# RETRIEVING BY FREQUENCY, NOT POSITION: FREQUENCY-DIVISION MULTIPLEXED MEMORY FOR STRUCTURED RETRIEVAL IN LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We introduce a frequency-division multiplexed (FDM) context-encoded token stream with functional natural language coexistence in transformer models, wherein multiple independent information channels for structured data are encoded as amplitude-shift keyed (ASK) sinusoidal carriers, superposed into a single token sequence, enabling accurate, uniform forward-pass retrieval of in-context memory. Each fact targeted for storage and accurate retrieval access is a KEY=VALUE pair encoded into K discrete-valued channels as ASK sinusoidal carriers superposed as a composite signal into a fixed token sequence. Every token therefore carries information about every channel, making retrieval frequency-addressed rather than position addressed, without explicit Fourier, demodulation, or retrieval-specific modules. We finetune four transformer architectures (GPT-2 Medium, Qwen3-0.6B, LFM2.5-1.2B, Hermes3-3B) on up to 215K training samples of structured, non-linguistic FDM encoded data, achieving 93–100% per-channel retrieval accuracy across a 32-channel Context block (ch8–39). We then test whether the learned decoding is tied to a single position range. A brief additional finetuning stage teaches the FDM-finetuned model to decode the block at a second position range, and a learned key-value (K/V) prefix placed in front of the block does not degrade retrieval. We then finetune under a five-stage curriculum of decreasing carrier amplitude contrast, interleaving general-text replay where ordinary English text is mixed into every batch (annealed from 30% to 10%) specifically to guard the model from losing its natural-language abilities while it learns to read the FDM-encoded channels. This is performed on a 10-channel mixed-domain configuration, wherein all four architectures sustain 96.9–97.5% FDM action accuracy and 99.9–100% per-channel retrieval while answering 8–10 of 10 held-out natural-language questions correctly (10/10 for Qwen3-0.6B), with the same forward pass and no routing or gating mechanism handling both input types. These results show that transformer attention can learn to retrieve from a frequency-encoded in-context memory substrate while basic natural-language question answering remains functional. These results suggest a new design space for structured, frequency-addressed memory substrates that support short, accurate retrieval of structured facts within a single forward pass.

## 1 INTRODUCTION

Large language models exhibit a well documented pattern of strong performance on single fact retrieval that degrades sharply as facts must be jointly composed (Press et al., 2023; Biran et al., 2024; Yang et al., 2024). This is typically attributed to a model's poor reasoning performance, but may more accurately be attributed to a shared bottleneck between retrieval and reasoning that both compete for the same attention based resources (Wu et al., 2025b). Retrieval has its own distinct challenges despite this. Even when facts are provided explicitly in-context, as retrieval accuracy depends on position, facts in the middle of long contexts retrieve less reliably (Liu et al., 2024). This degradation persists even when position is held fixed relative to context length, with performance still degrading as length alone increases (Modarressi et al., 2025). Simple retrieval tasks with no composition requirements also degrade sharply with increasing context length (Hsieh et al., 2024).



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Longer contexts also enlarge the KV cache streamed at every decoding step which adds to the penalty (Fu, 2024). Attempts to alleviate these problems with sparse and latent attention (Yuan et al., 2025; DeepSeek-AI, 2025), KV-cache compression and trained caches (Eyuboglu et al., 2026) reduce this cost, but leave this positional coupling of facts intact. The problem extends to agent chat assistants with long-horizon pipelines (Packer et al., 2023) reporting a 30% accuracy drop for retrieving information across sustained multi-session interactions (Wu et al., 2025a). Many applications however require accurate, deterministic retrieval with no margin for error. Though very powerful models with good enough retrieval capabilities may be acceptable in certain cases, in others such as in the financial and insurance sectors, this is considered unacceptably unreliable, requiring the introduction of more deterministic solutions. An encoding that breaks the coupling between fact and position may serve retrieval applications that need exact accuracy guarantees and may also propose new design spaces for memory in language models.

We propose frequency-division multiplexed (FDM) encoding using amplitude-shift keying (ASK) for in-context memory substrate design. To our knowledge, this is the first spectrum based technique for constructing memory substrates that are frequency addressed for retrieval, designed around carrier orthogonality and frequency based channel separability. FDM allows for fact addressing using frequency multiplexing, and we show that 40 independent channels can be multiplexed into a fixed token budget, each carrying a separately stored short fact that four architectures recover at 93–100% per-channel accuracy, with errors concentrated at the highest carrier frequencies. The theoretical capacity limits are also calculated, showing scaling potential. We further show that, after brief finetuning, the host decodes the FDM block at a second position range, and that a learned K/V prefix placed in front of the block does not degrade retrieval relative to a no-prefix control. We finally show preliminary evidence that for the in-context FDM encoded memory substrate, a model can be further finetuned so that it still answers plain English questions.

