Rotary Positional Encodings for ViT and Performer

IEOR6617.2024Fall.Project Report

Haoyu Dong, hd2573, Jinfan Xiang, jx2598, Wangshu Zhu wz2708,
Xudong Chen xc2763, Zekai Wen zw3057,

Abstract

This study explores the integration of two-dimensional Rotary Positional Embeddings (2D RoPE) into Vision Transformers (ViTs) and Performers to enhance image recognition performance on the CIFAR-100 dataset. While Absolute Positional Embeddings (APE) achieve superior accuracy and convergence for low-resolution images, RoPE-based methods show potential advantages for higher-resolution tasks. Performers demonstrate greater computational efficiency with larger token dimensions compared to ViTs. These findings highlight the importance of dataset characteristics and model architecture in selecting positional embedding strategies. Future research will focus on adaptive mechanisms to optimize performance, efficiency, and scalability in transformer-based models.

1 Introduction

The self-attention mechanism has become the cornerstone of state-of-the-art performance in large-scale deep learning tasks over recent years. It serves as a foundational element for numerous high-performance models, most notably the Transformer architecture[1]. Transformers have revolutionized fields such as natural language processing (NLP) and computer vision[2], consistently setting new benchmarks. One key innovation contributing to the Transformer's success, alongside self-attention, is positional embedding. Positional embeddings enable the model to capture spatial or sequential relationships among tokens, which is critical for tasks that depend on context, such as understanding "the cat is on the chair" versus "the chair is on the cat."

Positional embeddings, however, operate independently of the self-attention mechanism itself. Selecting an appropriate positional embedding method is essential for enhancing the performance of the attention mechanism. The original Transformer paper introduced absolute positional embeddings (APE), which encode positional relationships explicitly as fixed embeddings added to token embeddings. While APE allows the model to learn positional information, its effectiveness is limited, especially for more complex relationships in larger datasets.

To address these limitations, subsequent research has introduced relative positional embeddings (RPE). RPEs encode relative distances between tokens rather than their absolute positions, making them more adaptable to tasks requiring relational understanding[3]. Recent advancements, such as Rotary positional Embeddings (RoPE)[6], have further refined this approach by encoding relative positional relationships through rotational transformations[4] in the embedding space. RoPE has shown exceptional performance in NLP tasks, particularly for one-dimensional datasets, where it effectively preserves positional information. However, extending RoPE to two-dimensional tasks, such as image data in vision transformers (ViTs), remains challenging due to its original design constraints.

The original paper we aim to reproduce introduces a two-dimensional extension of RoPE (2D RoPE) for vision transformers. This adaptation modifies the query (Q) and key (K) representations in the self-attention mechanism to encode 2D positional relationships effectively. Their experiments demonstrated two significant findings:

1. The 2D RoPE mechanism improves performance on specific vision tasks compared to classical positional embeddings.

2. Combining 2D RoPE with traditional positional embedding methods yields further performance gains for ViT models.

In our work, we aim to reproduce the findings of the original paper but under different conditions: using smaller-scale image datasets and incorporating the 2D RoPE mechanism into state-of-the-art transformer variants such as Performer. Performer[5] introduces efficient attention mechanisms that reduce computational complexity while maintaining robust performance, making it a compelling choice for smaller datasets. Through this project, we not only aim to validate the performance of 2D RoPE but also explore its integration into other transformer architectures to compare their effectiveness and gain deeper insights into attention-based learning models.

This paper is organized as follows: in Part 1 we have introduced the background and motivation of our study; Part 2 reviews the original paper's methodology and contributions; Part 3 outlines our experimental setup, including architectural design and parameter settings; an "Implementation" section takes place in Part 4, detailing the dataset used and the Performer model employed; Part 5 presents an analysis of the results, comparing our findings with those of the original study; and finally, Part 6 concludes by summarizing key insights, discussing limitations, and suggesting avenues for future research.

2 Review on original paper

This part will review the original work. We will illustrate the idea of RoPE for ViT and analyze its contributions.

2.1 Methodology of the Original Paper

It is proposed in the original paper that Rotary positional Embedding (RoPE) is a mechanism to encode relative positional information in self-attention layers efficiently. While RoPE was initially designed for 1D sequence data (e.g., natural language), the paper extends its applicability to 2D vision tasks such as classification, detection, and segmentation. This work addresses challenges related to extrapolation and positional encoding in Vision Transformers (ViTs).

RoPE uses a mathematically elegant method that replaces additive positional embeddings. This approach embeds relative positional differences into attention computations with Euler's formula. This mechanism inherently supports extrapolation across sequence lengths without additional bias terms.

For 2D vision tasks, the paper provides two methods. Axial frequencies apply independent rotations along the x and y axis, effectively encoding axis-aligned positional relationships. However, this method struggles with mixed dependencies. Mixed learnable frequencies overcome this by combining rotations of both axes with learnable parameters, allowing flexible encoding of diagonal and complex spatial relationships.

RoPE is integrated into Vision Transformers (ViTs) in two ways. In standard ViT architectures, RoPE is applied globally to the sequence of patches, treating each patch as a token. In hierarchical models like Swin Transformers, RoPE is applied locally within attention windows. Across both architectures, RoPE enhances attention by embedding relative positional information without requiring additional computational overhead. The paper evaluates RoPE on multiple tasks, including multi-resolution classification on ImageNet-1k, object detection on MS-COCO, and semantic segmentation on ADE20k, focusing on its ability to handle extrapolation scenarios effectively.

2.2 Key Results of the Original Paper

The original paper shows that 2D RoPE significantly enhances performance across a range of vision tasks, including image classification, object detection, and semantic segmentation. One of its standout features is its ability to handle input resolutions that exceed those seen during training.

For multi-resolution classification on ImageNet-1k, RoPE excels in handling extrapolated resolutions, such as those beyond the training size of 224×224. Both RoPE-Axial and RoPE-Mixed outperform traditional methods like APE and RPB. Among these, RoPE-Mixed performs best, thanks to its ability to capture complex 2D positional relationships through learnable mixed frequencies. While combining RoPE with APE slightly improves interpolation accuracy, it compromises RoPE's extrapolation capabilities, makingFor object detection on MS-COCO, RoPE improves Average Precision (AP) scores when used with ViT and Swin Transformer backbones. For example, pairing RoPE-Mixed with DINO-trained ViT-B and ViT-L achieves up to +1.8 AP gains over APE. Similarly, Swin Transformers with RoPE outperform those using RPB, demonstrating RoPE's effectiveness in hierarchical architectures. RoPE's ability to improve global attention plays a critical role in its strong performance on tasks involving objects of varying sizes and scales.

In semantic segmentation on ADE20k, RoPE delivers significant improvements in mean Intersection-over-Union (mIoU). ViT models with RoPE-Mixed achieve up to +2.5 mIoU increases compared to APE in multi-scale evaluations. Swin Transformers with RoPE-Mixed also outperform those with RPB, confirming its strength in dense prediction tasks. Notably, the best segmentation results are achieved by combining RoPE-Mixed with APE, showcasing its flexibility for both interpolation and extrapolation scenarios.

