Sparse Cayley-STRING Relative Positional Encoding for Vision Transformers

Begum Cicekdag, Hongrong Yang, Soham Samal

December 15, 2025

Contents

1 Introduction & Motivation

Relative positional information is a key ingredient for strong Vision Transformer (ViT) performance: without it, self-attention cannot reliably capture the 2D structure of images. Among many approaches, rotary-style encodings are attractive because they inject relative geometry through norm-preserving rotations of the query and key representations, often generalizing well across resolutions.

Cayley-STRING extends rotary encodings by composing them with a learnable orthogonal, parameterized via the Cayley transform of a skew-symmetric generator. While this preserves desirable stability properties, the standard formulation learns a dense generator and applies the resulting transform with a per-head linear solve, introducing non-trivial overhead in lightweight ViTs.

This project studies whether structured sparsity in the skew-symmetric generator can retain most of the benefits of Cayley-STRING while substantially improving efficiency. We implement a small ViT with Cayley-STRING as a baseline and propose a sparse variant of the generator that reduces computational cost (and, in some cases, admits a closed-form update). We evaluate both approaches on MNIST and CIFAR-10, reporting not only best test accuracy but also training time per epoch and inference latency to characterize the practical accuracy-efficiency trade-off.

2 Background

2.1 Vision Transformer Architecture

A Vision Transformer[1] represents an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ as a sequence of patch tokens. The image is partitioned into non-overlapping patches of size $P \times P$, each patch is linearly projected into an embedding of dimension D, and a special classification token (`[CLS]`) may be prepended to aggregate global information. Let $N = (H/P)(W/P)$ be the number of patches. After patch embedding, the input sequence takes the form

$$\mathbf{X}_0 = [\mathbf{x}_{\text{cls}};\ \mathbf{x}_1;\ \cdots;\ \mathbf{x}_N] \in \mathbb{R}^{(N+1) \times D}. \tag{1}$$

A ViT then applies L Transformer blocks, each consisting of multi-head self-attention and an MLP, typically with residual connections and layer normalization:

$$\mathbf{X}'_\ell = \mathbf{X}_{\ell-1} + \mathrm{MSA}(\mathrm{LN}(\mathbf{X}_{\ell-1}))\,, \tag{2}$$

$$\mathbf{X}_\ell = \mathbf{X}'_\ell + \mathrm{MLP}(\mathrm{LN}(\mathbf{X}'_\ell))\,, \qquad \ell = 1, \ldots, L. \tag{3}$$

In each attention layer, queries, keys, and values are computed by linear projections of token embeddings:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \tag{4}$$

and attention weights are formed via scaled dot products:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V}, \tag{5}$$

where d_h is the per-head dimension. The final classifier typically uses the `[CLS]` token representation from the last layer.

2.2 Relative Positional Encodings for ViTs

Self-attention alone is invariant to permutations of the input tokens; positional information must be injected to recover spatial structure. In ViTs, a common approach is absolute positional embeddings, where each token index receives a learned vector added to $\mathbf{X}_0$. While effective, absolute embeddings can be sensitive to perturbations and do not explicitly encode relative geometry.

Relative positional encodings (RPEs) instead modify attention scores based on pairwise offsets between tokens. A generic RPE formulation augments the attention logits with a relative bias term:

$$\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \mathbf{B}\right), \tag{6}$$

where $\mathbf{B}_{ij}$ depends on the relative displacement between tokens i and j (e.g., differences in 2D coordinates on the patch grid). This can encode translation-like invariances more naturally than absolute embeddings.

Rotary position embeddings (RoPE)[2] provide an elegant RPE mechanism by applying position-dependent orthogonal rotations to queries and keys. For a token at position $\mathbf{r}_i$, RoPE defines a rotation matrix $\mathbf{R}(\mathbf{r}_i)$ and replaces

$$\mathbf{q}_i \leftarrow \mathbf{R}(\mathbf{r}_i)\mathbf{q}_i, \qquad \mathbf{k}_i \leftarrow \mathbf{R}(\mathbf{r}_i)\mathbf{k}_i. \tag{7}$$

Because inner products between rotated vectors depend on relative rotations, the resulting attention scores become functions of relative displacement. For images, a common extension is 2D RoPE, which assigns each patch token a coordinate (x_i, y_i) and applies rotary factors along each axis (via an axial construction that splits the head dimension into two parts).

