PosBERT: Disentangling Positional and Semantic Representations in Transformer Encoders

Anonymous ACL submission

Abstract

Modern Transformers rely on Relative Positional Encoding (RPE), which operates only within attention and does not persist positional information as a distinct feature in hidden representations. We show that this design choice limits the encoding of macroscopic document structure. To address this, we introduce **PosBERT**, an encoder Transformer that separates three information streams: semantics, an isolated Absolute Positional (AP) subspace, and a localized RP bias. In this setting: the AP subspace collapses into a two-dimensional, low-frequency manifold that captures document structure, with two principal components explaining 94% of its variance, compared to 23% in standard models. Attention heads further specialize into structure-oriented and semantic-oriented groups, yielding fine-grained interpretability. PosBERT matches standard entangled architectures on GLUE and MTEB, and SQuAD extractive question answering, demonstrating that disentangling position comes at no cost to linguistic performance. Beyond these results, the separation opens practical avenues: the low-dimensional AP manifold is a natural candidate for length extrapolation, while head specialization enables actionable error diagnostics and selective computation.

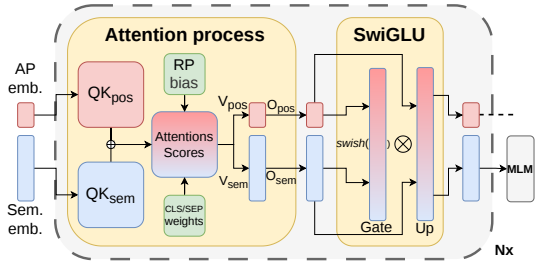


Figure 1: The PosBERT architecture. Absolute positional (AP) embeddings are displayed in red, and semantic embeddings in blue. Red-blue gradient components contain shared information. Separate RMSNorms are not displayed for readability.

1 Introduction

The shift from absolute to relative positional encoding discarded something valuable. Modern Transformers increasingly rely on Relative Positional Encoding (RPE)—via additive attention biases (Raffel et al., 2020; Press et al., 2022; Chi et al., 2022a; Li et al., 2024) or rotary transformations (RoPE) (Su et al., 2023)—which encodes position only during attention computation. Unlike earlier absolute positional (AP) embeddings (Vaswani et al., 2017; Devlin et al., 2019), RPE leaves no persistent positional trace in hidden representations. Moreover, RPE methods often rely on long-term decay heuristics that assume attention diminishes with distance, creating positional biases (Wu et al., 2025) and bottlenecking long-context tasks (Chen et al., 2025). Gu et al. (2026) further demonstrated this forced positional dependence is often unnecessary; many heads function perfectly without positional signals. While decoder-only models partially compensate via the implicit positional signal of causal masking (Haviv et al., 2022; Wang et al., 2024), this is indirect and limited. The fundamental question of how to best represent position remains open across architectures.

As we establish in this work, the lack of explicit AP information within hidden states impairs the encoding of macroscopic document structure. Combined with evidence that models learn positional and semantic signals in nearly orthogonal subspaces (Song and Zhong, 2024), this suggests explicitly separating these signals rather than entangling them. Disentanglement opens several practical avenues: (1) In RAG, semantic representations could be cached independently with position injected on-demand, or the AP subspace could inject non-linear structural signals (e.g., table layouts); (2 because the isolated AP space collapses to a low-frequency manifold, length extrapolation could be achieved by manipulating the AP subspace alone,

without disturbing learned semantic representations; (3) Separation clarifies attention head behavior, heads using neither AP nor RP perform purely semantic retrieval (Wu et al., 2024), enabling diagnostics that trace errors to either structural mislocalization or semantic confusion; (4) Head specialization allows selective computation, where structure- or semantic-oriented heads are independently activated or pruned per task. This leads us to the following research questions:

- Is explicit AP information necessary for encoding document structure, and does its absence in RPE-based models cause measurable degradation?
- When positional and semantic streams are separated, what structure emerges in the isolated AP space?
- Does disentanglement preserve, or even improve, downstream performance compared to standard entangled models?

We validate these ideas on encoder-based models. Lacking a causal mask, encoders rely solely on positional encoding for position information, providing a clean experimental setting. They also remain state-of-the-art for document-level representation and retrieval (Weller et al., 2025). While their smaller scale is practical for this work, the principles we uncover are architecture-agnostic. We introduce PosBERT[1], a Transformer encoder that processes semantic information, AP (as an isolated hidden subspace), and RP (as a learned additive bias) in separate streams. Our contributions are:

- We show that RPE- and RoPE-based encoders fail to encode macroscopic document structure, proving that persistent AP information in hidden states is necessary for this capacity (Section 3).
- We introduce the PosBERT architecture (Section 4), which isolates AP, RP, and semantic streams to prevent the positional decay observed in deeper layers of standard models.
- We reveal that the isolated AP space spontaneously collapses into a two-dimensional, low-frequency structural manifold (94% variance in 2 PCs), and that attention heads specialize into structure-oriented or semantic-oriented groups (Section 5).
- We demonstrate that this disentanglement matches standard architectures on GLUE and MTEB, while improving extractive QA on SQuAD (Section 6).

[1]models and experiments will be shared in camera ready

2 Related Works

Positional Encoding Positional encoding has evolved from absolute positional embeddings (APE) — fixed sinusoidal (Vaswani et al., 2017) or learned (Devlin et al., 2019; Liu et al., 2020) — added directly to token representations, to relative positional encoding (RPE) injected into attention logits. Additive RPE methods include T5's bucketed bias (Raffel et al., 2020), ALiBi's fixed decay (Press et al., 2022), and refinements such as KERPLE (Chi et al., 2022b), FiRE (Li et al., 2024) and Sandwich (Chi et al., 2023). Rotary Positional Encoding (RoPE) (Su et al., 2023) applies multiplicative rotations to keys and queries, with extensions like YaRN (Peng et al., 2023) improving long-context extrapolation. All these RPE methods share a key property: positional information is consumed during attention and does not persist in hidden states.