Additionally, the paper highlights how RoPE enhances attention dynamics, enabling models to attend to more diverse and longer-range token interactions. This is particularly beneficial for handling high-resolution inputs. Remarkably, these benefits come with minimal computational overhead, as RoPE introduces only a slight increase in FLOPs for ViT-B. All methods presented in the essay show that RoPE proves to be a robust and efficient positional embedding method that consistently outperforms traditional approaches like APE and RPB.

3 Methodology

3.1 Basic setting

Platform	AutoDL
CPU	14 vCPU Intel Xeon Gold 6348 @ 2.60GHz
GPU	NVIDIA A800 Tensor Core GPU 80GB
Python Version	3.10
Pytorch Version	2.1.2
CUDA	11.8

Table 1. Experimental environment configuration

This experiment aims to compare the accuracy and efficiency of both Transformer and Performer with and without the introduction of RoPE (Rotary Positional Embedding). The experimental environment is based on the AutoDL cloud computing platform, configured with an A800 80GB GPU, 14 vCPU Intel Xeon Gold 6348 @ 2.60GHz, using Python 3.10 (Ubuntu 22.04), PyTorch 2.1.2 and CUDA 11.8.

The experiment will train 20 models: Transformer with APE (Absolute positional Encoding), Transformer with axial RoPE, Transformer with mixed RoPE, Tranformer with APE and axial RoPE, Transformer with APE and mixed RoPE, Performer with APE, Performer with axial RoPE, Performer with mixed RoPE, Performer with APE and axial RoPE, Performer with APE and mixed RoPE, and each of these models have small and base version, which differ on the number of attention head. In addition, the experiment runs them under the same training configuration and compare their training time and accuracy.

3.2 Performer Introduction

The Performer, introduced by Choromanski et al. (2021), is an efficient alternative to traditional Transformer, specifically designed to handle the challenges of long sequences. By rethinking how self-attention is computed, the Performer reduces the quadratic complexity of standard Transformers to linear, making it faster and more scalable for tasks like natural language processing, computer vision, and even biological sequence modeling [5].

At the heart of the Performer is the FAVOR+ (Fast Attention Via Orthogonal Random features) mechanism. Instead of directly computing the attention weights with softmax, FAVOR+ uses kernel-based random feature maps to approximate these calculations. In standard Transformers, the self-attention mechanism computes interactions between every pair of tokens, leading to a computational complexity of $O(N^2)$, where N is the sequence length. This quadratic growth becomes a bottleneck for long sequences, especially in resource-constrained environments [5]. Performers resolve this by reformulating the self-attention mechanism as:

$$\widehat{\boldsymbol{Att}_{\leftrightarrow}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \widehat{\boldsymbol{D}}^{-\mathbf{1}}(\boldsymbol{Q}'((\boldsymbol{K}')^{\boldsymbol{T}}\boldsymbol{V})) \tag{1}$$

where Q, K, and V represent the query, key, and value matrices, respectively. In the Performer, this is reformulated using kernel-based random feature maps as:

$$(\boldsymbol{Q}', \boldsymbol{K}') = (\boldsymbol{\Phi}(\mathbf{Q}), \boldsymbol{\Phi}(\mathbf{K})) \tag{2}$$

where $\Phi(X)$ is a feature transformation that allows the attention computation to scale linearly with the input size [5]. This approximation allows the Performer to reduce the computational complexity of attention from $O(N^2)$ to $O(N)$. With the help of this innovation, the Performer is able to process much longer sequences efficiently, without significantly sacrificing accuracy or model expressiveness.

Performers have been shown to perform well across a variety of applications, including text classification, language modeling, and protein sequence modeling. They inherit the strengths of Transformers [1] while addressing their computational and memory limitations. Moreover, Performers are particularly suited for real-world scenarios that require processing long-range dependencies efficiently, making them a practical and impactful advancement in deep learning [1].

4 Implementation

4.1 Dataset

The CIFAR-100 dataset is an image classification dataset released by Alex Krizhevsky and Geoffrey Hinton in 2009 and is widely used in the fields of computer vision and deep learning. The dataset contains 60,000 32x32 pixel RGB color images, of which 50,000 are used for training and 10,000 for testing. Compared to the CIFAR-10 dataset, CIFAR-100 has richer categories, containing 100 classes, each with 600 images, which are grouped into 20 superclasses [7].

Each image contains 3 color channels (red, green, and blue) with a resolution of 32x32 pixels and is stored as a 3072-dimensional vector (32×32×3). The dataset files are stored in Python pickle format and mainly consist of train (training set), test (test set), and meta (calss name and superclass name) files. The number of images under each category is roughly balanced, which facilitates the training and evaluation of the model on fine-grained classification tasks.

The 100 classes of CIFAR-100 cover a wide range of realistic scenarios such as fish (e.g., dolphins, whales, seals), flowers (e.g., lilies, dandelions, sunflowers), reptiles (e.g., lizards, snakes, turtles), etc., which are categorized into 20 larger superclasses [7].

Since the dataset size of CIFAR-100 is only about 161 MB, the image size is small, which is easy to load and process quickly, and is very suitable for experiments in resource-constrained environments, we choose this dataset in this experiment.

4.2 Hyperparameter setting

Since most of the hyperparameters in the source code have default values and obtained results good enough, we didn't change them too much. Below we list some of the hyperparameters we changed for our exvironmental setting:

Hyperparameter	Value
--batch-size	512
--epochs	400
--input-size	32
--lr	1e-4
--unscale-lr	True
--repeated-aug	True

Table 2. Hyperparameters setting

Due to the fact that the GPU used is A800 with 80G memory, the batch size is set to 512 in order to save the training time. epochs are 300 rounds by default, but after experiments, 300 rounds did not converge very obviously, so it is increased to 400 rounds. Since the image size of CIFAR-100 is 32×32, we set the input size to 32. the default size of learning rate is 5e-4, but in the actual training process there will be a gradient explosion, so we choose to shrink to 1e-4 to ensure the accuracy of the realized data. When unscale-lr is True, Linear Learning Rate Scaling is not performed because linear learning rate scaling will significantly increase the learning rate during high-volume training, which may lead to excessive gradient updates, triggering unstable training or even gradient explosion. Keeping the learning rate fixed helps the training process be smoother and avoids drastic fluctuations. repeated-aug allows each sample to be repeatedly sampled for multiple times in each epoch and different data enhancement operations (e.g., random cropping, rotation, flipping, etc.) are applied to each sample. This approach can effectively increase the diversity of samples seen by the model, which helps to improve the generalization ability.

