Parameter-Efficient Transformer Embeddings via Functional Factorization

Henry Ndubuaku [1]

Abstract

Embedding tables in transformer sentence encoders are dense matrices whose parameter count scales with vocabulary size. We study a narrow question: can these tables be replaced by a shared functional parameterization in sentence-level representation learning? We propose **Parameter-Efficient Transformer Embeddings (PETE)**, which maps token IDs to normalized scalar coordinates, expands them with fixed Fourier features, and refines them using a lightweight residual MLP. PETE replaces independent token-specific vectors with a shared row-generation rule for the embedding matrix. In controlled NLI-to-STS-B experiments, PETE retains much of the performance of learned lookup embeddings while reducing total parameter count by up to $8\times$–$10\times$. In ultra-compact regimes, PETE maintains non-trivial semantic alignment with as few as 58K parameters. These results suggest that functional embedding parameterizations are a promising direction for compact sentence encoders, while also motivating future comparisons to explicit low-rank and hash-based embedding compression.

1. Introduction

Embedding tables are among the simplest components of transformer models, yet they can consume a large fraction of parameters in compact sentence encoders. A standard embedding layer learns an unconstrained matrix $E \in \mathbb{R}^{V \times d}$, assigning each token an independent vector. This design is flexible, but it scales directly with vocabulary size and does not exploit possible redundancy across token representations.

This paper asks whether such independence is necessary for sentence-level representation learning. In tasks such as semantic textual similarity, models are evaluated on global sentence meaning rather than token-level prediction. This suggests that the embedding layer may tolerate a stronger structural prior, provided that downstream transformer layers can refine the resulting representations.

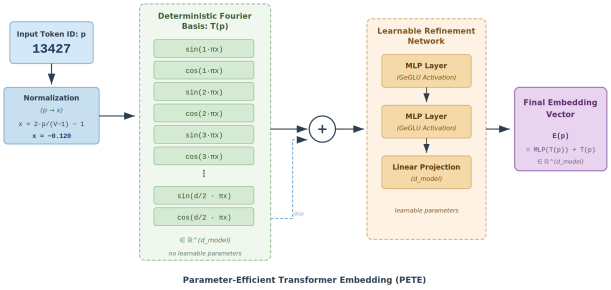

Figure 1. **Parameter-Efficient Transformer Embeddings (PETE).** Token IDs are mapped to normalized scalar coordinates, expanded with fixed Fourier features, and refined by a shared residual MLP to produce token embeddings without a learned lookup table.

Several factors make learned embedding tables inefficient in compact settings:

- **Sparsity:** Rare tokens may receive limited training signal, leading to under-optimized vectors (Svenstrup et al., 2017).
- **Redundancy:** Distinct token vectors may encode overlapping statistical or semantic roles (Lan et al., 2020).
- **Scaling with vocabulary size:** The parameter count of a learned lookup table grows as Vd, regardless of whether all token-specific degrees of freedom are needed.

We propose **Parameter-Efficient Transformer Embeddings (PETE)**, a structured embedding parameterization that replaces learned lookup tables with a shared function over token IDs. Token IDs are mapped to normalized scalar coordinates, lifted through fixed Fourier features, and refined by a lightweight residual MLP. PETE therefore replaces independent token-specific vectors with a compact row-generation rule for the embedding matrix.

From a compression perspective, PETE can be interpreted as a functional factorization of the embedding table. Instead

[1]Cactus Compute, London, United Kingdom. Correspondence to: Henry Ndubuaku <henry@cactuscompute.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

of learning every row of E independently, PETE generates row $E_{p,:}$ through a shared function $g_\theta(\phi(p))$, where $\phi(p)$ is a fixed Fourier feature map of the token index. This is not a classical low-rank factorization such as $E \approx AB$; the induced matrix is not guaranteed to have low algebraic rank. Rather, PETE is a low-capacity row-generation scheme that shares the motivation of low-rank methods: replacing an unconstrained dense matrix with a structured parameterization.

Scope. Our goal is not to claim that PETE universally replaces learned token embeddings. Instead, we study a deliberately narrow setting: compact sentence encoders trained for sentence-level semantic similarity. This setting is useful because sentence-level tasks can tolerate some loss of token-specific lexical precision while still requiring meaningful global representations. We therefore view PETE as an initial study of functional embedding compression rather than a general-purpose embedding replacement for all transformer workloads.

