Emergent Semantics Beyond Token Embeddings: Transformer LMs with Frozen Visual Unicode Representations

Anonymous authors
Paper under double-blind review

Abstract

Understanding the locus of semantic representation in large language models (LLMs) is crucial for interpretability and architectural innovation. The dominant paradigm posits that trainable input embeddings serve as foundational "meaning vectors." This paper challenges that view. We construct Transformer models where the embedding layer is entirely frozen, with vectors derived not from data, but from the visual structure of Unicode glyphs. These non-semantic, precomputed visual embeddings are fixed throughout training. Our method is compatible with any tokenizer, including a novel Unicode-centric tokenizer we introduce to ensure universal text coverage. Despite the absence of trainable, semantically initialized embeddings, our models converge, generate coherent text, and, critically, outperform architecturally identical models with trainable embeddings on the MMLU reasoning benchmark. We attribute this to "representational interference" in conventional models, where the embedding layer is burdened with learning both structural and semantic features. Our results indicate that high-level semantics are not inherent to input embeddings but are an emergent property of the Transformer's compositional architecture and data scale. This reframes the role of embeddings from meaning containers to structural primitives. We release all code and models to foster further research.

1 INTRODUCTION

Understanding where and how semantic abstractions arise in transformer language models is both a theoretical and practical concern for the development of scalable, robust, and interpretable AI systems. Traditionally, input token embeddings—learned representations from large corpora—are viewed as the primary locus of meaning: their algebraic properties have been cited as evidence of encodable semantic relationships ("king - man + woman = queen"), and advancing architectures routinely attribute model improvements to smarter or larger embedding matrices. This paradigm has shaped assumptions about what is required for "intelligent" representation learning.

The robustness of modern Large Language Models (LLMs) to orthographic variations, such as in 'I can wRiTe', highlights a fundamental question. The model's understanding is not derived from a single, semantically rich token for "write". Instead, it often relies on composing meaning from a sequence of semantically poor characteror subword-level tokens (e.g., 'w', 'R', 'i', 'T', 'e'). If the input vectors for these atomic units lack inherent semantic content, where does the model's understanding originate? This paper argues and empirically demonstrates that semantics are not a property of input embeddings but an emergent phenomenon of the Transformer architecture itself.

However, recent advances in byte and character-level modeling, as well as modular and multi-lingual architectures, suggest that the relationship between embeddings and high-level meaning is far subtler. In pursuit of a deeper understanding, we investigate the extreme case: Transformer language models where the input embedding layer is never trained and, in fact, deliberately devoid of semantic optimization. Instead, we fix this layer to a deterministic mapping based on visual Unicode glyph representations and token-level n-gram image composites.

Our experimental framework draws embeddings not from data-driven optimization, but from precomputed character images spanning the entire Unicode range. For multicharacter tokens (as in byte-pair and unigram models), we employ compositional image techniques to generate a fixed-length visual feature vector per token, achieving universal text coverage. The tokenizer is similarly Unicode-centric, balancing character-level granularity with multi-script n-gram tokens to maximize overlap across languages.

Instead of pursuing state-of-the-art benchmark scores, which often depend on massive scale, our primary goal is to isolate and test a fundamental hypothesis: can language abstraction emerge without trainable, semantically-rich input embeddings? We deliberately constrain model and dataset sizes to create a controlled environment for comparing the learning dynamics of frozen- versus trainable-embedding models. Our focus is on the conditions under which meaning emerges, not on outperforming large-scale systems.

Critically, we observe that despite this radical departure from conventional practice, these "frozen-embedding" models not only converge but display robust convergence and, surprisingly, outperform architecturally identical counterparts with trainable embeddings on reasoning benchmarks like MMLU, suggesting representational interference in conventional approaches. Emergent abstraction appears within the transformer blocks themselves, not in the input vectors. This has profound implications: it implies that semantic structure is an architectural phenomenon, not an initialization artifact. Our main contributions are:

- demonstrate that semantic understanding in Transformers can emerge without trainable input embeddings
- introduce a universal visual embedding scheme compatible with any tokenizer
- identify "representational interference" as a potential limitation of trainable embeddings
- release open-source implementations enabling reproducibility

The rest of this paper is organized as follows: Section II reviews related work; Section III details our method for constructing visual embeddings and Unicode tokenization; Section IV describes the experimental setup; Section V presents results and analysis; Section VI discusses implications for future architectures; and Section VII concludes. Code and all artifacts are released to foster further progress in both science and industry[1]

2 RELATED WORK

Semantic representation in neural language models has received extensive attention. Early word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) models established the conceptual centrality of "meaningful" dense vector spaces, with well-documented algebraic properties in vector arithmetic. Subsequent transformer-based LMs (Radford et al., 2019) extended this paradigm to context-dependent embeddings, further entrenching the notion that trainable input vectors are the source of semantic capability. However, several lines of research challenge this view.