Structure-Aware and Disentangled Positional Encoding Several approaches address the interaction between positional and semantic information. HoPE (Chen et al., 2025) removes the low-frequency components of RoPE to mitigate negative biases in long contexts. BiPE (He et al., 2024) uses dual encodings for inter-segment and intra-segment positions. CoPE (Golovneva et al., 2024) learns a gating mechanism to assign context-dependent positional encodings. DaPE (Zheng et al., 2024, 2025) conditions its positional bias on semantic information through a linear transformation. However, all these methods still entangle positional and semantic information within the same hidden representations. Our work is most closely related to **TUPE** (Ke et al., 2021), which separates AP and RP into distinct attention components. However, TUPE's AP component is a single shared learned projection applied identically across all layers, functionally equivalent to a static AP bias. In contrast, PosBERT maintains a dedicated AP hidden subspace that evolves through every layer, allowing the model to progressively refine structural representations and interact with the semantic space through controlled mixing in the feed-forward network.

Positional Space in Transformers Several studies examine latent space geometry. Wang and Chen (2020) show that learned APEs in models

like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are high-dimensional because they conflate positional and structural information. Song and Zhong (2024) reveal that positional and semantic bases are nearly orthogonal, where nearby-token attention is driven by position and matching behaviors by content. Positional processing is not uniformly distributed; Gu et al. (2026) show it often concentrates in a few shallow specialized heads, while Urrutia et al. (2025) observe a layerwise shift toward more symbolic heads in deeper layers. Relatedly, Wu et al. (2024) identify *retrieval heads* that bypass positional information entirely to perform purely semantic matching. PosBERT's disentangled design makes this specialization explicit.

3 Probing Positional Encodings for Structural Information

Understanding document structure requires a hierarchical coordinate system capturing a token's global absolute position, its structural segment, and its local intra-segment position. In this section, we probe whether different positional encodings support this three-level hierarchy.To isolate the impact of positional encoding, we train three variants of NeoBERT (Breton et al., 2024) under identical configurations (Section 4.3): RoPE-NeoBERT (vanilla architecture with rotary embeddings), AP-NeoBERT (learned absolute embeddings), and RP-NeoBERT (learned relative bias). Let model $\mathcal{M}$ map a document $\mathcal{D}$ to hidden states $H^{(l)} = \{h_1^{(l)}, \ldots, h_N^{(l)}\}$ at layer l, where $h_i^{(l)} \in \mathbb{R}^d$. We evaluate these variants using three linear probes, one per hierarchy level, defined as $f(h_i^{(l)}) = W^\top h_i^{(l)} + b$. Probes are optimized via Ridge regression and evaluated using R^2. We report averages over three runs on 50 texts (80-20 train-test split), ensuring all results have a standard deviation ≤ 0.002.

3.1 Token-level AP

The first level of the hierarchy is the most basic: can a model recover the global position of a token? While RoPE's rotations depend on absolute position, they are applied to queries and keys during attention and not persisted in the residual stream, making it unclear whether this information is recoverable from hidden states. The absolute position $y_i \in [0, 1]$ of token t_i is defined as its normalized index:

$$y_i = \frac{pos(t_i)}{\max(pos(\mathcal{D}))}$$

Results The results are reported in Table 1. As expected, AP-NeoBERT properly encodes this information, especially in early layers. Critically, AP information collapses in the final layer ($R^2 = 0.34$): because the MLM head operates on the full hidden state, the model repurposes positional dimensions for semantic prediction, overwriting structural information. This final-layer degradation directly motivates PosBERT's design, where the MLM objective is restricted to the semantic subspace, leaving the AP subspace free from this pressure. RP-NeoBERT can partially infer AP information in its embeddings. RoPE-NeoBERT, despite its rotations being absolute-position-dependent, is largely unable to encode AP information in its hidden states, confirming that the lack of persistence in the residual stream is the bottleneck.

Layer	**0**	**1**	**2**	**3**	**4**	**5**
RoPE	0.13	0.14	0.34	0.25	0.27	0.20
RP	0.17	0.50	0.46	0.42	0.42	0.22
AP	**0.99**	**0.99**	**0.97**	**0.90**	**0.84**	**0.34**

Table 1: Token-level AP probe results

3.2 Segment-level AP

The second level tests coarse structural membership: does a model know which segment (e.g., sentence or paragraph) a token belongs to? Let a document $\mathcal{D}$ be partitioned into a sequence of K segments $\mathcal{S} = \{S_0, S_1, \ldots, S_{K-1}\}$. We define a mapping function $\sigma(t_i) \to \{0, \ldots, K-1\}$ that assigns each token t_i to its segment index. The segment index is updated at every structural boundary (defined by terminal punctuation or newline characters). Unlike the absolute position probe, the target y_i for this task is the discrete segment ID:

$$y_i = \sigma(t_i)$$

This target measures the model's awareness of its current progress through the document's macroscopic structure, independent of the token's local position within a specific segment.

Results As displayed in Table 2, the results follow the same pattern as the token-level probe. AP-NeoBERT strongly encodes segment-level AP in early layers but loses the information in deeper

layers, while RP-NeoBERT and RoPE-NeoBERT struggle to encode this information.

Layer	0	1	2	3	4	5
RoPE	0.12	0.13	0.32	0.24	0.24	0.20
RP	0.17	0.47	0.40	0.37	0.37	0.23
AP	**0.88**	**0.87**	**0.85**	**0.79**	**0.74**	**0.32**

Table 2: Segment-level AP probe results

3.3 Intra-segment token AP

The third level tests local progress: does a model encode where a token sits within its own segment? Let a document $\mathcal{D}$ be partitioned into a sequence of segments $\mathcal{S} = \{S_0, S_1, \ldots, S_{K-1}\}$. For any token t_i belonging to a segment S_k of length L_k, we define the target variable $y_i^{rel} \in [0, 1]$ as the normalized progress within that segment:

$$y_i^{rel} = \frac{pos(t_i) - \min\{j \mid t_j \in S_k\}}{L_k - 1}$$

where $pos(t_i)$ is the absolute index of the token. This target transforms a discrete sawtooth count into a continuous linear slope, where 0.0 denotes the segment start and 1.0 the terminal boundary marker.

Results All positional encodings perform about equally on this task, as displayed in Table 3. We show later in this work (Section 5) that this task is actually handled almost entirely by the semantic space, which explains the equivalent performance across all positional encodings.