Our contributions are:

- We formulate embedding-table compression as a row-generation problem, replacing independent token vectors with a shared function over token IDs.
- We introduce PETE, a simple Fourier-feature and residual-MLP parameterization for generating transformer token embeddings.
- We provide a controlled comparison against learned lookup embeddings in small sentence encoders, showing substantial parameter reductions with modest degradation on STS-B.
- We discuss the relationship between functional embedding parameterizations and low-rank compression, including limitations and missing comparisons.

2. Related Work

Numerous research efforts have aimed to reduce embedding parameter inefficiency through compression, adaptive designs, and alternative representations. Cai et al. (2024) highlight the inefficiency of high-dimensional embedding representations in knowledge graphs and propose the Dimension Lifting Network, which replaces wide embeddings with a narrow embedding layer followed by a lifting network. Xu et al. (2024) introduce TensorGPT, which leverages Tensor-Train Decomposition to reduce embedding parameter counts with minimal performance degradation, though it requires a large embedding table to be trained initially before projection. Yan et al. (2021) propose an Adaptively-Masked Twins-based layer that dynamically adjusts embedding dimensions based on feature values.

Other approaches explore structured parameter sharing and compression through low-rank factorization, matrix decomposition, hashing, or quantization. Lan et al. (2020) develop ALBERT, which reduces redundancy by factorizing the embedding matrix and sharing parameters across layers. Shen et al. (2019) introduce Q-BERT, which applies Hessian-based quantization to reduce embedding precision with minimal performance loss. Hash-based embeddings (Svenstrup et al., 2017) replace direct lookup tables with hash functions, reducing memory while retaining token-specific access patterns.

Our work is also related to low-rank and structured representations in neural networks. Classical low-rank compression methods approximate a dense parameter matrix by a product of smaller matrices, while methods such as LoRA introduce trainable low-rank updates to frozen weights. PETE follows the same broad motivation of reducing unconstrained dense parameterization, but applies it to the token embedding table. Rather than factorizing a learned embedding matrix after training or introducing a learned low-rank adapter, PETE directly parameterizes embeddings as the output of a shared function over token indices.

This makes PETE a functional alternative to explicit low-rank embedding compression. The representational capacity of the embedding table is controlled by the fixed basis and shared MLP rather than by the vocabulary size. PETE therefore complements existing compression methods: it does not claim to replace classical matrix factorization, hashing, or quantization, but instead explores whether a simple row-generation rule is sufficient for compact sentence encoders.

PETE is also conceptually related to coordinate-based representations and Fourier feature mappings. In those settings, a signal is represented as the output of a function over coordinates. PETE applies a similar idea to the rows of a vocabulary embedding matrix: token IDs provide coordinates, fixed Fourier features provide a basis, and a shared MLP generates token vectors. Unlike sinusoidal positional encodings, which represent sequence positions, PETE parameterizes vocabulary entries themselves.

3. Functional Factorization of Embedding Tables

We propose a parameter-efficient token encoding strategy that replaces the standard learned embedding lookup table with a structured functional mapping from token IDs to continuous vectors. A standard embedding layer learns a free matrix

$$E \in \mathbb{R}^{V \times d_{\text{model}}}, \tag{1}$$

where each row $E_{p,:}$ is an independent parameter vector. The parameter count of this layer is

$$\#\text{params}_{\text{lookup}} = V d_{\text{model}}. \tag{2}$$

PETE replaces this unconstrained matrix with a structured row-generation rule:

$$E_{p,:} = g_\theta(\phi(p)), \tag{3}$$

where ϕ is a fixed feature map over token IDs and g_θ is a shared learned function. Equivalently, PETE models token embeddings as outputs of a shared function

$$f_\theta : [-1, 1] \to \mathbb{R}^{d_{\text{model}}}, \tag{4}$$

where $f_\theta = g_\theta \circ \phi$. Thus, the number of learned embedding parameters depends on the complexity of g_θ, not directly on the vocabulary size V.