Character - and byte-level models, such as ByT5 (Xue et al., 2022) and CANINE (Clark et al., 2022), forego word-level tokenization for universal coverage; yet, even here, embedding matrices are trainable and morph into semantic vector spaces during pretraining. Cross-lingual and modular approaches pursue shared embedding spaces for transfer (Conneau et al., 2018; Lample & Conneau, 2019), but rely on substantial alignment learning. Mixture-of-experts (Shazeer et al., 2017; Fedus et al., 2022) and adapter fusion (Pfeiffer et al., 2020) aim to modularize reasoning, yet remain dependent on trainable embeddings. Our input embedding matrix is never trained and strictly visual/surface oriented.

More radically, recent work in visual-language models (Tan & Bansal, 2022) investigates multimodal representations by grounding language in visual contexts. Models like CLIP (Radford et al., 2021) or Flamingo (Alayrac et al., 2022) learn joint embeddings for text and images. However, these approaches typically combine separate, powerful encoders for each modality and aim to align two rich semantic streams, rather than enforcing a non-semantic, frozen vector space for language as we propose.

[1]Our code and models are provided in the supplementary materials.

Our work differs in two key aspects: We forego data-driven embedding learning altogether, initializing the input layer with fixed, non-semantic, visual features derived from Unicode glyphs. Our primary contribution is demonstrating that high-level abstraction emerges even when the input layer is restricted to fixed, non-semantic, visual features, isolating the compositional layers as the locus of meaning.

3 METHOD

3.1 FROZEN VISUAL UNICODE EMBEDDINGS

Our method generates a fixed embedding matrix E of shape (V, d_model), where V is the vocabulary size and d_model is the model's hidden dimension. The vector for each token is derived through a deterministic, multi-step process:

Glyph Rendering: Each token in the vocabulary is rendered into a bitmap. Single-character tokens are rendered directly. Multi-character n-gram tokens are rendered by horizontally concatenating the glyphs of their constituent characters. For example, the token "ing" is rendered as a single image containing the sequence of 'i', 'n', 'g' glyphs. We use a standardized font (e.g., unifont-14.0.01) at a fixed-point size.

Image Standardization: The resulting variable-width bitmaps are resized to a fixed square resolution (e.g., 32x32 for d_model=1024) using bilinear interpolation. This creates a standardized visual representation for every token.

Vectorization: Each standardized HxH bitmap is flattened into a H^2-dimensional raw vector (binary components).

Projection: To project the high-dimensional image vector into the model's latent space, we apply Principal Component Analysis (PCA). A PCA model is fit on the raw vectors of the entire vocabulary, and the top d_model principal components are used to transform each raw vector into a d_model-dimensional embedding. This preserves the maximum variance of the visual features within the target dimensionality (floating points components). PCA was chosen for its determinism, computational efficiency, and ability to capture dominant visual variance in a fixed-dimensional space without introducing trainable parameters. This aligns with our core hypothesis of investigating emergent semantics absent of any data-driven feature learning at the input layer. While a learned visual encoder like a CNN could potentially extract richer features, it would re-introduce trainable parameters, confounding our experimental goal of isolating the Transformer's compositional layers.

Normalization and Freezing: The resulting embedding vectors are L2-normalized. The final matrix E is loaded into the Transformer's embedding layer, and its weights are frozen (i.e., requires_grad=False) for all subsequent training stages.

The approach supports tokenizers from arbitrary LMs (e.g., Mistral Nemo tokens text used in bvv241-nemo tokenizer), as new tokens are mapped via their textual form.

3.2 UNICODE-CENTRIC TOKENIZER CONSTRUCTION

To test our hypothesis across diverse languages and existing token schemes, we required a method to generate visual embeddings for any given vocabulary. We developed a Unicode-centric tokenization framework, bvv241, to serve this purpose.

The core approach is based on bijective mapping of Unicode codepoints to individual tokens, whenever possible, ensuring a 1:1 relationship between characters and tokens across a wide range of scripts. Reserved, surrogate, and private Unicode ranges are systematically exploited to store custom and multi-character tokens, allowing for efficient representation of commonly occurring substrings and entire words—particularly beneficial for high-frequency terms shared across models and languages.

The tokenizer's hybrid design draws from frequent multi-character patterns identified in top-performing, state-of-the-art models: approximately 19,000 such tokens are incorporated. This structure enables robust handling of diverse languages and scripts, mitigating the fragmentation common in byte-pair encoding (BPE) and unigram tokenizers—especially for underrepresented scripts and code-mixed text.