Layer	0	1	2	3	4	5
RoPE	0.29	0.68	0.72	0.67	**0.68**	0.49
RP	**0.50**	0.68	0.67	0.68	0.63	**0.50**
AP	0.20	**0.73**	**0.78**	**0.74**	0.64	0.48

Table 3: Intra-segment token AP probe results

These probing experiments establish two findings: (1) persistent AP information in hidden states is necessary to encode document structure — RP- and RoPE-based models fail to represent where tokens are and which segment they belong to; (2) even in AP-based models, positional information degrades in the final layers as the MLM objective repurposes these dimensions for prediction. This motivates an architecture that isolates positional information from the prediction objective entirely.

4 PosBERT: Architecture and Training

Having established that persistent AP information is necessary for structural encoding (Section 3), we now introduce PosBERT, a Transformer encoder with architecturally separated semantic, absolute-positional, and relative-positional information streams. PosBERT is based on NeoBERT (Breton et al., 2024); we describe below the modifications to each component. The full architecture is displayed in Figure 1.

4.1 Attention blocks

The attention blocks process the three information streams to produce jointly computed attention scores, which are then applied to both the AP and semantic value streams.

Input token representations Each token is represented by two embeddings: a d_{AP}-dimensional AP embedding and a d_{sem}-dimensional semantic embedding. We define the hidden size of the model as $d_{model} = d_{AP} + d_{sem}$.

RMSNorm The pre-attention and post-attention layer norms, using the RMS norm (Zhang and Sennrich, 2019), are separated between the AP and the semantic embeddings to avoid creating an interdependence between the two.

Relative positional bias PosBERT uses an additive relative-positional bias learned in each head. However, state-of-the-art biases (Press et al., 2022; Chi et al., 2022b; Li et al., 2024) rely on kernels using differences of position $i-j$, therefore forcing a long-term decay. Instead, we use the bucketed bias introduced in T5 (Raffel et al., 2020) (detailed in Appendix A.1), which offers higher precision for nearby tokens but becomes increasingly coarse and generalized as the distance between tokens increases. Each bucket has its own learned parameter $\rho^h_{\text{bucket}(i-j)}$, independent of its distance to the attending token. Additionally, following TUPE (Ke et al., 2021), we argue that the position of special tokens $\mathscr{S} = \{[\text{CLS}], [\text{SEP}]\}$ at the beginning and the end of a sequence is purely arbitrary and should not be used to compute their relative attention weights. Therefore, their RP bias corresponds to head-specific distance-agnostic learned parameters: $\rho^h_{i \leftarrow j}$ when both tokens are special, $\rho^h_{\leftarrow j}$ when attending *from* a special token j to a regular token, and $\rho^h_{i \leftarrow}$ when a regular token attends *to* a special token i. The RP bias in head h is:

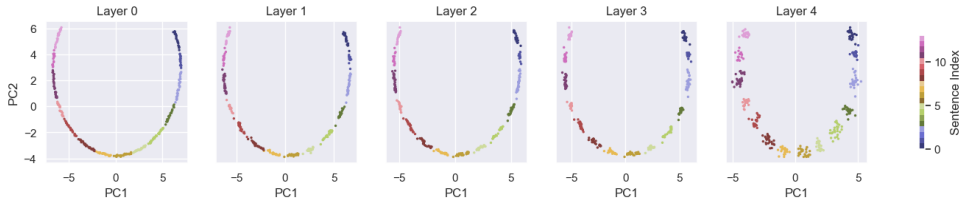


Figure 2: 2-dimensional PCA of the AP hidden states at each layer when encoding a long document. Each sentence (delimited by punctuation) is given a different color. Tokens from the same sentence cluster together, with clusters becoming increasingly distinct in deeper layers. The last layer is omitted because the AP subspace is not used for MLM prediction, leaving it unconstrained.

$$b^h_{i\leftarrow j} = \begin{cases} \rho^h_{i\leftarrow j} & \text{if } i \in \mathscr{S} \wedge j \in \mathscr{S} \\ \rho^h_{\leftarrow j} & \text{if } j \in \mathscr{S} \\ \rho^h_{i\leftarrow} & \text{if } i \in \mathscr{S} \\ \rho^h_{\text{bucket}(i-j)} & \text{otherwise} \end{cases} \tag{1}$$

Attention heads We first consider a single attention head h. Queries and keys are computed independently for each stream: $Q_{sem} = W_q^{sem} x_{sem}$, $K_{sem} = W_k^{sem} x_{sem}$, $Q_{AP} = W_q^{AP} x_{AP}$, $K_{AP} = W_k^{AP} x_{AP}$, where $W_q^{AP}, W_k^{AP} \in \mathbb{R}^{d_{model} \times d_{AP}}$ and $W_q^{sem}, W_k^{sem} \in \mathbb{R}^{d_{model} \times d_{sem}}$. A key design choice: we project both streams to the same size d_{model} despite the different input sizes (d_{AP} and d_{sem}), ensuring that the dot-product variances are matched and preventing the higher-dimensional space from dominating the attention computation. The attention weight matrices are $w^{h,sem}_{i\leftarrow j} = \frac{(Q_{sem})_i \cdot (K_{sem})_j}{\sqrt{d_{model}}}$ and $w^{h,AP}_{i\leftarrow j} = \frac{(Q_{AP})_i \cdot (K_{AP})_j}{\sqrt{d_{model}}}$. The attention logit $l^h_{i\leftarrow j}$ is computed by summing the RP bias, the semantic attention weight, and the AP attention weight (the latter only when $i, j \notin \mathscr{S}$):

$$l^h_{i\leftarrow j} = b^h_{i\leftarrow j} + w^{h,sem}_{i\leftarrow j} + w^{h,AP}_{i\leftarrow j} \mathbb{1}[i \notin \mathscr{S} \wedge j \notin \mathscr{S}] \tag{2}$$

Attention scores are obtained by applying the softmax, normalizing over all keys j: $A^h_{i\leftarrow j} = \frac{\exp(l^h_{i\leftarrow j})}{\sum_{j'} \exp(l^h_{i\leftarrow j'})}$. This process is illustrated in Figure 5.