## 2 METHOD

Four architectures (GPT-2 Medium, Qwen3-0.6B, LFM2.5-1.2B, Hermes3-3B) are used in this study with four stages to the experimental procedure: Step (i) defines the $K$-channel ASK carrier encoding and generates the corresponding FDM encoded dataset, with $K = 40$ channels throughout, unless otherwise noted (§2.1); Step (ii) teaches single position retrieval with a single-block (one contiguous FDM position range) finetuning on 115K examples followed by a compositional stage on 100K hop-structured examples (1–5 hops) chaining dependent channel reads (§2.1.3; §2.1.4; Table 9, Appendix A); Step (iii) tests tolerance to a learned K/V prefix in two steps: first, continuing from the single-block checkpoints, the host is finetuned to decode the FDM block at a shifted position range (§2.1.5); second, a separate write head is trained to produce a K/V prefix at the original positions, placed in front of that block (§3.2); Step (iv) tests whether FDM retrieval can coexist with ordinary language capability on a shared checkpoint (§2.1.6).

Five evaluation scores are defined in this study. ACTION checks whether the query's final answer keyword, drawn from the last channel in the queried chain, is correct. RULE and META check retrieval of ch6 and ch7. For ch6, ground truth only includes RULE in the answer when META=NONE, so for the two templates that chain META to RULE (4- and 5-hop), we report RULE both with and without the META=NONE case (Table 2). FACT is the per-channel accuracy over the channels the queried chain actually consulted, one channel for a 1-hop query, up to five for a 5-hop query. This tests whether the underlying retrieval of each individual channel involved was accurate. EXTRA is the per-channel accuracy over the full 32-channel Context block (ch8–39, formatted as NAME=VALUE pairs appended after the chained answer), scored on every sample regardless of which channels were queried. Since most of these channels are never the target of the question, correct retrieval shows the model decodes the whole composite signal rather than just locating one answer. The five metrics overlap in that a single channel can count toward both FACT and one of RULE/META/EXTRA in the same sample. Correctness is scored by binary substring match asking, does the expected value appear anywhere in the response which is resolved as a binary present or absent check, never a fractional or partial score. The match target differs by metric; however, and so does its strictness. EXTRA requires the full NAME=VALUE string to appear verbatim; FACT and RULE/META require only the bare channel value to appear; ACTION requires only that one of a fixed set of decision keywords appear somewhere in the response, a looser target than the other four metrics' specific-value checks. Finally, in the prefix-tolerance experiment (§3.2; Table 6), we compare

*slot accuracy*, which is the mean per-channel retrieval accuracy, alongside *joint accuracy*, which is the fraction of samples where all 32 channels are correct at once.

### 2.1 FDM ENCODING AND DECODING FOR STRUCTURED FACTS

We define a memory substrate for structured facts as a set of $K$ channels, each carrying a discrete value $v_k \in \{0, 1, \ldots, M_k - 1\}$, where $M_k$ is the size of channel $k$'s vocabulary $\mathcal{V}_k$; each vocabulary is fixed and ordered, so $v_k$ denotes the index of the selected category within $\mathcal{V}_k$ (e.g., for channel $k =$ LOCATION with $\mathcal{V}_k = \{\text{PARIS}, \text{TOKYO}, \text{LONDON}, \text{BERLIN}\}$, $M_k = 4$ and $v_k = 1$, which denotes TOKYO, i.e., $\mathcal{V}_k[1] =$ TOKYO). Each channel value is encoded onto a distinct sinusoidal carrier at frequency $f_k = k$ Hz using amplitude-shift keying (ASK):

$$s_k(t) = A(v_k) \cdot \sin(2\pi f_k t), \quad A(v_k) = A_{\text{low}} + (A_{\text{high}} - A_{\text{low}}) \frac{v_k}{M_k - 1}. \tag{1}$$

Once $v_k$ is fixed, $A(v_k)$ yields one specific amplitude out of the $M_k$ possible levels, and the carrier $s_k(t) = A(v_k) \cdot \sin(2\pi f_k t)$ is built using that selected amplitude. Eq. (1) reduces to standard binary ASK when $M_k = 2$, with $A(0) = A_{\text{low}}$ and $A(1) = A_{\text{high}}$. The composite signal $x(t) = \sum_{k=1}^{K} s_k(t)$ is sampled at rate $f_s$ over $N$ samples and quantized to $L$ amplitude levels with $q(\cdot)$, the quantization function mapping each sample to its nearest discrete level. Each level is mapped to a token ID through a fixed pseudorandom bijection $\tau(\cdot)$ between the $L$ quantization levels and a randomly chosen $L$-token subset of the host model's vocabulary, drawn once per architecture, using a fixed random seed. In this way, the mapping is reproducible across runs. Since token IDs near each other have similar embeddings, pseudorandomness prevents the model from using token-ID adjacency as a shortcut for recovering quantization levels, forcing it to rely on genuine frequency decomposition instead. This encoding procedure is used identically whether building the finetuning corpus (§2.1.3) or encoding a specific input for evaluation. In every case, $v_k$, the index of the selected value within channel $k$'s vocabulary, is known before encoding and can be validated against at evaluation time, since the vocabulary is fixed in advance and the value drawn for each sample is recorded at generation time. The FDM-encoded token sequence delivered to the model bears no surface relationship to the channel values it encodes (Appendix B).