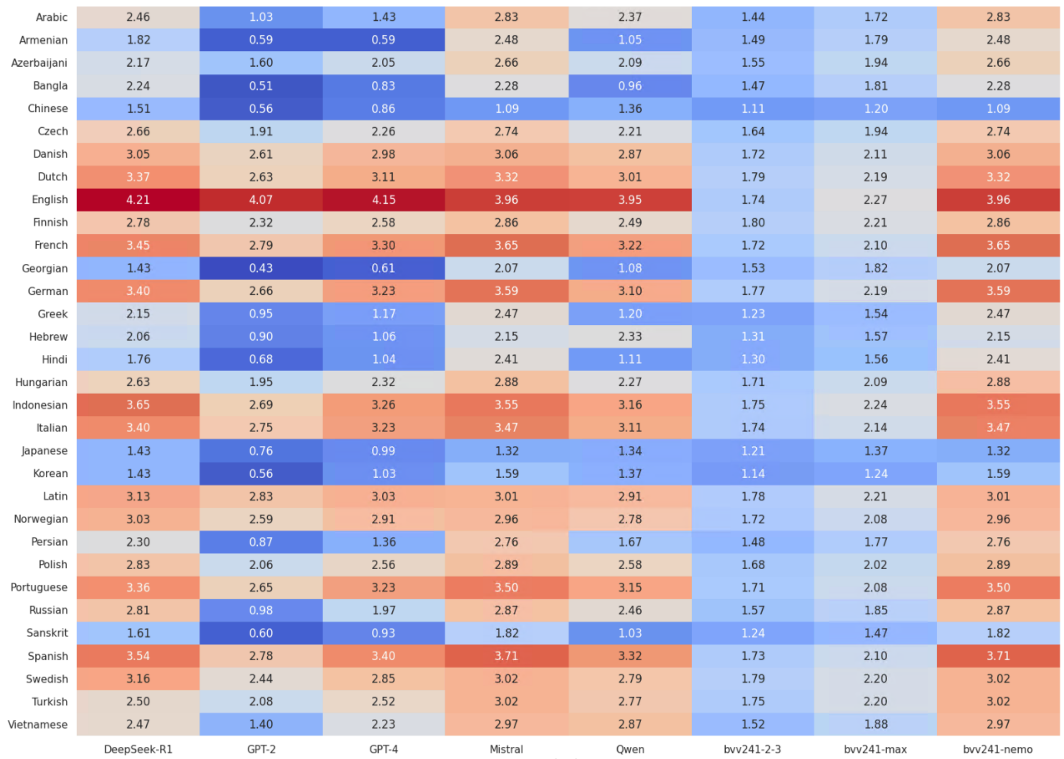

Figure 1: Average characters per token. Lower values indicate lower compression efficiency. Our bvv241 variants prioritize character-level granularity.

The bvv241-nemo variant is provided as a demonstration of the flexibility of the proposed methodology. While the underlying Unicode-centric mapping scheme is applicable to any tokenizer, bvv241-nemo specifically replicates the vocabulary, token structure, and exact token enumeration of the SOTA Mistral Nemo tokenizer, using its public vocabulary files as a template. This showcases how the approach can serve as a drop-in replacement or augmentation for existing BPE or unigram schemes.

Furthermore, since all token indices and their corresponding Unicode/codepoint representations are deterministic and exhaustively specified, this methodology enables the direct creation of a pre-initialized embedding matrix of shape vocab_size × n_embed. Such a matrix can be either used as a fixed (non-trainable) embedding or as a high-quality cold start, reducing the need for extensive embedding pre-training. Combining strict Unicode mapping with multi-character tokens, the bvv241 family achieves highly stable performance for Asian, Arabic, and underrepresented scripts—where differences from SOTA are minimal. For Indo-European (Latin and partially Cyrillic) languages, compression is less aggressive, but the model gains in predictability, recoverability, and linguistic inclusivity.

3.3 MODEL ARCHITECTURE

We employ a standard decoder-only Transformer architecture, similar to GPT-2. All models use a hidden dimension d_model=1024, multi-head attention, and GELU activations. The key modification is that the token embedding layer (nn.Embedding) has its weights frozen to the precomputed visual matrix. For comparison, we train baseline models with identical architectures but with a standard, randomly initialized and trainable embedding layer.

Table 1: Model Parameters and Vocabulary Statistics. Overview of models used in experiments. 'Frozen' models use our fixed visual embeddings. 'Trainable' models are baselines with standard, randomly initialized, trainable embeddings. All other architectural parameters (d_model=1024, layers, etc.) are held constant within each size class (0.2B, 0.4B, 0.5B) between frozen and trainable embeddings mode. The prefixes 'pro', 'max', and 'best' correspond to models with 0.2B, 0.4B, and 0.5B parameters, respectively.

Model Name	**Parameters and Vocab Size**	**Training Corpus and Embedding Type**
pro_bvv_en	0.2 B, 65536	English, Frozen
pro_bvv_unfrozen	0.2 B, 65536	English, Trainable
pro_bvv_ru	0.2 B, 65536	EN+Russian, Frozen
pro_bvv_zh	0.2 B, 65536	EN+Chinese, Frozen
best_bvv_ru	0.5 B, 131072	EN+Russian, Frozen
best_bvv_unfrozen_ru	0.5 B, 131072	EN+Russian, Trainable
best_bvv_zh	0.5 B, 131072	EN+Chinese, Frozen
best_bvv_unfrozen_zh	0.5 B, 131072	EN+Chinese, Trainable
max_bvv_ru	0.4 B, 131072	EN+Russian, Frozen
max_bvv_zh	0.4 B, 131072	EN+Chinese, Frozen
nemo_bvv_ru	0.4 B, 131072	EN+Russian, Frozen
nemo_bvv_zh	0.4 B, 131072	EN+Chinese, Frozen

4 EXPERIMENTAL SETUP

4.1 TRAINING DATASETS

The training corpus comprises a multilingual mix including subsets of Wikipedia (en, ru, zh), and SFT datasets (10% insertion in pretrain) SQuAD-v2, TriviaQA, HotpotQA, NQ-Open, BoolQ, CommonsenseQA, and ARC, totalling approximately 9B tokens. Wikipedia data constitutes roughly 5% of its total size.