The values are computed separately for each stream: $V_{AP} = W_v^{AP} x_{AP}$ and $V_{sem} = W_v^{sem} x_{sem}$, with $W_v^{AP} \in \mathbb{R}^{d_{AP} \times d_{AP}}$ and $W_v^{sem} \in \mathbb{R}^{d_{sem} \times d_{sem}}$. Unlike Q/K, the value projections remain within their respective spaces. The same attention scores A^h are applied to both value streams, and the results are processed independently by output projections $W_o^{AP} \in \mathbb{R}^{d_{AP} \times d_{AP}}$ and $W_o^{sem} \in \mathbb{R}^{d_{sem} \times d_{sem}}$.

For multi-head attention with n_{heads} attention heads of size $d_{head} = \frac{d_{model}}{n_{heads}}$, the process is similar, but head transformations are done in $\frac{d_{AP}}{n_{heads}}$- and $\frac{d_{sem}}{n_{heads}}$-dimensional spaces, which require d_{AP} and d_{sem} to be multiples of n_{heads}.

4.2 SwiGLU

NeoBERT, and most recent Transformer architectures, use SwiGLU (Shazeer, 2020) as a drop-in replacement for the post-attention feed-forward network. SwiGLU is defined as:

$$W_{down}\left(\text{swish}(W_{gate}x) \otimes W_{up}x\right) \tag{3}$$

with x the token embeddings, $W_{up}, W_{gate} \in \mathbb{R}^{d_{int} \times d_{model}}$ and $W_{down} \in \mathbb{R}^{d_{model} \times d_{int}}$ with d_{int} a hidden intermediate size usually defined as $d_{int} = 4 \cdot d_{model}$, $\otimes$ the Hadamard product, and $\text{swish}(x) = x \cdot sigmoid(x)$.

We implement this by allowing the up-projection W_{up} and the gating W_{gate} to operate on the concatenated vector $x = [x_{AP}; x_{sem}] \in \mathbb{R}^{d_{model}}$, projecting to a $d_{int} = d^{AP}_{int} + d^{sem}_{int}$ dimensional space. To maintain the architectural separation, we re-separate the information during the down-projection. Denoting by $x^{(l)}_{AP}$ and $x^{(l)}_{sem}$ the AP and semantic representations after the attention block at layer l:

$$\begin{aligned} h_{inter} &= \text{swish}(W_{gate}x) \otimes (W_{up}x) \\ x^{(l+1)}_{AP} &= x^{(l)}_{AP} + W^{AP}_{down}(h_{inter}[: d^{AP}_{int}]) \\ x^{(l+1)}_{sem} &= x^{(l)}_{sem} + W^{sem}_{down}(h_{inter}[d^{AP}_{int} :]) \end{aligned}$$

where $W^{AP}_{down} \in \mathbb{R}^{d_{AP} \times d^{AP}_{int}}$ and $W^{sem}_{down} \in \mathbb{R}^{d_{sem} \times d^{sem}_{int}}$, with $d^{AP}_{int} = 4 \cdot d_{AP}$ and $d^{sem}_{int} = 4 \cdot d_{sem}$.

The SwiGLU is the only component where semantic and positional information interact, serving as a controlled channel for semantic-conditioned structural updates (e.g., a "\n" token triggering a

segment boundary in the AP space). The dual down-projections re-segregate the information before the residual connection, preserving the two-stream architecture. This design corroborates previous work (Golovneva et al., 2024; Zheng et al., 2024) showing the benefit of conditioning positional representations on semantic content.

4.3 Training

Setup Following NeoBERT, we train PosBERT on the English corpus of the *FineWeb* dataset (Penedo et al., 2024) with a Masked Language Modeling (MLM) objective. We use an $L = 6$ layer architecture with $A = 6$ heads per layer and a hidden size of $d_{model} = 768$. For PosBERT, we split the embedding into $d_{AP} = 48$ and $d_{sem} = 720$, keeping $1/16$ of the dimensions for positional information. This ratio is motivated by the observation that positional information is inherently low-dimensional (Song and Zhong, 2024). The maximum positional embedding is set to 512. We train for 70 k steps on four H100 GPUs with an effective batch size of 256, totaling around 22B tokens, using an AdamW optimizer (Loshchilov and Hutter, 2019) with a 10k-step learning rate warm-up followed by cosine decay.

MLM decoding Since MLM is an intrinsically semantic task, the decoding head is applied only to the semantic part of the last hidden state ($\mathbb{R}^{d_{sem} \times d_{vocab}}$), rather than to the full d_{model}-dimensional representation. This is a critical design choice: it frees the AP subspace from prediction pressure, preventing the final-layer positional collapse observed in standard models (Section 3).

AP shifting We use random position shifting (Kiyono et al., 2021) to uniformly train all positions and to decorrelate tokens' semantic content from their position. Given a list of n tokens as input $[T_0, T_1, ..., T_n]$, instead of assigning them positions $[P_0, P_1, ..., P_n]$, we randomly sample $k \in [0, m - n]$ where m is the maximum position embedding and assign positions $[P_k, P_{k+1}, ..., P_{k+n}]$. We now analyze what PosBERT learns in these separated spaces.

5 Analysis of PosBERT's Learned Representations

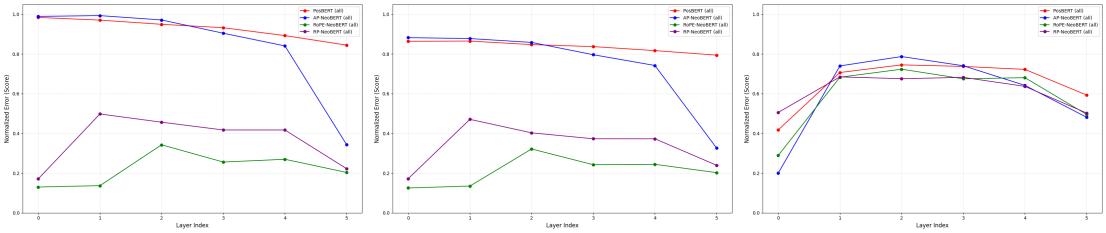
PosBERT encodes structural information in a low-dimensional AP subspace distinct from its semantic representations, and its attention heads spontaneously specialize into structure-driven and semantic-driven groups. We demonstrate these properties through visualization, probing, and spectral analysis.

