Reviving Shift Equivariance in Vision Transformers

Peijian Ding[1] **Davit Soselia**[1] **Thomas Armstrong**[1] **Jiahao Su**[2] **Furong Huang**[1]

Abstract

Shift equivariance, integral to object recognition, is often disrupted in Vision Transformers (ViT) by components like patch embedding, subsampled attention, and positional encoding. Attempts to combine convolutional neural network with ViTs are not fully successful in addressing this issue. We propose an input-adaptive polyphase anchoring algorithm for seamless integration into ViT models to ensure shift-equivariance. We also employ depth-wise convolution to encode positional information. Our algorithms enable ViT, and its variants such as Twins to achieve 100% consistency with respect to input shift, demonstrate robustness to cropping, flipping, and affine transformations, and maintain consistent predictions even when the original models lose 20 percentage points on average when shifted by just a few pixels with Twins' accuracy dropping from 80.57% to 62.40%.

1. Introduction

Inductive bias helps guide machine learning algorithm design by reducing the optimal model's search space. Convolutional Neural Networks (CNNs) owe their success to their shift equivariance inductive bias, mirroring human object recognition abilities. However, Transformers, though successful in computer vision and natural language processing, lack shift-equivariance due to patch embedding, positional embedding, and subsampled attention.

Attempts to integrate CNNs into vision transformers have only partially addressed this issue. Although Dai et al. (2021) propose a relative attention method that combines depthwise convolution with attention to achieve shift equivariance, this approach requires computing full global attention, and shift equivariance is not maintained when downsampled attention is required for computational efficiency (Chu et al., 2021; Tu et al., 2022; Ding et al., 2022). Both MaxViT (Tu et al., 2022) and Twins transformer (Chu et al., 2021) utilize depth-wise convolution to encode positional information, but their block attention (or window attention) and strided convolution are not shift equivariant.

Our work introduces modules that fully incorporate CNN's shift-equivariant inductive bias into vision transformers. We propose a nonlinear operator - the polyphase anchoring algorithm - ensuring shift-equivariance by choosing the maximum polyphase as anchors for strided convolution and subsampled attentions. We also employ depthwise convolution with circular padding to encode positional information.

Our contributions include versatile modules that improve vision transformer models' performance, an adaptive nonlinear operator ensuring shift-equivariance, and complete shift-equivariance capabilities for vision transformers, backed by theoretical and empirical evidence.

2. Approach

To achieve model-wise shift-equivariance, we first detect the modules that lack shift equivariance. We then introduce a polyphase anchoring algorithm to ensure shift-equivariance for strided convolution and two popular types of subsampled attention — window attention, a widely used local subsampled attention (Liu et al., 2021; 2022; Tu et al., 2022; Chu et al., 2021), and global subsampled self-attention, a popular choice for global subsampled attention (Chu et al., 2023; 2021). Finally, we use depthwise convolution with circular padding to guarantee shift-equivariance in positional encoding. As the composition of shift-equivariant functions remains shift-equivariant, we obtain a shift-equivariant model.

2.1. Detecting modules lacking shift equivariance

To create a shift-equivariant model, we identify and address non-shift-equivariant modules in vision transformers (ViTs). We find that *patch embedding*, *positional encoding*, and *subsampled attentions* like window and global subsampled self-attention lack shift-equivariance. In contrast, normalization and MLP layers are shift-equivariant.

To tackle these issues, we introduce a polyphase an-

[1]University of Maryland [2]ByteDance Ltd. Correspondence to: Peijian Ding <pding@umd.edu>, Davit Soselia <dsoselia@umd.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

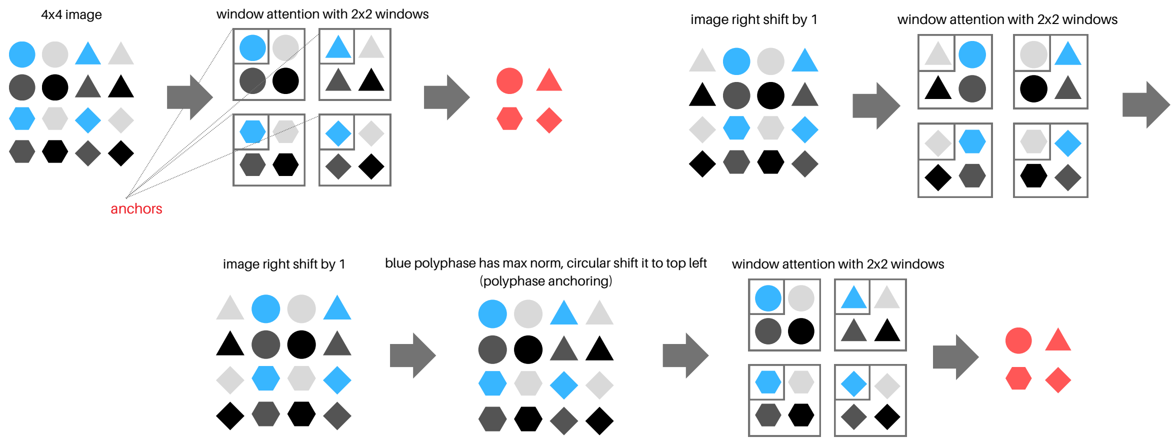
4x4 image
window attention with 2x2 windows
anchors
image right shift by 1
window attention with 2x2 windows
image right shift by 1
blue polyphase has max norm, circular shift it to top left
(polyphase anchoring)
window attention with 2x2 windows

Figure 1. The maximum polyphase is colored in blue. Each shape with distinct color represents a token. We also illustrate the concept of *anchors* — the top left coordinate in each window. The red and yellow shapes indicate that window attention produces inconsistent predictions on shifted image whereas the composition of polyphase anchoring and window attention does not.

