Beyond Addition: HDC Binding for Transformers' Position Encoding applied to Time Series Classification

Kenny Schlegel[1], Jose Juan Pena Gomez[2], Dmitri A. Rachkovskij[2,3], Stefan Streif[1], Evgeny Osipov[2]
[1]*Chemnitz University of Technology*, Chemnitz, Germany
[2]*Luleå University of Technology*, Luleå, Sweden
[3]*Institute of Information Technologies and Systems*, Kyiv, Ukraine
Email: kenny.schlegel@etit.tu-chemnitz.de

***Abstract*—Transformers integrate positional information by adding Positional Encodings (PE) to token embeddings, which is particularly relevant for time series classification (TSC). We revisit PE from a Hyperdimensional Computing (HDC) perspective and cast positional encoding as a representation design problem with two axes: (i) the operation used to integrate token and position information and (ii) the similarity structure of the positional vectors. We compare additive superposition to HDC binding, instantiated by Circular Convolution, and evaluate sinusoidal, random, and Fractional Power Encoding (FPE) positional vectors. Across the UEA multivariate time series classification benchmark [2], binding consistently improves performance over addition, and circular-convolution binding combined with similarity-aware FPE achieves the best average performance (64.43% mean accuracy vs. 56.81% for the additive baseline). These findings suggest that positional integration is a first-class design choice for time-series Transformers.**

***Index Terms*—Hyperdimensional Computing, Vector Symbolic Architectures, Time Series Classification, Transformers, Positional Encoding**

I. Introduction

Modeling temporal structure with Transformers requires explicit positional information, since self-attention is inherently permutation invariant. In Transformer's sequence processing, positional information is typically injected by *additive* superposition of positional encodings (PEs) and token embeddings, following the original formulation [3]. Despite its widespread use, this design choice is rarely questioned and lacks a principled justification.

Recent studies in time series classification have shown that Transformer variants with additive positional encodings can struggle to capture temporal dependencies, sometimes outperformed by simpler linear models [4]. This suggests that the limitation lies not only in the choice of PE itself, but in how positional information is integrated into token representations. In particular, their additive superposition preserves similarity to both content and position but does not form a distinct composite representation, which may lead to ambiguities and degraded temporal structure.

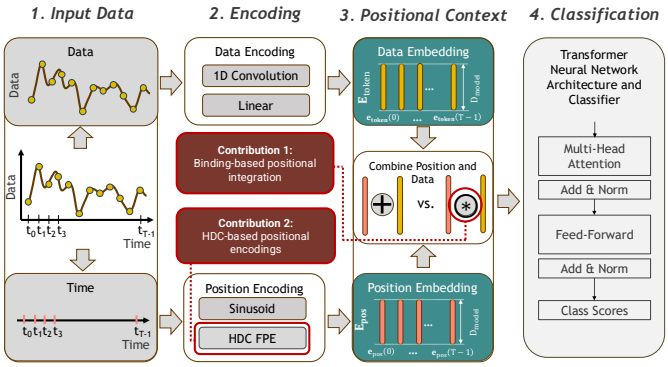

Fig. 1: Overview of the two design axes: (1) the integration operation between token and positional embeddings and (2) the representation and similarity structure of positional vectors. Replacing addition with HDC binding and using similarity-aware encodings yields more expressive representations for time series classification.

Motivated by these limitations, several recent Transformer architectures incorporate positional information through multiplicative or interaction-based mechanisms rather than simple addition. Examples include attention-level formulations and Rotary Positional Embeddings (RoPE), which encode position via structured algebraic interactions between content and position-dependent terms [5], [6]. While these approaches demonstrate the benefits of moving beyond additive positional encodings, they neither provide a systematic analysis of the positional *integration operation* itself, nor its interaction with the geometry of positional representations.

