figure 7: Illustration on how we obtain Gaussian attention bias.

* Generate a 2D Gaussian table using two learnable parameters:

$$f(x, y) = A_l^2 \exp\left(-\left(\frac{(x-x_c)^2}{2\sigma_l^2} + \frac{(y-y_c)^2}{2\sigma_l^2}\right)\right),$$

Design Choice

* Design as additive bias.
- It can be seamlessly plugged into any type of RPE.
- e.g. RelPosBias or RelPosMlp

* Use learnable parameters.
- Hyperparameter-free!
- Allow layer-wise freedom.

* Allow the learnability of the original RPE.
- Benefit from enriched expression in self-attention.

* Use a single Gaussian table.
- Sliced Gaussians are shifted versions of each other.
- Inspired by the use of relative coordinates.

* Do not use constant term in Gaussian.
- Softmax is invariant to constant translation.

* Share it across multiple heads of SA.
- But we observed a negligible effect.
- Validated from the ablation study.

* Do not apply weight decay to the two parameters.
- In PyTorch, explicitly specify not to apply weight decay.

Experiments

* We consistently observed improved performance after applying Gaussian Attention Bias.

*** Image Classification**
- ImageNet-1K

Dataset	Model	RPE w/o GAB	RPE w/ GAB	Difference
ImageNet-1K	ViT-S/16 (R)	80.567	80.724	+0.157
	ViT-M/16 (R)	81.224	81.249	+0.025
	ViT-B/16 (R)	81.381	81.484	+0.103

Table 2: Top-1 accuracy on the ImageNet-1K dataset. All the accuracies in this paper are expressed in percentage units. "GAB" indicates Gaussian attention bias.

*** Image Classification**
- Oxford-IIIT Pet, Caltech-101, Stanford Cars, Stanford Dogs

Dataset	Model	RPE w/o GAB	RPE w/ GAB	Difference
Oxford-IIIT Pet	ViT-S/16 (R)	91.486	92.780	+1.294
	ViT-M/16 (R)	92.810	92.960	+0.150
	ViT-B/16 (R)	93.381	93.743	+0.362
Caltech-101	ViT-S/16 (R)	88.403	90.202	+1.799
	ViT-M/16 (R)	89.132	89.983	+0.851
	ViT-B/16 (R)	89.254	89.570	+0.316
Stanford Cars	ViT-S/16 (R)	80.126	83.079	+2.953
	ViT-M/16 (R)	80.731	83.890	+3.159
	ViT-B/16 (R)	80.154	82.612	+2.458
Stanford Dogs	ViT-S/16 (R)	81.535	82.507	+0.972
	ViT-M/16 (R)	85.088	85.714	+0.626
	ViT-B/16 (R)	89.256	90.185	+0.929

Table 3: Test accuracy with and without Gaussian attention bias on other datasets.

*** Object Detection and Semantic Segmentation**
- COCO 2017, ADE20K

Backbone	RPE Method	COCO		ADE20K	
		AP^{box}	AP^{mask}	mIoU	aAcc
Swin-S	RelPosBias w/o GAB	48.12	43.03	46.16	81.82
	RelPosBias w/ GAB	48.23	43.13	46.41	82.09
	Difference	+0.11	+0.10	+0.25	+0.27

Table 4: Experimental results in terms of object detection and semantic segmentation.

*** Ablation Study**
- Comparison of head-shared and head-wise versions.

	RPE w/o GAB		RPE w/ GAB			
	Baseline		Head-shared		Head-wise	
Dataset	Val	Test	Val	Test	Val	Test
Oxford-IIIT Pet	93.682	91.486	93.923	92.780	93.773	92.509
Caltech-101	89.959	88.403	91.296	90.202	91.126	90.591
Stanford Cars	81.294	80.126	84.205	83.079	84.411	82.928
Stanford Dogs	82.777	81.535	83.188	82.507	83.501	81.438

Table 4: Comparison of head-shared and head-wise Gaussian attention bias. ViT-S/16 (R) was used for these experiments.