Table 1. ImageNet1K training from scratch

Model	image size	#param.	Epochs	Acc. IN1K	Consis.	Acc. Rand. S
ViT_S	224^2	22M	300	75.52	86.61	74.98
ViT_S-poly	224^2	22M	300	**76.37**	**100**	**76.37**
ViT_B	224^2	87M	300	73.85	85.60	73.01
ViT_B-poly	224^2	86M	300	**74.62**	**100**	**74.62**
Twins_B	224^2	56M	300	80.57	91.25	79.90
Twins_B-poly	224^2	56M	300	**80.59**	**100**	**80.59**

choring algorithm for strided convolution, window attention, and global subsampled self-attention, ensuring shift-equivariance. Additionally, we use depthwise convolution with circular padding to guarantee shift-equivariance in positional encoding. With these modifications, we deliver a fully shift-equivariant model, as the composition of shift-equivariant functions remains shift-equivariant.

2.2. Polyphase anchoring algorithm

Inspired by the concept of adaptive polyphase sampling presented in Chaman and Dokmanic (2021), we propose the polyphase anchoring algorithm, an efficient technique that can be seamlessly integrated with various types of subsampled attention operators (Tu et al., 2022; Chu et al., 2021; Liu et al., 2021; 2022; Dong et al., 2021; Ding et al., 2022) to ensure shift-equivariance. The algorithm is implemented as an autograd function in PyTorch, making it simple to incorporate into deep learning models.

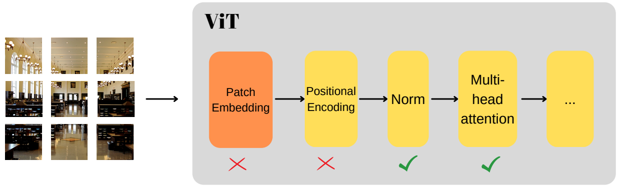

Figure 2. The figure highlights that, in the context of ViT, patch embedding and positional encoding do not exhibit shift-equivariant properties.

Polyphase anchoring identifies the maximum L_p norm polyphase and shifts the input accordingly, so that the maximum polyphase aligns with the anchor positions of window attention, as illustrated in Figure 1. *Anchors* S_A of window attention represent the set of coordinates at the top-left of each window, as depicted in Figure 1.

$$S_A = \{(i, j) \mid i \equiv 0 \pmod s, j \equiv 0 \pmod s\} \quad (1)$$

Table 2. Robustness experiments on ImageNet1K

Model	Acc. Crop	Acc. Flip	Acc. Affine	Worst-of-30 Batch 1	Worst-of-30
ViT_S	75.09	75.50	69.85	53.80	68.90
ViT_S-poly	**76.08**	**76.34**	69.54	**76.20**	**76.02**
ViT_B	73.31	73.83	68.64	53.20	67.42
ViT_B-poly	**74.36**	**74.64**	**70.46**	**74.40**	**74.20**
Twins_B	80.51	80.60	75.88	62.40	73.86
Twins_B-poly	80.43	80.56	**76.12**	**80.78**	**80.78**

where $i \leq H, j \leq W, i, j \in \mathbb{Z}$, $s \times s$ denotes the size of the window in window attention, (i, j) is a coordinate on a 2D grid.

Algorithm 1 demonstrates polyphase ordering. For brevity, we define the polyphase X_{pq} mathematically here. Let X_{pq} be tokens in a polyphase defined by coordinate (p, q) and stride size s, where $X_{pq} = X[:, p :: s, q :: s]$.

Algorithm 1 Polyphase anchoring

```
**Input** $X \in R^{\cdots \times H \times W}$, stride size $s \in \mathbb{Z}$
$\hat{X}_{pq} = \arg\max_{\{X_{pq} | p, q \in \{0, \ldots, s-1\}\}} ||X_{pq}||$
$\hat{X} = g_{pq} \cdot (X)$ where $g_{pq}$ circularly shifts $X$ by $(-p, -q)$ along the last two dimensions.
**Output**: $\hat{X}$
```

The polyphase anchoring algorithm is a nonlinear operator that conditionally shifts the input based on its maximum L_p norm polyphase. This guarantees shift-equivariance in strided convolution, subsampled attention like window attention, and GSA. As a result, the lack of shift-equivariance in patch embedding modules and subsampled attention modules in ViT variants, such as Twins (Chu et al., 2021), is addressed.

We provide theoretical guarantee that the composition of polyphase anchoring with strided convolution, window attention, and global subsampled self-attention results in shift-equivariant operations. Consequently, we effectively address the lack of shift equivariance in patch embedding modules and subsampled attention modules for ViT variants such as Twins (Chu et al., 2021). Detailed proofs are provided in Appendix.

2.3. Positional Embedding

Positional embedding like absolute (Dosovitskiy et al., 2020) and relative positional embedding (Liu et al., 2021; 2022) can disrupt shift-equivariance. Using zero-padded depthwise convolution, as introduced by Chu et al. (2023), we can promote shift-equivariance. With circularly-padded depthwise convolution replacing positional encoding, shift-equivariance is achieved.

Formally, we define depthwise convolution operation using input tensor $\mathbf{X}$ and a set of depthwise filters $\mathbf{W}$. Assuming circular padding, each channel's convolution is shift-equivariant, making depthwise convolution shift-equivariant overall.