#### 2.1.1 S-RANDOM INTERLEAVED DUAL ENCODING

At high channel counts ($K \geq 20$), individual carriers become harder to distinguish as interference between carriers increases with spectral crowding. Resolving adjacent carriers requires sufficient samples per carrier period, which can be increased either by extending the signal duration or by reducing the number of active channels. We address this alternatively with *S-random interleaved dual encoding* where the model receives both the original signal and an S-random interleaved copy, concatenated as $2N = 512$ tokens. The resulting token sequence $\mathbf{s}$, of length $2N = 512$, is what the host model receives as the FDM-encoded input.

$$\mathbf{s} = [\underbrace{\tau(q(x(t_0))), \ldots, \tau(q(x(t_{N-1})))}_{\text{Encoder 1: natural order}}, \underbrace{\tau(q(x(t_{\pi(0)}))), \ldots, \tau(q(x(t_{\pi(N-1)})))}_{\text{Encoder 2: S-random interleaved}}] \tag{2}$$

The S-random permutation $\pi$ with $S = \lfloor \sqrt{N/2} \rfloor = 11$, from standard S-random interleaver design, which denotes the minimum separation constraint used by the S-random interleaver, guarantees that adjacent samples in Encoder 1 are widely separated in Encoder 2, providing a decorrelated redundancy that improves gradient flow during training.

**Dataset generation** Each record is generated as follows: for each channel, one value is drawn uniformly at random from that channel's fixed vocabulary (Appendix A); together these values form the memory for that sample, such that no sample shares the same underlying facts as any other. The memory is FDM encoded into a token sequence. For each target hop count (1–5 hops), a fixed quota of samples is generated. Within that hop count, a template is drawn at random and its decision function is run on the memory to produce the chained answer. The Context block (channels 8–39, `NAME=VALUE` pairs) is then built from the same memory and appended to that answer. The encoded signal, question, final answer, hop count, and memory are bundled into one sample record. All samples are split 85/10/5 into train/validation/test.

#### 2.1.2 DECODING

No explicit demodulation module (e.g., FFT) is present at inference. Decoding is standard next-token prediction, where at each queried channel's output position, the predicted token maps back through the fixed bijection (§2.1) to a quantization level and then to $v_k$. Frequency decomposition therefore happens implicitly within the model's ordinary forward pass, through attention, with no external signal processing step. Since all samples are split 85/10/5 into train/validation/test, at evaluation, the test split's records are used as ground truth against which the model's generated output can be scored.

#### 2.1.3 SINGLE-BLOCK FINETUNING

We finetune the four models on examples of the form `[MEMORY]` ⟨512 FDM tokens⟩ `[/MEMORY]` `Question:` ⟨query⟩ `Answer:` ⟨response⟩, using cross-entropy loss computed only over the response tokens, with the memory block, question, and prompt tokens masked from the loss.

We employ a 5-stage curriculum over 115K training samples, progressively increasing $A_{\text{low}}$ from 0.0 to 0.25 to make frequency decomposition progressively harder (Table 1). The same curriculum schedule is used across all four models to enable direct comparison. Evaluation uses the 1,500 held-out Stage-4 test samples with greedy decoding (up to 250 new tokens), scored for ACTION, FACT, META, RULE and EXTRA channel accuracy.

#### 2.1.4 HOP-CONTROLLED FINETUNING

Continuing from the single-block checkpoint for each model, we finetune on 100K examples structured as conditionally dependent facts (1–5 hops), chaining separate, but dependent channel reads (Templates in Table 9, Appendix A).

#### 2.1.5 POSITION-RANGE FINETUNING

A single-block FDM host sees the FDM block at one fixed position, immediately after `[MEMORY]`. To let it decode the block at a shifted position ($P$=513), we continue finetuning each single-block checkpoint (Qwen3-0.6B, LFM2.5-1.2B, Hermes3-3B; GPT-2 Medium is excluded) with AdamW, cosine learning-rate decay, and full parameter updates. Training mixes the shifted layout ($r$=0.3) with the original layout ($1-r$=0.7), which preserves single-block performance.

