Rethinking and Improving Relative Position Encoding for Vision Transformer

Team E: Avirup Ghosh, Shiyi Gong, Shenyang Liu, Rishabh Patel

Published 2021 IEEE/CVF

Contents

- Background and previous work
- Four new relative position encoding methods
- Implementation details and Analysis on Relative Position Encoding
- Comparison on image classification and object detection and conclusion

Background and previous work

- Self-Attention
- Absolute Position Encoding and Relative Position Encoding (RPE)
- Shaw's RPE
- RPE in Transformer-XL
- Huang's RPE
- RPE in SASA
- RPE in Axial-Deeplab

Self-attention

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{x}_j \mathbf{W}^V). \tag{1}$$

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

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K)^T}{\sqrt{d_z}}. \tag{3}$$

Absolute Position Encoding and Relative Position Encoding (RPE)

- Absolute Positional encodir $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{p}_i,$$

- Relative Position Encoding (RPE)

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{x}_j \mathbf{W}^V + \mathbf{p}_{ij}^V), \qquad (5)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})},$$

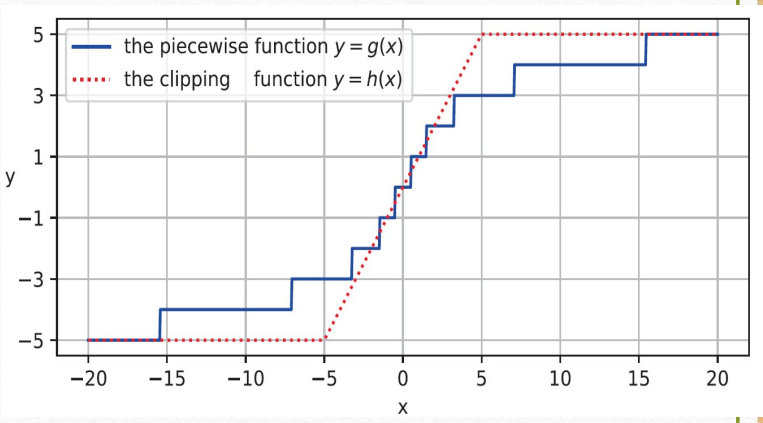
$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q + \mathbf{p}_{ij}^Q)(\mathbf{x}_j \mathbf{W}^K + \mathbf{p}_{ij}^K)^T}{\sqrt{d_z}}. \qquad (6)$$

Shaw's RPE

$$clip(x, k) = max(-k, min(k, x)),$$

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{x}_j \mathbf{W}^V + \mathbf{p}^V_{clip(i-j,k)}),$$

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K + \mathbf{p}^K_{clip(i-j,k)})^T}{\sqrt{d_z}},$$



RPE in Transformer-XL

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q + \mathbf{u})(\mathbf{x}_j \mathbf{W}^K)^T + (\mathbf{x}_i \mathbf{W}^Q + \mathbf{v})(\mathbf{s}_{i-j} \mathbf{W}^R)^T}{\sqrt{d_z}},$$

Huang's RPE

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q + \mathbf{p}_{ij})(\mathbf{x}_j \mathbf{W}^K + \mathbf{p}_{ij})^T - \mathbf{p}_{ij}\mathbf{p}_{ij}^T}{\sqrt{d_z}}, \quad (11)$$

RPE in SASA

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K + concat(\mathbf{p}^K_{\delta\tilde{x}}, \mathbf{p}^K_{\delta\tilde{y}}))^T}{\sqrt{d_z}}, \quad (12)$$

$$\delta\tilde{x} = \tilde{x}_i - \tilde{x}_j$$

$$\delta\tilde{y} = \tilde{y}_i - \tilde{y}_j$$

RPE in Axial-Deeplab

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V + p_{ij}^V)$$

$$e_{ij} = \frac{\left(x_i \mathrm{W}^{\mathrm{Q}} + p_{ij}^Q\right)\left(x_j \mathrm{W}^K + p_{ij}^K\right)^T - p_{ij}^Q p_{ij}^{K^T}}{\sqrt{d_z}}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})},$$

New relative position encoding methods

- Bias Mode
- Contextual Mode
- A Piecewise Index Function
- 2D Relative Position Calculation

Bias and Contextual Mode

$$e_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K)^T + b_{ij}}{\sqrt{d_z}}, \tag{13}$$