4.2 BASELINES AND COMPARISONS

Baselines include classic transformer LMs with trainable embedding layers matched for architecture, parameter count, and tokenizer coverage. Our primary comparisons are between models with 'Frozen' embeddings and their 'Trainable' counterparts (e.g., 'best_bvv_ru' vs. 'best_bvv_unfrozen_ru'), which share identical architecture and training data.

4.3 EVALUATION

We report standard language model metrics:

- Train and validation loss/perplexity,
- MMLU,
- ARC and CommonsenseQA.

All metrics are computed on held-out test sets.

4.4 IMPLEMENTATION DETAILS

Training was performed on H100 80 Gb with batch sizes up to 16 and block sizes of 1024. Gradient accumulation and mixed precision were used for efficiency.

5 RESULTS

Before presenting our results, it is crucial to frame their context. This study does not aim to achieve state-of-the-art performance on benchmarks like MMLU. Our models are intentionally constrained in scale (up to 0.5B parameters) and trained on a modest 9B token dataset. The primary objective is to demonstrate the feasibility of learning with non-semantic embeddings and to compare the learning dynamics between our proposed frozen-embedding models and traditional trainable-embedding baselines under identical conditions. The following metrics should therefore be interpreted as evidence of emergent abstraction, not as a challenge to large-scale models.

5.1 MODEL CONVERGENCE

A primary concern was whether a model with non-semantic, frozen embeddings could converge at all. Figure 2 shows the training loss curves for a 0.5B parameter frozen-embedding model (best_bvv_ru) and Figure 3 shows its trainable-embedding counterpart (best_bvv_unfrozen_ru). Both models exhibit stable convergence profiles, with nearly identical loss trajectories. This demonstrates that the Transformer architecture is capable of learning effectively even when the input layer provides only fixed structural information.

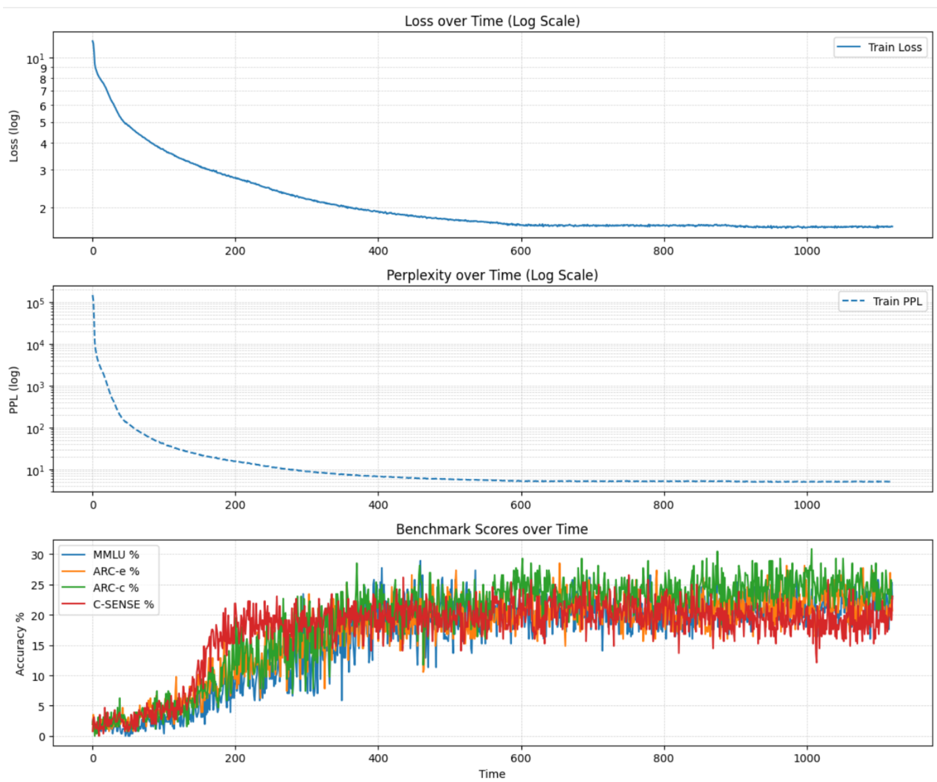

Figure 2: Learning curves (best_bvv_ru) for frozen embedding.