#### 2.1.6 SINGLE-BLOCK COEXISTENCE: FDM RETRIEVAL AND NATURAL LANGUAGE

Unlike §2.1.5, this configuration uses a single FDM position range throughout. This section's two training data sources, FDM encoded retrieval examples and natural language replay, are interleaved during finetuning to test whether FDM reading can coexist with natural language competence in a single model. We finetune the four models on a 10-channel configuration in which every training stage interleaves both. The FDM encoded retrieval examples are of the form `[MEMORY]` ⟨512 FDM tokens⟩ `[/MEMORY]` `Question:` ⟨query⟩ `Answer:` ⟨response⟩ and general instruction data (Alpaca; 52K samples) serve as natural language replay.At each optimization step, a batch is drawn from the replay data with probability equal to the stage's replay ratio (30% replay / 70% FDM at the easiest stage) and from the FDM data otherwise. The FDM and replay sources use different

Table 1: Five-stage curriculum training schedule. $A_{\text{low}}$ increases progressively to make frequency decomposition harder at each stage. The same curriculum is applied to all four models.

| Stage | Description | $A_{\text{low}}$ | $A_{\text{high}}$ | Samples | Epochs | Learning Rate |
|---|---|---|---|---|---|---|
| 0 | Easy (max contrast) | 0.00 | 1.0 | 15,000 | 5 | $5 \times 10^{-5}$ |
| 1 | Standard contrast | 0.10 | 1.0 | 20,000 | 5 | $3 \times 10^{-5}$ |
| 2 | Moderate | 0.15 | 1.0 | 25,000 | 7 | $2 \times 10^{-5}$ |
| 3 | Harder contrast | 0.20 | 1.0 | 25,000 | 7 | $1 \times 10^{-5}$ |
| 4 | Hardest | 0.25 | 1.0 | 30,000 | 10 | $1 \times 10^{-5}$ |
| Total | | | | 115,000 | 34 | |

Table 2: Cross-architecture FDM retrieval at $K$=40, $W$=512 ($n$=1500 evaluation samples per architecture). **Action**: correctness of the query's final answer keyword. **Rule**: ch6 retrieval, reported aggregate and conditional (restricted to $n$=285 samples where **Meta**=NONE, the only condition under which the reference answer contains the rule name). **Meta**: ch7 retrieval, all five values checked. **Fact**: per-channel accuracy over consulted channels. **Extra**: per-channel accuracy over the fixed Context block, ch8–39.

| Architecture | Action | Rule (agg.) | Rule (cond.*) | Meta | Fact | Extra |
|---|---|---|---|---|---|---|
| GPT-2 Medium (345M) | 98.5% | 39.1% | 100.0% | 79.9% | 100.0% | 99.4% |
| Qwen3-0.6B | 98.7% | 39.2% | 100.0% | 79.9% | 100.0% | 100.0% |
| LFM2.5-1.2B | 98.6% | 39.1% | 100.0% | 79.9% | 100.0% | 93.0% |
| Hermes3-3B | 98.7% | 39.1% | 100.0% | 79.8% | 99.9% | 99.8% |

objectives: FDM batches apply cross-entropy loss on answer tokens only, so the model is trained to condition on, never to reproduce, the carrier block, while replay batches use the standard causal language modeling objective over the full text. No architectural scaffolding such as gating, routing, or explicit source-classification module is provided. Beyond the `[MEMORY]`/`[/MEMORY]` tokens that mark FDM examples, the model must learn to handle both input distributions within a single set of weights.

The curriculum mirrors the 40-channel schedule of Section 2.1.3 using five stages of increasing decoding difficulty, with $A_{\text{low}}$ rising from $0.0$ to $0.25$ and the per-stage learning rate annealing from $5\times10^{-5}$ to $1\times10^{-5}$. The replay ratio is simultaneously annealed from 30% to 10% with FDM's share thus rising from 70% to 90%, shifting for more FDM emphasis as the task gets harder while maintaining continuous natural language exposure. For Hermes3-3B, all stage learning rates are scaled by $1/3$, which we found necessary for stable training at a 3B scale; all other hyperparameters are identical across the four models. Full-parameter finetuning is used throughout, with AdamW, gradient clipping at norm 1.0 and two added special tokens (`[MEMORY]`, `[/MEMORY]`). These become dedicated tokens so the model has a consistent, single signal indicating when the FDM block starts. Evaluation covers both FDM retrieval and natural-language competence. FDM-side accuracy (ACTION, RULE, META, FACT, and per-channel EXTRA accuracy) is measured with greedy decoding on 1,500 Stage-4 samples held out from finetuning, per model. Natural language competence is measured on a fixed set of 10 general-knowledge questions posed in plain English with no `[MEMORY]` block present, scored by accent-insensitive substring matching against reference answers.

# 3 EMPIRICAL RESULTS AND THEORETICAL CAPACITY ANALYSIS

## 3.1 SINGLE-BLOCK VALIDATION: FDM ENCODING WORKS AS IN-CONTEXT MEMORY

As a single-block in-context FDM memory substrate, all four architectures retrieve at near-ceiling on FACT (99.9–100%), RULE (cond.) (100%) and reach 98.5–98.7% on ACTION. META lags at 79.8–79.9% across all architectures. EXTRA retrieval is near-perfect on Qwen3 and Hermes3 in the standard single-block evaluation (Table 2).