- Bias
- Contextual

$$b_{ij} = r_{ij}, \tag{14}$$

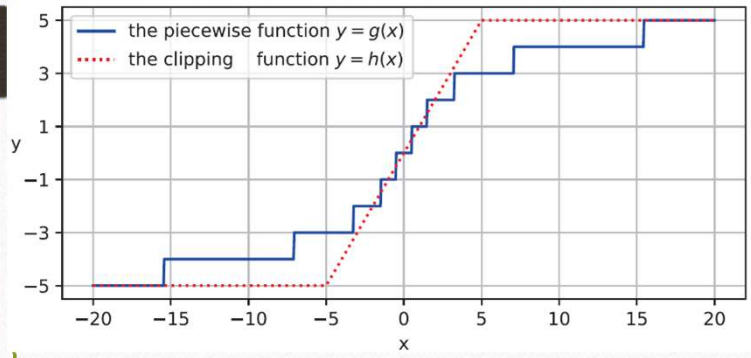
$$b_{ij} = (\mathbf{x}_i \mathbf{W}^Q) \mathbf{r}_{ij}{}^T, \tag{15}$$

$$b_{ij} = (\mathbf{x}_i \mathbf{W}^Q)(\mathbf{r}_{ij}^K)^T + (\mathbf{x}_j \mathbf{W}^K)(\mathbf{r}_{ij}^Q)^T, \tag{16}$$

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} (\mathbf{x}_j \mathbf{W}^V + \mathbf{r}_{ij}^V), \tag{17}$$

A Piecewise Index Function

- mapping a relative distance into an integer in finite set



$$clip(x, k) = max(-k, min(k, x)), \tag{9}$$

$$g(x) = \begin{cases} [x], & |x| \leq \alpha \\ sign(x) \times min(\beta, [\alpha + \frac{\ln(|x|/\alpha)}{\ln(\gamma/\alpha)}(\beta - \alpha)]), & |x| > \alpha \end{cases} \tag{18}$$

2D Relative Position Calculation

- Euclidean method:

$$\mathbf{r}_{ij} = \mathbf{p}_{I(i,j)}, \tag{19}$$

$$I(i,j) = g(\sqrt{(\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2}\ ), \tag{20}$$

- Quantization method

$$I(i,j) = g(quant(\sqrt{(\tilde{x}_i - \tilde{x}_j)^2 + (\tilde{y}_i - \tilde{y}_j)^2})\ ). \tag{21}$$

2D Relative Position Calculation

- Cross method

$$\mathbf{r}_{ij} = \mathbf{p}^{\tilde{x}}_{I^{\tilde{x}}(i,j)} + \mathbf{p}^{\tilde{y}}_{I^{\tilde{y}}(i,j)}, \tag{22}$$

$$I^{\tilde{x}}(i,j) = g(\tilde{x}_i - \tilde{x}_j), \tag{23}$$

$$I^{\tilde{y}}(i,j) = g(\tilde{y}_i - \tilde{y}_j), \tag{24}$$

- Product method

$$\mathbf{r}_{ij} = \mathbf{p}_{I^{\tilde{x}}(i,j),I^{\tilde{y}}(i,j)}. \tag{25}$$

Experiments

- Compare each of the proposed methods
- Compare proposed methods to SOTA image classification and object detection
- Visualization of RPE

Implementation details

- DeiT with same training settings
- RPE added to keys
- Ratio for piecewise function constants
  - $\alpha:\beta:\gamma = 1:2:8$

$$g(x) = \begin{cases} [x], & |x| \leq \alpha \\ sign(x) \times min(\beta, [\alpha + \frac{\ln(|x|/\alpha)}{\ln(\gamma/\alpha)}(\beta - \alpha)]), & |x| > \alpha \end{cases} \quad (18)$$

- Add 1 to β (# buckets) for RPE of classification token

Directed vs. Undirected
Bias vs. Contextual

Method based on DeiT-S [21]	Is Directed	Mode	Top-1 Acc(%)	Δ Acc(%)
Original [21]	-	-	79.9	–
Euclidean	×	bias	80.1	+0.2
		contextual	80.4	+0.5
Quantization	×	bias	80.3	+0.4
		contextual	80.5	+0.6
Cross	✓	bias	80.5	+0.6
		contextual	80.8	+0.9
Product	✓	bias	80.5	+0.6
		contextual	80.9	+1.0

Shared v.s. Unshared

Mode	Shared	#Param. (M)	MACs (M)	Top-1 Acc(%)
Bias	×	22.05	4613	80.54 ± 0.06
	✓	22.05	4613	80.05 ± 0.04
Contextual	×	22.28	4659	80.99 ± 0.16
	✓	22.09	4659	80.89 ± 0.04