In this work, we study PEs in Transformers from the perspective of Hyperdimensional Computing (HDC). Both

This work is based on the Master's thesis of Jose Juan Pena Gomez [1]. The work of KS was part of the research project RAHD (grant no. 100686370) and was funded by the European Union and the Free State of Saxony with funds from the Regional Development Fund (ERDF). The work of DR was supported in part by the Swedish Foundation for Strategic Research (SSF, grant nos. UKR22-0024, UKR24-0014). The work of EO and DR was supported in part by the Swedish Research Council (VR grant no. 2022-04657). EO and DR acknowledge the computational resources provided by the National Academic Infrastructure for Supercomputing in Sweden, partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Transformers and HDC are built on (high-dimensional) vector representations, where information is encoded through vector transformations and compared via similarity measures based on inner products. HDC provides a principled framework for such representations and distinguishes between two fundamental algebraic operations: *superposition* and *binding*. While superposition corresponds to additive aggregation, binding operations associate content with positional roles to form new composite representations, using multiplicative(-like) operations such as Circular Convolution. These operations are used in HDC to construct compositional representations, in particular enabling the association of content with positional roles while preserving meaningful similarity structure.

We therefore cast positional encoding in Transformers as a *representation design problem with two orthogonal axes* (Figure 1): (i) the operation used to integrate token and positional information, and (ii) the similarity structure induced by the positional vectors themselves. Focusing on absolute positional representations and encoder-only Transformers for multivariate TSC, we perform a controlled study of binding-based positional integration and systematically generated PEs.

Our contributions are as follows:

1) We formulate PE in Transformers as a two-axis representation design problem, separating the positional integration operation from the representation and similarity structure of positional vectors.
2) We systematically evaluate additive superposition against HDC binding with sinusoidal, random, and Fractional Power Encoding (FPE) positional vectors, including an ablation over the FPE bandwidth parameter β.
3) We conduct a controlled benchmark study on the UEA multivariate time series archive [2] and compare the best binding-based configurations to representative Transformer baselines and advanced positional mechanisms.

The remainder of the paper is organized as follows. Section 2 reviews related work on Transformers, PEs, and Hyperdimensional Computing. Section 3 describes the proposed binding operations and PE strategies. Section 4 details the experimental setup and evaluation protocol. Section 5 presents the results and analysis. Section 6 concludes the paper. To support reproducibility, the implementation and results used in this study are publicly available at GitHub.

II. Related Work

A. Transformers for Sequence Modeling

The Transformer architecture [3] has become the cornerstone of modern sequence modeling, initially revolutionizing Natural Language Processing (NLP) tasks such as machine translation. Its self-attention mechanism allows for parallel computation of dependencies across sequence elements, overcoming the limitations of recurrent neural networks.

The success of Transformers has extended beyond NLP to domains like computer vision and time series analysis. For time series classification (TSC), Transformers offer advantages in handling long sequences and capturing global patterns, but face challenges in representing temporal relationships.

B. Positional Encodings in Transformer Networks

To address the permutation invariance of self-attention, positional encodings (PEs) are added to input embeddings. The original sinusoidal PEs [3] use sine and cosine functions of different frequencies to encode absolute positions (see Section III-C). However, these have limitations such as anisotropy and poor distance awareness [7], [8]. Subsequent developments include learnable PEs, relative PEs that encode pairwise relationships, and task-specific variants. For time series, tAPE [9] adapts sinusoidal encodings for better isotropy, while RoPE [6] applies rotational transformations to query and key vectors for relative position encoding. Advanced attention mechanisms like MLA [10] integrate positional information structurally within the attention computation.

C. Hyperdimensional Computing

Hyperdimensional Computing [11] is a computing paradigm based on distributed representations in very high-dimensional spaces, where randomly sampled hypervectors are quasi-orthogonal with high probability. Information is represented by hypervectors $\mathbf{x} \in \mathbb{R}^D$ (with D in hundreds, thousands, or more) and manipulated using a small set of algebraic operations that preserve interpretable similarity relations.

In this work, we focus on two core operations that enable compositional hyperdimensional representations [12], [13]: (i) *superposition* ($+$), which aggregates multiple hypervectors and (ii) *binding* ($\circledast$), which associates two hypervectors into a new representation. Binding can be realized in different ways depending on the underlying vector space, including Circular Convolution, component-wise multiplication, or matrix-based transformations. These realizations differ in their algebraic properties and exhibit distinct trade-offs with respect to robustness and computational complexity, as systematically compared in [14], [15].