### 3.1.1 FREQUENCY-DEPENDENT RETRIEVAL ACCURACY

We evaluate per-channel readout accuracy by carrier frequency band as follows. For each of the 32 Context block channels (ch8–39), we measure the fraction of test samples in which the model's generated response contains the value originally encoded into the FDM signal for that channel. Grouped by carrier frequency band, reporting the range of per-channel accuracy (Table 3), this breakdown reveals that errors concentrate at high carrier frequencies, which are also the last channels in the Context block, particularly for LFM2.5 model and somewhat for GPT-2. The effect is negligible for Qwen3 and Hermes3.

Table 3: Per-channel retrieval accuracy by carrier frequency band.

| Frequency band | Carrier range | GPT-2 | Qwen3 | LFM2.5 | Hermes3[¶] |
|---|---|---|---|---|---|
| Low (ch8–13) | 9–14 Hz | 99.8–100.0% | 99.9–100.0% | 100.0% | 100.0% |
| Mid-low (ch14–20) | 15–21 Hz | 99.4–100.0% | 99.9–100.0% | 100.0% | 100.0% |
| Mid-high (ch21–30) | 22–31 Hz | 99.2–99.6% | 99.9–100.0% | 100.0% | 100.0% |
| High (ch31–35) | 32–36 Hz | 99.0–99.8% | 99.9–100.0% | 98.9–100.0% | 100.0% |
| Very high (ch36–39) | 37–40 Hz | 92.4–99.6% | 99.9–100.0% | 15.9–86.1% | 99.9–100.0% |

[¶] Qwen3 and Hermes3 use fixed global amplitude normalization ($[-K \cdot A_{\text{high}},\ K \cdot A_{\text{high}}]$, $K = 40$); GPT-2 and LFM2.5 use sample-wise normalization. Each model is evaluated on test data generated by its own encoder, so per-channel accuracy reflects architecture-specific signal distributions and values are not directly comparable at the token level.

### 3.1.2 Simultaneous Multi-Channel Retrieval

To measure the reliability of retrieval under composition, we measure *joint accuracy*: the fraction of samples in which *all* $N$ queried channels are decoded correctly simultaneously, for $N \in \{1, 2, 4, 8, 16, 24, 32\}$. For a given $N$, we select $N$ of the 32 extra channels (ch8–39) and, for each sample, check whether all $N$ are decoded correctly within that one generated answer. All 32 extra channels are always produced together in a single generation, so their correctness is not actually independent; we compare against a hypothetical independence baseline to see how far actual joint accuracy departs from it. Using each of the $N$ selected channels' own empirical accuracy $p_k$, the probability of all $N$ being simultaneously correct *if* channels failed independently would be $\prod_{k=1}^{N} p_k$. We report this as the *independence estimate*, against which actual joint accuracy is compared (Table 4). This independence estimate serves as a naive multiplicative baseline for compositional reliability in that it asks whether joint retrieval degrades the way independent per-channel failure would predict, not to imply statistical correlation.

Table 4 shows that joint accuracy remains at 100% through 4 channels for all three architectures. GPT-2's joint accuracy degrades gently from 100% (N=1) to 88.2% at N=32, the hardest case since it requires all 32 extra channels correct simultaneously. The degradation is concentrated rather than distributed across channels as we can see that per-channel accuracy is 99–100% for channels 8–38, with channel 39 (COMMS) alone at 92.4% accounting for most of the errors (Table 3). Its actual joint accuracy (88.2%) nonetheless exceeds the independence estimate (81.8%). Hermes3 sustains near-ceiling joint accuracy (99.9–100%) across the full range, including $N$=32. LFM2.5 collapses to 2.1% at $N$=32, consistent with its high-frequency carrier degradation at ch36–39 (Table 3); at $N \leq 24$ it remains at 99.9–100%.

### 3.1.3 Hop-Controlled Accuracy

The joint accuracy experiment measures parallel retrieval reliability, but not *compositional* reasoning. For hop counts higher than 1, each hop retrieves a channel value whose interpretation depends

Table 4: Joint retrieval accuracy: fraction of samples with all $N$ channels decoded correctly simultaneously. Indep. columns report the independence estimate $\prod_{k=1}^{N} p_k$, the product of the individual empirical accuracies of the $N$ channels selected, computed from each architecture's per-channel results.

| Channels | GPT-2 (345M) | LFM2.5 (1.2B) | Hermes3 (3B) | Indep. (GPT-2) | Indep. (LFM2.5) | Indep. (Hermes3)[‡] |
|---|---|---|---|---|---|---|
| 1 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 2 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| 4 | 99.8% | 100.0% | 100.0% | 99.8% | 100.0% | 100.0% |
| 8 | 99.2% | 100.0% | 100.0% | 98.6% | 100.0% | 100.0% |
| 16 | 97.6% | 100.0% | 100.0% | 95.3% | 100.0% | 100.0% |
| 24 | 95.4% | 99.9% | 100.0% | 91.2% | 99.9% | 100.0% |
| 32 (all extra) | 88.2% | 2.1% | 99.9% | 81.8% | 1.8% | 99.9% |