2.3 Cayley-STRING Relative Positional Encodings

Cayley-STRING[3] builds on RoPE by introducing additional learnable orthogonal mixing to improve expressivity in higher dimensions while preserving geometric structure. In the Transformer setting, the position-dependent map can be expressed as a composition

$$\mathbf{R}(\mathbf{r}_i) \;=\; \mathrm{RoPE}(\mathbf{r}_i)\,\mathbf{P}, \qquad \mathbf{P} \in O(d_h), \tag{8}$$

where $\mathbf{P}$ is a learnable orthogonal matrix (per head). Orthogonality is desirable because it preserves norms and stabilizes training, while still allowing meaningful rotations of the query/key features.

To parameterize $\mathbf{P}$ without explicitly enforcing constraints during optimization, Cayley-STRING uses the Cayley transform. Given a skew-symmetric generator $\mathbf{S}$, $\mathbf{S}^\top = -\mathbf{S}$, the Cayley transform defines

$$\mathbf{P} = (\mathbf{I} - \mathbf{S})(\mathbf{I} + \mathbf{S})^{-1}. \tag{9}$$

The baseline approach learns a dense $\mathbf{S}$, which provides $O(d_h^2)$ degrees of freedom but requires solving a linear system to apply $\mathbf{P}$ to $\mathbf{q}_i$ and $\mathbf{k}_i$, which incurs non-negligible overhead. This computational cost motivates structured variants that restrict $\mathbf{S}$ to reduce overhead while maintaining the orthogonality guarantee induced by the Cayley transform.

In our experiments, we adopt Cayley-STRING as the positional mechanism inside a small ViT and compare the standard dense generator against a structured sparse alternative, measuring both predictive performance and runtime efficiency on MNIST and CIFAR-10.

3 Methodology

3.1 Learnable Block-Diagonal Skew-Symmetric Variant

We propose a structured and sparse parameterization of the skew-symmetric generator $\mathbf{S}$ that significantly reduces computational cost while preserving the orthogonality guarantee.

Specifically, for each attention head, we constrain $\mathbf{S}$ to be block-diagonal with 2×2 skew-symmetric blocks:

$$\mathbf{S} = \operatorname{diag}\left(\begin{bmatrix} 0 & a_1 \\ -a_1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & a_2 \\ -a_2 & 0 \end{bmatrix}, \dots \right), \tag{10}$$

where $\{a_i\}$ are learnable scalar parameters. If the head dimension d is odd, the remaining dimension is left unchanged.

This construction ensures skew-symmetry by design and reduces the number of learnable parameters from $\mathcal{O}(d^2)$ to $\mathcal{O}(d)$ per head.

The use of 2×2 blocks is motivated by the intrinsic structure of real skew-symmetric matrices. Any real skew-symmetric matrix can be decomposed into independent planar rotations, each acting on a two-dimensional subspace. Therefore, a 2×2 skew-symmetric block is the smallest non-trivial unit capable of generating an orthogonal transformation via the Cayley transform.

Using larger blocks (e.g., 4×4) would couple multiple rotation planes and substantially increase computational complexity, typically requiring matrix inversion or more expensive operations. In contrast, the 2×2 block-diagonal construction preserves the essential rotational degrees of freedom while admitting a closed-form Cayley update, enabling efficient training and inference. This design provides a practical balance between expressiveness and computational efficiency for small Vision Transformer models.