For a two-layer MLP with hidden width h, PETE uses $O(d_{\text{model}}h + hd_{\text{model}})$ learned embedding parameters, independent of V. This changes the scaling behavior of the embedding layer from vocabulary-dependent lookup to vocabulary-independent row generation. The goal is not to perfectly recover token-specific lexical meaning at the embedding layer, but to provide a compact representation that downstream transformer layers can refine.

PETE is not a classical linear low-rank factorization of the form $E \approx AB$. Instead, it is a nonlinear functional factorization in which the rows of E are generated by a shared low-capacity function. This places PETE in the broader family of structured low-dimensional parameterizations rather than in strict matrix-rank compression.

Our approach is motivated by two observations:

1. Embedding layers in sentence encoders can be overparameterized relative to their contribution to sentence-level performance.

2. Subword token IDs produced by widely used tokenizers such as Byte-Pair Encoding (BPE) and WordPiece are often assigned approximately in descending order of frequency, introducing weak statistical structure over the index space.

We emphasize that token ID proximity does *not* imply semantic similarity. The ordering is imperfect, corpus-dependent, and noisy. PETE does not assume semantic monotonicity of token IDs; instead, the ordering provides a deterministic coordinate system over the vocabulary.

Each token ID $p \in \{0, \ldots, V-1\}$ is mapped to a normalized scalar coordinate:

$$x_p = 2\left(\frac{p}{V-1}\right) - 1. \tag{5}$$

This embeds the discrete vocabulary into a bounded interval independent of vocabulary size.

The scalar coordinate x_p is lifted into a higher-dimensional representation using a fixed Fourier basis. The resulting feature vector $\mathbf{T}(p) \in \mathbb{R}^{d_{\text{model}}}$ is defined component-wise as:

$$T_i(p) = \begin{cases} \sin(\pi x_p (\lfloor i/2 \rfloor + 1)), & i \text{ even}, \\ \cos(\pi x_p (\lfloor i/2 \rfloor + 1)), & i \text{ odd}. \end{cases} \tag{6}$$

Fourier features provide a parameter-free basis for representing continuous inputs with controllable frequency resolution. In this setting, they offer three practical advantages: zero additional parameters, deterministic computation, and support for both slowly and rapidly varying functions over token indices.

Because token ID ordering is only weakly correlated with token statistics and not directly semantic, the fixed Fourier features alone may be insufficient. We therefore introduce a lightweight shared MLP to refine the representation:

$$\mathbf{E}(p) = \text{MLP}(\mathbf{T}(p)) + \mathbf{T}(p). \tag{7}$$

The residual connection preserves the structured Fourier basis while allowing the MLP to learn nonlinear corrections. This constrains the embedding table to be a low-capacity deformation of a fixed feature representation, enforcing parameter sharing across the vocabulary.

What PETE does not claim. PETE does not assume that token ID proximity corresponds to semantic similarity, nor does it claim that natural language has a one-dimensional latent vocabulary structure. It also does not provide a universal replacement for learned embeddings in generative or token-level tasks. The claim is narrower: in compact sentence encoders, a shared functional embedding parameterization can preserve useful sentence-level representations while greatly reducing parameters.

4. Experiments

We evaluate PETE in the setting of *sentence representation learning*. Models are trained on Natural Language Inference (NLI) data and evaluated on the Semantic Textual Similarity Benchmark (STS-B), a standard transfer benchmark for sentence embeddings. Specifically, sentence encoders are trained on SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) and evaluated on the STS-B validation set (Cer et al., 2017), both without task-specific fine-tuning and with supervised fine-tuning.

Performance is reported using Pearson and Spearman correlation coefficients. All controlled comparisons use the same transformer encoder architecture and differ only in the token embedding parameterization:

- **Lookup baseline:** A standard learned embedding table of size $V \times d_{\text{model}}$.

- **PETE:** Token embeddings generated by normalized token coordinates, fixed Fourier features, and a shared residual MLP.

Unless otherwise specified, all other architectural components are held fixed, including the number of layers and attention heads, hidden dimensionality, feed-forward expansion ratio, positional encoding (RoPE; (Su et al., 2023)), normalization (RMSNorm; (Zhang & Sennrich, 2019)), and activation function (GeGLU; (Shazeer, 2020)). This isolates the effect of the embedding parameterization while keeping downstream model capacity fixed.