Ensuring shift-equivariance in patch embedding, positional embedding, and subsampled attention leads to a fully shift-equivariant model. Moreover, adding a shift-invariant pooling operation in the classification head can yield a truly shift-invariant model.

3. Experiments

In this section, we demonstrate that we can construct a 100% shift-equivariant ViT and Twins transformers. Models employing our algorithm exhibit superior accuracy in fair comparisons, improved robustness under shifting, cropping, flipping, and random patch erasing, 22.4% relative percentage point gain (or 41.6% increase) from ViT small under worst-of-30 shift attack, and 100% consistency under shift attacks.

Settings. We evaluate six architectures, including ViT base, ViT small, Twins (Chu et al., 2021) base, and their shift-equivariant counterparts using polyphase anchoring and depthwise convolution on ImageNet-1k. We use Twins-SVT introduced by Chu et al. (2021). To ensure fair comparisons, we train each model and its counterpart from scratch on ImageNet-1k under identical training settings including hyperparameters and data augmentation strategy.

3.1. Accuracy and Consistency on ImageNet-1k

Evaluation. We measure performance using accuracy, consistency, and accuracy under small random shift from 0 to 15 pixels. Consistency (Chaman and Dokmanic, 2021) measures the likelihood of the model assigning the image and its shifted copy to the same class.

Results. Table 1 shows the comparison of ViT and Twins transformer models against their shift-equivariant counterparts using models training from scratch. Shift-equivariant models demonstrate superior accuracy under random shifts, and 100% consistency.

3.2. Robustness tests on ImageNet-1k

Evaluations. We evaluate the the models under random cropping, horizontal flipping, patch erasing, and affine transformations. Additionally, we perform a worst-of-k shift attack for each batch of images, we keep the shift within a small range of $(-15, 15)$, to keep it inconsequential to human perception, and use the worst-case shift for evaluation. Some of these metrics are sensitive to the batch size used since the worst shift is chosen per batch. We use a batch size of 64 for all metrics and additionally evaluate worst-of-30 with a batch size of 1 for 2000 samples.

Results. As shown in Table 2, shift-equivariant models obtain comparable or better accuracy than their respective counterparts. Under the worst-of-k shift attack, our models achieve significantly improved accuracy and consistency, while having slight-to-high gains on the other transformations.

3.3. Stability and shift-equivariance tests on ImageNet-1K

Output logits variance measures the variability of the model's logits predictions with respect to a range of small random shifts from -5 to 5. It quantifies the spread or dispersion of the logits (L) as a function of the input shift. Mathematically, the output logits variance can be calculated as follows:

$$\text{Variance} = \frac{1}{N}\sum_{i=1}^{N}\left(L(x_i) - \bar{L}\right)^2 \tag{2}$$

where Variance represents the output logits variance, N is the total number of samples, x_i denotes the input sample, $L(x_i)$ corresponds to the logits prediction for the input x_i, and $\bar{L}$ represents the mean logits prediction for the given range of input shifts. The output logits variance concerning small shift perturbations is almost zero for ViT_S/16-poly and Twins_B-poly, indicating that the output logits remain unchanged under small input shifts. Conversely, the output logits variance is nonzero for nearly 50% of the input images, suggesting that the model's assigned probability for the input label alters in response to minor pixel shifts.

Shift-equivariance tests are unit tests that measure if the features are shift-equivariant. Let $\mathcal{M}$ be a machine learning model that takes an input $\mathbf{X}$ and produces a feature map $\mathbf{F} = \mathcal{M}(\mathbf{X})$. For a given translation $g \in G$, define the shifted input $\mathbf{X}'$ as $\mathbf{X}' = g \cdot \mathbf{X}$. Let $\mathbf{F}' = \mathcal{M}(\mathbf{X}')$ be the feature map obtained by applying the model to the shifted input. The feature shift-equivariance test can be defined as follows:

$$\text{shift-equivariance}(\mathcal{M}) = \begin{cases} 0, & \text{if } \mathbf{F}' = g' \cdot \mathbf{F} \\ \|\mathbf{F}' - \mathbf{F}\|, & \text{otherwise} \end{cases} \tag{3}$$

where $g' \in G$ is translation in the feature space, $\|\cdot\|$ is L_2 norm. Both the polyphase models of ViT and Twins successfully pass all shift equivariance tests in the feature space, while the original ViT and Twins models fail to do so and exhibit substantial norm differences in the feature space.

4. Related Work

Data augmentation encourages shift-equivariance by adding shifted copies of images to the training set but lacks guarantees. In CNNs, it has been shown that models learn invariance to transformations only for images similar to typical training set images (Azulay and Weiss, 2018).

Regularization during training encourages shift-equivariance and invariance using soft constraints. A pretraining objective during self-supervised learning can be added to predict transformations applied to the input (Dangovski et al., 2022). A loss function based on cross-correlation of embedded features encourages equivariance (Xie et al., 2022) but does not provide theoretical guarantee.

Architectural design can also result in shift-equivariance and invariance. For CNNs, anti-aliasing strategies (Zhang, 2019) and adaptive polyphase sampling (APS) address the lack of shift-equivariance due to downsampling. The former lacks guarantees, while the latter is computationally expensive, needing full convolution computations on subsampled attentions.