3.2 Closed-form Cayley Transform

For each 2×2 block in Eq. (10), the Cayley transform admits a closed-form expression. Given

$$\mathbf{S}_i = \begin{bmatrix} 0 & a_i \\ -a_i & 0 \end{bmatrix}, \tag{11}$$

the corresponding orthogonal block $\mathbf{P}_i$ is

$$\mathbf{P}_i = \frac{1}{1 + a_i^2} \begin{bmatrix} 1 - a_i^2 & -2a_i \\ 2a_i & 1 - a_i^2 \end{bmatrix}. \tag{12}$$

Applying Eq. (12) to each block allows the full Cayley transform to be computed without explicitly forming matrices or solving linear systems. As a result, the transformation can be implemented using only element-wise operations.

3.3 Integration with Attention

The proposed block-diagonal Cayley transform is applied independently to the query and key representations for each attention head:

$$\tilde{\mathbf{Q}} = \mathbf{PQ}, \quad \tilde{\mathbf{K}} = \mathbf{PK}, \tag{13}$$

where $\mathbf{P}$ is the orthogonal matrix implicitly defined by the block-diagonal construction. The transformed queries and keys are then used in standard scaled dot-product attention.

This design preserves the orthogonality and rotational structure encouraged by Cayley-based positional encodings, while substantially reducing both training and inference cost.

3.4 Banded Skew-Symmetric Generator

As an intermediate design point between the dense baseline and the block-diagonal construction, we consider a banded parameterization of the skew-symmetric generator $\mathbf{S}$. This was motivated by [4], which uses structured parameterizations (e.g. projections) to improve efficiency. For each attention head, we restrict $\mathbf{S}$ to have nonzero entries only within a fixed bandwidth b around the main diagonal:

$$\mathbf{S}_{ij} = 0 \quad \text{if} \quad |i - j| > b. \tag{14}$$

The diagonal entries are always constrained to zero to maintain skew-symmetry.

This parameterization preserves local interactions while significantly reducing the number of learnable parameters from $\mathcal{O}(d^2)$ to $\mathcal{O}(bd)$ per head. After applying the band mask, skew-symmetry is enforced explicitly by

$$\mathbf{S} = \mathbf{A} - \mathbf{A}^\top, \tag{15}$$

where $\mathbf{A}$ denotes the masked unconstrained parameter matrix.

Unlike the block-diagonal variant, the banded generator still produces a dense orthogonal transformation after the Cayley transform, but with lower computational overhead due to the reduced effective degrees of freedom. The bandwidth b therefore controls a smooth trade-off between expressiveness and efficiency. In our experiments, we evaluate small bandwidths ($b = 2, 4$), which already capture most of the useful rotational structure for the head dimensions considered.

3.5 Top-k Sparse Skew-Symmetric Generator

We further explore a sparsity-driven alternative in which the skew-symmetric generator is constrained using a Top-k selection mechanism. For each attention head, we retain only the k largest-magnitude entries of an unconstrained parameter matrix and set all remaining entries to zero. Formally, let $\mathbf{A}$ denote the raw learnable matrix and let Z_k index the k largest entries of $|\mathbf{A}|$ (excluding the diagonal). The masked generator is defined as

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{A}_{ij}, & (i,j) \in Z_k, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Skew-symmetry is then enforced via $\mathbf{S} = \mathbf{A} - \mathbf{A}^\top$.

To ensure structural consistency, the sparsity mask is symmetrized so that if (i, j) is retained, then (j, i) is also kept. Diagonal entries are explicitly excluded, as they cancel under skew-symmetrization. This yields a generator with at most $\mathcal{O}(k)$ nonzero entries per head, independent of the head dimension.

The Top-k variant provides fine-grained control over sparsity and serves as a complementary perspective to banded constraints. However, because the Cayley transform produces a dense orthogonal matrix regardless of generator sparsity, reducing k primarily affects computational cost and training dynamics rather than the form of the resulting transformation. In our experiments, we consider moderate values of k (24, 48) that balance sparsity with stability.