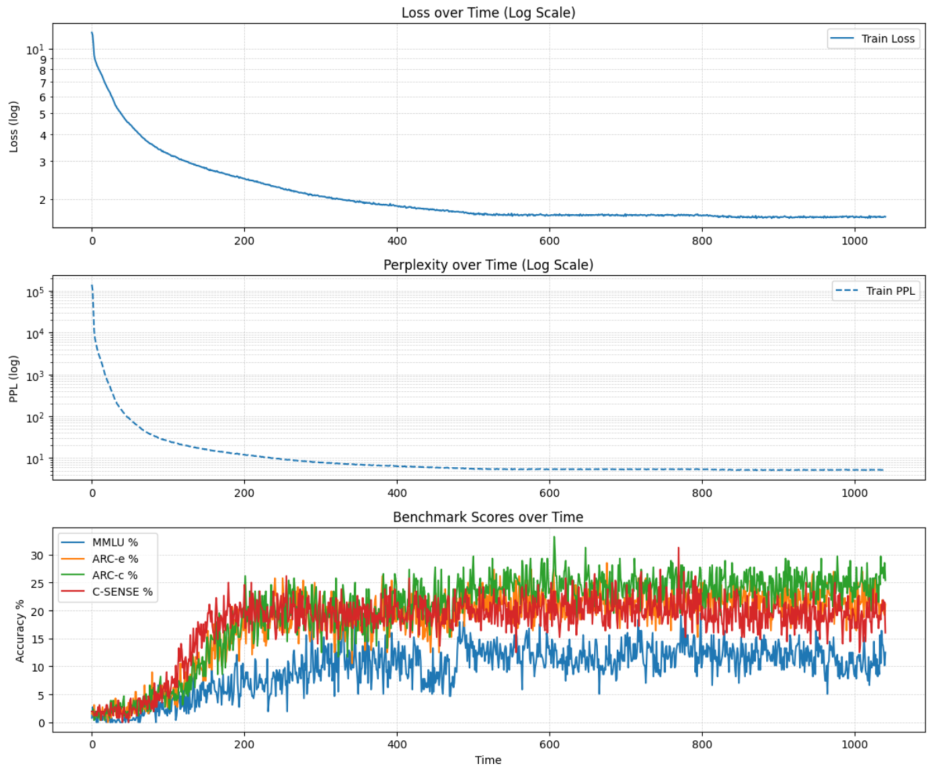

Figure 3: Learning curves (best_bvv_unfrozen_ru) for trainable embedding.

5.2 PERFORMANCE OF FROZEN VS. TRAINABLE EMBEDDING MODELS

Our primary finding challenges the conventional wisdom that trainable, semantically-rich embeddings are necessary for high-level reasoning. As shown in Figure 4 and Figure 5, while our frozen-embedding models demonstrate robust convergence across all tasks, they show a striking and consistent performance advantage on the MMLU reasoning benchmark compared to their architecturally identical counterparts with trainable embeddings.

For instance, best_bvv_ru (0.5B params, frozen embeddings) achieves an MMLU score of 22.29, whereas best_bvv_unfrozen_ru (0.5B params, trainable embeddings) scores only 11.37 — a nearly 2x performance gap. This pattern holds across different model sizes and languages.

We hypothesize this phenomenon, which we term "representational interference," stems from the dual burden placed on trainable embeddings. A trainable embedding layer must simultaneously learn low-level orthographic features (e.g., character shapes, token boundaries) and high-level semantic abstractions within the same parameter space. This creates a conflict during optimization, leading to a suboptimal representation that is neither a perfect structural encoder nor a pure semantic vector.

In contrast, our frozen visual embeddings provide a stable, information-rich structural foundation. By offloading the task of representing token form to a fixed, deterministic layer, we free the Transformer's attention and feed-forward layers to focus exclusively on their core competency: composing these structural primitives into high-level meaning. The model does not waste capacity learning what a token looks like; it only learns what to do with it.

This hypothesis is further supported by the t-SNE visualizations of the embedding spaces below.

Figure 4: Performance Comparison: MMLU sub tasks with score greater than 27%.