In the following, we use binding based on Circular Convolution and its Fourier-domain implementation, as used in Holistic Reduced Representations (HRR) [16]. This operation is defined for real-valued vectors and forms the basis for similarity-preserving positional encoding strategies (Fractional Power Encoding), as introduced in Section III-C. The benefits of temporal position encoding via binding, combined with similarity-preserving positional vectors based on Fractional Power Encoding, have already been demonstrated in prior work. In particular, this approach has shown strong performance for time series representation and classification by preserving graded temporal similarity and supporting structured sequence encoding [17]–[19]. These properties suggest that binding-based, similarity-preserving PEs might not be limited to hyperdimensional models, but also have potential as an alternative PE mechanism in Transformer architectures.

III. Method

We study positional information in Transformers for time series as a *representation design problem*. In contrast to

standard approaches that treat PEs as an additive incorporation, we explicitly decompose positional integration into two orthogonal design choices: (1) the algebraic operation used to associate token and positional representations, and (2) the representational structure of the positional encoding vectors themselves. Both are motivated by Hyperdimensional Computing (HDC), with a particular emphasis on Fractional Power Encoding (FPE) as a method of HDC. Throughout, we consider an encoder-only Transformer for multivariate time series classification.

A. Problem Setting and Notation

Let $X \in \mathbb{R}^{B\times T\times N}$ denote a batch of multivariate time series, with batch size B, sequence length T, and N input channels. An input embedding layer maps each time step to a D_{model}-dimensional token embedding $E_{\text{token}} \in \mathbb{R}^{B\times T\times D_{\text{model}}}$. A PE produces position encoding vectors $E_{\text{pos}} \in \mathbb{R}^{T\times D_{\text{model}}}$. The central question addressed in this work is how E_{token} and E_{pos} should be combined to form the input representation $Z \in \mathbb{R}^{B\times T\times D_{\text{model}}}$ passed to the Transformer encoder stack. We use bold lowercase symbols to denote instances of embedding vectors at a specific time step $t \in \{0,\dots,T-1\}$, i.e. $\mathbf{e}_{\text{token}}(t), \mathbf{e}_{\text{pos}}(t) \in \mathbb{R}^{D_{\text{model}}}$ refers to the token and positional embedding vectors at time t.

B. Input Embedding Layers

Before entering the Transformer encoder stack, the input time series data $X \in \mathbb{R}^{B\times T\times N}$ (potentially multivariate, with N channels) is transformed into the model's token embedding $E_{\text{token}} \in \mathbb{R}^{D_{\text{model}}}$. We consider two types of this transformation: (1) Linear projection and (2) 1-D convolution.

(1) The transformation by the linear projection can be performed by a learnable linear layer $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{N\times D_{\text{model}}}$ without bias, which maps the input features $\mathbf{x}_t \in \mathbb{R}^N$ at each time step t to the D_{model}-dimensional embedding space. This mapping is applied independently at each time step:

$$E_{\text{token}} = X\mathbf{W}_{\text{emb}} \cdot \sqrt{D_{\text{model}}}, \tag{1}$$

(2) The transformation can be done by 1-D convolution, where a one-dimensional convolutional layer is used to capture local temporal patterns before the self-attention mechanism, following ideas similar to ConvTran [9]. A convolution with D_{model} output channels, kernel size k_c, stride s_c, and padding p_c is applied:

$$E_{\text{token}} = \text{Conv1d}(X) \cdot \sqrt{D_{\text{model}}}, \tag{2}$$

The scaling by $\sqrt{D_{\text{model}}}$ used in both cases was introduced in the original Transformer and stabilizes embedding magnitudes before positional information is integrated.

C. Integration and Generation of Positional Embeddings

Strategies for integration of tokens with position embeddings. The first design axis concerns the strategy for *integration* of token and positional embeddings. The standard Transformer architecture integrates positional information by component-wise addition, which in HDC corresponds to the superposition operation:

$$\mathbf{z}(t) = \mathbf{e}_{\text{token}}(t) + \mathbf{e}_{\text{pos}}(t). \tag{3}$$

We introduce Circular Convolution, one of the bindings types in HDC [16], as a new integration position strategy in Transformers. It is akin to an interaction (association) between the token and its positional embedding.