(1) PosBERT trades off absolute position for structure As exhibited in its attention patterns (Figure 4, second row), PosBERT learns to use the AP embedding space as structural encoding, separating sentences and paragraphs into blocks. A PCA of the AP hidden states (Figure 2) reveals a growing separation between sentences across layers: tokens from the same sentence cluster together, and these clusters become increasingly distinct in deeper layers. This confirms that the AP space progressively abstracts raw position into document structure.

(2) PosBERT keeps structural information throughout all its layers When evaluated on the probes from Section 3, PosBERT not only scores on par with AP-NeoBERT but demonstrates a mostly constant retention of structural information throughout all layers, completely bypassing the final-layer collapse observed in the baseline model (Figure 3). This is a direct consequence of the decoupled architecture: since the MLM objective is applied exclusively to the semantic space, the AP subspace is free from task-specific interference and can maintain its structural encoding through the final layer.

(3) PosBERT separates AP and semantic information into functionally distinct spaces By evaluating our structural probes independently on the separated hidden states, we demonstrate that PosBERT actively prevents positional information from bleeding into the semantic space. As shown in Table 4, global document hierarchy (token AP (Tok.) and segment AP (Seg.)) is strongly and almost exclusively encoded within the AP space. Local intra-segment positioning (Intra-seg.), however, is handled primarily by the semantic space, explaining the performance parity between AP, RP, and RoPE on this specific task. Furthermore, we observe no layer-wise improvement when probing the semantic space for global AP features, verifying that structural information remains safely isolated within its dedicated manifold throughout the model's forward pass.

(4) The learned AP embeddings collapse into a 2D low-frequency structural manifold PosBERT's isolated AP space abstracts global docu-



(a) Token-level AP probing (b) Segment-level AP probing (c) Intra-segment token AP probing

Figure 3: Results of structural probing for PosBERT (Section 4) and the three NeoBERT variants (Section 3). From left to right: token-level AP (Section 3.1), segment-level AP (Section 3.2), and intra-segment token AP (Section 3.3).

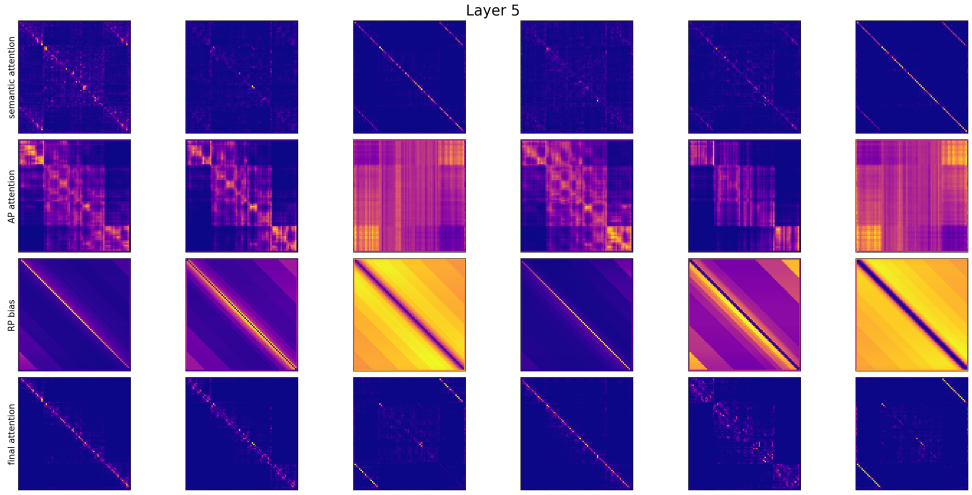


Figure 4: Softmax applied independently to the last layer's attention weights for semantic (1st row), AP (2nd row) and RP (3rd row) components, as well as the combined attention scores (Eq. 2, 4th row) on an example of shape DOC-A + DOC-B + DOC-A. Each column corresponds to an attention head. Color scale differs across plots.

	Tok.	Seg.	Intra-seg.
AP	**0.91**	**0.80**	0.06
Semantic	0.36	0.35	**0.64**
Both	0.93	0.83	0.65

Table 4: Average probing results for Token-level (Tok.), Segment-level (Seg.) and intra-segment (Intra-seg.) absolute position on the AP embedding, the semantic embedding, and both embeddings concatenated.

ment structure by discarding the high-frequency noise typical of standard positional embeddings. We demonstrate this via two decompositions. First, PCA, following Wang and Chen (2020), shows PosBERT's learned AP embeddings collapse into a two-dimensional manifold, with the first two principal components (PCs) capturing 94.1% of the total spatial variance. In contrast, the AP-NeoBERT baseline remains entangled, capturing only 23.4% in its first two PCs. Second, a Type-II Discrete Cosine Transform (DCT) confirms PosBERT's dominant PCs operate at low frequencies, concentrating 98.4% and 99.0% of their spectral power within the first four bins. The residual third component (2.3% variance) acts as high-frequency noise. Conversely, AP-NeoBERT's top PCs are dominated by high-frequency signals, holding under 0.2% of their power in these low-frequency bins. Appendix B.2 further details PosBERT's representational simplicity compared to standard encoder architectures.

(5) Attention heads cluster into AP-oriented and semantic-oriented groups PosBERT's disentangled architecture reveals a strict functional taxonomy: attention heads specialize almost exclusively as either AP-driven structural routers or semantic-driven information seekers. Notably, we find no purely RP-oriented heads. Instead, the RP bias acts as a localized support mechanism for the semantic cluster: as shown in Figure 4, heads 3 and 6 act as *retrieval* heads. The semantic attention (first row) attends to semantically similar tokens, while the RP bias (third row) disallows tokens from attending to themselves, enabling the model to focus entirely on distant context.