5. Conclusions and Discussions

This work fully revives shift equivariance in vision transformers using versatile modules — polyphase anchoring and depthwise convolution. We detect common modules in vision transformers that lack shift equivariance and propose input-adaptive nonlinear operator that ensures shift-equivariance in patch embedding layer and subsampled attention blocks. While this study prioritized comparative analysis over achieving state-of-the-art accuracy due to computational constraints, we demonstrated superior performance in certain areas like consistency in classification and robustness to worst-case shift, cropping, flipping, and affine transformation. Future work includes leveraging industrial-scale resources for improved performance.

References

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *CoRR*, abs/1805.12177, 2018. URL http://arxiv.org/abs/1805.12177.

Anadi Chaman and Ivan Dokmanic. Truly shift-invariant convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3773–3783, June 2021.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=5kTlVBkzSRx.

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers, 2023.

Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes, 2021.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gKLAAfiytI.

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 74–92. Springer, 2022.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Nate Gruver, Marc Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution, 2022.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *ECCV*, 2022.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

Yuyang Xie, Jianhong Wen, Kin Wai Lau, Yasar Abbas Ur Rehman, and Jiajun Shen. What should be equivariant in self-supervised learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4110–4119, 2022. doi: 10.1109/CVPRW56347.2022.00456.

Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008, 2021.

Richard Zhang. Making convolutional networks shift-invariant again, 2019.

A. Appendix

To effectively restore shift-equivariance in vision transformers, it is necessary to examine each individual module and identify the components responsible for disrupting the shift-equivariance property. To achieve this, we must first establish formal definitions for equivariance and self-attention.

A.1. Mathematical background

A.1.1. Equivariance

Equivariance serves as a formal concept of consistency under transformations (Gruver et al., 2022). A function $f : V_1 \to V_2$ is considered equivariant to transformations from a symmetry group G if applying the symmetry to the input of f produces the same result as applying it to the output:

$$\forall g \in G : f(g \cdot x) = g' \cdot f(x) \tag{4}$$

Here, $\cdot$ denotes the linear mapping of the input by the representation of group elements in G. Throughout this paper, all instances of $\cdot$ adhere to this definition. When $g = g'$, the function is referred to as G-equivariant. If g' is the identity, the function is G-invariant. For cases where $g \neq g'$, the function is considered generally equivariant. General equivariance is a valuable concept when the input and output spaces have different dimensions. When G represents the translation group, the above definition yields shift-equivariance.

A.1.2. Self-attention

A self-attention operator A_s exhibits permutation-equivariance. Let X represent the input matrix, and T_π denote any spatial permutation. We can express this as:

$$A_s(T_\pi(X)) = T_\pi(A_s(X)). \tag{5}$$

A_s is the self-attention operator with parameter matrices $W_q \in \mathbb{R}^{d \times d_k}$, $W_k \in \mathbb{R}^{d \times d_k}$, and $W_v \in \mathbb{R}^{d \times d_v}$:

$$\begin{aligned} A_s &= \text{SoftMax}(XW_q(XW_k)^T)XW_v && (6) \\ &= \text{SoftMax}(QK^T)V. && (7) \end{aligned}$$

A.2. Detecting modules lacking shift equivariance

We elaborate section 2.1 from the main manuscript in this section and detect modules that lack shift equivariance. Vision transformers consist of a patch embedding layer, positional encoding, transformer blocks, and MLP layers. We analyze each of these modules in ViT and its variants, discovering the following:

- Patch embedding layer (strided convolution) is not shift-equivariant due to downsampling.
- Absolute positional encoding (Dosovitskiy et al., 2020) and relative positional embedding (Liu et al., 2021; 2022) are not shift-equivariant.
- Normalization, global self-attention, and MLP layers are shift-equivariant.
- Subsampled attentions such as window attention (Liu et al., 2021; 2022; Tu et al., 2022; Chu et al., 2021) and global subsampled self-attention (Chu et al., 2021) are not shift-equivariant.

Patch embedding converts image patches into sequence vector representations through strided convolution. However, strided convolution is not shift-equivariant due to downsampling, as addressed by Zhang (2019); Chaman and Dokmanic (2021). Figure 1 illustrates that the image patching layer, or strided convolution, is not shift-equivariant. When an image is shifted by a pixel, the pixels in each window change, leading to different computations.

Positional encoding is a method for incorporating spatial location information into tokens, the representations of input image patches. Popular positional encoding techniques such as absolute positional encoding in ViT (Dosovitskiy et al., 2020)