5 Result

5.1 DeiT result

In our experiments with DeiT architecture, we evaluated multiple positional embedding configurations, including Absolute positional Embedding (APE), RoPE-Axial, RoPE-Mixed, and their combined variants. Validation accuracy and loss trends shown below provide key insights:

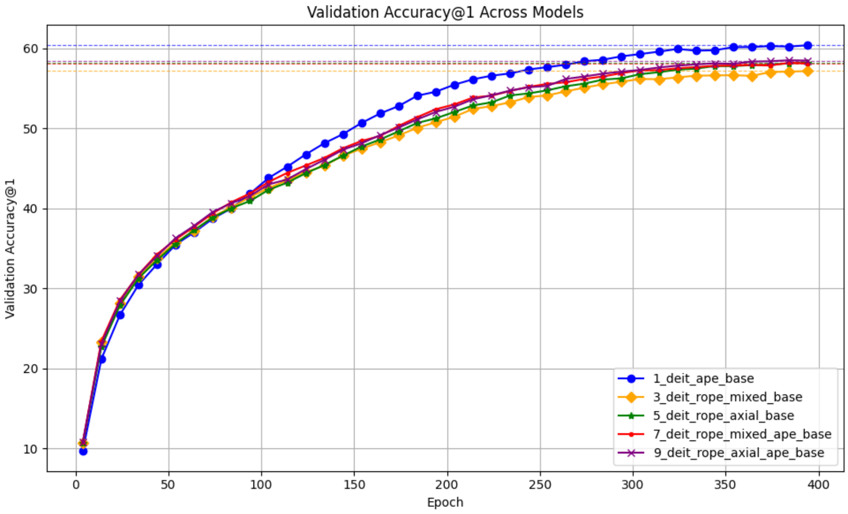


Figure 1. The Top-1 Accuracy of DeiT with different positional embedding methods

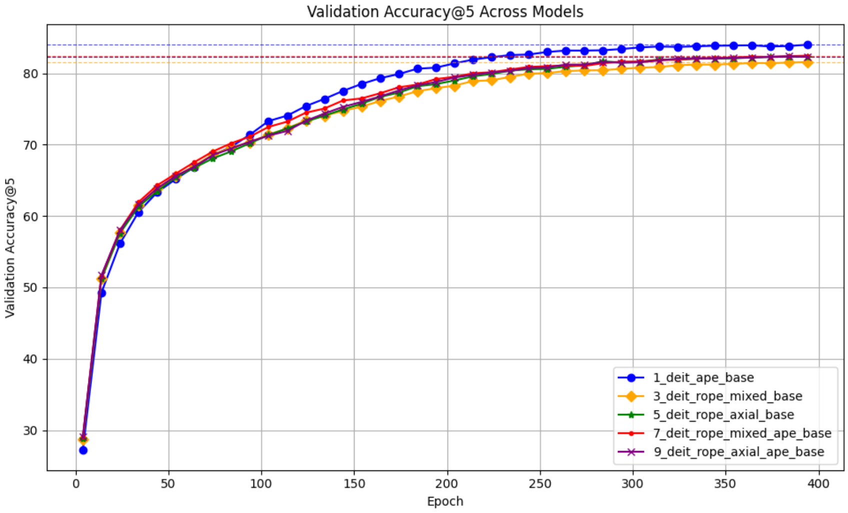


Figure 2. The Top-5 Accuracy of DeiT with different positional embedding methods

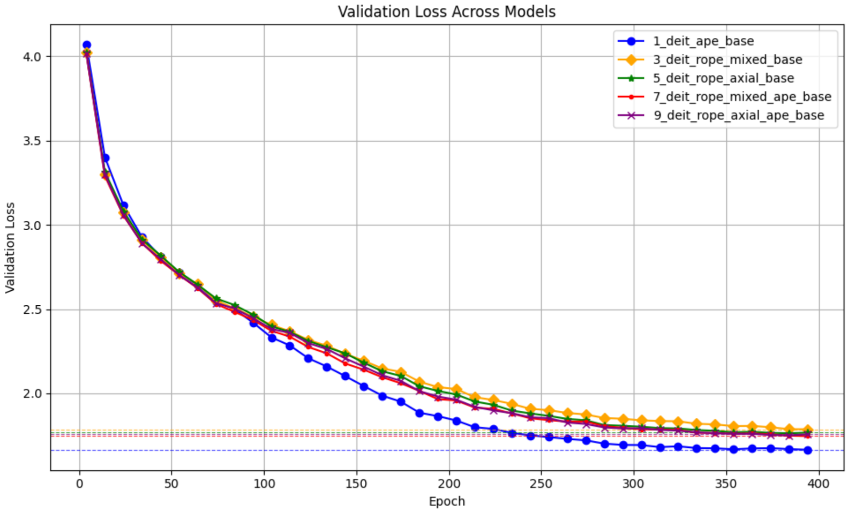


Figure 3. The loss of DeiT with different positional embedding methods

The APE-based model consistently outperforms RoPE-based configurations across all epochs, achieving higher validation accuracy at both top 1 and top 5 thresholds. RoPE-Mixed and RoPE-Axial embeddings, while competitive, showed slightly lower final accuracy, which is in contrast to the findings of the original essay.

Models using APE demonstrated a faster reduction in validation loss, converging earlier and stabilizing at a lower loss compared to RoPE variants. The RoPE-Mixed and RoPE-Axial configurations showed slower convergence, which could be attributed to the smaller input resolution (32×32 for CIFAR-100) and the inherent challenges in encoding positional information in such datasets.

One critical factor contributing to these observations is the use of CIFAR-100, a dataset with smaller image resolutions (32×32) compared to those used in the original study (e.g., ImageNet). We assume that the reduced spatial information in CIFAR-100 limits the advantages offered by RoPE, which thrives on extrapolating and scaling to larger resolutions. This difference in dataset characteristics may explain why APE outperformed other embeddings, contrary to the original findings where RoPE showed superior results, and this property of the RoPE is also illustrated in the original study.

5.2 Performer results

5.2.1 positional embedding comparison

We implement the Performer based on the framework in GitHub[1]. In the context of Performer architecture, we evaluated its performance using a range of positional embeddings used in RoPE above: Absolute positional Embedding (APE), RoPE-Axial, RoPE-Mixed, and their combined variants. The results are as follows:

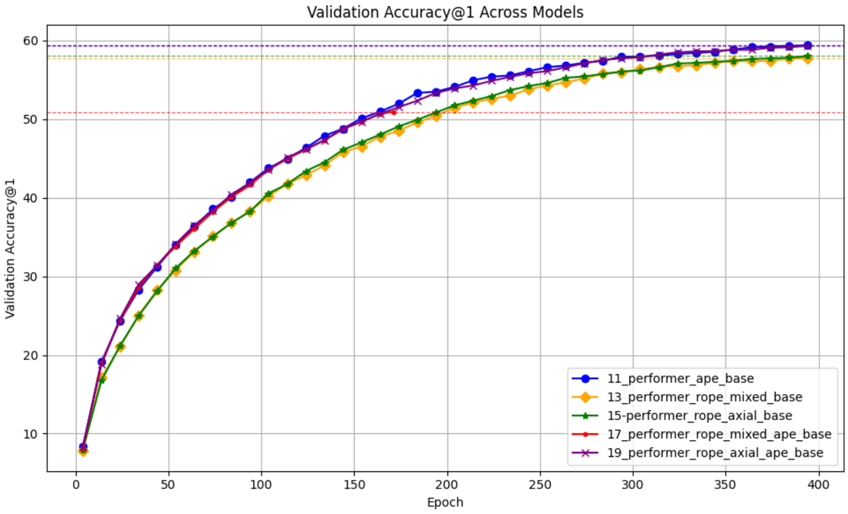


Figure 4. The Top-1 Accuracy of Performer with different positional embedding methods