Piecewise v.s. Clip

Image Classification with DeiT

Function	Mode	Top-1 Acc(%)	Top-5 Acc(%)
clip	bias	80.1	94.9
	contextual	80.9	95.5
piecewise	bias	80.0	95.0
	contextual	80.9	95.5

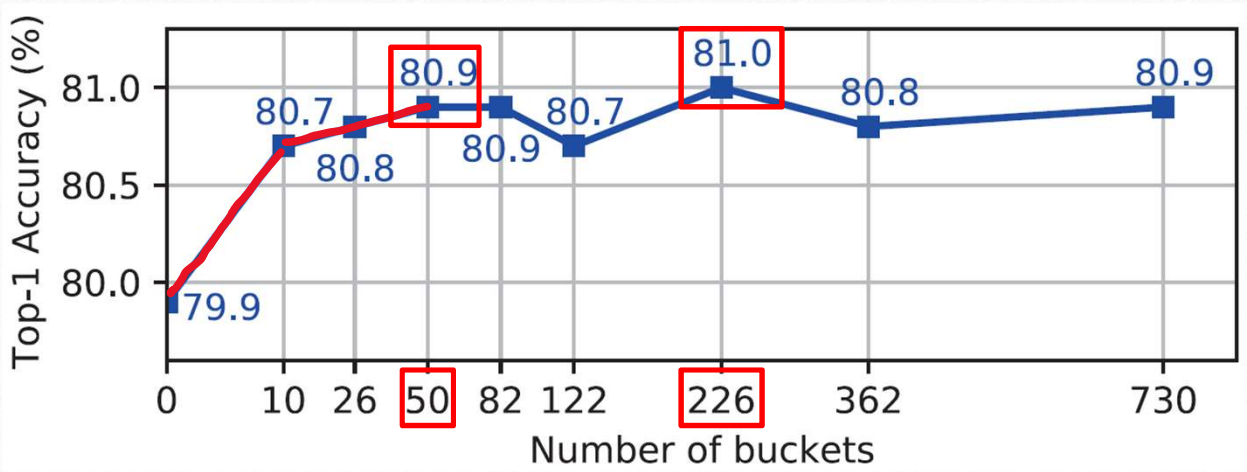
$$clip(x, k) = max(-k, min(k, x)), \tag{9}$$

$$g(x) = \begin{cases} [x], & |x| \leq \alpha \\ sign(x) \times min(\beta, [\alpha + \frac{\ln(|x|/\alpha)}{\ln(\gamma/\alpha)}(\beta - \alpha)]), & |x| > \alpha \end{cases} \tag{18}$$

Object Detection with DETR

5	sinusoid	ctx clip	9×9	150	40.4	60.9	42.4	19.1	43.7	59.8
6	sinusoid	contextual	9×9	150	40.8	61.5	42.5	18.5	44.4	60.5

Number of buckets



Component-wise analysis

#	Abs Pos.	p_{ij}^{Q}	p_{ij}^{K}	p_{ij}^{V}	Top-1	Top-5
1 [21]	learnable	×	×	×	79.9	95.0
2	×	×	×	×	77.6(-2.3)	93.8
3	×	✓	×	×	80.9(+1.0)	95.4
4	×	×	✓	×	80.9(+1.0)	95.3
5	×	×	×	✓	80.2(+0.3)	95.0
6	×	✓	✓	×	81.0(+1.1)	95.5
7	×	✓	✓	✓	81.3(+1.4)	**95.7**
8	learnable	✓	×	×	80.9(+1.0)	95.5
9	learnable	×	✓	×	80.9(+1.0)	95.5
10	learnable	×	×	✓	80.2(+0.3)	95.1
11	learnable	✓	✓	×	81.1(+1.2)	95.4
12	learnable	✓	✓	✓	**81.4**(+1.5)	95.6

Complexity Analysis

- Inefficient Implementation:

$$y_{ij} = (\mathbf{x}_i \mathbf{W}) \mathbf{p}_{I(i,j)}{}^T.$$

- Efficient Implementation:

$$z_{i,t} = (\mathbf{x}_i \mathbf{W}) \mathbf{p}_t{}^T, t \in \{I(i,j) | i,j \in [0,n)\},$$