For a token embedding $\mathbf{e}_{\text{token}}(t)$ and a position embedding $\mathbf{e}_{\text{pos}}(t)$ at time step t, the integrated positional embedding is defined as

$$\mathbf{z}(t) = \mathbf{e}_{\text{token}}(t) \circledast \mathbf{e}_{\text{pos}}(t), \tag{4}$$

where $\circledast$ denotes the Circular Convolution over the embedding dimension. In practice, Circular Convolution is computed efficiently using the convolution theorem, $\mathbf{z}(t) = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{e}_{\text{token}}(t)) \odot \mathcal{F}(\mathbf{e}_{\text{pos}}(t)))$, where $\mathcal{F}$ denotes the discrete Fourier transform applied along the embedding dimension and $\odot$ is the Hadamard product.

Positional embeddings generation strategy. The second design axis concerns the *generation* of positional embeddings. Positional embeddings differ in the way they induce a similarity function between time steps under the chosen similarity measure, e.g. cosine similarity:

$$\texttt{sim}(t, t+\Delta) = \frac{\langle \mathbf{e}_{\text{pos}}(t), \mathbf{e}_{\text{pos}}(t+\Delta)\rangle}{\|\mathbf{e}_{\text{pos}}(t)\|\,\|\mathbf{e}_{\text{pos}}(t+\Delta)\|}. \tag{5}$$

There exist different methods for producing position embeddings with different similarity preservation properties depending on the type of *similarity kernel* induced by particular generation method. The methods considered in this work are: 1. random generation, 2. sinusoidal, and 3. Fractional Power Encoding. Figure 2 illustrates similarity kernels for different methods.

a) Random Encoding: Random encodings assign an independent random vector to each position. For sufficiently large D_{model}, vectors associated with distinct positions are approximately orthogonal in expectation, and therefore have low similarity across time steps. As a result, random encodings provide a useful control case: they inject position identifiers without imposing any structured or graded notion of temporal distance.

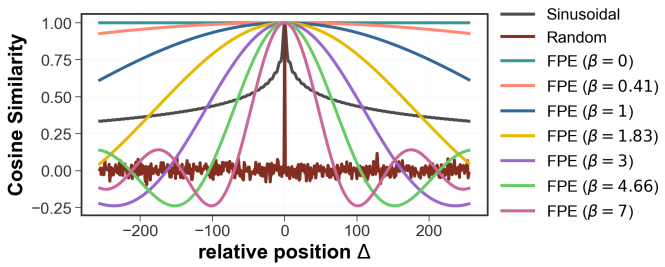

Fig. 2: Cosine similarity kernel induced by positional encodings. The similarity kernel $\text{sim}(0, \Delta)$ shows the cosine similarity between a fixed reference position ($t = 0$) and positions at offset Δ.

b) Sinusoidal Encoding (baseline): Sinusoidal encoding is a traditional positional encoding in Transformers as introduced in [3]. We use this encoding as the baseline. The sinusoidal encodings are constructed for a discrete sequence index $t \in \{0, \ldots, T-1\}$ and a frequency index $i \in \{0, \ldots, \lfloor D_{\text{model}}/2 \rfloor - 1\}$, and the angular frequencies $\omega_i = 10000^{-2i/D_{\text{model}}}$ as

$$\mathbf{e}_{\text{pos}}(t)_{2i} = \sin(t\,\omega_i), \qquad \mathbf{e}_{\text{pos}}(t)_{2i+1} = \cos(t\,\omega_i). \quad (6)$$

c) Fractional Power Encoding (FPE): Fractional Power Encoding (FPE), originally introduced in [20] and further developed in [21]–[23], constructs *similarity-preserving* positional embeddings that encode graded temporal distance through a controlled similarity kernel. FPE starts from a unitary base vector in the Fourier domain $(\mathbf{c})_j = e^{i\theta_j}, \quad |(\mathbf{c})_j| = 1$, where the phases θ_j are sampled independently from a fixed distribution (e.g., uniform on $[-\pi, \pi)$ in the present case), which determines the shape of the induced kernel. The positional representation at a (possibly non-integer) position $t \in \mathbb{R}$ is obtained by fractional exponentiation:

$$\mathbf{c}(t, \beta) = \mathbf{c}^{\beta t}, \qquad (\mathbf{c}(t, \beta))_j = e^{i\theta_j \beta t}. \quad (7)$$