3.6 Comparison of Structured Generators

Taken together, the block-diagonal, banded, and Top-k constructions represent a spectrum of structured parameterizations for the Cayley-STRING generator. The block-diagonal 2×2 variant admits a closed-form Cayley transform and completely eliminates matrix inversion. The banded and Top-k variants retain the standard Cayley transform but reduce the effective dimensionality of the generator. These designs allow us to systematically study the trade-offs between expressiveness, computational efficiency, and empirical performance within a unified Cayley-based framework.

3.7 Parameters

For fair comparison, we adopt dataset-specific training configurations. On MNIST, models are trained for 10 epochs using AdamW with a learning rate of 2×10^{-3} and weight decay 1×10^{-4}. On CIFAR-10, we train for 25 epochs with a learning rate of 4×10^{-3} and the same weight decay. We all save the best model rather than the final model during training process.

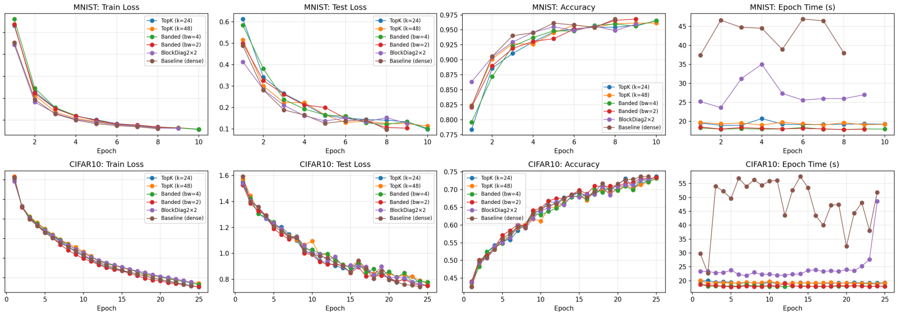

Figure 1: Training results of baseline and the proposed variants.

Table 1: Performance and efficiency comparison on MNIST and CIFAR-10.

Dataset	Model	Params (M)	Best Acc.	Train s/epoch	Train s_total	ms/batch	ms/img
MNIST	Baseline (dense)	0.117	0.9680	42.92	343.40	9.13	0.071
MNIST	BlockDiag2×2	0.115	0.9595	27.43	246.89	7.18	0.056
MNIST	Banded (bw=2)	0.117	0.9681	18.09	162.85	9.22	0.072
MNIST	Banded (bw=4)	0.117	0.9657	18.03	180.28	9.26	0.072
MNIST	TopK ($k = 24$)	0.117	0.9650	19.35	193.50	11.76	0.092
MNIST	TopK ($k = 48$)	0.117	0.9615	19.36	193.62	11.75	0.092
CIFAR-10	Baseline (dense)	0.118	0.7369	47.59	1142.24	9.35	0.073
CIFAR-10	BlockDiag2×2	0.116	0.7299	24.33	583.87	7.32	0.057
CIFAR-10	Banded (bw=2)	0.118	0.7345	18.12	453.05	9.25	0.072
CIFAR-10	Banded (bw=4)	0.118	0.7324	18.07	451.65	9.22	0.072
CIFAR-10	TopK ($k = 24$)	0.118	0.7374	19.28	481.93	11.70	0.091
CIFAR-10	TopK ($k = 48$)	0.118	0.7316	19.22	480.56	11.72	0.092

Note: Params (M) denotes total learnable parameters (in millions). *Best Acc.* is the highest test accuracy achieved during training. *Train s/epoch* is the average wall-clock training time per epoch, and *Train s_total* is the total training time summed over all epochs. *ms/batch* and *ms/img* report inference latency measured on the test set (milliseconds per batch / per image). The Baseline uses a dense skew-symmetric generator with a matrix solve for the Cayley transform, while BlockDiag2×2 uses independent 2×2 skew-symmetric blocks with a closed-form Cayley update (no linear solve). Banded and TopK variants restrict the generator via banded sparsity or Top-k masking, respectively, while retaining the full Cayley transform.

