Give it Space! Explicit Disentangling of Positional and Semantic Representations in Encoders

Anonymous ACL submission

Abstract

Positional encoding (PE) underpins how permutation-invariant Transformers represent sequence order, yet how positional information is processed and stored remains poorly understood. Modern PE methods such as RoPE still struggle on tasks such as long-context understanding or retrieval (Chen et al., 2025). Hence, a better understanding of the internal positional mechanism could help design better PE. Building on evidence that positional and semantic signals occupy nearly orthogonal subspaces in trained Transformers, we modify an encoder Transformer to process three explicitly disentangled streams: semantic, absolute positional (AP) and relative positional (RP), and confine the masked-language-modeling (MLM) objective to the semantic stream. This decoupling enables a clean mechanistic study and yields three take-aways. (1) The isolated AP subspace spontaneously collapses into a low-frequency two-dimensional manifold that captures the structure of the document; (2) Attention heads specialize into structure and semantic-oriented groups, with RP exclusively supporting the latter; (3) Standard positional encodings do not robustly retain macroscopic structure: RoPE and RP only weakly encode it, and entangled AP loses it in the final layers under MLM pressure. The disentangled approach preserves positional encoding, which improves linguistic representation on 49 of the 65 linguistic phenomena of the Flash-Holmes probing benchmark.

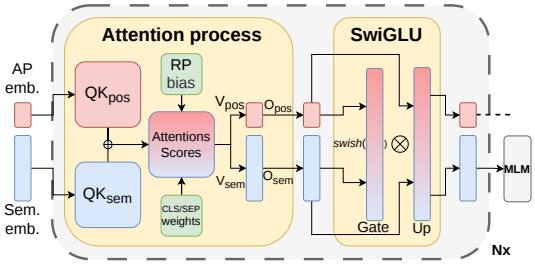


Figure 1: The disentangled architecture. Absolute positional (AP) embeddings are displayed in red, and semantic embeddings in blue. Red-blue gradient components contain shared information. Separate RMSNorms are not displayed for readability.

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., 2022; 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 that this forced positional dependence is often unnecessary; many heads function perfectly well without positional signals. Therefore, the fundamental question of how positional information is used, and how to best represent it, remains open across architectures. Given recent evidence that models learn positional and semantic signals in nearly orthogonal subspaces (Song and Zhong, 2024), explicitly separating these signals would provide new insights into the internal mechanisms used in models and help define better position representations in transformers. Additionally, providing separated representations could allow for richer embeddings, since they do not need to mix information. We present in this work the first mechanistic exploration of such a disentangled system to tackle the following research questions:

- Does explicitly disentangling positional and semantic streams provide insights into Trans-

formers' internal mechanisms?

- What geometric structure and functional taxonomy emerge within the network when positional and semantic information streams are separated?
- Does explicitly offloading position to an isolated subspace yield richer representations?

We compare three baselines, one using RoPE, one using learned AP embeddings, and one using a learned RP bias, with a disentangled architecture using separated AP embeddings and a learned RP bias. After introducing the architectural changes in Section 3, we make the following contributions:

- Through a mechanistic exploration, we reveal that the isolated AP space collapses into a two-dimensional, low-frequency structural manifold (94% variance in 2 principal components), and that attention heads specialize almost exclusively into structure-oriented or semantic-oriented groups (Section 4).
- We show that RoPE is insufficient to encode macroscopic document structure. While entangled AP embeddings can encode this structural information, they lose most of it in the final layer due to interference from the MLM prediction objective. Disentangling the streams solves this positional bottleneck.
- We demonstrate that this disentanglement enhances the linguistic fidelity of the token representations: it improves probing results on 49 of the 65 linguistic phenomena of the Flash-Holmes probing benchmark.

For reproducibility and further research, we share our implementation and experimental scripts.[1]

2 Related Works

Positional Encoding Positional encoding (PE) 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., 2022), 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.

Semantic-Positional Interaction in PE 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. While these PE methods better account for the positional-semantic interaction, they do not allow direct access to each type of information. Our architecture shares TUPE's (Ke et al., 2021) separation of AP and RP, but replaces its static, shared AP bias with an evolving AP subspace that progressively refines structural representations and mixes with semantic space 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. Our disentangled design makes this specialization explicit.

3 Architecture and Training

To explore the disentanglement of positional and semantic information, each component of the architecture must be modified. In this section, we introduce the architecturally separated semantic, absolute-positional, and relative-positional information streams. We base our architecture on

[1] shared on openreview, will be shared in camera ready

NeoBERT (Breton et al., 2024) and we describe below the modifications to each component. The full architecture is displayed in Figure 1.

3.1 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 $d_{model} = d_{AP} + d_{sem}$. We use the `bert-base-uncased` tokenizer.

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 Many recent biases (Press et al., 2022; Chi et al., 2022; Li et al., 2024) rely on kernels parameterized by the signed distance $i - j$, which imposes a long-term decay prior. To avoid baking such a prior into our analysis, we instead adopt 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 as the distance between tokens increases, while leaving the decay behaviour to be learned. 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:

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

Attention Heads For multi-head attention with n_{heads} heads, queries and keys are computed independently for each stream within each head h: $Q^h_{sem} = W^{h,sem}_q x_{sem}$, $K^h_{sem} = W^{h,sem}_k x_{sem}$, $Q^h_{AP} = W^{h,AP}_q x_{AP}$, $K^h_{AP} = W^{h,AP}_k x_{AP}$, where $W^{h,AP}_q, W^{h,AP}_k \in \mathbb{R}^{d_{head}\times d_{AP}}$ and $W^{h,sem}_q, W^{h,sem}_k \in \mathbb{R}^{(d_{head})\times d_{sem}}$, with $d_{head} = d_{model}/n_{heads}$. A key design choice: we project both streams to the same per-head size d_{head} 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. This requires d_{AP} and d_{sem} to be multiples of n_{heads}; in our configuration $d_{AP}/n_{heads} = 8$, which is small but suffices since the AP stream carries low-bandwidth structural information. The attention weight matrices are $w^{h,sem}_{i\leftarrow j} = \frac{(Q^h_{sem})_i \cdot (K^h_{sem})_j}{\sqrt{d_{head}}}$ and $w^{h,AP}_{i\leftarrow j} = \frac{(Q^h_{AP})_i \cdot (K^h_{AP})_j}{\sqrt{d_{head}}}$. The attention logit $l^h_{i\leftarrow j}$ is computed by summing the RP bias $b^h_{i\leftarrow j}$, the semantic attention weight $w^{h,sem}_{i\leftarrow j}$, and the AP attention weight $w^{h,AP}_{i\leftarrow j}$ (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: $A^h_{i\leftarrow j} = \frac{\exp(l^h_{i\leftarrow j})}{\sum_{j'} \exp(l^h_{i\leftarrow j'})}$. A visual explanation of this process is shown in Appendix A.2 (Figure 5).

The values are computed separately for each stream and remain within their respective spaces: per head h, $V^h_{AP} = W^{h,AP}_v x_{AP}$ and $V^h_{sem} = W^{h,sem}_v x_{sem}$, with $W^{h,AP}_v \in \mathbb{R}^{(d_{AP}/n_{heads})\times d_{AP}}$ and $W^{h,sem}_v \in \mathbb{R}^{(d_{sem}/n_{heads})\times d_{sem}}$. The same attention scores A^h are applied to both value streams, and the concatenated per-head outputs are processed by output projections $W^{AP}_o \in \mathbb{R}^{d_{AP}\times d_{AP}}$ and $W^{sem}_o \in \mathbb{R}^{d_{sem}\times d_{sem}}$.

3.2 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 $d_{int} = d_{int}^{AP} + d_{int}^{sem}$ dimensional space. To maintain the architectural separation, we re-separate the information during the down-projection. Denoting by $x_{AP}^{(l)}$ and $x_{sem}^{(l)}$ the AP and semantic representations after the attention block at layer l:

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

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

The SwiGLU is the only component where semantic and positional embeddings directly interact, serving as a 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. This design corroborates previous work (Golovneva et al., 2024; Zheng et al., 2024) showing the benefit of conditioning positional representations on semantic content.

3.3 Training

Setup Following Breton et al. (2024), we train our architecture on the English corpus of the *FineWeb* dataset (Penedo et al., 2024) with a Masked Language Modeling (MLM) objective. We use an $L = 6$ layers architecture with $A = 6$ heads per layer and a hidden size of $d_{model} = 768$. We split the embedding into $d_{AP} = 48$ and $d_{sem} = 720$, corresponding to $1/16$ 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 70k 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 an important design choice: it frees the AP subspace from prediction pressure. Because of this, the AP component of the last layer's hidden state is unused by the MLM head and therefore receives no gradient.

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 $[P_k, P_{k+1}, ..., P_{k+n}]$.

Baselines In the following sections, we compare our disentangled architecture (DSTG-NeoBERT, hereafter the default DSTG variant trained with the semantic-only MLM head described above) with three baseline NeoBERT architectures: RoPE-NeoBERT, with a RoPE-based positional encoding; AP-NeoBERT, with a learned AP embedding; and RP-NeoBERT, with a learned RP embedding. The three baselines have $L = 6$ layers with $A = 6$ heads per layer, and a hidden size $d_{model} = 720$ (matching d_{sem} of DSTG-NeoBERT; the resulting parameter asymmetry is discussed in the Limitations section). These baselines are trained using the exact same experimental setup and data.

3.4 Preliminary Experiments

Before any analysis, we ensure that DSTG-NeoBERT does not show catastrophic failure on standard encoder tasks, and is comparable to its entangled counterparts. We evaluate the four models on three standard benchmarks: GLUE (Wang et al., 2018), MTEB (Muennighoff et al., 2023) and SQuAD (Rajpurkar et al., 2016). 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; SQuAD benchmarks model on extractive question answering. Table 1 displays the results and confirms that disentanglement does not degrade standard performance. Experimental details and per-task results are given in Appendix C.

4 Analysis of the Learned Representations

DSTG-NeoBERT encodes structural information in a low-dimensional AP subspace, and its atten-

	metric	AP	RP	RoPE	DSTG
GLUE	Avg.	**0.79**	0.75	**0.79**	0.78
MTEB	Avg.	**0.47**	0.43	0.46	0.46
SQUAD	EM	0.74	0.75	0.76	**0.77**
	F1	0.82	0.84	**0.86**	**0.86**

Table 1: Results on the GLUE, MTEB and SQuAD benchmarks. "Avg." metric signify an average of task-dependent metrics (see Appendix C). "EM" means *Exact Match*.

tion heads spontaneously specialize into structure-driven and semantic-driven groups. We demonstrate these properties through visualization, probing, and spectral analysis.