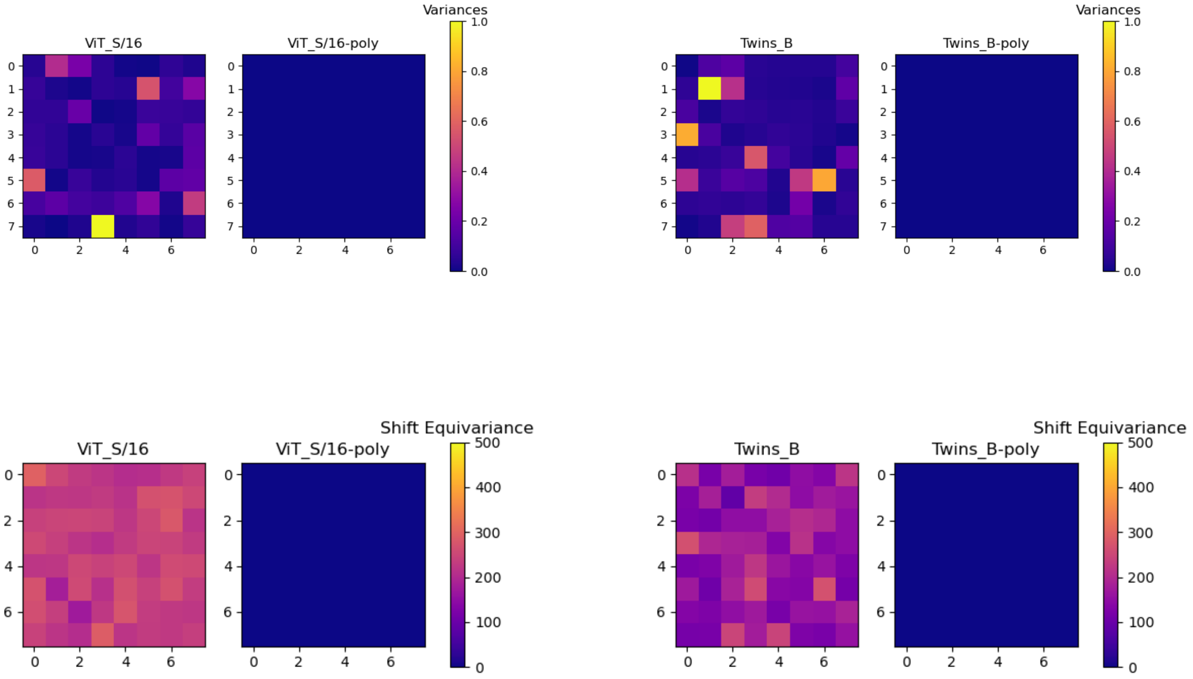

Figure 3. The top figures display the output logits variance for ViT_S, Twins_B, and their shift-equivariant counterparts, while the bottom figures provide a comparison of shift-equivariance tests between ViT_S, Twins_B, and their respective shift-equivariant versions.

and relative positional encoding in Swin (Liu et al., 2021; 2022) are not shift-equivariant. Absolute positional encoding (Dosovitskiy et al., 2020) adds the absolute positional information to input tokens by considering an input image as a sequence or a grid of patches (Dosovitskiy et al., 2020; Wang et al., 2021). Trivially, absolute positional embedding is not shift-equivariant because the same absolute positional information is added to the input tokens regardless of shift, as shown in Figure 4 The relative positional embedding introduced by Liu et al. (2021) is the following:

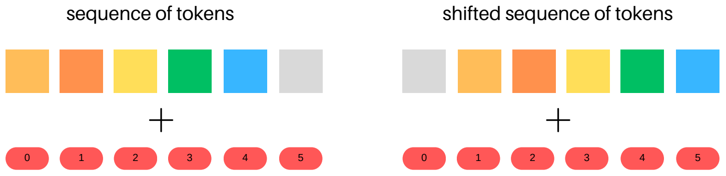

Figure 4. The figure illustrates that the identical absolute positional encoding is applied to both the input and its circularly shifted counterpart, resulting in a lack of shift-equivariance.

$$\text{Attention}(Q, K, V) = \text{SoftMax}(\frac{QK^T}{\sqrt{d}} + B)V, \tag{8}$$

where $B \in R^{M^2 \times M^2}$ is the relative position bias term for each head; $Q, K, V \in R^{M^2 \times d}$ are the query, key and value matrices; d is the query/key dimension, and M^2 is the number of patches in a window. Although self-attention is permutation equivariant, self-attention with relative position bias is not shift-equivariant. Further mathematical deductions are provided in section A.3.1.

Normalization layers standardize the input data or activations of preceding layers to stabilize training and enhance model performance by ensuring consistent scales and distributions. Both batch normalization and layer normalization are shift-equivariant. Trivially, normalizing a shifted input along batch and feature dimensions is equivalent to shifting the normalized input.

MLP layers, or Multi-Layer Perceptron layers, are a sequence of feedforward neural network layers that perform a linear transformation followed by a non-linear activation function. An MLP layer is shift-equivariant. In layer l of an MLP model, we have:

$$h^{(l)} = \phi(xW^{(l)} + b^{(l)}) \tag{9}$$

where x is a row vector, W is a weight matrix, and b is a bias term. Given an input matrix X whose row vectors are tokens, it is obvious that the MLP layer is shift equivariant with respect to input tokens.

In the ViT architecture, we have identified that MLP layers and normalization layers are shift-equivariant, while patch embedding and positional encoding are not. ViT variants (Liu et al., 2021; 2022; Chu et al., 2023; Dong et al., 2021; Wang et al., 2021; Tu et al., 2022) introduce additional challenges for shift-equivariance, as they typically employ subsampled attention operations to reduce the quadratic computational complexity with respect to the number of tokens in global self-attention.

Subsampled attentions are streamlined versions of global self-attentions that can be classified into two categories: local and global (Liu et al., 2021; 2022; Chu et al., 2021; Zhang et al., 2021; Tu et al., 2022). Local attention is typically employed in conjunction with subsampled global attention to encode substantial spatial information while avoiding excessive computational costs (Tu et al., 2022; Chu et al., 2021; 2023; Zhang et al., 2021). However, the use of subsampled attentions often results in a lack of shift-equivariance due to downsampling. Consequently, addressing the shift-equivariance issue in these subsampled attentions is crucial.

The most prevalent local attention mechanism is window attention, while a popular subsampled global attention variant is the global subsampled self-attention (GSA) introduced by Chu et al. (2021). The polyphase anchoring algorithm from the main manuscript directly tackles the lack of shift-equivariance. This approach is designed to maintain spatial information while reducing computational complexity, thus promoting shift-equivariance in subsampled attention mechanisms.