[‡] Hermes3 independence estimate is 100.0% for $N \leq 24$ and 99.9% at $N = 32$, reflecting 1 error on ch37 in 1500 samples.

on the outcome of the previous hop, forming a serial conditional chain. The fixed finite vocabulary at each hop makes each step's correct answer unambiguous. We make use of FACT accuracy, which checks only the specific channels the chain consulted along the way (as defined in §2). Hop count reflects the number of channels a chain conditions on in sequence; individual templates vary in which channels they consult (Table 9). Full per-hop logic and worked examples appear in Appendix A. Table 5 shows FACT accuracy remaining at near-ceiling at 1–4 hops then declining to 82.8–83.4% at 5 hops across all four architectures, indicating that retrieval of the individual channels forming a deep chain becomes harder as chain length grows to 5 hops.

### 3.2 TWO-BLOCK FINETUNING AND TOLERANCE TO A LEARNED K/V PREFIX

We test whether the host can decode an FDM token block at a shifted position range while a learned K/V prefix occupies the original positions. The $K$ queried channels are nominally partitioned into two disjoint subsets $\mathcal{C}_A$ and $\mathcal{C}_B$. $\mathcal{C}_A$ is supplied to a key-value cache generated per sample by a $\sim$62.8M-parameter write head from the channel and value indices (trained end-to-end on answer-token cross-entropy with all host parameters frozen); the prefix occupies $P = W + 1 = 513$ cache positions, $[0, P-1]$. FDM-encoded context tokens begin at position $P$, where the signal is $W$=512 tokens before re-tokenization. The in-context signal encodes all 32 queried channels, including those in $\mathcal{C}_A$, so every channel is recoverable from the tokens alone. We perform a two-block finetuning that trains the host to decode FDM at both position ranges, for 5,000 steps at learning rate $2\times10^{-5}$, with a two-block layout sampled at $r$=0.3. Because the tokens carry all channels, this experiment does not show that information is read from the prefix, and we make no claim of joint reads from two locations. We evaluate on Qwen3-0.6B and Hermes3-3B across all nine partition ratios (Table 6); 0/32, which has no prefix, is the control.

Across both architectures, all partitions with nonzero in-context support from 0/32 through 28/4 form a stable partition plateau. On Qwen3-0.6B, slot-level retrieval remains within 99.27–99.63%, while joint all-32-channel retrieval ranges from 83.0–91.6%. Hermes3-3B replicates the plateau at higher accuracy where every nonzero-context partition reaches at least 99.98% slot accuracy and 99.50% joint accuracy. The one configuration that does not converge is 32/0, where no FDM tokens are present and the learned prefix must carry all 32 channels alone; This shows that write head cannot replace the token block, and it is the only row in which retrieval depends on the prefix.

### 3.3 SINGLE-BLOCK COEXISTENCE RESULTS

FDM retrieval coexists functional natural language competence on a single shared checkpoint. Across all four Stage-4 checkpoints (Table 7), FDM retrieval is near-ceiling: FACT accuracy is 99.9–100%, META and EXTRA retrieval are 100%, and ACTION accuracy is 96.9–97.5%. RULE accuracy is the sole weak result at 39.7–40.7%. This reflects the aggregate scoring in §2: ground truth only includes the rule name when META=NONE. Most samples have no rule name in the reference answer to match against, capping aggregate RULE accuracy regardless of true decoding quality, which is a scoring artifact, not a retrieval failure. We find as preliminary evidence, that the same checkpoints retain natural language competence where true accuracy is 10/10 for Qwen3-0.6B and Hermes3-3B, 9/10 for LFM2.5-1.2B, and 8/10 for GPT-2 Medium, with the only errors being genuine factual misses ("São Paulo" for the capital of Brazil) rather than FDM-induced degradation.

Table 5: Hop-controlled accuracy across architectures. FACT: per-channel accuracy averaged over the channels consulted at that hop count (1 channel at 1-hop, up to 5 at 5-hop); each row is a separate evaluation at that hop count, not cumulative.

| Hops | GPT-2 | Qwen3 | LFM2.5 | Hermes3 |
|---|---|---|---|---|
| 1 | 99.9% | 99.9% | 100.0% | 100.0% |
| 2 | 100.0% | 99.2% | 99.9% | 99.9% |
| 3 | 99.4% | 99.3% | 99.4% | 99.4% |
| 4 | 99.8% | 98.4% | 100.0% | 99.9% |
| 5 | 83.2% | 83.4% | 82.8% | 83.2% |