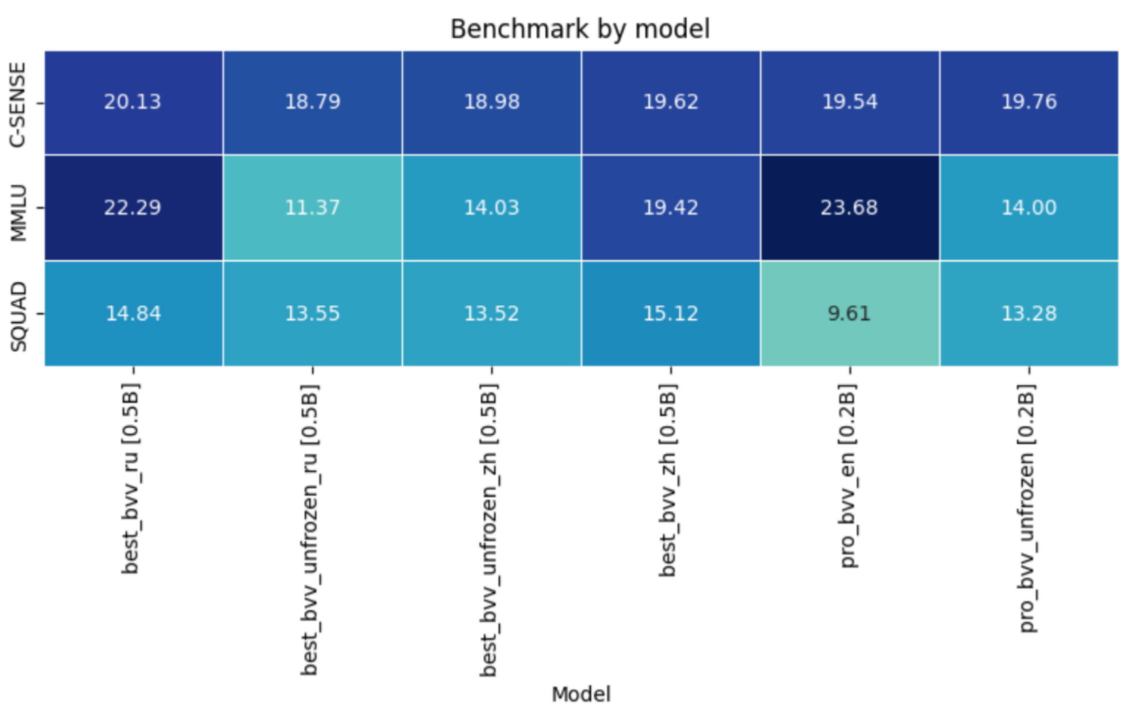

Figure 5: Performance Comparison: Frozen vs. Trainable Embedding Models.

5.3 EMERGENT SEMANTICS VISUALIZATION OF INPUT TOKEN EMBEDDINGS

Our fixed visual embedding vectors naturally cluster by surface/formal features (e.g., length), rather than by meaning (Figure 7). In contrast, trainable embeddings demonstrate soft semantic grouping, though the distributed latent space remains globally unstructured (Figure 6).

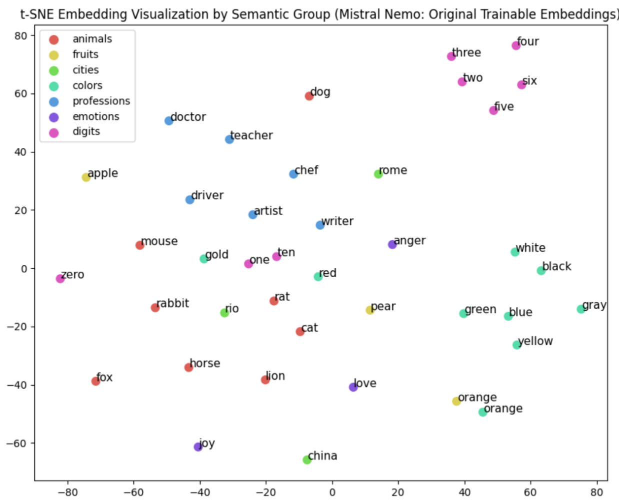

Figure 6: t-SNE visualization of input token embeddings. In the trainable Mistral Nemo embeddings, semantic groups (e.g., numbers, professions, animals) tend to lie closer to one another, yet, globally, the point cloud remains mostly uniform.

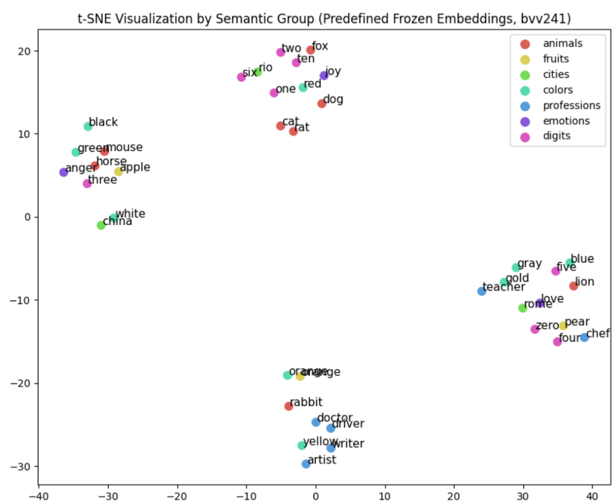

Figure 7: t-SNE visualization of the frozen visual embeddings. The prominent clustering directly corresponds to token length. This phenomenon is a direct artifact of our embedding generation process: multi-character tokens, when rendered and resized to a fixed-resolution bitmap, produce images with higher average pixel density. As Principal Component Analysis (PCA) is designed to capture axes of maximum variance, it naturally identifies this structural feature as a dominant component. The resulting clusters are therefore based on a physical token property (length/density), not semantic content, reinforcing the conclusion that semantic understanding must emerge entirely within the subsequent Transformer layers.

This illustrates that any emergence of true semantic structure must occur entirely within the network's compositional layers.

5.4 ABLATION STUDY: THE ROLE OF EMBEDDING INITIALIZATION

This ablation study compares the performance of traditional trainable embeddings, frozen embeddings initialized with Unicode-based visual representations, and frozen embeddings initialized with random black-and-white bitmaps, which serve as a randomized baseline.