A.3. Theoretical guarantees

A.3.1. Positional Encoding

In section 2.3 of the main manuscript, we mentioned that relative positional encoding (Liu et al., 2021; 2022) is not shift-equivariant. We reiterate the definition below and provide a counterexample. Relative positional encoding is defined as:

$$A_r = \text{SoftMax}(XW_Q(XW_K)^T + B)XW_V \tag{10}$$

Counterexample: Let B be a $n \times n$ square matrix with two standard basis vectors e_1 and e_2 and everywhere else zero.

$$B = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{11}$$

Let P_π be the matrix representation for the linear transformation T_π that circularly shifts the input signals s.t

$$P = \begin{pmatrix} (e_n)^T \\ (e_1)^T \\ (e_2)^T \\ \vdots \\ (e_{n-1})^T \end{pmatrix}. \tag{12}$$

For relative positional encoding to be shift-equivariant, we must have $A_r(T_\pi(X)) = T_\pi(A_r(X))$.

$$\begin{aligned} LHS = A_r(T_\pi(X)) &= \text{SoftMax}(T_\pi(X)W_Q(T_\pi(X)W_K)^T + B)T_\pi(X)W_V && (13) \\ &= \text{SoftMax}(P_\pi XW_Q(P_\pi XW_K)^T + B)P_\pi XW_V && (14) \end{aligned}$$

$$\begin{aligned} RHS = T_\pi(A_r(X)) &= P_\pi \text{SoftMax}(XW_Q(XW_K)^T) + B)P_\pi^T P_\pi XW_V && (15) \\ &= \text{SoftMax}(P_\pi XW_Q(XW_K)^T)P_\pi^T + P_\pi BP_\pi^T)P_\pi XW_V && (16) \end{aligned}$$

Assume $P_\pi XW_V$ is right-invertible: $\exists Q$ s.t $(P_\pi XW_V)Q = I$. Multiply both LHS and RHS by Q and apply logarithmic function.

$$LHS = P_\pi XW_Q(XW_K)^T)P_\pi^T + B + \log(S_1) \tag{17}$$

$$RHS = P_\pi XW_Q(XW_K)^T)P_\pi^T + P_\pi BP_\pi^T + \log(S_2) \tag{18}$$

For $LHS = RHS$, the following much hold:

$$B = P_\pi BP_\pi^T + C, \tag{19}$$

where C is a constant matrix. However,

$$P_\pi BP_\pi^T = \begin{pmatrix} 0 & e_2 & e_3 & \dots \end{pmatrix} \tag{20}$$

QED.

Tangent from the solutions proposed in the main paper, we reveal that relative positional encoding is shift equivariant under specific conditions. More concretely, if the bias term is shift equivariant, relative positional encoding is shift equivariant (Liu et al., 2021; 2022). Let T_π denote the spatial translation of the input X, and A_r denote a self-attention operator with relative position bias. We have:

$$\begin{aligned} A_r(T_\pi(X)) &= \text{SoftMax}(T_\pi(X)W_Q(T_\pi(X)W_K)^T + B)T_\pi(X)W_V && (21) \\ &= \text{SoftMax}(P_\pi XW_Q(P_\pi XW_K)^T + B)P_\pi XW_V && (22) \\ &= \text{SoftMax}(P_\pi XW_Q(XW_K)^T P_\pi^T + P_\pi P_\pi^T B)P_\pi XW_V && (23) \\ &= \text{SoftMax}(P_\pi XW_Q(XW_K)^T P_\pi^T + P_\pi BP_\pi^T)P_\pi XW_V && (24) \\ &= P_\pi \text{SoftMax}(XW_Q(XW_K)^T) + B)P_\pi^T P_\pi XW_V && (25) \\ &= T_\pi(A_r(X)) && (26) \end{aligned}$$

Although it is not directly related to the solutions proposed in the main manuscript, this finding demonstrates that shift-equivariance can be ensured in relative positional encoding through constraining the bias term to be shift-equivariant.

A.3.2. Polyphase anchoring

In section 2.2 of the main manuscript, we claimed that the composition of polyphase anchoring with strided convolution, window attention, and global subsampled self-attention respectively results in shift-equivariant operations. We provide proofs for those claims in this section.

Lemma A.1. *Polyphase anchoring operator P is general equivariant with respect to $\forall g \in G$, where G is the symmetry group of translations, and $P : V \to V$ is a nonlinear operator that conditionally shift the input X. $\forall g \in G$, $\exists g' \in G$ s.t:*

$$P(g \cdot X) = g' \cdot P(X), \tag{27}$$

where $\cdot$ denotes the linear mapping of the input by the representation of group elements in G.

Proof: let $X \in \mathbb{R}^{\dots \times H \times W}$,

$$P(g \cdot X) = g_{|(g \cdot X)} \cdot g \cdot X \tag{28}$$

where $g_{|(g \cdot X)}$ is some translation conditioned on input $g \cdot X$.

$$P(X) = g_{|X} \cdot X \tag{29}$$

where $g_{|X}$ is some translation conditioned on input X. Since $g_{|X}, g_{|(g\cdot X)}, g \in G$, $\exists g' \in G$ s.t

$$g_{|(g\cdot X)} \cdot g \cdot X = g' \cdot g_{|X} \cdot X \tag{30}$$
$$P(g \cdot X) = g' \cdot P(X) \tag{31}$$

QED.