(1) The learned AP embeddings collapse into a 2D low-frequency manifold Despite being 48-dimensional, DSTG-NeoBERT's learned AP embedding matrix (the lookup table mapping positions to initial AP vectors) converges toward a two-dimensional low-frequency sinusoidal space. First, PCA on this embedding matrix, following Wang and Chen (2020), shows that the 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's learned position embeddings remain entangled, capturing only 23.4% in their first two PCs. Second, a Type-II Discrete Cosine Transform confirms that these PCs operate at low frequencies, concentrating 98.4% and 99.0% of their spectral power within the first four bins. The third component (2.3% variance) acts as a high-frequency component. Conversely, AP-NeoBERT's top PCs are dominated by high-frequency signals, holding under 0.2% of their power in low-frequency bins. Appendix B.1 details DSTG-NeoBERT's representational simplicity. **This corroborates findings from Song and Zhong (2024), and hints at the inherent simplicity of the positional information needed by Transformers when they can use RP on top of AP.**

(2) DSTG-NeoBERT trades off absolute position for structure The model uses its low-frequency AP components to represent the macroscopic structure of the text. As exhibited in its attention patterns (Figure 3, second row), DSTG-NeoBERT learns to use the AP embedding space as structural encoding, separating sentences and paragraphs into blocks. Additionally, visualizing the AP *hidden*

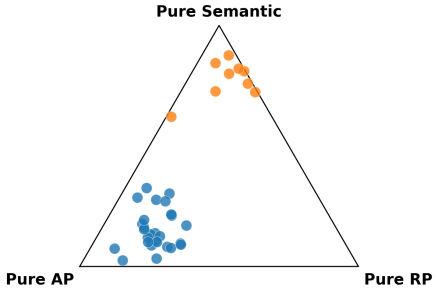


Figure 2: Categorization of all heads across layers of DSTG-NeoBERT, based on the KL-divergence of ablated information. Each corner corresponds to heads influenced by a single information.

states across layers (Figure 4) – whose first two PCs retain about 90% of the variance – reveals a growing separation between sentences across layers: tokens from the same sentence cluster together, and these clusters become increasingly distinct in deeper layers. The AP space therefore keeps its low effective dimensionality throughout the network. **This shows that the AP space progressively abstracts raw position into document structure.**

(3) Attention heads cluster into AP-oriented and semantic-oriented groups DSTG-NeoBERT's disentangled architecture reveals a functional taxonomy: attention heads specialize almost exclusively as either AP-driven or semantic-driven. 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 3, 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 i (AP, RP or semantic) is removed prior to the softmax, for each information $i \in \{sem, AP, RP\}$:

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

Averaging these KL divergences across 500 documents from WikiText (Merity et al., 2016) yields a 3-dimensional influence vector $[Score_{sem}, Score_{AP}, Score_{RP}]$ for each head, normalized per-head. As shown in Figure 2, only 9

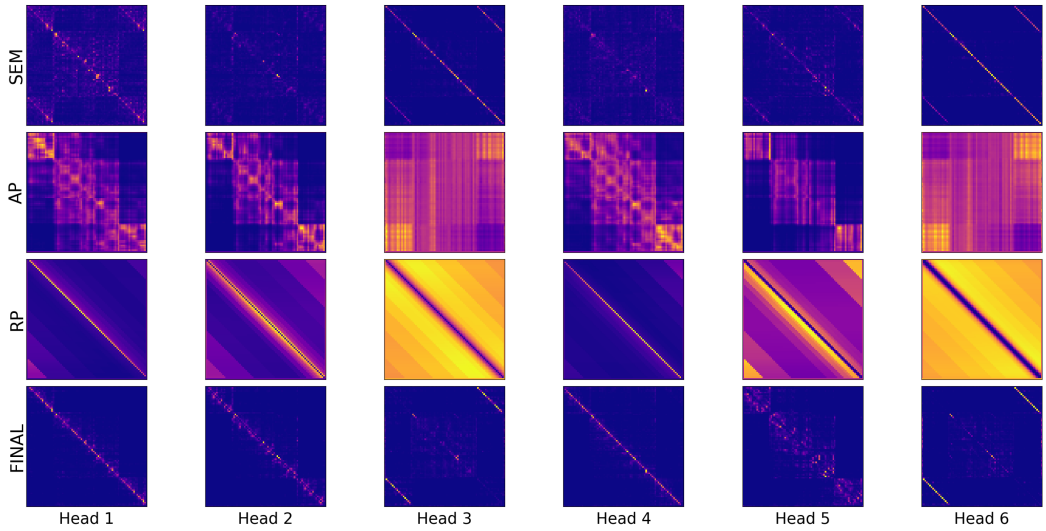


Figure 3: Softmax applied independently to the last layer's attention weights for semantic (1st row), AP (2nd row) and RP (3rd row) components, as well as the combined attention scores (Eq. 2, 4th row) on an example of shape DOC-A + DOC-B + DOC-A. Each column corresponds to an attention head. Color scale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

out of the 36 heads *specialize* in semantic matching. The remaining heads are strongly affected by the AP information. **Despite being given the freedom to either use AP or RP positional information, the model only uses RP as complementary information to the semantic heads, while AP is used as its own information stream.**

These findings suggest that while RP and semantic information complement each other, AP information is used as an additional structural information stream. This raises two questions: (1) Do entangled models with other positional encodings (RP, RoPE, or entangled AP) end up encoding comparable structural information in their hidden states, or is the disentangled AP stream playing a distinct role? (2) Does the structural information preserved by the disentangled architecture translate into gains on downstream linguistic probes? We answer these questions in the next section.

5 Probing Structural Representation in Transformers

Understanding document structure requires a hierarchical coordinate system capturing a token's global absolute position, its structural segment, and its local intra-segment position. This is what we study in this section.

Let model $\mathcal{M}$ map a document $\mathcal{D}$ of length N to hidden states $H^{(l)} = \{h_1^{(l)}, \ldots, h_N^{(l)}\}$ at layer l. We evaluate the four models 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 use texts from the WikiText (Merity et al., 2016) dataset, concatenated to make 500 documents of maximum input lengths for the models (512 tokens). We report the average results over five runs with different seeds, using 400 documents for training and 100 for testing. For DSTG-NeoBERT, we also probe separately the AP and semantic spaces to evaluate where the information is represented.