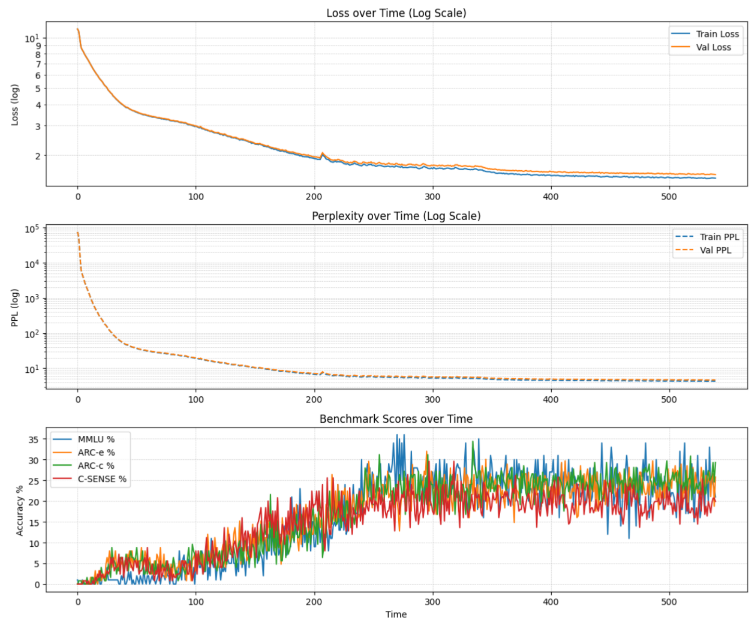

Figure 8: Learning curves for the pro_bvv_en model using a frozen nn.Embedding layer.

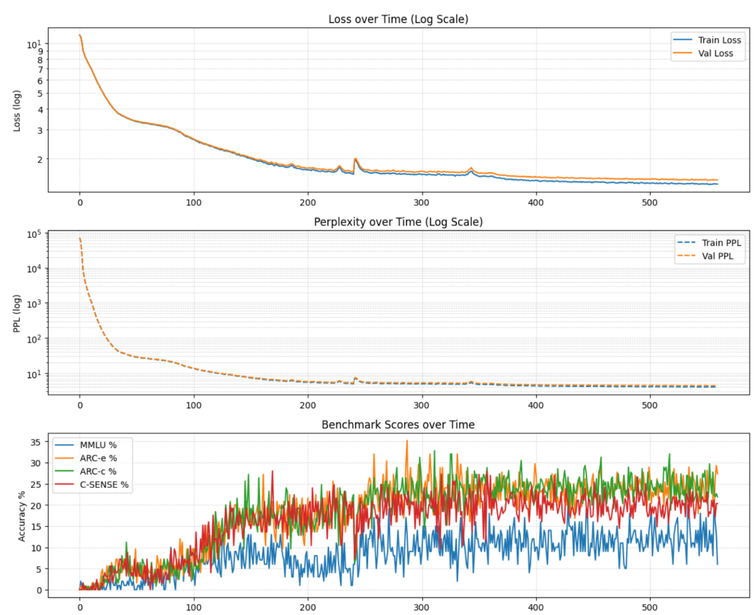

Figure 9: Learning curves for the pro_bvv_unfrozen model using a standard trainable nn.Embedding layer.

This ablation study reveals a critical insight: even when frozen embeddings are initialized from random noise, the model still converges, albeit at a significantly slower rate (Figure 10). This finding suggests that the semantic structures observed in trainable embeddings are not a prerequisite for learning, but rather an

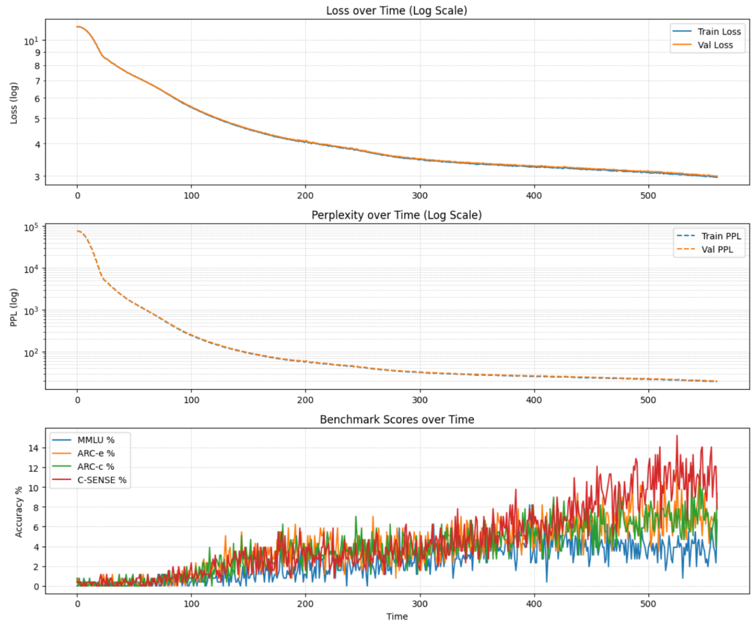

Figure 10: Learning curves for the pro_bvv_random model using a frozen nn.Embedding layer where, instead of visual Unicode representations of tokens, images with random black-and-white pixel noise were used. Unexpectedly, we observe convergence during training, but with a 5 to 10 times slowdown.