Corollary A.2. *$P(g \cdot X) = g' \cdot P(X)$ where g' translates $P(X)$ by an integer multiple of stride size s. Stride size is the distance between two consecutive tokens in the same polyphase on a 2D grid.*

Proof: let $X \in \mathbb{R}^{\cdots \times H \times W}$, $X[:, i, j] \in \mathbb{C}$ denote a token located at (i, j) coordinate on a 2D grid.

By definition of polyphase anchoring, tokens in the maximum polyphase are at the anchor positions s.t

$$P(X)[:, 0 :: s, 0 :: s] = \arg \max_{P(X)[:,i::s,j::s] \in \{P(X)[:,i::s,j::s] | i,j \in \mathbb{Z}, i,j<s\}} \|P(X)[:, i :: s, j :: s]\|, \tag{32}$$

where $P(X)[:, 0 :: s, 0 :: s]$ denotes the polyphase or subsampled grid starting from top left at $(0, 0)$ with stride size s. (This notation aligns with regular PyTorch usage.) Assuming that maximum polyphase is unique, $P(X)[:, 0, 0]$ $P(g \cdot X)[:, 0, 0]$ both belong to the same polyphase. Since coordinate distance between tokens in the same polyphase is a integer multiple of stride size, we must have $P(g \cdot X) = g' \cdot P(X)$, where g' translate $P(X)$ by a multiple of stride size s on a 2D grid. QED.

Lemma A.3. *Given a window attention operator A_w and polyphase anchoring operator P, the composition of these operators is general shift-equivariant $\forall g \in G$, where G is the translation group, and for $s, w \in \mathbb{Z}$ such that $s = w$, where s is the stride size in the polyphase and $w \times w$ is the window size. This can be expressed as:*

$$A_w(P(g \cdot X)) = g' \cdot A_w(P(X)) \tag{33}$$

Proof: let $X \in \mathbb{R}^{C \times H \times W}$, $X = \begin{bmatrix} X_{00} & \cdots & X_{0n} \\ \vdots & \vdots & \vdots \\ X_{m0} & \cdots & X_{mn} \end{bmatrix}$ where $m = \frac{H}{w}$, $n = \frac{W}{w}$, and $X_{ij} \in R^{C \times w \times w}$, $i \in \{0, \cdots, m\}, j \in \{0, \cdots, n\}$. We call $w \times w$ window size and X_{ij} tokens in the window (i, j). The window attention operator

$$A_w(X) = \begin{bmatrix} A_s(X_{00}) & \cdots & A_s(X_{0n}) \\ \vdots & \vdots & \vdots \\ A_s(X_{m0}) & \cdots & A_s(X_{mn}) \end{bmatrix}$$

, where A_s is the self attention operator

$$A_s(X) = \text{SoftMax}(XW_q(XW_k)^T)XW_v = \text{SoftMax}(Q(K)^T)V$$

.

$$A_w(P(X)) = A_w\left(\begin{bmatrix} \hat{X}_{00} & \cdots & \hat{X}_{0n} \\ \vdots & \vdots & \vdots \\ \hat{X}_{m0} & \cdots & \hat{X}_{mn} \end{bmatrix}\right)$$

, where $\{\hat{X}_{00}[:, 0, 0], \cdots, \hat{X}_{mn}[:, 0, 0]\}$ are tokens in the maximum polyphase because the polyphase anchoring algorithm conditionally shifts the input data so that the maximum polyphase is $\hat{X}[:, 0 :: w, 0 :: w]$.

$$
\begin{aligned}
A_w(P(g \cdot X)) &= A_w(g' \cdot P(X)) \\
&= A_w \left(g' \cdot \begin{bmatrix} \hat{X}_{00} & \cdots & \hat{X}_{0n} \\ \vdots & \vdots & \vdots \\ \hat{X}_{m0} & \cdots & \hat{X}_{mn} \end{bmatrix} \right) \\
&= \left(\begin{bmatrix} g' \cdot A_s(\hat{X}_{ij}) & \cdots & g' \cdot A_s(\hat{X}_{i(j-1)}) \\ \vdots & \vdots & \vdots \\ g' \cdot A_s(\hat{X}_{(i-1)j}) & \cdots & g' \cdot A_s(\hat{X}_{(i-1)(j-1)}) \end{bmatrix} \right) \quad \text{(Corollary A.2)} \qquad (34) \\
&= g' \cdot \left(\begin{bmatrix} A_s(\hat{X}_{00}) & \cdots & A_s(\hat{X}_{0n}) \\ \vdots & \vdots & \vdots \\ A_s(\hat{X}_{m0}) & \cdots & A_s(\hat{X}_{mn}) \end{bmatrix} \right) \\
&= g' \cdot A_w(P(X))
\end{aligned}
$$

QED.

Lemma A.4. *Let P be the polyphase anchoring operator and $*_s$ represent the strided convolution operator. $\forall g \in G$, where G is the translation group, and $\forall s_1, s_2 \in \mathbb{Z}$ s.t $s_1 = s_2$, where s_1 is the stride size in the polyphase and s_2 is the stride size of convolution, the composition of strided convolution and polyphase anchoring is general shift-equivariant:*

$$
h *_s P(g \cdot X) = g' \cdot (h *_s P(X)) \tag{35}
$$

Here, X denotes the input signal, h is the convolution filter, and $\cdot$ signifies the linear mapping of the input by the representation of group elements in G. Furthermore, $P : V \to V$ is a nonlinear operator acting on the input space V. Mathematically, strided convolution $*_s$ can be represented as a full convolution followed by a downsampling operation

$$
h *_s X = P^{(s)}_{0,0}(h * X)
$$

where $P^{(s)}_{m,n}(\cdot)$ is a function with a matrix as input and its down-sampled sub-matrix as output. This function select the elements on the grid defined by m, n, s, where (m, n) denotes the upperleft position of the grid, and s denotes the sub-sampling stride.

$$
\begin{aligned}
LHS = h *_s P(g \cdot X) &= P^{(s)}_{0,0}(h * P(g \cdot X)) && (36) \\
&= P^{(s)}_{0,0}(h * (g' \cdot P(X))) && (37) \\
&= P^{(s)}_{0,0}(g' \cdot (h * P(X))) && (38) \\
&= g' \cdot (P^{(s)}_{0,0}(h * P(X))) && (39) \\
&= RHS && (40)
\end{aligned}
$$

where $g' \in G$ translates input by an integer multiple of stride size s. QED.