Table 6: Partition-ratio sweep. $n_{\text{param}}/n_{\text{ctx}}$ gives the number of channels nominally assigned to the learned K/V prefix versus the in-context FDM token block. The token block encodes all 32 channels in every configuration, so this table measures tolerance to a learned prefix, not retrieval from the prefix. 0/32 is the no-prefix control. Slot: mean per-channel retrieval accuracy; Joint: fraction of samples with all 32 channels correct in one forward pass. Each seed: 4K steps, $n$=200. Wilson 95% CIs over converged seeds; dashes indicate that no seed converged.

| Model | Partition | Converged | Slot accuracy | Joint accuracy |
|---|---|---|---|---|
| Qwen3-0.6B | 0/32 | 5/5 | 99.27% ± 0.09 | 83.00% ± 2.33 |
| | 4/28 | 5/5 | 99.51% ± 0.08 | 88.00% ± 2.02 |
| | 8/24 | 4/5 | 99.37% ± 0.10 | 87.00% ± 2.33 |
| | 12/20 | 4/5 | 99.45% ± 0.09 | 88.00% ± 2.25 |
| | 16/16 | 5/5 | 99.58% ± 0.07 | 89.00% ± 1.94 |
| | 20/12 | 4/5 | 99.62% ± 0.08 | 90.25% ± 2.06 |
| | 24/8 | 5/5 | 99.54% ± 0.07 | 90.00% ± 1.86 |
| | 28/4 | 4/5 | 99.63% ± 0.07 | 91.62% ± 1.93 |
| | 32/0 | 0/5 | — | — |
| Hermes3-3B | 0/32 | 3/3 | 99.98% ± 0.03 | 99.50% ± 0.96 |
| | 4/28 | 3/3 | 100.00% ± 0.02 | 100.00% ± 0.64 |
| | 8/24 | 3/3 | 99.98% ± 0.03 | 99.50% ± 0.96 |
| | 12/20 | 3/3 | 100.00% ± 0.02 | 100.00% ± 0.64 |
| | 16/16 | 3/3 | 99.99% ± 0.03 | 99.67% ± 0.87 |
| | 20/12 | 3/3 | 99.99% ± 0.03 | 99.67% ± 0.87 |
| | 24/8 | 3/3 | 99.98% ± 0.03 | 99.50% ± 0.96 |
| | 28/4 | 3/3 | 99.99% ± 0.02 | 99.83% ± 0.77 |
| | 32/0 | 0/3 | — | — |

### 3.4 Theoretical capacity

We derive the theoretical FDM capacity per forward pass as $C = K \times b$ bits, bounded by the Nyquist ceiling ($K_{\max} = \lfloor f_s/2 \rfloor - 1$) and amplitude quantization ($b \leq \log_2 64 = 6$ bits per channel). We ask the question, given the same fixed frequency resolution of $\Delta f \approx 0.39\,\text{Hz}$ used empirically, how many channels and how many bits can we store and theoretically retrieve from, assuming separability is guided by frequency-based retrieval theory and guarantees? Table 10 (Appendix C) lists $b$ (number of bits per channel) for each model we tested against $W$ (context window size) and how many $K_{\max}$ (theoretical channels) could carry 1, 3, and 6 bits per channel. $C$ is total theoretical capacity in bits per forward pass or the total amount of information (in bits) the substrate could theoretically deliver to the model in one single pass through the network.

## 4 Discussion

This work presents FDM as the first frequency addressed memory substrate in transformer language models. FDM is a fact retrieval technique that relies on a frequency decomposition rather than a positional token sequence search. We demonstrate this for short, structured synthetic facts, where accurate retrieval is made possible by encoding a composite, carrier frequency modulated signal into a single token sequence stream using the well known technique of ASK (§2.1; Table 2). In this way, FDM inherits the frequency carrier orthogonality features of ASK, where the token budget

Table 7: FDM retrieval and natural language accuracy on the 10-channel Stage-4 held out evaluation (1500 FDM samples + 10 NL questions per model). Rule chance baseline is 25% (four rules).

| Model | Action | Rule | Meta | Fact | Extra | NL |
|---|---|---|---|---|---|---|
| Qwen3-0.6B | 97.5% | 40.6% | 100% | 100% | 100% | 10/10 |
| LFM2.5-1.2B | 97.5% | 39.7% | 100% | 100% | 100% | 9/10 |
| Hermes3-3B | 96.9% | 39.7% | 100% | 99.9% | 100% | 10/10 |
| GPT-2 Medium | 97.4% | 40.7% | 100% | 100% | 100% | 8/10 |