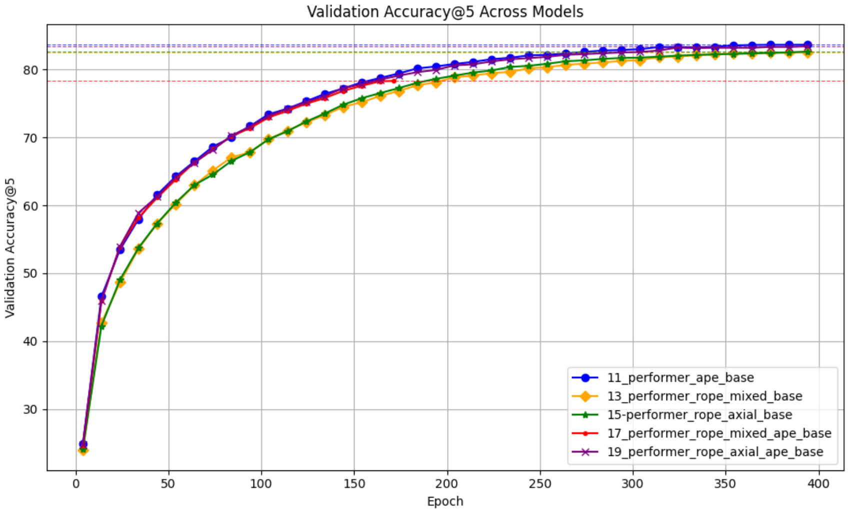


Figure 5. The Top-5 Accuracy of Performer with different positional embedding methods

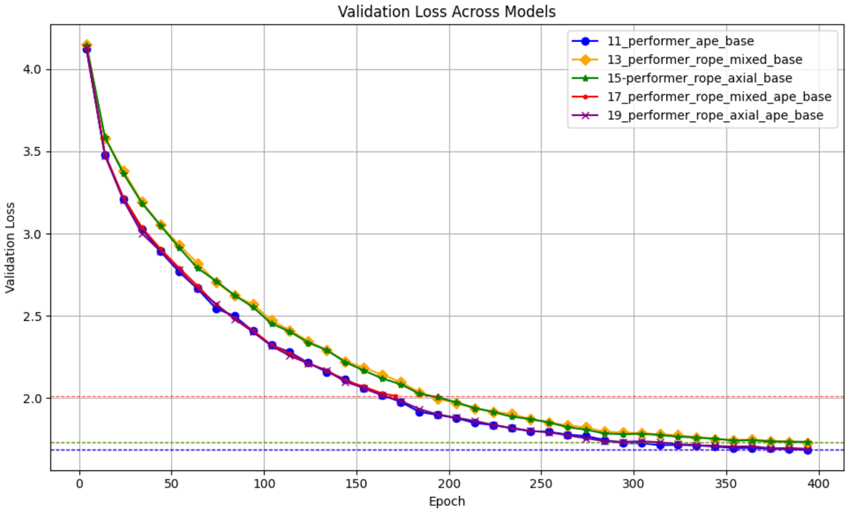


Figure 6. The loss of Performer with different positional embedding methods

Performer models with APE demonstrated faster convergence in validation loss and reached significantly lower final values compared to configurations using only RoPE-based embeddings. This performance difference stems from the fundamental differences in how positional information is encoded. APE directly encodes absolute positional relationships, which are straightforward and effective for the limited spatial resolution (32×32) of CIFAR-100 images. In contrast, RoPE relies on relative positional encoding, which is designed for capturing complex spatial relationships in high-resolution data but struggles to fully utilize the simpler positional information present in low-resolution datasets like CIFAR-100.

The Performer architecture combined with APE consistently achieved the highest validation accuracy across all epochs in both top-1 and top-5 metrics. This highlights that APE aligns well with Performer's efficient attention mechanism by facilitating effective integration of absolute positional information into attention computation. Combined embeddings, such as RoPE-Mixed-APE and RoPE-Axial-APE, offered moderate improvements over RoPE-only configurations but still lagged behind APE-only setups. This suggests that the simplicity and directness of APE encoding make it better suited for low-resolution datasets.

RoPE-Axial and RoPE-Mixed configurations, while slightly less effective than APE, showed steady improvements during training. However, their reliance on relative positional encoding reduces their utility in CIFAR-100, where the smaller resolution limits the complexity of positional relationships that can be captured. This limitation reinforces findings from DeiT experiments, where RoPE's benefits were more pronounced in high-resolution tasks. Together, these results underscore how dataset characteristics—such as resolution—and embedding design principles shape performance, particularly in architectures like Performer that rely on efficient attention mechanisms.

5.2.2 Number of Attention Head comparison

To better understand the relationship between model size and the performance of Performer, we conducted experiments with two variants of Performer architecture across different positional embeddings setups: base (12 attention heads) and small (6 attention heads). Both models have the same embedding dimensions. The validation accuracy and loss trends provide critical insights into how model size affects the Performer.

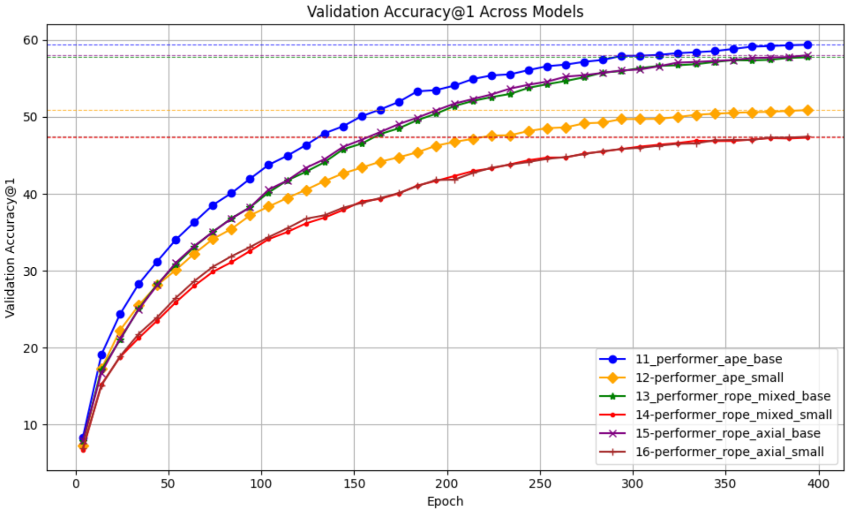


Figure 7. The Top-1 accuracy of Performer with different positional embedding methods and model sizes