We empirically establish this taxonomy by quantifying the weight of each representational space through information ablation. We compute KL-divergence between the true attention probability distribution (A^h) and the ablated distribution ($A^h_{\setminus i}$) generated when a component (AP, RP or semantic) is removed prior to the softmax, for each information $i \in \{sem, AP, RP\}$:

$$Score_i = D_{KL}(A^h \| A^h_{\setminus i}) \tag{4}$$

Averaging these divergence scores across 50 document examples yields a normalized 3-dimensional influence vector for each head. As detailed in Appendix B.1, clustering these influence vectors mathematically exposes the two functional groups described above.

6 Experiments

Having analyzed PosBERT's internal representations, we now verify that disentanglement does not sacrifice downstream performance. We evaluate PosBERT against RoPE-NeoBERT, AP-NeoBERT, and RP-NeoBERT on three benchmarks: GLUE (Wang et al., 2018), MTEB (Muennighoff et al., 2023), and SQuAD (Rajpurkar et al., 2016).

GLUE and MTEB. GLUE tests natural language inference and semantic similarity across 8 datasets; MTEB evaluates zero-shot dense embeddings across 56 English datasets spanning 7 task types. Following a standardized hyperparameter grid-search (Appendix C.1), PosBERT achieves average scores of 0.78 on GLUE (Table 6) and 0.46 on MTEB (Appendix C.2), matching the entangled RoPE-NeoBERT (0.79, 0.46) and AP-NeoBERT (0.79, 0.47) baselines, and outperforming RP-NeoBERT (0.75, 0.43). Full per-task results are in Appendix C.1. This confirms that forcing positional and semantic information into a shared representation is architecturally unnecessary for standard linguistic reasoning and embedding quality.

SQuAD. Extractive question answering requires high-resolution token-level representations to accurately predict span boundaries, making it a particularly relevant test for PosBERT's structural encoding. As shown in Table 5, PosBERT achieves an Exact Match (EM) of 0.77 and an F1 score of 0.87, slightly above other baselines. probing (Section 5) demonstrates that PosBERT's AP subspace preserves structural information through the final layer. In contrast, standard architectures lose this information when the MLM objective repurposes positional dimensions for prediction. Preserving this structural encoding supports span boundary detection.

Model	EM	F1
PosBERT	0.77	0.87
RoPE-NeoBERT	0.76	0.86
AP-NeoBERT	0.74	0.82
RP-NeoBERT	0.75	0.84

Table 5: Results on the SQuAD benchmark

7 Conclusion

In this work, we demonstrated that the shift to purely relative positional encoding discards valuable information: RPE-based encoders fail to persist spatial coordinates in their hidden states, impairing macroscopic document understanding. To address this, we introduced PosBERT, an encoder architecture that architecturally disentangles processing into separate semantic, Absolute Positional (AP), and localized RP streams.

Our analyses revealed that this separation has striking consequences. The isolated AP space spontaneously collapses into a two-dimensional, low-frequency structural manifold (94% variance in 2 PCs, compared to 23% in standard models), completely bypassing the final-layer positional decay observed in entangled architectures. Attention heads specialize into structure-driven and semantic-driven groups, yielding fine-grained interpretability. Crucially, this disentanglement comes at no cost: PosBERT matches standard architectures on GLUE and MTEB, and outperforms them on SQuAD extractive question answering.

PosBERT demonstrates that respecting the natural orthogonality of positional and semantic signals yields both high performance and transparency. The remarkably low dimensionality of the learned AP space suggests that significant optimizations are possible on the positional stream — from compression and quantization to structured parameterizations that exploit its sinusoidal regularity. More broadly, the separation opens practical avenues: caching position-independent semantic representations for RAG, leveraging head specialization for selective computation, and manipulating the low-dimensional AP manifold for length extrapolation. Scaling to larger models and extending the approach to decoder-only architectures are promising directions for future work.

Limitations

While PosBERT successfully demonstrates the theoretical and empirical benefits of disentangling positional and semantic information, this exploratory work has several notable limitations. Our empirical validation is currently constrained to a relatively small scale. The PosBERT model and its baselines were trained from scratch using a 6-layer, 6-head architecture trained on approximately 22 billion tokens. While this scale is sufficient to show the emergence of the low-frequency structural manifold and the functional taxonomy of attention heads, we have not yet verified whether these disentanglement properties hold, or whether the training dynamics shift, when scaled to massive parameter counts (e.g., 7B+ parameters) or larger token budgets. Additionally, our probing evaluations are conducted on a modest set of 50 documents, which, while consistent across models, limits the statistical power of the analysis. Furthermore, our architectural modifications and probing experiments are specifically designed for and evaluated on encoder-only Transformers. We intentionally focused on bidirectional models because they cannot implicitly infer position through a causal attention mask, ensuring a cleaner separation of variables for our analysis. Consequently, the applicability of this strict three-stream separation to decoder-only architectures remains an open question for future work.

References

Lola Le Breton, Quentin Fournier, John X Morris, Mariam El Mezouar, and Sarath Chandar. 2024. NeoBERT: A Next-Generation BERT.

Yuhan Chen, Ang Lv, Jian Luan, Bin Wang, and Wei Liu. 2025. HoPE: A novel positional encoding without long-term decay for enhanced context awareness and extrapolation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23044–23056, Vienna, Austria. Association for Computational Linguistics.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. 2022a. KERPLE: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems*, volume 35, pages 8386–8399. Curran Associates, Inc.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. 2022b. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399.

Ta-Chung Chi, Ting-Han Fan, Alexander Rudnicky, and Peter Ramadge. 2023. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13522–13537, Toronto, Canada. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *Preprint*, arXiv:2003.10555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual Position Encoding: Learning to Count What's Important. *Preprint*, arXiv:2405.18719.

Zihan Gu, Ruoyu Chen, Han Zhang, Hua Zhang, and Yue Hu. 2026. Deconstructing positional information: From attention logits to training biases. In *The Fourteenth International Conference on Learning Representations*.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1382–1390, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhenyu He, Guhao Feng, Shengjie Luo, Kai Yang, Liwei Wang, Jingjing Xu, Zhi Zhang, Hongxia Yang, and Di He. 2024. Two Stones Hit One Bird: Bilevel Positional Encoding for Better Length Extrapolation. *Preprint*, arXiv:2401.16421.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking Positional Encoding in Language Pre-training. *Preprint*, arXiv:2006.15595.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3309–3321, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised

Learning of Language Representations. *Preprint*, arXiv:1909.11942.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. 2024. Functional Interpolation for Relative Positions Improves Long Context Transformers. *Preprint*, arXiv:2310.04418.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. Learning to Encode Position for Transformer with Continuous Dynamical Model. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6327–6335. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. *Preprint*, arXiv:1711.05101.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. YaRN: Efficient Context Window Extension of Large Language Models. *Preprint*, arXiv:2309.00071.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. *Preprint*, arXiv:2108.12409.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Noam Shazeer. 2020. GLU Variants Improve Transformer. *Preprint*, arXiv:2002.05202.