emergent artifact. We hypothesize that this phenomenon stems from a trainable embedding layer being implicitly forced to solve two conflicting tasks: 1) Token Discrimination, the primary goal of assigning a unique vector to each token, and 2) Semantic Proximity, the secondary goal of clustering vectors for contextually similar tokens. These objectives are often at odds. In essence, during backpropagation, gradients from high-level contextual errors "imprint" semantic relationships onto the embedding vectors, forcing them to serve as a low-dimensional projection surface for knowledge learned by the entire network. Our proposed frozen-embedding approach resolves this conflict by decoupling the tasks. By providing a stable, unique identifier for each token, the fixed vectors (whether visually or randomly initialized) perfectly satisfy the discrimination objective. This liberates the Transformer's compositional layers—the attention and feed-forward networks—to exclusively manage the learning of semantic relationships. We posit that this separation of concerns reduces optimization conflicts and allows the architecture to learn more efficiently, as evidenced by the superior performance on reasoning benchmarks.

6 DISCUSSION

Our findings suggest that the role of input embeddings in Transformers may be widely misunderstood. Meaning appears not to be a property of the embeddings, but a process enacted by the architecture.

6.1 RE-EVALUATING EMBEDDINGS: FROM MEANING CONTAINERS TO STRUCTURAL PRIMITIVES

The success of our frozen-embedding models, particularly on MMLU, challenges the view of embeddings as "meaning vectors." Instead, they can be seen as "structural primitives." The "representational interference" hypothesis suggests a fundamental inefficiency in standard LLMs. By tasking the embedding layer with both structural and semantic learning, we may be creating an optimization bottleneck that harms downstream reasoning. Forcing the model to learn, for instance, that the token "king" is semantically close to "queen" but structurally dissimilar from "monarch" within the same vector space is a difficult, potentially conflicting task.

Our method resolves this by offloading the structural representation to a fixed, deterministic layer. The model doesn't waste capacity learning what a token looks like; it only learns what to do with its form in context. This may be particularly advantageous for tasks requiring orthographic or morphological awareness, which could be implicitly tested in MMLU's diverse sub-tasks.

By contrast, our frozen visual embeddings resolve this conflict. They provide a stable, information-rich, and purely structural foundation. The model does not expend capacity learning what a token looks like; it only learns what to do with it. This frees the Transformer's attention and MLP layers to focus exclusively on their core competency: composing these structural primitives into high-level meaning.

This suggests a hidden inefficiency in standard models. Consider a task requiring orthographic reasoning (e.g., "How many 'e's in 'strawberries'?"). If "strawberries" is a single token, its semantic embedding contains no direct information about its character composition. The model must perform a costly "internal detokenization," using its deep layers to implicitly reconstruct the word's structure from memory. Our approach mitigates this by providing structural information at the input layer, potentially making the model more computationally efficient for a class of reasoning tasks. The performance gap on MMLU suggests that decoupling structural representation from semantic composition is a more effective learning strategy, especially under constrained data and compute.

6.2 IMPLICATIONS FOR ARCHITECTURE AND EFFICIENCY

This paradigm opens several avenues for future architectures:

- **Modular and Standardized Embeddings:** A universal, algorithmically-derived visual embedding matrix could become a standard component, allowing researchers to swap out different Transformer backbones for direct comparison without confounding factors from different embedding initializations.
- **Efficient MoE and Model Fusion:** Mixture-of-Experts (MoE) models could share a common frozen embedding layer, reducing parameter count and simplifying routing logic.
- **Improved Cross-Lingual Transfer:** A single Unicode-based visual embedding space is inherently multilingual. This could provide a more robust foundation for zero-shot cross-lingual transfer, as the model learns to operate on visual script features common to many languages.

6.3 LIMITATIONS AND FUTURE WORK

This study was intentionally constrained in scale. Future work should test whether these findings hold for models at the 10B+ parameter scale. Additionally, while our method covers the entire Unicode text space, its generalization to truly novel modalities (e.g., mathematical formulas, chemical structures) remains an open question.

7 CONCLUSION

We have demonstrated that transformer LMs with frozen, purely visual Unicode-based embeddings can learn rich linguistic and logical abstraction. Semantics in neural LMs is not a property of the embedding matrix, but an emergent phenomenon of the model's layered composition and self-attention. This insight opens new avenues for model standardization, modularity, and efficient scaling in multi-lingual and multi-domain NLP.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Nando de Freitas, Oriol Vinyals, and Andrew Zisserman. Flamingo: a visual language model for few-shot learning, 2022.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. CANINE: Pre-training an efficient tokenization-free encoder for language representation. In *Transactions of the Association for Computational Linguistics*, volume 10, pp. 726–745, 2022.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 3637–3650, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Blog Post 1, OpenAI, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4141–4156, 2022.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. In *Transactions of the Association for Computational Linguistics*, volume 10, pp. 291–306, 2022.