The embeddings are then computed using the inverse Fourier transform as

$$\mathbf{e}_{\text{pos}}(t) = \mathcal{F}^{-1}(\mathbf{c}(t, \beta)). \quad (8)$$

The resulting similarity kernel is translation invariant, and the inverse bandwidth parameter β controls the decay of similarity with increasing positional offset $|\Delta|$, as grounded in the theory of convolutive powers [20]. The time index in our approach is normalized by the sequence length as $\tau = t/T \in [0, 1]$, and τ is used in Equation (8). A value of $\beta = 0$ yields no decay, while larger β results in a sharper similarity kernel (Figure 2).

D. Transformer Architecture for Classification

All experiments use a shallow encoder-only Transformer. The combined embeddings Z are processed by one encoder layer, each consisting of multi-head self-attention and a feed-forward network with residual connections and layer normalization.

For time series classification, the final sequence representation is obtained by global average pooling over the temporal dimension, followed by a linear classification head. This architecture isolates the effects of positional integration from deeper architectural complexity.

IV. Experimental Setup

A. Benchmark Datasets

For evaluation, we conducted experiments on the UEA Multivariate Time Series Classification Archive [2]. It provides a standardized benchmark for evaluating algorithms on multivariate time series data. Released in 2018 as a collaborative effort between the University of East Anglia and the University of California, Riverside, the archive includes 27 datasets covering a diverse range of domains such as human activity recognition, motion capture, EEG/MEG signals, and spectrogram-based audio data. All time series are formatted to equal length, contain no missing values, and include predefined train/test splits.

In addition, we include a synthetic dataset similar to [18] with multivariate time series, where the class label is determined by the position of a peak in the channels (first vs. second half of the sequence). All channels contain Gaussian noise. This setup isolates temporal position as the key discriminative factor and is used only for evaluation in Section V-C.

B. Model Variants and Controlled Factors

We study positional encoding as a representation design problem along two orthogonal axes (cf. Figure 1):

1) **Positional integration operation:** `additive` superposition (baseline) versus HDC binding by Circular Convolution `CC`.
2) **Positional embedding representation:** (i) **Null (no PE)**, i.e., no positional information is injected ($Z = E_{\text{token}}$), (ii) additive sinusoidal encodings [3] as the standard baseline, (iii) random encodings as a near-orthogonal position control, and (iv) Fractional Power Encoding (FPE) as a similarity-aware alternative (Section III-C). For FPE we evaluate multiple inverse bandwidth values $\beta \in \{0, 0.41, 1, 1.83, 3, 4.66, 7\}$ to vary the decay of positional similarity with distance.

To probe whether conclusions depend on the token embedding layer, we consider two input embedding implementations (Section III-B) to obtain E_{token}: a linear projection (1) and a 1D convolutional embedding (2).

C. Model and Training Configuration

The default model is a single-layer encoder-only Transformer with 8 attention heads, an embedding dimension of 64, and a feed-forward dimension of 256. All models are trained using AdamW with a target effective batch size of 64, a dropout rate of 0.1, and a maximum of 100 epochs.

To ensure stable and consistent training across configurations, mixed-precision training and gradient clipping are applied. Learning rates are selected using a preliminary learning rate range test to identify suitable starting values. Early stopping based on the validation loss with a patience of 10 epochs is used, and the best checkpoint is selected according to the highest validation accuracy.

D. Evaluation Metrics and Aggregation

We report the mean accuracy (Acc) across datasets (higher is better) and mean rank (lower is better). For each dataset, we repeat training and evaluation for 10 random seeds and report the seed-averaged test accuracy. Ranks are computed per dataset from these averages and then averaged across datasets and the mean rank is computed separately within the set of methods compared in each table.