Jiajun Song and Yiqiao Zhong. 2024. Uncovering hidden geometry in Transformers via disentangling position and context. *Preprint*, arXiv:2310.04861.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Preprint*, arXiv:2104.09864.

Felipe Urrutia, Jorge Salas, Alexander Kozachinskiy, Cristian Buc Calderon, Hector Pasten, and Cristobal Rojas. 2025. Decoupling positional and symbolic attention behavior in transformers. *Preprint*, arXiv:2511.11579.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jie Wang, Tao Ji, Yuanbin Wu, Hang Yan, Tao Gui, Qi Zhang, Xuanjing Huang, and Xiaoling Wang. 2024. Length generalization of causal transformers without position encoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14024–14040, Bangkok, Thailand. Association for Computational Linguistics.

Yu-An Wang and Yun-Nung Chen. 2020. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849, Online. Association for Computational Linguistics.

Orion Weller, Kathryn Ricci, Marc Marone, Antoine Chaffin, Dawn Lawrie, and Benjamin Van Durme. 2025. Seq vs Seq: An Open Suite of Paired Encoders and Decoders. *Preprint*, arXiv:2507.11412.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval Head Mechanistically Explains Long-Context Factuality. *Preprint*, arXiv:2404.15574.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. 2025. On the Emergence of Position Bias in Transformers. *Preprint*, arXiv:2502.01951.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and 1 others. 2024. Dape: Data-adaptive positional encoding for length extrapolation. *Advances in Neural Information Processing Systems*, 37:26659–26700.

Chuanyang Zheng, Yihang Gao, Han Shi, Jing Xiong, Jiankai Sun, Jingyao Li, Minbin Huang, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. 2025. DAPE v2: Process attention score as feature map for length extrapolation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10628–10666, Vienna, Austria. Association for Computational Linguistics.

A PosBERT Architecture Details

A.1 T5 Positional Bias

This section provides details on the T5 bucketed bias used in PosBERT's relative positional component (Section 4). T5 (Raffel et al., 2020) introduces a learned relative position bias inside self-attention. For a query token at position i and a key token at position j, the model considers their relative distance and associates it with a learned scalar bias, which is learned per attention head. In fact, T5 does not learn one parameter for every possible distance; distances are first mapped to a finite set of buckets. Nearby distances are represented more finely, while larger distances are grouped together more coarsely. T5 uses 32 relative-position embeddings (i.e. 32 buckets), with bucket ranges that grow logarithmically up to a relative offset of 128; beyond that, all larger distances are mapped to the same bucket. This means that small offsets such as one or two positions apart can receive distinct learned biases, whereas larger offsets such as 40, 50, or 60 tokens apart may share the same bucket, and any offset larger than 128 is treated identically from the viewpoint of a single layer.

These learned scalar biases are added directly to the attention logits before the softmax. Using the usual attention notation, this can be written as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B\right)V,$$

where $B \in \mathbb{R}^{n\times n}$ is the matrix of relative position biases, n is the sequence length, and each entry B_{ij} depends on the bucketed relative distance between query position i and key position j for the corresponding attention head.

A.2 Attention Process

Figure 5 displays a graphical example of the attention logits computation described in Section 4.

B Complementary Analysis of PosBERT's Learned Representations

B.1 Decomposition of Information in Attention Heads

As described in Section 5, attention heads cluster into two distinct groups: AP-driven heads and semantic-driven heads. The three-score vector $[\mathrm{Score}_{sem}, \mathrm{Score}_{AP}, \mathrm{Score}_{RP}]$ gives us a coordinate system to compare the attention heads. We show the two clusters in Figure 6. Only 9 out of the 36 attention heads are semantic-driven. The rest focuses on absolute position and structure.

B.2 Comparison of Learned AP Embeddings

This section extends the PCA and DCT analysis of Section 5 by comparing PosBERT's learned AP embeddings to those of AP-NeoBERT and other encoder-based Transformers with learned AP embeddings: BERT (Devlin et al., 2019), ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2020). Figure 7 shows the cumulative variance of Principal Components (PCs) for each model. PosBERT is significantly lower-dimensional than other systems. The principal components of each model are displayed in Figure 8. PosBERT displays strongly sinusoidal patterns with a much higher explained variance.

C Experiment Details

C.1 GLUE

Metrics Following Wang et al. (2018), we use different metrics for the different datasets. For STSB, we provide Pearson correlation. For Cola, Matthews correlation. For QQP, the F1 score. The accuracy is given for all other tasks.

Hyperparameters and training We run the hyperparameter search with learning rates in $\{6e\text{-}6, 5e\text{-}6, 2e\text{-}5\}$, batch sizes in $\{4, 16, 32\}$, and weight decays in $\{1e\text{-}2, 1e\text{-}5\}$. Following NeoBERT, we finetune on the training split

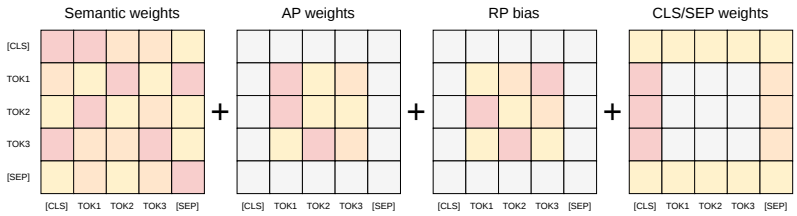


Figure 5: PosBERT attention weights mechanism. Grayed-out squares correspond to discarded weights. From left to right: (1) The semantic space can match all tokens. (2) The AP space can match all but [CLS] and [SEP] tokens. (3) The RP bias learns a per-bucket weight (all buckets are of size 1 in this example), [CLS] and [SEP] are discarded. Notice that the weights are not symmetrical nor necessarily decaying with distance. (4) The learned ρ weights for [CLS] and [SEP] applied to the sum.