All experiments use the BERT WordPiece tokenizer (Devlin et al., 2019) with vocabulary size $V = 30{,}522$. For PETE, token IDs are deterministically mapped to normalized scalar coordinates in $[-1, 1]$. No token-specific parameters are learned at the embedding stage.

Models are trained using a contrastive sentence embedding objective inspired by InfoNCE. Each training example consists of paired sentences (s_i, s_j) drawn from NLI data. Premise–hypothesis pairs labeled as entailment are treated as positive pairs, while other sentences within the batch serve as negatives. Given sentence embeddings $\mathbf{h}_i$ and $\mathbf{h}_j$, cosine similarity is scaled by a learnable temperature parameter τ, and the objective encourages higher similarity for positive pairs relative to all in-batch negatives.

All models are trained using AdamW (Loshchilov & Hutter, 2019) with shared hyperparameters unless otherwise stated:

- Batch size: 128
- Learning rate: 2×10^{-5}
- Warmup steps: 1,000
- Total training steps: 122,700
- Precision: mixed precision (FP16)

To avoid introducing confounding regularization effects, dropout is disabled for both PETE and baseline models unless explicitly stated.

5. Results

5.1. Controlled Lookup-vs-PETE Comparison

Table 1 reports zero-shot performance on the STS-B validation set after training on SNLI and MNLI. Across multiple model sizes, PETE retains much of the performance of learned lookup embeddings while using substantially fewer parameters. For models with moderate capacity, PETE reduces total parameter count by approximately $8\times$–$10\times$, with a modest drop in correlation scores.

Table 1. Zero-shot STS-B validation scores without task-specific fine-tuning. PETE is compared against learned lookup embeddings under matched transformer architectures. The table is compacted to fit within one ICML column.

Model	L/H	d	Params	Spearman	Pearson
PETE	1/4	256	1.1M	74.93	74.54
Transformer	1/4	256	8.9M	77.01	76.80
PETE	1/8	512	4.7M	75.21	74.65
Transformer	1/8	512	20.1M	77.50	76.78
PETE	2/4	256	2.2M	76.38	76.02
Transformer	2/4	256	9.9M	77.34	76.89

Table 2. Contextual comparison with compact sentence-encoder baselines after STS-B fine-tuning. These numbers are not controlled head-to-head comparisons because training objectives, pretraining data, and optimization details differ across methods.

Model	Params	Pearson	Spearman	Avg
PETE	≈58K	69.0	69.5	69.25
PETE	≈396K	78.0	78.0	78.00
PETE	≈1.5M	79.7	79.7	79.70
PETE	≈3.6M	**81.7**	**81.9**	**81.80**
BERT-Tiny	≈4M	74.3	73.6	73.95
BERT-Mini	≈11M	74.3	73.6	73.95
TinyBERT	14.5M	–	80.4	–
MobileBERT-Tiny	15.1M	–	80.1	–
BERT-Small	≈29M	78.8	77.0	77.90

These results provide evidence that, in compact sentence encoders, independent token-specific embedding vectors are not always necessary for useful sentence-level semantic representations. The controlled comparison is limited to the NLI-to-STS-B setting, and should not be interpreted as a general claim about all transformer tasks.

5.2. Ultra-Compact Regimes

To explore the limits of parameter efficiency, we evaluate PETE variants designed for ultra-low-capacity settings. These models further reduce parameter count by employing compact feed-forward layers and reduced intermediate dimensions while retaining the same PETE embedding mechanism.

Table 2 places ultra-compact PETE variants in the context of commonly used compact transformer baselines. These results are included only to contextualize the scale of PETE models. The controlled comparison in this paper is the PETE-versus-lookup comparison in Table 1.

Despite operating with as few as 58K parameters, PETE maintains non-trivial semantic alignment. At parameter counts around 3–4M, PETE approaches the performance range of substantially larger compact models in this setting. However, because the external baselines differ in pretraining, distillation, architecture, and optimization, this table should

be read as contextual rather than as evidence of superiority.