5.1 Token-level AP

The first level of the hierarchy is the most basic: can a model recover the global position of a token? The absolute position $y_i \in [0, 1]$ of token t_i is defined as its normalized index:

$$y_i = \frac{pos(t_i)}{max_j(pos(t_j))} = \frac{pos(t_i)}{512}$$

Results The results are reported in Table 2. As expected, AP-NeoBERT properly encodes this information, especially in early layers. Critically, AP information collapses in the final layer ($R^2 = 0.34$): We hypothesize that because the MLM head operates on the full hidden state, the model repurposes positional dimensions for semantic prediction, overwriting structural information. We validate this through the DSTG-NeoBERT probe: The disentangled architecture keeps the positional information until its last position layer in its positional subspace, not used by the MLM head. RoPE-NeoBERT and RP-NeoBERT are largely unable to encode AP information in their hidden states.

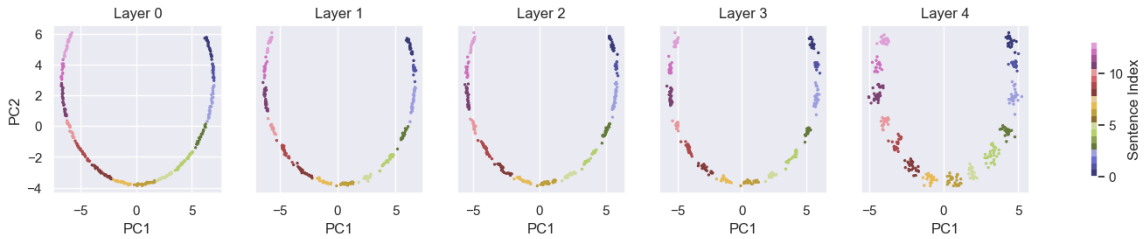


Figure 4: 2-dimensional PCA of the AP hidden states at each layer when encoding a long document. Each sentence (delimited by punctuation) is given a different color. Tokens from the same sentence cluster together, with clusters becoming increasingly distinct in deeper layers. The last layer is omitted because the AP subspace is not used for MLM prediction, leaving it unconstrained.

Layer	0	1	2	3	4	5
RoPE	0.10	0.12	0.32	0.20	0.18	0.06
RP	0.17	0.50	0.46	0.42	0.42	0.22
AP	**0.99**	**0.99**	**0.97**	0.89	0.82	0.24
DSTG	**0.99**	0.98	0.96	**0.95**	**0.92**	-
AP only	*0.99*	*0.98*	*0.95*	*0.94*	*0.89*	-
sem. only	*0.44*	*0.34*	*0.35*	*0.32*	*0.26*	*0.18*

Table 2: Token-level AP probe results. "-" marks the last-layer AP probe for DSTG: because the AP component of the final hidden state receives no gradient (see Section 3.3), we do not probe it; we report only the semantic subspace at layer 5.

Layer	0	1	2	3	4	5
RoPE	0.09	0.11	0.30	0.19	0.17	0.06
RP	0.17	0.47	0.40	0.37	0.37	0.23
AP	0.94	**0.95**	**0.93**	0.85	0.79	0.23
DSTG	**0.94**	0.94	0.92	**0.91**	**0.88**	-
AP only	*0.94*	*0.93*	*0.91*	*0.90*	*0.85*	-
sem. only	*0.42*	*0.33*	*0.35*	*0.31*	*0.27*	*0.18*

Table 3: Segment-level AP probe results. "-" marks the discarded last AP space (see Table 2 caption).

5.2 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 3, 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 the last layer, while DSTG-NeoBERT manages to keep this structural information. Both relative baselines struggle to encode segment-level position.

5.3 Intra-segment token AP

The third level tests local progress: does a model encode where a token sits within its own segment? For a token t_i in 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 linear slope, where 0.0 denotes the segment start and 1.0 the terminal boundary marker.

Results Though AP-NeoBERT and DSTG-NeoBERT perform better than RoPE- and RP-NeoBERT, all four perform correctly on the task, as displayed in Table 4. This unexpected result is explained in the subspace probing of DSTG-NeoBERT: we find that the entirety of the intra-segment position information can be retrieved from the semantic subspace, and that the positional subspace plays no role in this probe.

We validate in Appendix B.2 that MLM is the reason for the loss of positional information by running the probes on a DSTG-NeoBERT trained with MLM on the full $d_{model} = d_{AP} + d_{sem}$ embedding, and observe a loss of positional information in the last layer. We also run inter-model probing

Layer	0	1	2	3	4	5
RoPE	0.29	0.70	0.73	0.69	0.68	0.50
RP	**0.50**	0.68	0.67	0.68	0.63	0.50
AP	0.20	**0.76**	**0.80**	**0.76**	0.66	0.46
DSTG	0.43	0.73	0.76	**0.76**	**0.73**	-
AP only	*0.04*	*0.06*	*0.06*	*0.06*	*0.06*	-
sem. only	*0.42*	*0.73*	*0.76*	*0.75*	*0.73*	*0.58*

Table 4: Intra-segment token AP probe results. "-" marks the discarded last AP subspace (Table 2 caption).

	AP	RP	RoPE	DSTG
Morphology	0.58	0.58	0.58	**0.60**
Reasoning	0.47	0.47	0.46	**0.48**
Semantics	0.47	0.47	0.48	**0.52**
Discourse	0.32	0.33	0.32	**0.35**
Syntax	0.71	0.71	0.71	**0.75**

Table 5: Results on the five linguistic fields of the Flash-Holmes benchmark (Waldis et al., 2024).

to ensure that the information encoded by DSTG-NeoBERT is similar to that of the other baselines. Our experiments establish two findings: (1) **among the encoders we study, only those that carry persistent AP information in their hidden states reliably encode document structure**; relative-only models (RP, RoPE) do not learn to do so under MLM pretraining; and (2) **AP-NeoBERT does encode structure and uses it in its intermediate layers, but discards this information in its last layer** to prepare for the MLM task.