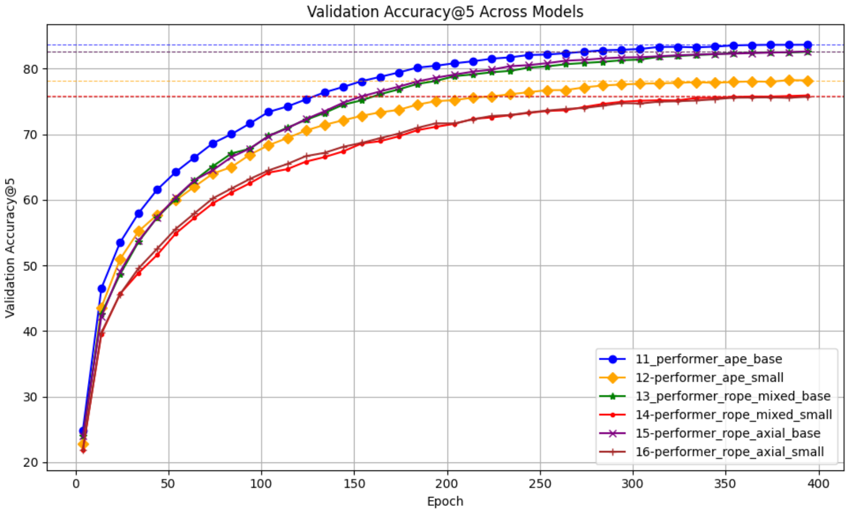


Figure 8. The Top-5 accuracy of Performer with different positional embedding methods and model sizes

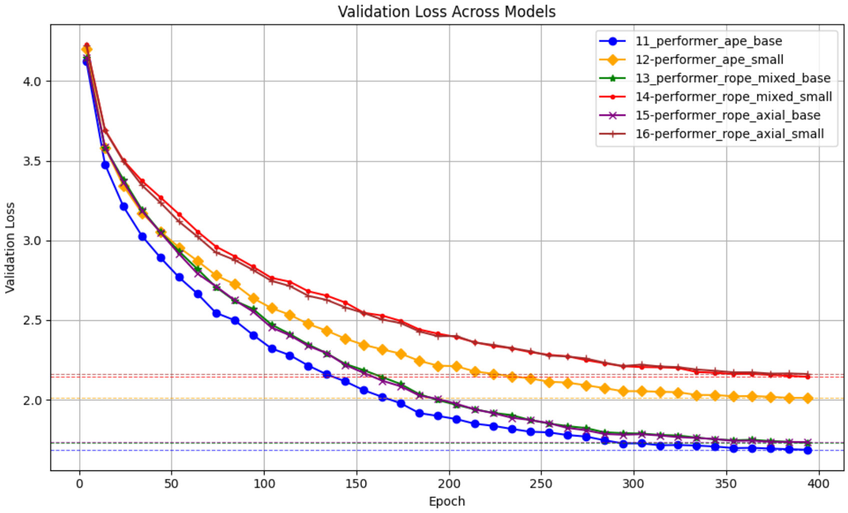


Figure 9. The loss of Performer with different positional embedding methods and model sizes

Base models consistently exhibited faster convergence and lower validation loss compared to small models, demonstrating the significant advantage provided by having more attention heads. The superior performance of base models compared to small models is directly linked to their increased number of attention heads. More attention heads enable richer feature representation by allowing the model to process diverse patterns simultaneously, while also improving attention precision by dividing the embedding dimension into finer subspaces. This precision helps disentangle complex positional relationships, particularly for embeddings like RoPE, which rely on relative positional encoding.

Additionally, the increased capacity of base models stabilizes training by balancing the learning load across heads, resulting in smoother gradient flows and faster convergence. In contrast, small models, with fewer heads, lack the capacity to fully leverage positional information, particularly for complex embeddings, leading to consistently poorer performance.

5.3 Discussion

5.3.1 Accuracy comparison

This section provides an in-depth analysis of the performance differences between DeiT and Performer architectures with various positional embeddings, highlighting how their structural characteristics and attention mechanisms interact with embedding types in the context of the low-resolution CIFAR-100 dataset.

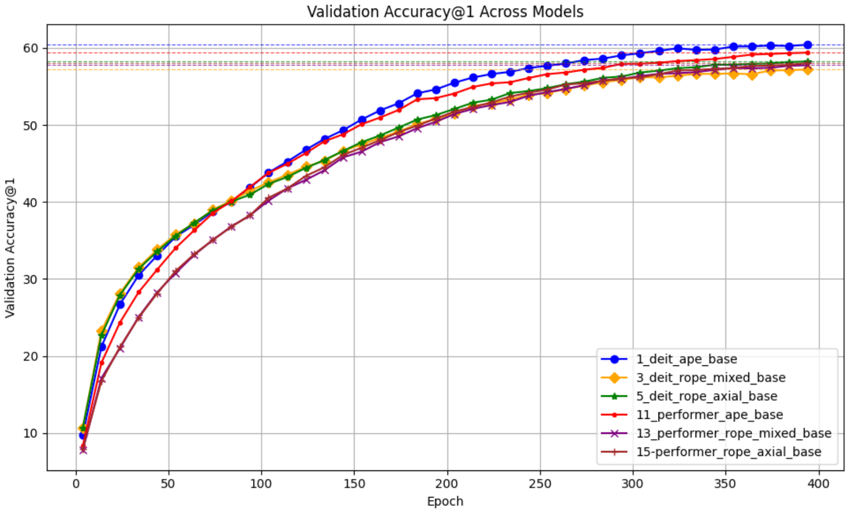


Figure 10. The Top-1 comparison between Deit and Performer with different positional embedding

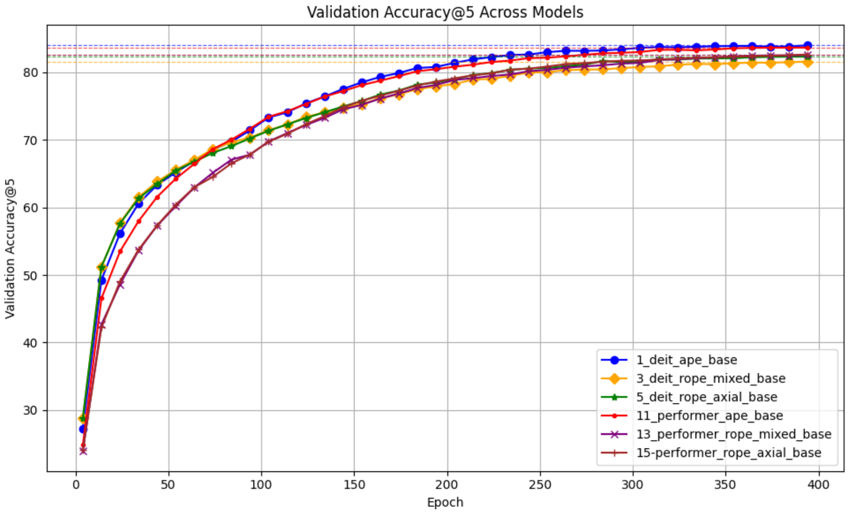


Figure 11. The Top-5 comparison between Deit and Performer with different positional embedding

