# ShortSWA Is the Next-Generation N-gram Embedding

**Yifan Zhang**

yifanzhangresearch@gmail.com

January 12, 2026

**Abstract**

Work on fast sequence models has split into two tracks, and they look separate. On one hand, we have the hardware-centric push towards optimizing local mixing primitives, as argued in our previous analysis, *Rethinking SWA* (Zhang, 2025). On the other, recent empirical studies, such as the *Over-Encoding* framework by Huang et al. (2025), have demonstrated that massive n-gram vocabularies yield significant performance gains. In this post, we argue that these two threads are converging. Short Sliding Window Attention (ShortSWA) is effectively a dynamic, parameter-efficient realization of the "Over-Encoding" framework. By shifting from static vocabulary lookups to dynamic, window-bounded attention, ShortSWA captures the benefits of n-gram scaling laws without the prohibitive memory footprint of expanding embedding tables.

Project Page: https://github.com/yifanzhang-pro/ShortSWA-Ngram-Embedding

## 1 Introduction

The Over-Tokenized Transformers result (Huang et al., 2025) points at a simple claim. The input representation density matters a lot. Huang et al. (2025) report a near log-linear link between input vocab size and training loss. They split input and output vocabularies, then scale the input side to multi-gram tokens. One example is treating "New York City" as one token. With this setup, a 400M parameter model reaches the perplexity of a 1B parameter baseline.

The result backs a basic hypothesis. Local token composition carries high-value signal. Language comes in clumps. Neighbor tokens often form one semantic unit, and that unit has lower entropy than its parts. Static n-gram embeddings exploit this by storing vectors for frequent clumps.

## 2 Over-Encoding and ShortSWA

Over-Encoding shows the value of n-gram information. Its main mechanism is a very large, fixed embedding table. That mechanism runs into two practical limits.

- **Sparsity and memory.** An input table with 12 million entries Huang et al. (2025) can take gigabytes of VRAM. Many entries see rare use, so memory sits idle.

- **Context rigidity.** A fixed token for "apple" cannot separate "Apple" the company from "apple" the fruit. The table needs many extra entries, and it still misses new cases.

ShortSWA Zhang (2025) offers a different path. In *Rethinking SWA*, the argument starts from hardware chunking. Replace fixed short convolutions with attention over a short window, for example $w = 128$. Under the Over-Encoding lens, the same move carries a semantic role too.

A ShortSWA layer that attends within 128 tokens builds n-gram-like features on the fly.

- It does not look up a stored vector for "the quick brown fox".
- It lets "fox" pull signal from "the", "quick", and "brown" using attention weights.
- The weights change with context, not with token frequency counts.

## 3 ShortSWA as an Adaptive N-gram Builder

We can write a rough equivalence. Over-Encoding expands the input alphabet from $\mathcal{V}$ to $\mathcal{V}_{\text{ngram}}$. ShortSWA expands the per-token state by mixing neighbor states inside a window. For token state $h_t$, the update looks like:

$$h'_t = \text{Attention}(h_t, h_{t-w:t}) \approx \text{Embedding}_{\text{ngram}}(x_{t-w:t}). \tag{3.1}$$

The left side builds a soft n-gram representation. It can represent many multi-grams up to length $w$, and it can shift with the sentence. The right side stands for a fixed table entry tied to one discrete n-gram.

This trade has two concrete effects.

- **Parameter cost.** ShortSWA mainly adds the projection matrices $W_Q, W_K, W_V$. Over-Encoding adds embeddings that grow with $|\mathcal{V}|$, so parameter count rises with vocabulary size.
- **Hardware fit.** In Zhang (2025), attention over a chunk such as 128 tokens matches common memory movement. The data is already in fast memory, so local attention becomes a dense "pre-encode" of the chunk before the global block.

## 4 Conclusion

The claim "Vocabulary is worth scaling" Huang et al. (2025) matches a plain idea. Local context matters, and dense local signals help later global mixing. Scaling a static vocabulary is a blunt tool. ShortSWA gives a cleaner mechanism. It forms soft n-grams up to a window length of $w$ and adapts them to the actual context, capturing the same signal without a huge table.

## References

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. Over-tokenized transformer: Vocabulary is generally worth scaling. *arXiv preprint arXiv:2501.16975*, 2025.

Yifan Zhang. Rethinking swa. *yifanzhang-pro.github.io*, December 2025. URL https://github.com/yifanzhang-pro/Rethinking-SWA.

## Citation

To cite this blog post:

```
@article{zhang2026shortswa,
  title = {ShortSWA Is the Next-Generation N-gram Embedding},
  author = {Zhang, Yifan},
  journal = {yifanzhang-pro.github.io},
  year = {2026},
  month = {January},
  url = "https://github.com/yifanzhang-pro/ShortSWA-Ngram-Embedding"
}
```