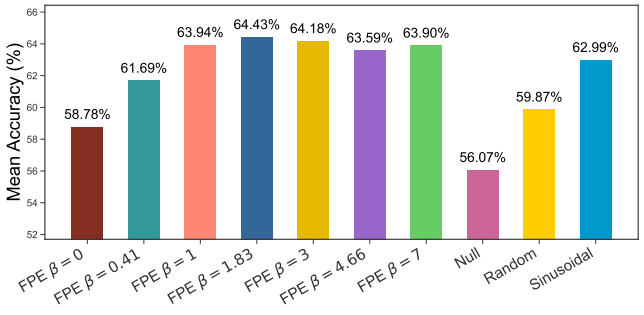

Fig. 3: Mean test accuracy across all datasets for different positional encoding schemes using *Circular Convolution* binding. FPE β denotes Fractional Power Encoding (FPE) with the bandwidth parameter β.

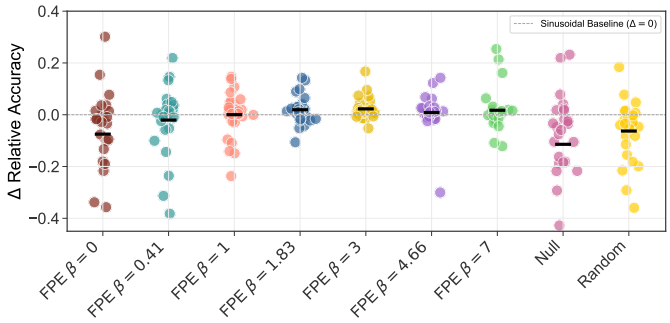

Fig. 4: Per-dataset relative accuracy $\Delta\text{Acc}_{\text{rel}}$ of alternative positional encoding schemes vs the sinusoidal baseline under *circular convolution* binding. Each point corresponds to one dataset; positive values indicate improved performance relative to the baseline.

Improvements relative to a baseline (typically the sinusoidal PE with the same integration operation) are quantified using absolute and relative accuracy changes,

$$\Delta\text{Acc} = \text{Acc} - \text{Acc}_{\text{base}}, \qquad \Delta\text{Acc}_{\text{rel}} = \frac{\text{Acc} - \text{Acc}_{\text{base}}}{\text{Acc}_{\text{base}}}.$$

Dataset-averaged means and maxima of these quantities are reported.

E. Evaluation Questions

The experiments are structured to answer three questions: (i) whether replacing addition by binding improves performance, (ii) whether similarity-aware positional vectors (FPE) yield additional gains beyond sinusoidal or random encodings, and (iii) how the best binding-based configurations compare to established Transformer baselines and advanced positional mechanisms.

V. Results

A. Effect of Binding Operations

Table I reports results obtained with original *sinusoidal positional encodings* and compares additive integration to HDC binding via Circular Convolution. It shows that replacing additive composition by HDC Circular Convolution binding yields consistent improvements for both input embedding layers (linear and 1D convolution) and achieves better mean ranks than the corresponding additive baselines. This indicates that the positional integration operation is a first-class design choice for time-series Transformers rather than an implementation detail.

TABLE I: Dataset-averaged performance (Acc in %). Mean rank is computed per dataset based on accuracy (rank 1 = best) and averaged across datasets (lower is better). ΔAcc (%) and $\Delta\text{Acc}_{\text{rel}}$ denote absolute and relative improvements over the corresponding additive baseline.

Model Configuration	Mean Acc	Mean Rank	Mean ΔAcc	Mean $\Delta\text{Acc}_{\text{rel}}$	Max ΔAcc	Max $\Delta\text{Acc}_{\text{rel}}$
Linear + Additive	56.81	3.09				
Linear + Circ. Conv.	61.93	2.09	5.12	0.09	47.82	1.158
1D Conv. + Additive	54.88	2.74				
1D Conv. + Circ. Conv.	**63.20**	**2.07**	8.32	0.151	50.28	10.431