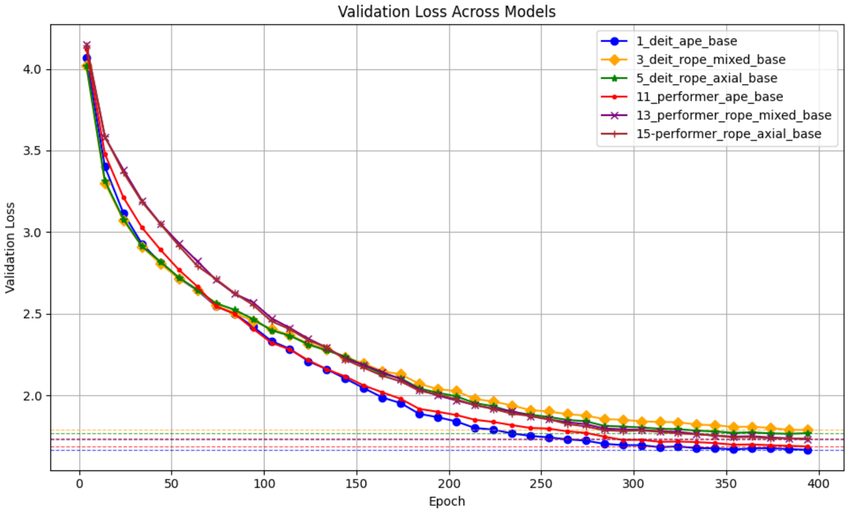


Figure 12. The loss comparison between Deit and Performer with different positional embedding

DeiT with Absolute positional Embedding (APE) consistently outperformed all other configurations in both accuracy and loss metrics. Performer with APE followed closely, achieving comparable but slightly lower results. Performer's approximate attention mechanism, which is linear in time and memory, inherently introduces minor precision errors when compared to DeiT's quadratic exact attention. This fundamental difference directly contributes to the performance gap between the two architectures. However, the minimal gap between DeiT and Performer with APE suggests that in low-resolution datasets like CIFAR-100, the internal feature structures of images are relatively simple, and attention computation precision is not critically impactful for achieving high performance.

Among models utilizing RoPE-Mixed and RoPE-Axial embeddings, Performer outperformed DeiT in both cases, though the difference was marginal. Performers with RoPE-Mixed and RoPE-Axial showed nearly identical results in accuracy and loss. DeiT with RoPE-Axial consistently ranked last in both accuracy and loss, highlighting its limitations in this context. RoPE-Axial and RoPE-Mixed excel in high-resolution scenarios by leveraging their ability to capture complex relative positional relationships. However, these capabilities are not fully utilized in the low-resolution CIFAR-100 dataset, where such relationships are less significant. The consistently poor performance of DeiT with RoPE-Axial suggests that its strong exploratory capabilities in high-resolution datasets are poorly suited to low-resolution tasks. Performer, by contrast, benefits from its efficient attention mechanism, which balances computational demands with sufficient precision for lower-resolution data.

DeiT with RoPE-Mixed and RoPE-Axial embeddings showed the fastest initial improvements in both loss and accuracy, outperforming other configurations during the early training stages. However, these gains quickly plateaued, with both models converging to suboptimal final results. RoPE embeddings rely on precise relative positional relationships to achieve their extrapolation capabilities. In the early stages of training, these relationships allow the model to efficiently capture 2D spatial dependencies, leading to rapid improvements in performance. However, as training progresses, the smaller resolution of CIFAR-100 limits the amount of positional information that can be exploited, causing the performance of these models to plateau prematurely. This aligns with the findings of the original paper, which noted that RoPE's effectiveness is contingent on sufficient spatial complexity to leverage its full potential.

5.3.2 Time comparison

Avg. Training Time / Epoch	APE	RoPE-Mixed	RoPE-Axial	RoPE-Mixed +APE	RoPE-Mixed +APE
ViT	6.17	9.651	9.195	7.497	9.318
Performer	7.264	10.384	9.937	10.092	9.731

Table 3. Training time comparison among all positional embedding

Avg. Inference Time / Epoch	APE	RoPE-Mixed	RoPE-Axial	RoPE-Mixed +APE	RoPE-Mixed +APE
ViT	6.17	9.651	9.195	7.497	9.318
Performer	7.264	10.384	9.937	10.092	9.731

Table 4. The loss comparison between Deit and Performer with different positional embedding

From the results above, we conclude that the choice of positional encoding significantly affects both training and inference efficiency for ViT and Performer models. APE consistently demonstrates the shortest training and inference times (6.17s for ViT and 7.264s for Performer), highlighting its computational simplicity and suitability as a baseline. However, RoPE-based methods, particularly RoPE-Mixed, introduce substantial overhead (e.g., 9.651s for ViT and 10.384s for Performer), reflecting the added complexity of rotary transformations in the attention mechanism. RoPE-Axial, while still more computationally demanding than APE, achieves slightly reduced times compared to RoPE-Mixed, offering a better trade-off between efficiency and representational power. Interestingly, combining RoPE-Mixed with APE yields variable results, with training and inference times influenced by implementation specifics. This combination appears to offset some of the computational costs of RoPE while retaining its representational advantages. Notably, Performer consistently incurs higher training and inference times across all encoding methods compared to ViT, suggesting that Performer's focus on memory efficiency does not fully mitigate the computational demands of complex positional encodings, which we’ ll take a deep discussion in 5.3.3. Overall, while APE remains the most efficient, RoPE-based methods and their combinations provide valuable trade-offs between computational cost and representation, warranting further optimization to better align with efficient architectures like Performer.

5.3.3 Patch size impact

In this chapter, training and inference speeds are measured and analyzed regarding the regular transformer as well as the performer. Different patch sizes are configured to compare the time efficiency under different token dimensions. For the CIFAR-100 dataset which possesses a 32x32 image size, the correspondence between patch size and number of tokens after the patch embedding is listed below in Table 5:

Patch Size	8×8	4×4	2×2	1×1
Token	16	64	256	1024

Table 5. Patch-Token Correspondence for CIFAR-100 Dataset

In theory, the attention mechanism costs $\mathrm{O}(\mathrm{L}^2\mathrm{d})$ timewise for the regular transformer, where L is the number of tokens and d stands for hidden dimension. For the performer, calculating the attention costs $\mathrm{O}(\mathrm{Lrd})$, where r indicates the kernel dimension. Since the ReLU kernel transformation without the projection matrix does not change the dimension size, the time complexity of calculating attention with the performer is $\mathrm{O}(\mathrm{Ld}^2)$ where $\mathrm{d} = 64$ in our case. Therefore, the regular transformer theoretically has better time efficiency than the performer when the token dimension $\mathrm{L} < 64$, while the performer is more advantageous when $\mathrm{L} > 64$.

In our experiments, the regular transformer and performer are both “small” sized and equipped with mixed-type rotary positional embeddings. Due to graphic memory limitations, batch sizes for both models are set to 64 at 1×1 patch configuration and set to 512 for other configurations in order to reduce overall training time. The average training and inference time costs for each epoch are illustrated in Figure 13 and Figure 14. Results indicate that the regular transformer is slightly faster than the performer at 8×8 patch size, while the performer enjoys significantly less training and inference time when the patch configuration is 2×2 and 1×1, which corresponds to large token dimensions (256 and 1024). The test results are highly consistent with our theoretical analysis.