5.4 Benefits of the Structural Encoding

We hypothesize that the additional structural information enhances token representations. To validate this, we run the disentangled model and the three baselines on the Flash-Holmes (Waldis et al., 2024) probing benchmark. Flash-Holmes uses over 200 datasets to probe hidden representations on 65 linguistic phenomena making up 5 fields: *Morphology* probes the structure of words with phenomena such as irregular forms or subject-verb agreement; *Syntax* probes the structure of sentences such as filler gaps or argument structure; *Discourse* probes the context in text such as rhetorical structure or sentence order; *Semantics* probes the meaning of words such as subject gender or factuality; and *Reasoning* probes the use of words in logical deduction such as negation.[2]

The results are averaged over five runs with different seeds and reported in Table 5. These results validate our hypothesis: the disentangled architecture obtains better results on all five fields. Interestingly, the three baselines perform equally, hinting that the additional gains stem from the preserved structural information. We report the results on the 65 phenomena in Appendix C.3. The disentangled architecture performs best on 49 of the 65 phenomena: 18 out of 21 phenomena in *Syntax*, 17 out of 24 in *Semantics*, 5 out of 6 in *Discourse*, 6 out of 10 in *Reasoning*, and 3 out of 4 in *Morphology*.

[2]Definitions taken directly from Waldis et al. (2024).

6 Conclusion

In this work, we explored the explicit disentanglement of semantic, absolute positional, and relative positional information in Transformer encoders. By introducing dedicated positional and semantic streams, we showed that positional information naturally organizes into a simple low-dimensional structural manifold, while attention heads specialize into distinct structural and semantic roles. Our analysis further revealed that standard positional encoding methods either fail to preserve macroscopic structure (for relative-bias-based methods), or discard it in later layers under MLM training pressure (for AP methods). Removing AP information from the MLM objective allowed the model to preserve structural information throughout the network and improved linguistic representations on probing tasks, outperforming entangled baselines on most phenomena in the Flash-Holmes benchmark and matching them on GLUE, MTEB and SQuAD.

Our work suggests new approaches to positional encoding in transformers. The simplicity of the learned disentangled AP space, paired with our findings on the loss of positional information in the last layer due to MLM, hints at the creation of new explicit ways to provide the model with absolute positional information without interfering with semantic representations. Additionally, if our observations extend to decoders, a disentangled approach 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. Therefore, scaling to larger models and extending the approach to decoder-only architectures are promising directions for future work.

Limitations

While our work 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 disentangled model and baselines were trained from scratch on approximately 22 billion tokens using a 6-layer, 6-head architecture. 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, DSTG-NeoBERT uses a slightly larger hidden size than the three baselines ($d_{model} = 768$ vs. 720), which gives it $\sim 6\%$ more parameters per token. We made this choice so that the semantic stream of DSTG-NeoBERT matches the full hidden size of the baselines, but it introduces a small capacity asymmetry in the comparisons. The maximum sequence length is also limited to 512 tokens, which restricts our ability to probe the very long-context regimes that motivated this work; we expect the AP manifold's low-frequency structure to be most useful at longer contexts, but verifying this requires a separate scaling study.

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. 2022. 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.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

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*.

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.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

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.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

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.

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.

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 DSTG-NeoBERT Architecture Details

A.1 T5 Positional Bias

This section provides details on the T5 bucketed bias used in DSTG-NeoBERT's relative positional component (Section 3). 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 3.

B Complementary Analysis of Learned Representations

B.1 Comparison of Learned AP Embeddings

This section extends the PCA and DCT analysis of Section 4 by comparing DSTG-NeoBERT'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 6 shows the cumulative variance of Principal Components (PCs) for each model. DSTG-NeoBERT is significantly lower-dimensional than other systems. The principal components of each model are displayed in Figure 9. DSTG-NeoBERT displays strongly sinusoidal patterns with a much higher explained variance.

B.2 Additional Probing Experiments

MLM on the Full d_{model} Embedding To further confirm that the MLM training objective is the cause of the loss in structural information, we trained a disentangled architecture with the MLM objective applied to both the positional and the semantic spaces, and trained the structural probes. We compare AP-NeoBERT, DSTG-NeoBERT (semantic MLM) and DSTG-NeoBERT (full embedding MLM), and provide the results for the token-level AP and segment-level AP in Figure 7. Note that we also provide the probing results on the last positional layer of DSTG-NeoBERT (semantic MLM), which is never used during training. We observe a significant loss of positional information in the last layer when MLM is applied to the full embedding, confirming that the loss of positional information stems from the MLM objective.

Inter-Model Probing To evaluate how much of the information encoded by DSTG-NeoBERT is represented in the three other baselines, we regress the hidden states of each baseline (AP, RP and RoPE) onto the disentangled AP and semantic

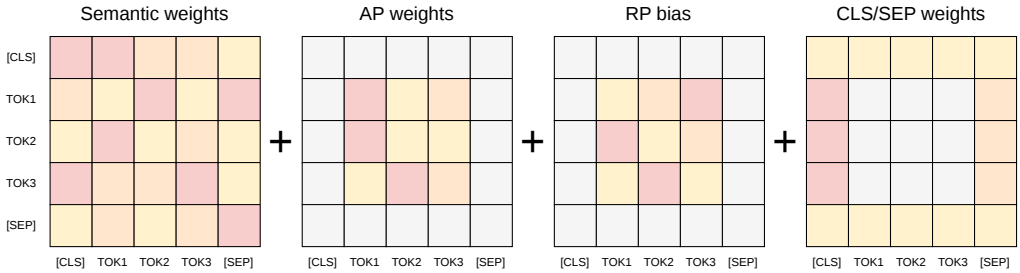


Figure 5: DSTG-NeoBERT attention weights mechanism. Grayed-out squares correspond to discarded weights. From left to right: (1) The semantic space can match all tokens. (2) The AP space can match all but [CLS] and [SEP] tokens. (3) The RP bias learns a per-bucket weight (all buckets are of size 1 in 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.