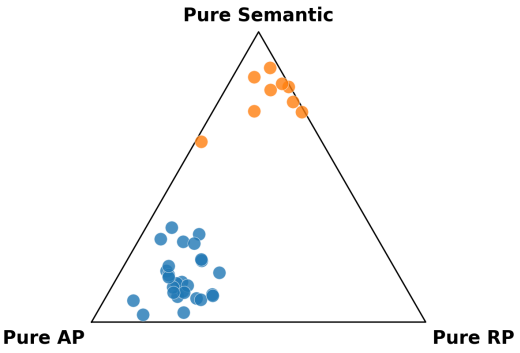


Figure 6: Categorization of attention heads in PosBERT. Each circle represents an attention head.

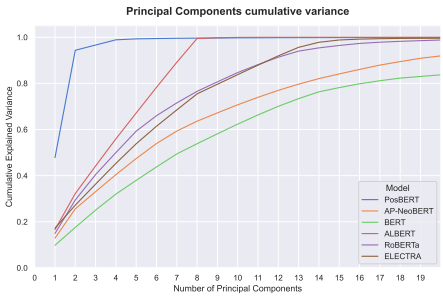


Figure 7: Cumulative explained variance of singular vectors for different encoder-based models with learned AP embeddings.

and evaluate on the validation split every $n = \min(500, \text{len(dataloader)}//10)$ steps. Training is stopped after 15 evaluations without improvement, or after 10 epochs. Following standard practice, WNLI is excluded from the evaluation. For each model, we report only the performance of the best hyperparameter combination. Results are displayed in Table 6.

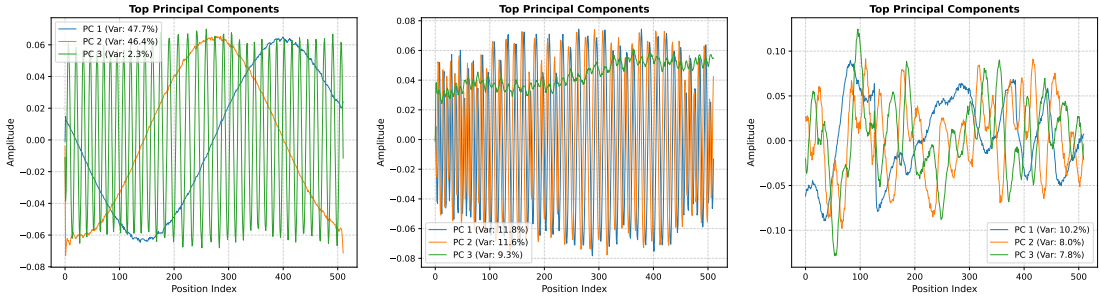
Full per-task results for the experiments described in Section 6.

C.2 MTEB

Metrics Similar to GLUE, metrics are different for each task. We use tham in metrics advised in Muennighoff et al. (2023). For classification, the main metric is the accuracy. For clustering, the main metric is the v-measure (Rosenberg and Hirschberg, 2007). For pair classification, it is the average precision. For reranking, MAP is used. Retrieval uses nDCG@10. STS and summarization uses Spearman correlation. Results are displayed in Table 7.

C.3 SQuAD

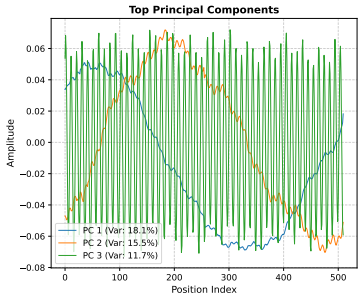
For SQuAD, we report the best best results of exact match (EM) and F1-score with a gridsearch of learning rates in $\{1\text{-}5, 3e\text{-}5, 5e\text{-}5\}$. For all four models, the best results for a learning rate of $3e$-5.



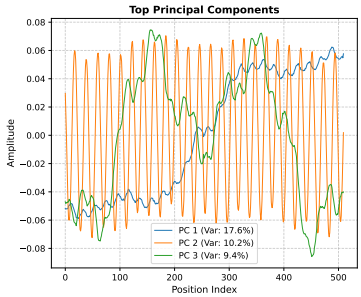
(a) PosBERT

(b) AP-NeoBERT

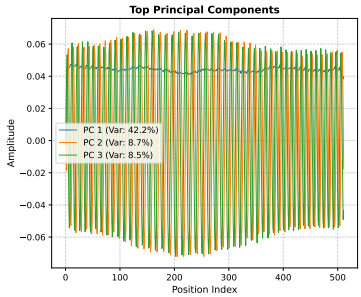
(c) BERT



(d) ALBERT



(e) ELECTRA



(f) RoBERTa

Figure 8: The three first PCs of the AP embeddings of different models.

Model	MNLI	QNLI	QQP	RTE	SST	MRPC	coLA	STS	Avg.
PosBERT	0.81	0.87	0.87	0.60	0.92	0.85	0.48	0.87	0.78
RoPE-NeoBERT	0.81	0.89	0.87	0.67	0.93	0.85	0.48	0.87	0.79
AP-NeoBERT	0.80	0.88	0.86	0.68	0.92	0.85	0.45	0.87	0.79
RP-NeoBERT	0.79	0.87	0.86	0.61	0.90	0.82	0.36	0.85	0.75

Table 6: Results on the GLUE Benchmark

Model	Class.	Clust.	PairClass.	Rerank.	Retri.	STS	Summ	Avg.
PosBERT	0.63	0.30	0.69	0.52	0.14	0.63	0.30	0.46
RoPE-NeoBERT	0.63	0.31	0.70	0.51	0.13	0.65	0.30	0.46
AP-NeoBERT	0.63	0.31	0.73	0.52	0.15	0.67	0.28	0.47
RP-NeoBERT	0.59	0.30	0.65	0.50	0.12	0.61	0.29	0.43

Table 7: Results on the MTEB Benchmark