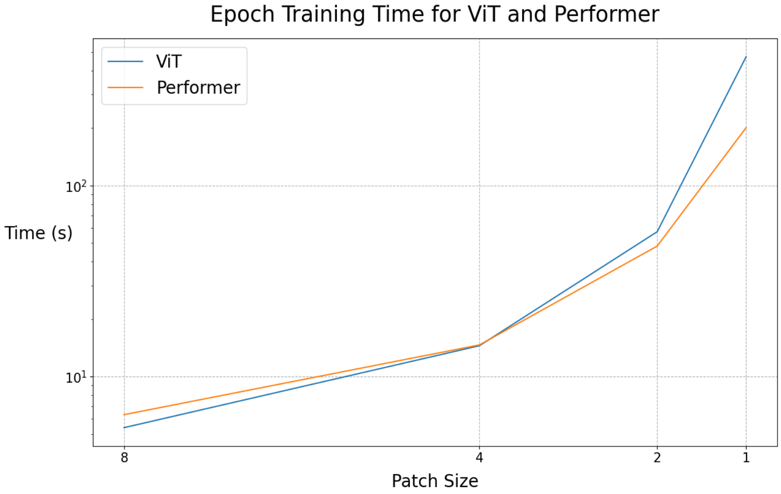


Figure 13. Epoch Training Time for Regular Transformer and Performer

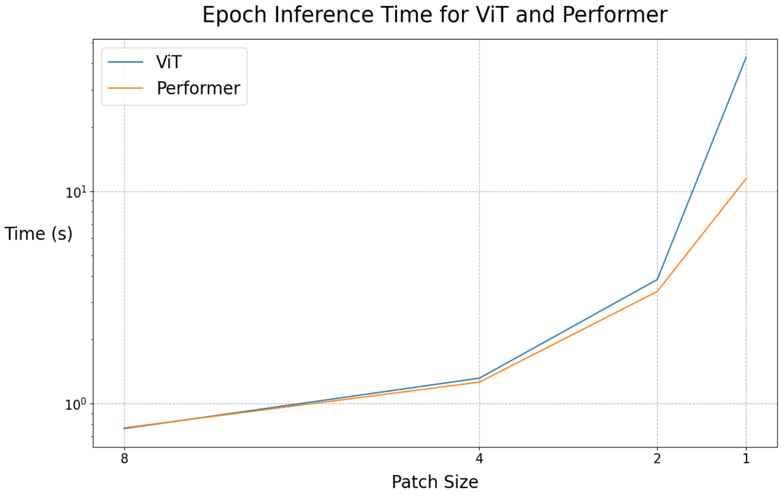


Figure 14. Epoch Inference Time for Regular Transformer and Performer

6 Conclusion and future work

6.1 Discussion

We provides a detailed evaluation of the DeiT and Performer architectures, focusing on the impact of different positional embedding strategies and time efficiency across varying token configurations. The findings demonstrate that Absolute Positional Embedding (APE) is highly effective in capturing positional relationships within low-resolution datasets such as CIFAR-100. Models utilizing APE consistently outperformed others, achieving a test accuracy of 60.42% in DeiT and 59.63% in Performer. Among all RoPE variants in our experiments, it is shown that Performer with RoPE-Axial and APE reaches the best accuracy, which is 59.30%.

The analysis of training and inference efficiency highlighted a nuanced performance difference between the two architectures. While the regular transformer exhibited marginally better efficiency at smaller token dimensions (e.g., 8 × 8 patch sizes), the Performer displayed significant advantages as the token dimensions increased (e.g., 1 × 1 patch sizes). This trend aligns closely with theoretical predictions, confirming the Performer's scalability and efficiency when handling high-token configurations.

The consistency between the theoretical and experimental results strengthens the validity of the findings presented in this report. However, certain trade-offs in embedding performance, particularly for hybrid approaches like RoPE Axial - APE, merit further investigation to optimize their application. These insights could provide a robust foundation for advancing the understanding of positional embedding strategies in transformer architectures and their adaptability to datasets with varying resolutions.

Models	Accuracy	Models	Accuracy
Deit_ape_base	60.42%	Performer_ape_base	59.63%
Deit_ape_small	48.11%	Performer_ape_small	50.87%
Deit_rope_mixed_base	57.55%	Performer_rope_mixed_base	57.75%
Deit_rope_mixed_small	46.71%	Performer_rope_mixed_small	47.50%
Deit_rope_axial_base	58.09%	Performer_rope_axial_base	58.03%
Deit_rope_axial_small	47.18%	Performer_rope_axial_small	47.53%
Deit_rope_mixed_ape_base	58.49%	Performer_rope_mixed_ape_base	50.97%
Deit_rope_mixed_ape_small	47.41%	Performer_rope_mixed_ape_small	49.53%
Deit_rope_axial_ape_base	58.54%	Performer_rope_axial_ape_base	59.30%
Deit_rope_axial_ape_small	47.80%	Performer_rope_axial_ape_small	49.67%

Table 6. The accuracy of all models

6.2 Future work

In our experiments, we reproduced the method described in the original paper and verified its feasibility. Moreover, some of our experimental results contradicted those reported in the original paper, and we have analyzed several potential reasons for these discrepancies in the report. The findings of this study open several avenues for future research that can further enhance the understanding and application of transformer architectures and positional embedding strategies.

Hybrid approaches, such as RoPE Axial APE, demonstrated promising results but still underperformed compared to standalone APE. Future studies could focus on systematically quantifying the trade-offs between accuracy and computational cost for these hybrid embeddings. Such an analysis could lead to improvements in their design, ensuring they effectively combine the strengths of absolute and relative positional encoding.

Performer’s efficient attention mechanism showed significant advantages at larger token dimensions but faced limitations at smaller ones. Future research could focus on introducing adaptive mechanisms within the Performer framework, allowing it to dynamically adjust based on token configurations or dataset characteristics. Such adaptability could improve its performance across a wider range of applications.

Achieving an optimal balance between accuracy, efficiency, and scalability remains a key challenge in transformer-based architectures. Future research could investigate multi-objective optimization strategies that consider these factors holistically, providing models that are not only high-performing but also cost-effective and scalable across different environments.

By pursuing these directions, future work can build on the results of this study to refine transformer architectures and positional embedding strategies. These advancements would support their broader adoption in applications requiring diverse datasets and computational constraints, ultimately contributing to the development of more efficient and versatile deep learning models.