B. Effect of Positional Encoding

Beyond the choice of binding versus addition, performance depends on the similarity structure induced by the positional vectors. To isolate the effect of the positional encoding itself, the following ablation fixes the token embedding to the linear projection and chooses only the positional encoding and the operation itself. We evaluate FPE-based positional encodings with $\beta \in \{0, 0.41, 1, 1.83, 3, 4.66, 7\}$ (as in [18]) and compare them to sinusoidal and random encodings. Under Circular Convolution binding, FPE achieves higher mean accuracy and lowest mean rank than sinusoidal encodings (Figure 3 and Table II) and yields systematic per-dataset improvements over the sinusoidal baseline (Figure 4), with larger β performing best in our setting. Interestingly, $\beta = 0$ yields strong gains on some datasets but underperforms on average, indicating that certain tasks benefit from absent temporal ordering. This highlights the importance of dataset-specific similarity kernels and motivates adaptive or learned β in future work. This is consistent with the similarity kernel: unlike sinusoidal encodings whose similarity does not decay to zero, FPE induces a kernel whose similarity decays toward zero with increasing $|\Delta|$ (Figure 2).

TABLE II: Dataset-averaged accuracy changes for Circular Convolution with different PEs, measured relative to the sinusoidal baseline.

Positional Encoding	Mean Acc	Mean Rank	Mean ΔAcc %	Mean $\Delta\text{Acc}_{\text{rel}}$	Max ΔAcc %	Max $\Delta\text{Acc}_{\text{rel}}$
Sinusoidal	61.93	5.87				
Random	59.87	7.48	−2.06	−0.033	5.18	0.1840
Null	56.07	7.85	−5.86	−0.095	8.40	0.2320
FPE $\beta = 0$	58.78	4.21	−4.21	−0.067	10.91	0.3014
FPE $\beta = 0.41$	61.69	6.37	−0.24	−0.004	7.29	0.2191
FPE $\beta = 1$	63.94	4.28	+2.01	+0.032	8.25	0.1464
FPE $\beta = 1.83$	**64.43**	4.38	+2.50	+0.040	5.88	0.1426
FPE $\beta = 3$	64.18	**4.12**	+2.25	+0.036	6.05	0.1671
FPE $\beta = 4.66$	63.59	4.40	+1.66	+0.027	6.07	0.1426
FPE $\beta = 7$	63.90	4.88	+1.97	+0.032	8.00	0.2537

C. Synthetic Dataset Results

To further isolate the effect of positional encoding, we evaluate the models on a controlled synthetic dataset (cf. Section IV-A), where the class label depends solely on the temporal position of a peak. This setup removes confounding factors and directly tests whether positional information is properly encoded. The results in Table III show that neither additive integration nor Circular Convolution with sinusoidal encodings is sufficient to solve this task. As illustrated in Figure 2, sinusoidal encodings induce a similarity that does not decay to zero, leading to ambiguity between distant positions. High accuracy is only achieved with Circular Convolution combined with similarity-aware FPE encodings, highlighting the importance of both integration and positional representation.

TABLE III: Synthetic dataset accuracy (single dataset). `Add Sin`: additive + sinusoidal; `C. C. Sin`: Circular Convolution + sinusoidal; β: FPE with parameter β (with Circular Convolution).

	Add Sin	**C.C. Sin**	$\beta=$ 0	$\beta=$ 0.41	$\beta=$ 1	$\beta=$ 1.83	$\beta=$ 3	$\beta=$ 4.66	$\beta=$ 7
Acc (%)	54.45	55.75	54.90	68.55	96.45	97.44	96.30	**97.75**	79.10

D. Comparison to Established Baselines and Advanced PE Mechanisms

Finally, Table IV contextualizes the best configurations against representative time-series Transformer baselines and advanced positional mechanisms. Binding-based models with structured positional representations achieve the strongest competitive mean accuracies, while substantially outperforming the baseline.

TABLE IV: Comparison to representative baselines and advanced positional mechanisms. The best result is in bold.

	Best FPE	**C.C. Sin**	**Conv Tran [9]**	**MLA [10]**	**RoPE [6]**	**Null**
Mean Acc (%)	**64.43**	61.93	60.20	61.29	62.34	56.08

VI. Conclusion

This work revisited positional encoding in Transformers for multivariate TSC from an HDC perspective. Instead of treating PE as a fixed additive component, we frame it as a joint design problem over the integration operation and the similarity structure of positional vectors.

Our results show that replacing addition with Circular Convolution binding consistently improves performance, with similarity-aware FPE yielding the best overall results. Although addition has $\mathcal{O}(D)$ complexity and Circular Convolution $\mathcal{O}(D \log D)$ via FFT, the latter remains practical for typical embedding dimensions.