6. Discussion

Connection to low-rank compression. PETE supports the broader view that dense neural parameter matrices can contain more degrees of freedom than are necessary for some downstream tasks. Classical low-rank compression constrains a matrix through an explicit product such as $E \approx AB$. PETE instead constrains the embedding table through a shared row-generating function. This makes PETE a functional alternative to explicit low-rank embedding compression rather than a direct matrix-rank method.

Parameter efficiency. The results indicate that PETE is well-suited to the specific setting studied here: compact sentence encoders trained for semantic similarity. By enforcing a shared functional mapping from token IDs to embeddings, PETE substantially reduces parameter count while preserving much of the performance of learned lookup embeddings. This suggests that the embedding layer can be a useful target for stronger structural priors in sentence-level models.

Embedding structure. PETE introduces a strong structural prior by generating token representations from a normalized scalar coordinate through fixed Fourier features and a shared MLP. This prior reduces memory, but it also restricts expressivity. The method is therefore best understood as a structured regularizer for sentence encoders, not as a claim that token semantics are intrinsically one-dimensional.

Limitations of the empirical study. The present study is intentionally limited. We do not compare against parameter-matched low-rank embedding factorizations, hash embeddings, product quantization, or ALBERT-style factorized embeddings. We also do not evaluate robustness to randomized token ID permutations or multilingual tokenization. These experiments are important for determining whether PETE's gains come from the tokenizer's frequency-induced index structure, the Fourier basis, or the shared MLP itself. Our results should therefore be interpreted as evidence that functional embedding parameterizations are promising, not as a definitive comparison against all embedding compression methods.

PETE may also be less suitable for tasks requiring precise token-level lexical identity, such as named entity recognition, syntactic tagging, or autoregressive generation. The enforced smoothness and parameter sharing across token IDs may remove information that such tasks require. The behavior of PETE under byte-level, character-level, multilingual, or morphologically rich tokenization schemes remains an open question.

Finally, while Fourier features provide a simple and parameter-free basis, we do not claim they are optimal. Alternative fixed or learned bases may yield better trade-offs between parameter efficiency and representational flexibility.

7. Conclusion

PETE explores a simple alternative to learned embedding tables: generating token embeddings through a shared Fourier-feature parameterization. In controlled sentence-embedding experiments, this functional factorization reduces parameter count substantially while preserving much of the performance of learned lookup embeddings. The results do not establish PETE as a universal embedding replacement, but they suggest that functional row generation is a promising design point for compact sentence encoders and a useful complement to explicit low-rank embedding compression.

Impact Statement

This paper presents work whose goal is to improve the parameter efficiency of machine learning models. More compact sentence encoders may reduce memory usage and energy costs during deployment, especially on resource-constrained devices. PETE does not introduce new data collection procedures or user-facing capabilities beyond those of standard sentence representation models. Its broader societal impacts are therefore expected to be similar to those of existing embedding, retrieval, and sentence representation systems.

References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In Màrquez, L., Callison-Burch, C., and Su, J. (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075/.

Cai, B., Xiang, Y., Gao, L., Wu, D., Zhang, H., Jin, J., and Luan, T. From wide to deep: Dimension lifting network for parameter-efficient knowledge graph embedding, 2024. URL https://arxiv.org/abs/2303.12816.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S. M., Cer, D., and Jurgens, D. (eds.), *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancou-

ver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001/.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations, 2020. URL https://arxiv.org/abs/1909.11942.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Shazeer, N. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Q-bert: Hessian based ultra low precision quantization of bert, 2019. URL https://arxiv.org/abs/1909.05840.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Svenstrup, D., Hansen, J. M., and Winther, O. Hash embeddings for efficient word representations, 2017. URL https://arxiv.org/abs/1709.03933.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In Walker, M., Ji, H., and Stent, A. (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Xu, M., Xu, Y. L., and Mandic, D. P. Tensorgpt: Efficient compression of large language models based on tensor-train decomposition, 2024. URL https://arxiv.org/abs/2307.00526.

Yan, B., Wang, P., Zhang, K., Lin, W., Lee, K.-C., Xu, J., and Zheng, B. Learning effective and efficient embedding via an adaptively-masked twins-based layer, 2021. URL https://arxiv.org/abs/2108.11513.

Zhang, B. and Sennrich, R. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.