$$y_{ij} = z_{i,I(i,j)}.$$



Comparison on image classification

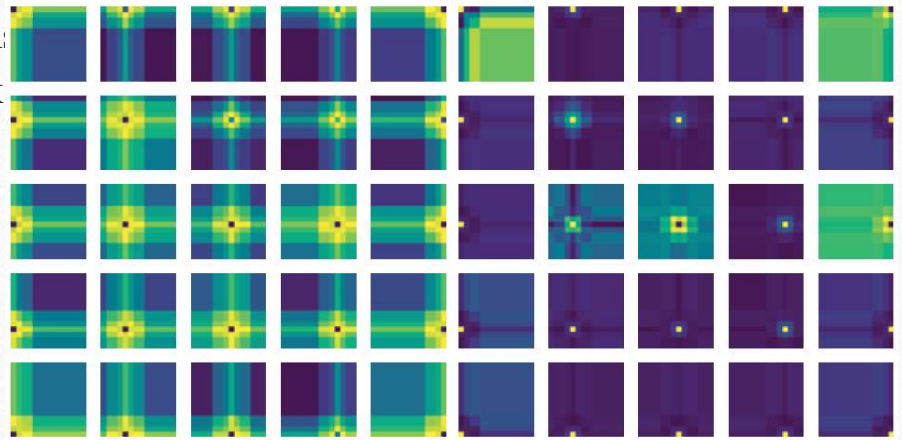
Model	#Param.	Input	MACs (M)	Top-1 Acc (%)
Convnets				
ResNet-50 [10]	25M	224^2	4121	79.0
RegNetY-4.0GF [14]	21M	224^2	4012	79.4
EfficientNet-B1 [20]	8M	240^2	712	79.1
EfficientNet-B5 [20]	30M	456^2	10392	83.6
Transformers				
ViT-B/16 [7]	86M	384^2	55630	77.9
ViT-L/16 [7]	307M	384^2	191452	76.5
DeiT-Ti [21]	5M	224^2	1261	72.2
CPVT-Ti(0-5) [3]	6M	224^2	1262	73.4
DeiT-Ti with iRPE-K(**Ours**)	6M	224^2	1284	73.7
DeiT-S [21]	22M	224^2	4613	79.9
CPVT-S(0-5) [3]	23M	224^2	4616	80.5
DeiT-S(Shaw's) [17, 21]+	22M	224^2	4659	80.9
DeiT-S(Trans.-XL's) [4, 21]+	23M	224^2	4828	80.8
DeiT-S(Huang's) [11, 21]+	22M	224^2	4706	81.0
DeiT-S(SASA's) [16, 21]*	22M	224^2	4639	80.8
DeiT-S with iRPE-K(**Ours**)	22M	224^2	4659	80.9
DeiT-S with iRPE-QK(**Ours**)	22M	224^2	4706	81.1
DeiT-S with iRPE-QKV(**Ours**)	22M	224^2	4885	81.4
DeiT-B [21]	86M	224^2	17592	81.8
CPVT-B(0-5) [3]	86M	224^2	17598	81.9
DeiT-B with iRPE-K(**Ours**)	87M	224^2	17684	82.4

Comparison on object detection

#	Abs Pos.	Rel Pos.	#buckets	epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1 [1]	sinusoid	none	-	150	39.5	60.3	41.4	17.5	43.0	59.1
2	none	none	-	150	30.4(-9.1)	52.5	30.2	9.4	31.2	50.5
3	sinusoid	bias	9×9	150	40.6(+1.1)	61.2	42.8	19.0	43.9	60.2
4	none	contextual	9×9	150	38.7(-0.8)	60.1	40.4	18.2	41.8	56.7
5	sinusoid	ctx clip	9×9	150	40.4(+0.9)	60.9	42.4	19.1	43.7	59.8
6	sinusoid	contextual	9×9	150	40.8(+1.3)	61.5	42.5	18.5	44.4	60.5
7	sinusoid	contextual	15×15	150	40.8(+1.3)	61.7	42.6	18.5	44.2	61.2
8 [1]	sinusoid	none	-	300	40.6	61.6	-	19.9	44.3	60.2
9	sinusoid	contextual	9×9	300	42.3(+1.7)	62.8	44.3	20.7	46.2	61.1

Visualization

- Visualize the extra for different position
- RPE makes patches focus more in block



(a) block 0 (b) block 10

Conclusion

- Review existing relative position encoding methods
- Propose four methods dedicated to visual transformers.
- The experiments show effectiveness of their methods on both classification and detection tasks.
- Their methods are easily adapted to other models.

Reference

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to- end object detection with transformers. In ECCV, 2020. 1,2, 7, 8

[2] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In ICCV, 2021. 1

[3] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? arXiv preprint arXiv:2102.10882, 2021. 1, 7, 8

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In ACL, 2019. 1, 3, 7, 8

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 2, 6, 7

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 1