Overall, the findings suggest that positional encoding should be designed jointly in terms of operation and positional geometry rather than treated as a fixed convention. Future work includes extending binding to other Transformer backbones, learning data-dependent β, and exploring simpler binding operations.

References

[1] J. J. P. Gomez, "Beyond Addition: Enhancing Time Series Transformers with Hyperdimensional Binding," 2025.

[2] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive," pp. 1–36, Oct. 2018.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5999–6009, 2017.

[4] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?" 2022.

[5] G. Ke, D. He, and T.-Y. Liu, "Rethinking Positional Encoding in Language Pre-training," 2020.

[6] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[7] Y. Liang, R. Cao, J. Zheng, J. Ren, and L. Gao, "Learning to Remove: Towards Isotropic Pre-trained BERT Embedding," in *Artificial Neural Networks and Machine Learning – ICANN 2021*. Springer International Publishing, 2021, pp. 448–459.

[8] Y. A. Wang and Y. N. Chen, "What do position embeddings learn? An empirical study of pre-trained language model positional encoding," *Conference on Empirical Methods in Natural Language Processing EMNLP)*, vol. 2, pp. 6840–6849, 2020.

[9] N. M. Foumani, C. W. Tan, G. I. Webb, and M. Salehi, "Improving position encoding of transformers for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 38, no. 1, pp. 22–48, 2024.

[10] D. et al., "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model," Jun. 2024.

[11] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, no. 2, pp. 139–159, 2009.

[12] T. Plate, "Distributed representations," *Cognitive Science*, pp. 1–15, Jan. 2003.

[13] D. Kleyko, D. A. Rachkovskij, E. Osipov, and A. Rahimi, "A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part I: Models and Data Transformations," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–40, Jul. 2023.

[14] K. Schlegel, P. Neubert, and P. Protzel, "A comparison of vector symbolic architectures," *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4523–4555, Dec. 2021.

[15] E. P. Frady, D. Kleyko, and F. T. Sommer, "A Theory of Sequence Indexing and Working Memory in Recurrent Neural Networks," *Neural Computation*, vol. 30, no. 6, pp. 1449–1513, 2018.

[16] T. A. Plate, "Holographic Reduced Representations," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 623–641, 1995.

[17] K. Schlegel, P. Neubert, and P. Protzel, "HDC-MiniROCKET: Explicit Time Encoding in Time Series Classification with Hyperdimensional Computing," in *2022 International Joint Conference on Neural Networks (IJCNN)*. Padua, Italy: IEEE, Jul. 2022, pp. 1–8.

[18] K. Schlegel, D. A. Rachkovskij, D. Kleyko, R. W. Gayler, P. Protzel, and P. Neubert, "Structured temporal representation in time series classification with ROCKETs and hyperdimensional computing," *Data Mining and Knowledge Discovery*, vol. 39, no. 6, p. 90, Oct. 2025.

[19] M. Unger, K. Schlegel, P. Protzel, and P. Neubert, "On the choice of Vector Symbolic Architectures for Time Series Classification with HDC-MiniROCKET," in *2025 International Joint Conference on Neural Networks (IJCNN)*, Rome, Italy, 2025.

[20] T. A. Plate, "Distributed Representations and Nested Compositional Structure," Ph.D. dissertation, University of Toronto, 1994.

[21] B. Komer, T. C. Stewart, A. R. Voelker, and C. Eliasmith, "A neural representation of continuous space using fractional binding," in *41st Annual Meeting of the Cognitive Science Society*, 2019.

[22] A. R. Voelker, P. Blouw, X. Choo, N. S. Y. Dumont, T. C. Stewart, and C. Eliasmith, "Simulating and predicting dynamical systems with spatial semantic pointers," *Neural Computation*, vol. 33, no. 8, pp. 2033–2067, 2021.

[23] E. P. Frady, D. Kleyko, C. J. Kymn, B. A. Olshausen, and F. T. Sommer, "Computing on Functions Using Randomized Vector Representations (in brief)," in *Neuro-Inspired Computational Elements Conference*. ACM, 2022, pp. 115–122.