4 Results

Table 1 and Figure 1 compare the dense Cayley-STRING baseline with three structured variants: block-diagonal 2×2, banded sparse, and Top-k sparse generators, evaluated on MNIST and CIFAR-10. Across both datasets, all variants achieve comparable classification accuracy, while exhibiting substantial differences in training and inference efficiency.

Accuracy. On MNIST, all models reach high accuracy, with only marginal differences between variants. The dense baseline achieves the highest peak accuracy, but the gap relative to structured variants is small. This is expected given the simplicity of MNIST and the limited head dimension ($d_h = 12$), where even restricted rotational parameterizations provide sufficient expressivity. Similarly, on CIFAR-10, the dense Cayley-STRING model slightly outperforms the sparse variants, but the accuracy differences remain slim across all configurations.

Interestingly, increasing sparsity, either by reducing the band width or by lowering k in the Top-k variant, does not lead to a significant degradation in performance. This suggests that, for small ViT models and moderate-scale datasets, the full $O(d_h^2)$ degrees of freedom of a dense skew-symmetric generator are not strictly necessary to capture useful relative positional structure.

Efficiency. In contrast to accuracy, efficiency metrics show clear and consistent differences. Both training time per epoch and inference latency are substantially lower for structured variants than for the dense baseline. The dense Cayley-STRING model requires solving a linear system per head and per forward pass, which dominates runtime even for relatively small head dimensions. All sparse variants reduce this overhead by restricting the generator $\mathbf{S}$, leading to faster training and lower inference latency.

Among the structured approaches, the block-diagonal 2×2 variant is the most efficient. Because each block admits a closed-form Cayley transform, this variant avoids linear solves entirely and relies only on element-wise operations. As a result, it achieves the lowest inference latency across both datasets, while maintaining competitive accuracy.

The banded and Top-k variants occupy an intermediate regime. Although they still require a linear solve, the sparsity of $\mathbf{S}$ reduces constant factors and leads to measurable speedups over the dense baseline. Differences between bandwidths (bw = 2 vs. bw = 4) and Top-k values ($k = 24$ vs. $k = 48$) are minor, reflecting the small head dimension and the fact that the Cayley transform produces a dense orthogonal matrix regardless of generator sparsity.

Effect of Sparsity The similarity in performance across banded and Top-k variants highlights an important property of Cayley-based parameterizations. While the generator $\mathbf{S}$ is sparse or structured, the resulting orthogonal matrix $\mathbf{P}$ is dense. Consequently, different sparsity patterns in $\mathbf{S}$ primarily affect computational cost rather than the qualitative form of the transformation applied to queries and keys. This explains why varying the bandwidth or the Top-k threshold has limited impact on accuracy in our experiments.

Overall, the results demonstrate a clear accuracy-efficiency trade-off. The dense Cayley-STRING baseline provides the strongest performance, but at a significantly higher computational cost. Structured sparse variants retain most of the accuracy benefits while substantially reducing training and inference time. In particular, the block-diagonal 2×2 variant offers the most favorable trade-off in this regime, achieving near-baseline accuracy with the lowest runtime overhead. These findings suggest that structured Cayley-STRING parameterizations are a practical alternative to dense generators in small Vision Transformers, especially when efficiency is a primary concern.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: `https://arxiv.org/abs/2010.11929`.

[2] Jianlin Su et al. "RoFormer: Enhanced Transformer with Rotary Position Embedding". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021. URL: `https://arxiv.org/abs/2104.09864`.

[3] Connor Schenck, Liyang Yu, Zhen Huang, et al. "Learning the RoPEs: Better 2D and 3D Position Encodings with STRING". In: *arXiv preprint arXiv:2502.02562* (2025). arXiv: `2502.02562 [cs.CV]`. URL: `https://arxiv.org/abs/2502.02562`.

[4] Zhen Li et al. "Learning to Search Efficiently in High Dimensions". In: *Advances in Neural Information Processing Systems*. 2011.