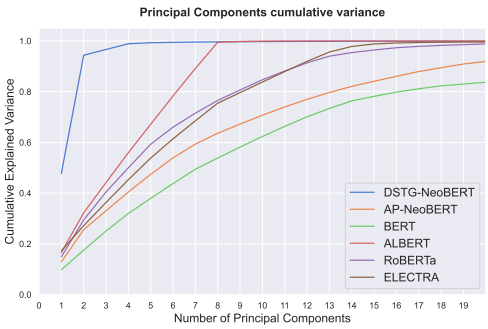


Figure 6: Cumulative explained variance of singular vectors for different encoder-based models with learned AP embeddings.

representations of DSTG-NeoBERT. Results are shown in Figure 8. While only AP-NeoBERT encodes the positional information, all three models achieve comparable regression R^2 when predicting the semantic representations of DSTG-NeoBERT. Given that the probe is linear, it is not expected to achieve an R^2 score close to 1 on the semantic embeddings probe.

Task	AP	RP	RoPE	DSTG
sentence order (F1)	0.72	0.72	0.65	**0.89**
top-constituent (F1)	0.35	0.35	0.37	**0.47**
POS tagging (F1)	0.62	0.62	0.64	**0.71**
Island effect (F1)	0.73	0.73	0.72	**0.79**
Readability (Pearson)	0.22	0.22	0.20	**0.27**

Table 6: Linguistic Phenomena of Flash-Holmes with the best improvements.

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.

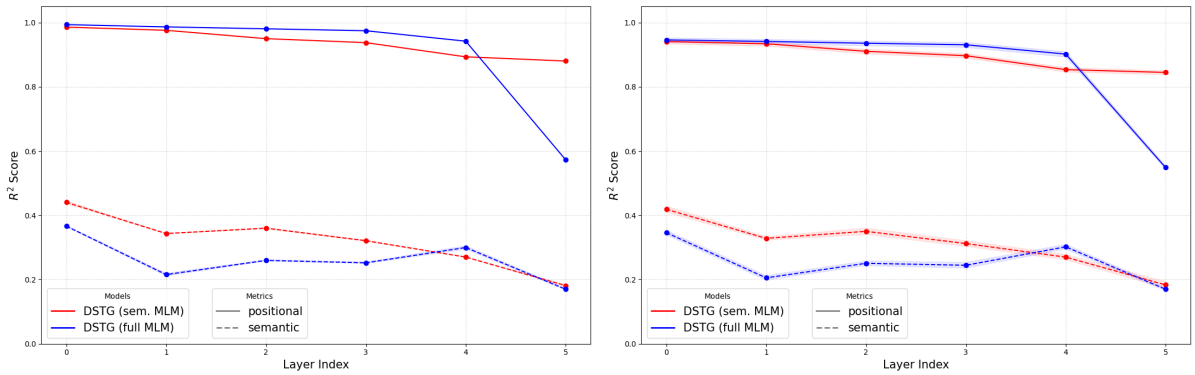
Hyperparameters and Training We run the hyperparameter search with learning rates in $\{6e\text{-}6, 5e\text{-}6, 2e\text{-}5\}$, batch sizes in $\{4, 16, 32\}$, and weight decays in $\{1e\text{-}2, 1e\text{-}5\}$. Following NeoBERT, we finetune on the training split and evaluate on the validation split 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 7.

C.2 MTEB

Metrics Similar to GLUE, metrics differ for each task. We use the metrics recommended in Muennighoff et al. (2023). For classification, the main metric is 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 use Spearman correlation. Results are displayed in Table 8.

C.3 Flash-Holmes

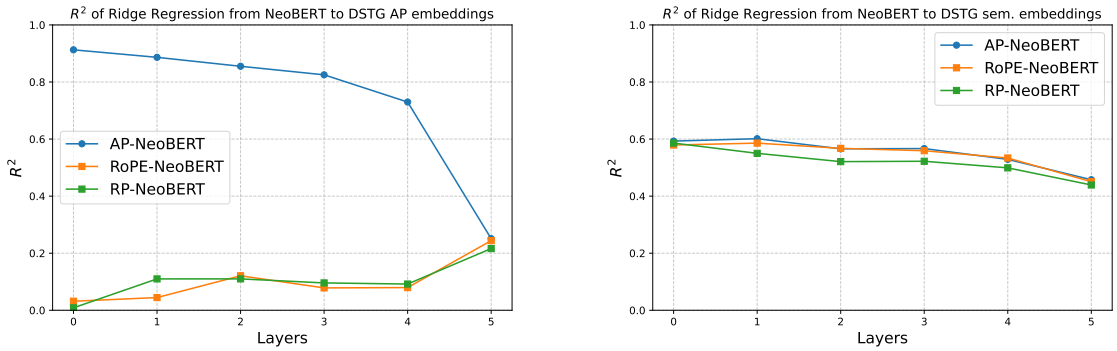
Results A deeper analysis of the results on Flash-Holmes shows that the highest gains are found in phenomena related to macroscopic structure. Table 6 shows the results for the five phenomena with the highest gains. *top-constituent* prediction probes highest-level grammatical phrases (subject, verb...). *Island effect* refers to why it is ungrammatical to



(a) token-level AP probe

(b) segment-level AP probe

Figure 7: Results of the structural probes on DSTG-NeoBERT with MLM on semantic only (red) and DSTG-NeoBERT with MLM on both semantic and positional (blue), for their positional (solid line) and semantic (dotted line) embeddings.