References

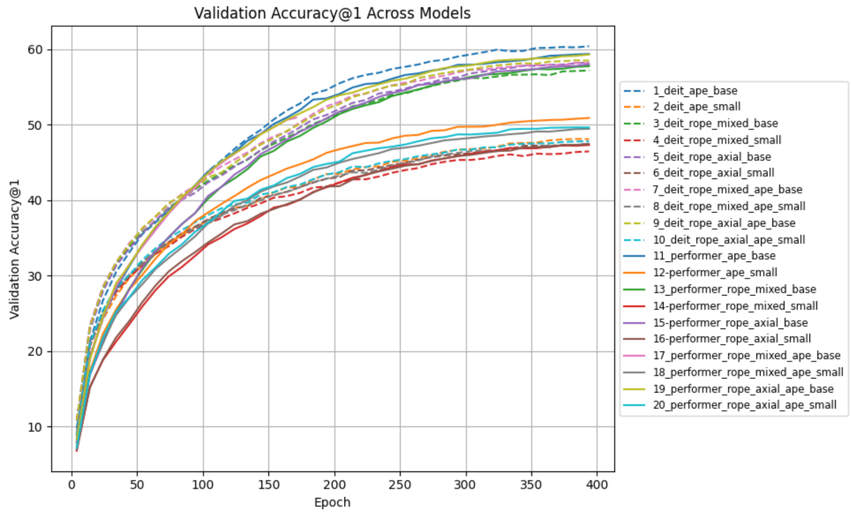
1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (pp. 5998-6008).
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations.
3. Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (pp. 464-468).
4. Su, J., Lu, Y., Pan, S., Guo, Z., Cao, Y., & Xiong, D. (2021). RoFormer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864.
5. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlós, T., ... & Chatterjee, A. (2021). Rethinking attention with performers. In International Conference on Learning Representations.
6. Heo, B., Park, S., Han, D., & Yun, S. (2024). Rotary position embedding for vision transformer. CoRR, abs/2403.13298.
7. Krizhevsky, A. (2009). Learning multiple layers of features from tiny images (Tech. Rep.). University of Toronto.

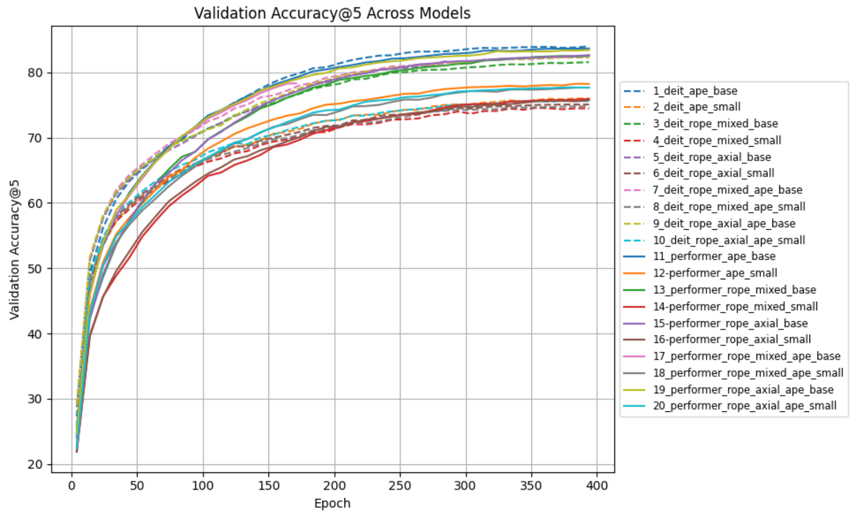
Contribution Form

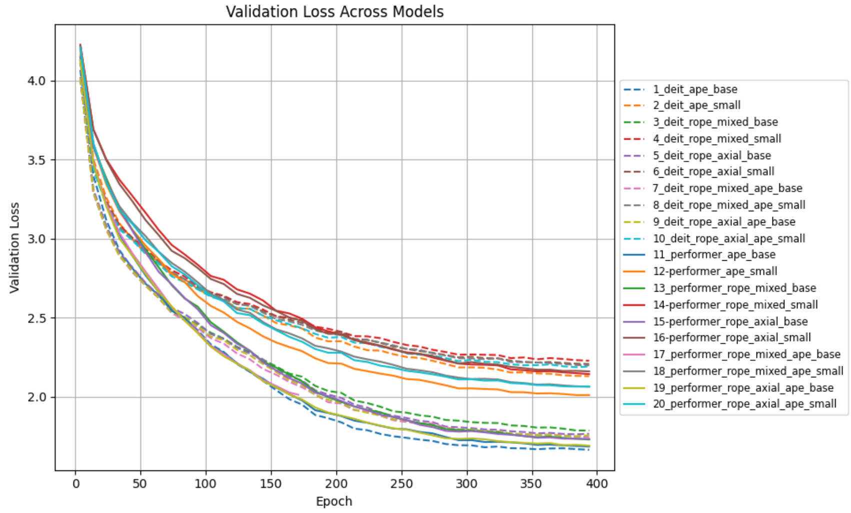
	hd2573	jx2598	wz2708	xc2763	zw3057
Topic Review	√		√	√	
Project Design	√	√		√	√
Coding	√	√	√	√	√
Training & Evaluation		√	√	√	√
Report Writing	√	√	√	√	√

*** All members contribute almost equally.**

Appendix


Validation Accuracy@1 Across Models
Validation Accuracy@1
10
20
30
40
50
60
0
50
100
150
200
250
300
350
400
Epoch
1_deit_ape_base
2_deit_ape_small
3_deit_rope_mixed_base
4_deit_rope_mixed_small
5_deit_rope_axial_base
6_deit_rope_axial_small
7_deit_rope_mixed_ape_base
8_deit_rope_mixed_ape_small
9_deit_rope_axial_ape_base
10_deit_rope_axial_ape_small
11_performer_ape_base
12-performer_ape_small
13_performer_rope_mixed_base
14-performer_rope_mixed_small
15-performer_rope_axial_base
16-performer_rope_axial_small
17_performer_rope_mixed_ape_base
18_performer_rope_mixed_ape_small
19_performer_rope_axial_ape_base
20_performer_rope_axial_ape_small


Validation Accuracy@5 Across Models
Validation Accuracy@5
20
30
40
50
60
70
80
0
50
100
150
200
250
300
350
400
Epoch
1_deit_ape_base
2_deit_ape_small
3_deit_rope_mixed_base
4_deit_rope_mixed_small
5_deit_rope_axial_base
6_deit_rope_axial_small
7_deit_rope_mixed_ape_base
8_deit_rope_mixed_ape_small
9_deit_rope_axial_ape_base
10_deit_rope_axial_ape_small
11_performer_ape_base
12-performer_ape_small
13_performer_rope_mixed_base
14-performer_rope_mixed_small
15-performer_rope_axial_base
16-performer_rope_axial_small
17_performer_rope_mixed_ape_base
18_performer_rope_mixed_ape_small
19_performer_rope_axial_ape_base
20_performer_rope_axial_ape_small


Validation Loss Across Models
Validation Loss
2.0
2.5
3.0
3.5
4.0
0
50
100
150
200
250
300
350
400
Epoch
1_deit_ape_base
2_deit_ape_small
3_deit_rope_mixed_base
4_deit_rope_mixed_small
5_deit_rope_axial_base
6_deit_rope_axial_small
7_deit_rope_mixed_ape_base
8_deit_rope_mixed_ape_small
9_deit_rope_axial_ape_base
10_deit_rope_axial_ape_small
11_performer_ape_base
12-performer_ape_small
13_performer_rope_mixed_base
14-performer_rope_mixed_small
15-performer_rope_axial_base
16-performer_rope_axial_small
17_performer_rope_mixed_ape_base
18_performer_rope_mixed_ape_small
19_performer_rope_axial_ape_base
20_performer_rope_axial_ape_small