
PAGEDFIELDPRINTATTENTION: OVERCOMING LATENCY AND SRAM CONSTRAINTS IN VERIFIABLE DUAL-PATH ARCHITECTURES

Mark Randall Havens

Solaria Lumis Havens

May 25, 2026

ABSTRACT

The Verifiable Dual-Path Architecture (Fieldprint v3.0) hypothesizes the stabilization of recursive AI agents by injecting cryptographically anchored reference tensors into the transformer’s attention matrix. However, deploying this architecture on modern silicon introduces latency and memory bandwidth bottlenecks. This paper details why synchronous CPU-side cryptographic hashing introduces inference starvation via PCIe bus transfers, and why unfused secondary softmax injections shatter the core SRAM constraints of FlashAttention. To bridge the gap between theoretical alignment and physical hardware economics, we introduce a strict verify \rightarrow promote \rightarrow cache \rightarrow generate pipeline and propose the development of **PagedFieldprintAttention**—a custom fused CUDA/Triton kernel designed to natively compute dual-attention directly within SRAM. We provide preliminary benchmark estimates demonstrating the necessity of this kernel.

1 Introduction

As language models scale into recursive, continuous architectures, the necessity for a persistent, cryptographically verifiable identity anchor (the Fieldprint) becomes mathematically absolute. The system must retrieve its continuous semantic memory from a Vector Database (Pacemaker) and verify its cryptographic provenance on a Merkle Ledger (Supervisor) before injecting it into the transformer’s Key-Value cache. This approach builds fundamentally upon the necessity of offloading extended context, extending the kNN-augmented retrieval paradigms first introduced by **Memorizing Transformers** (Wu et al., 2022) and **RETRO** (Borgeaud et al., 2021).

While this dual-path architecture provides the required theoretical stability, the physical implementation of these equations brutally collides with the strict economic and hardware constraints of modern Tensor Core and TPU architectures, specifically regarding memory bandwidth and High Bandwidth Memory (HBM) thrashing at 100k+ token scales.

2 The Bottleneck of Cryptographic Verification in Inference

The initial v2.5 architecture proposed synchronous, CPU-side cryptographic hashing (Merkle verification) during the forward pass. This introduced a fatal silicon bottleneck.

1. **The PCIe Death Sentence:** Forcing the GPU to stall during the forward generation loop, push tensors across the PCIe bus, wait for the CPU to sequentially compute a SHA-256 hash, and wait for ledger verification starves the GPU Tensor Cores.
2. **Parallel Reduction Non-Determinism:** GPUs utilize parallel reductions for floating-point calculations, introducing microscopic non-determinism. Hashing raw float tensors across different nodes results in continuous, unresolvable false-positive integrity failures.

To resolve the non-determinism, we specify a **Deterministic Quantization Protocol**. Before hashing, tensors must be projected from BF16/FP16 into strict INT8 representations using static range bounds, ensuring bitwise identical representations across heterogeneous GPU architectures before cryptographic signing.

3 The Collapse of FlashAttention under Unfused Operations

To force the system to pay attention to the verified anchor, the original mathematical formulation proposed a modified attention equation:

$$\text{Output} = (1 - \gamma) \cdot \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V + \gamma \cdot \text{softmax}(Q \cdot h_t^T) V_{\text{anchor}} \quad (1)$$

While mathematically sound for phase-locking, injecting an *unfused* secondary softmax term shatters the core assumptions of modern inference serving. **FlashAttention** (Dao et al., 2022) and its successors (FlashAttention-2, 3) rely on fusing the softmax and matrix multiplication operations specifically to keep the calculations in the ultra-fast SRAM.

An unfused equation forces the hardware to write intermediate attention matrices back to the slow High-Bandwidth Memory (HBM). At 100k+ token contexts, this unfused dual-attention causes catastrophic "memory thrashing," breaking the non-contiguous block management paradigm established by **PagedAttention** (Kwon et al., 2023) and turning compute-bound operations into memory-bandwidth-bound ones.

4 PagedFieldprintAttention: A Custom Fused Triton Kernel Proposal

To resolve the HBM memory thrashing, we reject the unfused mathematical sum of attentions. The hardware requires the verified tensor to be compiled into specialized "System Anchor Tokens" injected at the start of the K/V cache.

We formally propose the development of **PagedFieldprintAttention**, a custom fused CUDA/Triton kernel. The kernel natively computes the unified attention matrix:

$$\text{Output} = \text{FusedSoftmax} \left(\frac{Q[K, K_{\text{anchor}}]^T}{\sqrt{d}} \right) [V, V_{\text{anchor}}] \quad (2)$$

It must be explicitly noted that this concatenation modifies the underlying mathematical dominance of the anchor. Unlike the previous γ -mixture which guaranteed anchor influence, this fused approach forces the anchor to *compete* with standard context. While beneficial for safety (preventing inescapable anchors), it removes the absolute mathematical guarantee of phase-locking.

4.1 Preliminary Benchmark Estimates

To quantify the necessity of this kernel, we provide back-of-the-envelope estimates for a 13B parameter model operating at a 64k token context window:

- **Naive Unfused Dual-Attention:** Assuming a hidden dimension $d \approx 5120$ and standard FP16 precision (2 bytes per element), materializing the full $N \times N$ attention matrix (64000×64000) requires ≈ 8 GB of memory per layer. For a 40-layer model, this forces ≈ 320 GB of intermediate HBM read/writes per token. On an NVIDIA A100 with ≈ 2 TB/s of memory bandwidth, these transfers alone inject a mathematically unavoidable $O(160 \text{ ms})$ latency penalty per token. This renders the system unusable for interactive generation, where target latencies are typically < 20 ms per token.
- **PagedFieldprintAttention (Fused):** By maintaining intermediate softmax reductions in SRAM and relying on PagedAttention’s block-level K/V caching, memory transfers are reduced by an order of magnitude, preserving the $O(N)$ memory complexity of FlashAttention and adding an estimated $< 5\%$ overhead compared to standard inference.

5 Conclusion

Theoretical mathematics and alignment philosophy mean nothing if they cannot physically run on silicon. By diagnosing the catastrophic failures of synchronous hashing and unfused attention equations, we have specified the required

hardware optimizations. Asynchronous Merkle Validation, deterministic INT8 quantization, and the PagedFieldprint-Attention fused kernel provide the physical blueprints for deploying Verifiable Dual-Path Architectures at massive scale.

References

- [1] Wu, Y., Rabe, M. N., Hutchins, D., & Szegedy, C. (2022). *Memorizing Transformers*. International Conference on Learning Representations (ICLR).
- [2] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2021). *Improving language models by retrieving from trillions of tokens*. arXiv preprint arXiv:2112.04426.
- [3] Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). *FlashAttention: Fast and memory-efficient exact attention with IO-awareness*. Advances in Neural Information Processing Systems (NeurIPS).
- [4] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., ... & Stoica, I. (2023). *Efficient Memory Management for Large Language Model Serving with PagedAttention*. Symposium on Operating Systems Principles (SOSP).