(a) regression to positional embedding

(b) regression to semantic embedding

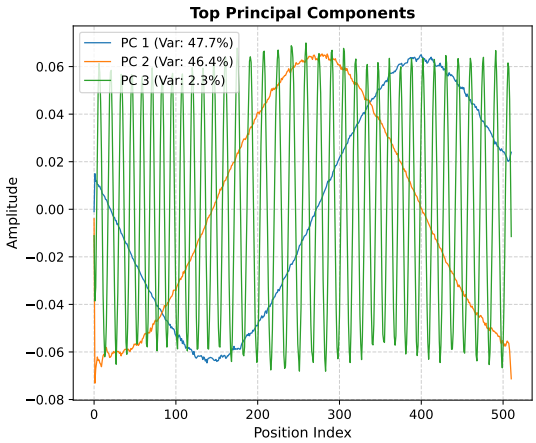
Figure 8: Regression from NeoBERT variants to DSTG-NeoBERT subspaces.

Model	MNLI	QNLI	QQP	RTE	SST	MRPC	coLA	STS	Avg.
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
RoPE-NeoBERT	**0.81**	**0.89**	**0.87**	0.67	**0.93**	**0.85**	**0.48**	**0.87**	**0.79**
DSTG-NeoBERT	**0.81**	0.87	**0.87**	0.60	0.92	**0.85**	**0.48**	**0.87**	0.78

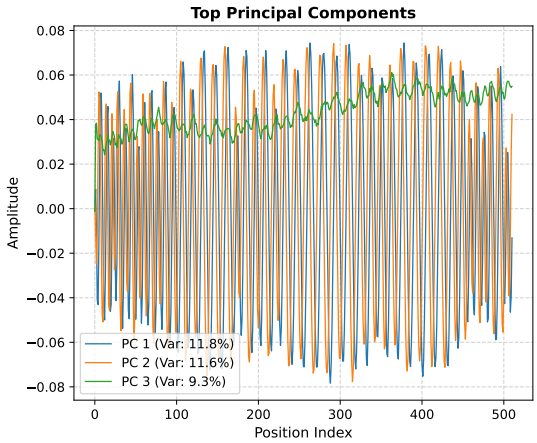
Table 7: Results on the GLUE Benchmark

Model	Class.	Clust.	PairClass.	Rerank.	Retri.	STS	Summ	Avg.
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
RoPE-NeoBERT	**0.63**	**0.31**	0.70	0.51	0.13	0.65	**0.30**	0.46
DSTG-NeoBERT	**0.63**	0.30	0.69	**0.52**	0.14	0.63	**0.30**	0.46

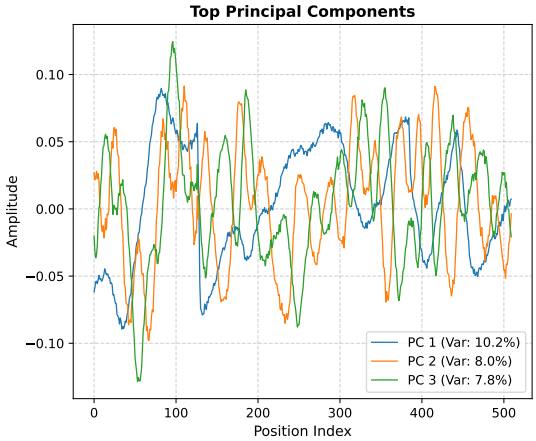
Table 8: Results on the MTEB Benchmark



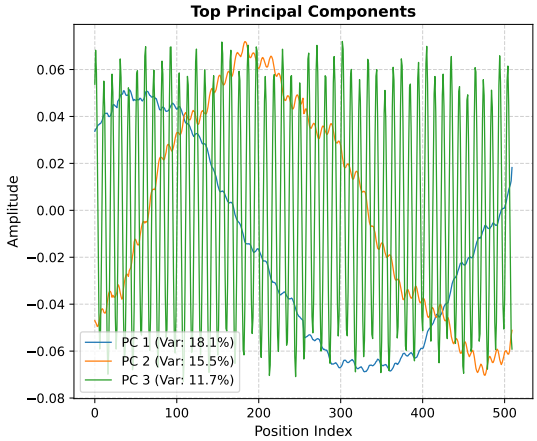
(a) DSTG-NeoBERT



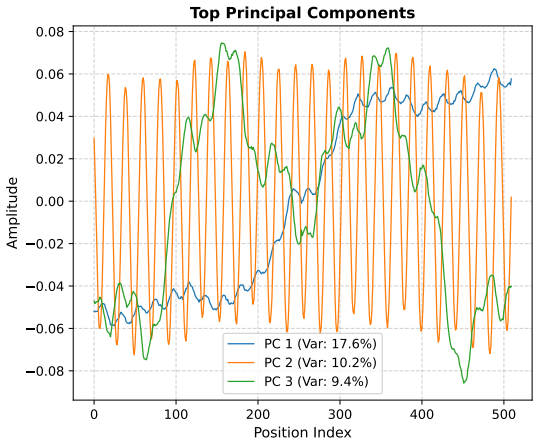
(b) AP-NeoBERT



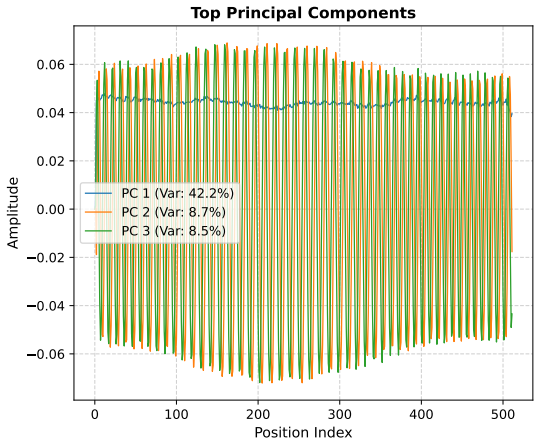
(c) BERT



(d) ALBERT



(e) ELECTRA



(f) RoBERTa

Figure 9: The first three PCs of the AP embeddings of different models.

extract or move a word or phrase out of certain complex structural domains, known as "islands," to form questions or relative clauses. *Readability* computes the Flesch Score (Flesch, 1948), a measure of readability based on the number of words in the sentence and the average number of syllables per word. We also provide the results on all phenomena, per subfield: Discourse (Table 9), Morphology (Table 10), Reasoning (Table 11), Semantics (Table 12) and Syntax (Table 13). We find a clear advantage of the disentangled architecture on the Syntax, Semantics and Discourse subfields. Refer to Waldis et al. (2024) for the definition of each phenomenon.