Lemma A.5. *For a global subsampled self-attention operator A_g (Chu et al., 2021) combined with a polyphase anchoring operator P, general shift-equivariance is achieved for $\forall g \in G$, where G is the translation group, and for $s_1, s_2 \in \mathbb{Z}$ such that $s_1 = s_2$. Here, s_1 is the stride size in the polyphase, and s_2 is the stride size in the global self-subsampled attention. This can be expressed as:*

$$
A_g(P(g \cdot X)) = g' \cdot A_g(P(X)) \tag{41}
$$

In global subsampled self-attention, we have:

$$
A_g(X) = \text{SoftMax}(QK_s^T)V_s, \tag{42}
$$

where K_s and V_s are subsampled from the full keys K and values V using strided convolution.

$$\begin{aligned}
LHS &= A_g(P(g \cdot X)) && (43)\\
&= \text{SoftMax}(P(g \cdot X)W_q(h *_s P(g \cdot X)W_k)^T)h *_s P(g \cdot X)W_v && (44)\\
&= \text{SoftMax}(g' \cdot P(X)W_q(h *_s (g' \cdot P(X))W_k)^T)h *_s (g' \cdot P(X))W_v && (45)\\
&= \text{SoftMax}(g' \cdot P(X)W_q(g' \cdot (h *_s P(X))W_k)^T)g' \cdot (h *_s P(X))W_v && (46)\\
&= \text{SoftMax}(P_{g'}P(X)W_q(P_{g'}(h *_s P(X))W_k)^T)P_{g'}(h *_s P(X))W_v && (47)\\
&= \text{SoftMax}(P_{g'}P(X)W_q((h *_s P(X))W_k)^T P_{g'}^T)P_{g'}(h *_s P(X))W_v && (48)\\
&= P_{g'}\text{SoftMax}(P(X)W_q((h *_s P(X))W_k)^T)P_{g'}^T P_{g'}(h *_s P(X))W_v && (49)\\
&= P_{g'}\text{SoftMax}(P(X)W_q((h *_s P(X))W_k)^T)(h *_s P(X))W_v && (50)\\
&= g' \cdot A_g(P(X)) = RHS, && (51)
\end{aligned}$$

where $P_{g'}$ is the matrix representation of a group element $g' \in G$ from the symmetry group of translations. QED.

A.3.3. COMPOSITION OF EQUIVARIANT FUNCTIONS

In section 2.1, we conduct a comprehensive analysis of Vision Transformers (ViT) and their derivatives, focusing on the aspect of shift-equivariance. We identify specific modules within these models that do not preserve shift-equivariance, which is integral to maintaining spatial coherence in vision tasks. In response to this discovery, we propose and implement a series of corrective measures, facilitating the design of fully shift-equivariant Vision Transformer architectures. Importantly, our approach capitalizes on the property that a composite function constructed from shift-equivariant functions retains shift-equivariance. This results in models that preserve spatial information across the entire network architecture.

Lemma A.6. *Composition of two equivariant functions with respect to transformations in symmetry group is equivariant.*

Proof: Let G be a symmetry group and let $f : X \to Y$ and $h : Y \to Z$ be equivariant functions, i.e., for all $x \in X$ and $g \in G$, we have:

$$f(g \cdot x) = g \cdot f(x)$$

and

$$h(g' \cdot y) = g' \cdot h(y)$$

where $\cdot$ denotes the group action of G on X, Y and Z. We want to show that $h \circ f : X \to Z$ is also equivariant, i.e., for all $x \in X$ and $g \in G$, we have:

$$(h \circ f)(g \cdot x) = g \cdot (h \circ f)(x)$$

We start with the left-hand side:

$$\begin{aligned}
(h \circ f)(g \cdot x) &= h(f(g \cdot x)) && \text{(definition of composition)}\\
&= h(g \cdot f(x)) && \text{(by equivariance of } f\text{)}\\
&= g \cdot h(f(x)) && \text{(by equivariance of } h\text{)}\\
&= g \cdot (h \circ f)(x) && \text{(definition of composition)}
\end{aligned}$$

where we used $g' \cdot h(y) = h(g' \cdot y)$, the associative property of group action, and the equivariance of f and h.

Therefore, we have shown that $(h \circ f)(g \cdot x) = g \cdot (h \circ f)(x)$ for all $x \in X$ and $g \in G$, which means that $h \circ f$ is equivariant with respect to the group action of G.

A.4. Experiment

Figure 3 demonstrates that ViT_S/16-poly and Twins_B-poly both pass all shift-equivariance tests and their output logits variance concerning small shift perturbations are almost zero. On the other hand, ViT_S/16 and Twins_B fail all shift-equivariance tests and demonstrate unstable output probabilities with respect to small input shift.