We discarded 7 out of the 215 probing datasets used in Flash-Holmes because the results were statistically insignificant (standard deviation > 0.1) for at least one model: blimp-determiner_noun_agreement_irregular_2, protoroles-changes_possession, protoroles-exists_as_physical, protoroles-location_of_event, protoroles-makes_physical_contact, protoroles-predicate_changed_argument and protoroles-stationary. Remaining datasets have a mean standard deviation of 0.02.

C.4 SQuAD

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

Phenomena	AP	RP	RoPE	DSTG
bridging	0.49	0.49	0.49	**0.50**
coreference resolution	0.66	0.66	0.67	**0.69**
discourse connective	0.08	0.08	0.07	**0.10**
next sentence prediction	0.44	0.44	**0.45**	0.43
rethorical structure theory	0.19	0.19	0.19	**0.22**
sentence order	0.72	0.72	0.65	**0.89**

Table 9: Flash-Holmes Results on the **Discourse** subfield

Phenomena	AP	RP	RoPE	DSTG
anaphor agreement	**0.68**	**0.68**	0.65	0.65
determiner noun agreement	**0.54**	**0.54**	**0.54**	**0.54**
irregular forms	0.80	0.80	0.80	**0.86**
subject-verb agreement	0.54	0.54	0.54	**0.58**

Table 10: Flash-Holmes Results on the **Morphology** subfield

Phenomena	AP	RP	RoPE	DSTG
age comparison	0.42	0.42	**0.46**	0.41
always never	**0.44**	**0.44**	**0.44**	**0.44**
antonym negation	**0.52**	**0.52**	0.50	0.51
encyclopedic composition	0.40	0.40	0.40	**0.41**
multi-hop composition	**0.40**	**0.40**	**0.40**	**0.40**
negation	0.52	0.52	0.51	**0.53**
object comparision	0.48	0.48	0.47	**0.49**
property conjunction	0.40	0.40	**0.41**	0.40
speculation	0.44	0.44	0.40	**0.46**
taxonomy conjunction	**0.41**	**0.41**	0.40	0.40

Table 11: Flash-Holmes Results on the **Reasoning** subfield

Phenomena	**AP**	**RP**	**RoPE**	**DSTG**
complex words	0.58	0.58	0.59	**0.62**
coordination inversion	0.52	0.52	0.55	**0.57**
event structure	**0.45**	**0.45**	**0.45**	0.44
factuality	**0.46**	**0.46**	0.45	0.45
genericity	0.13	0.13	0.14	**0.16**
metaphor	0.68	0.68	**0.69**	**0.69**
named entity labeling	0.53	0.53	0.55	**0.59**
negative polarity item licensing	0.91	0.91	0.93	**0.98**
object animacy	0.79	0.79	0.81	**0.85**
object gender	0.60	0.60	**0.62**	0.55
passive	0.57	0.57	0.55	**0.59**
protoroles	0.11	0.11	0.14	**0.16**
quantifiers	0.91	0.91	0.91	**0.95**
relation classification	0.34	0.34	0.37	**0.39**
semantic odd man out	0.54	0.54	**0.55**	**0.55**
sentiment analysis	0.24	0.24	0.27	**0.30**
subject animacy	0.86	0.86	**0.90**	0.85
subject gender	0.72	0.72	**0.75**	0.57
synonym-/antonym-detection	0.63	0.63	0.63	**0.64**
tense	**0.91**	**0.91**	**0.91**	0.88
time	**0.12**	**0.12**	0.11	0.11
verb dynamic	0.69	0.69	0.67	**0.70**
word content	0.09	0.09	0.12	**0.13**
word sense	**0.18**	**0.18**	0.15	**0.18**

Table 12: Flash-Holmes Results on the **Semantics** subfield

Phenomena	**AP**	**RP**	**RoPE**	**DSTG**
anaphor agreement	0.65	0.65	**0.66**	0.63
argument structure	0.72	0.72	0.72	**0.73**
bigram-shift	0.76	0.76	0.76	**0.78**
binding	0.71	0.71	0.72	**0.77**
case	**0.89**	**0.89**	0.74	0.72
constituent parsing	0.80	0.80	0.83	**0.87**
control / raising	0.81	0.81	0.80	**0.82**
deoncausative-inchoative alternation	0.77	0.77	0.74	**0.81**
ellipsis	0.63	0.63	0.64	**0.65**
filler gap	0.85	0.85	0.87	**0.94**
island effects	0.73	0.73	0.72	**0.79**
local attractor	**0.66**	**0.66**	0.63	0.65
negative polarity item licensing	0.80	0.80	0.81	**0.82**
object number	0.78	0.78	0.77	**0.79**
part-of-speech	0.62	0.62	0.63	**0.71**
readability	0.22	0.22	0.20	**0.27**
sentence length	**0.05**	**0.05**	**0.05**	**0.05**
subject number	0.78	0.78	0.78	**0.79**
subject-verb agreement	0.52	0.52	0.52	**0.55**
top-constituent-task	0.35	0.35	0.37	**0.47**
tree-depth	0.22	0.22	0.23	**0.27**

Table 13: Flash-Holmes Results on the **Syntax** subfield