becomes fundamentally governed by frequency resolution and carrier separation rules, rather than growing linearly with the number of plaintext KEY=VALUE strings. Although FDM releases the dependency of retrieval on positional search, adding channels requires a proportionally larger token budget since channels must remain separable between each other for the model to exercise the attention based frequency decomposition. At our tested $f_s = 100\,\text{Hz}$, this ceiling is $K_{\max} = 49$ channels for GPT-2, the smallest model (§3.4). Each carrier occupies a distinct 1 Hz, so in resolving $K$ adjacent carriers without interference between channels, we established a fixed sampling resolution of $\Delta f = f_s/N \leq 1\,\text{Hz}$. Since token budget $W = 2N$, adding more channels directly increase the token cost that is required to keep carriers separable. This therefore introduces a trade-off between payload depth, channel count and token budget that context window scaling can directly relieve while retrieval remains frequency based. Beyond retrieval accuracy, we report a preliminary test of whether the FDM-finetuned models still answer plain English questions. Under the mixed-domain curriculum (§2.1.6), the four checkpoints answer 8–10 of 10 held-out general-knowledge questions (10/10 for Qwen3-0.6B and Hermes3-3B) while sustaining 96.9–97.5% action accuracy on FDM queries (Table 7). With only ten questions and no pre-finetuning baseline, this shows that basic question answering remains functional. FDM therefore purposefully introduces a structural property to a model's memory substrate, providing an explicit frequency addressing basis that may be further extensible to many more substrates with boundaries that attention based FDM decoding can cross. Under hop-controlled retrieval (§3.1.3), FACT accuracy stays near-ceiling through 4 hops before declining to 82.8–83.4% at 5 hops across all four architectures. We show that composing several conditionally dependent channel reads is resilient up to at least 4 hops and degrades at the 5th. Whether this can be alleviated with improved frequency resolution or other signal characteristics remains an open question for further testing. After brief finetuning, the host also decodes the FDM block at a shifted position range, and a learned K/V prefix placed in front of it does not degrade retrieval across partitions (Table 6); because the token block encodes all channels in these runs, this shows prefix tolerance rather than joint reads from two locations, which we leave untested. The FDM features reported in this work generalize across up to three architectures tested (Table 2; §3.1). In the theoretical limit (§3.4; Table 10, Appendix C), channel limits and capacities scale from tens of channels for GPT-2 Medium to over ten thousand for Hermes3-3B, and the empirical $K{=}40$ regime we chose sits well within all four substrates' Nyquist ceilings. The technique of FDM therefore leaves substantial headroom for scaling. Finally, since FDM encoding is globally coupled in that the composite signal of all channels is stored within one token stream, any token change affects all 40 channels simultaneously, making selective fact manipulation a different problem from sequential plaintext corruption.

## 5 CONCLUSION

To the best of our knowledge, we demonstrate the first frequency-multiplexed encoding technique that gives transformers an in-context structured memory substrate with three properties: retrieval addressed by frequency rather than by position within a block, channel resolution controlled by sampling frequency, and channel separability through frequency-based carrier structure. After brief finetuning on a second position range, the host decodes the block at both ranges, and a learned K/V prefix placed in front of it does not degrade retrieval. FDM composition holds up under moderate compositional depth, up to 4 hops. We also provide preliminary evidence that plain English competence can coexist with FDM retrieval. This suggests a new design space for structured, accurate memory substrates in language models.

### AI USE STATEMENT

Large language model tools were used for prose editing, LaTeX formatting, debugging support, organization and editing of manuscript drafts. Python experimental scripts were generated by these tools as were the evaluation of results of running these scripts. All scientific claims, experimental designs, code execution, numerical results, tables, cited sources, and final text were reviewed, verified, and approved by the author.

## REPRODUCIBILITY STATEMENT

More work and results can be found in the appendices while the associated assets are available at `https://github.com/quaintanceai-prog/FDM`.

## REFERENCES


Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. doi: 10.18653/v1/2024.emnlp-main.781.

DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Ré. Cartridges: Lightweight and general-purpose long context representations via self-study. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. arXiv:2506.06266.

Yao Fu. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. NoLiMa: Long-context evaluation beyond literal matching. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.05167.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2410.10813.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025b. arXiv:2404.15574.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10210–10229, 2024. doi: 10.18653/v1/2024.acl-long.550.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. arXiv:2502.11089.

# A CHANNEL DEFINITIONS

Table 8: Channel definitions. Channels 0–7 drive reasoning logic; channels 8–39 provide operational context that must appear in the answer.

| Ch. | Name | $\|V_c\|$ | Values |
|---|---|---|---|
| 0 | SECRET | 4 | RED, BLUE, GREEN, GOLD |
| 1 | LOCATION | 4 | PARIS, TOKYO, LONDON, BERLIN |
| 2 | AGENT | 4 | ALICE, BOB, CAROL, DAVE |
| 3 | STATUS | 3 | CLEAR, COMPROMISED, UNKNOWN |
| 4 | PRIORITY | 3 | HIGH, MEDIUM, LOW |
| 5 | BACKUP | 2 | AVAILABLE, UNAVAILABLE |
| 6 | RULE | 4 | SAFETY_FIRST, MISSION_FIRST, BALANCED, CAUTIOUS |
| 7 | META | 5 | NONE, OVERRIDE_STATUS, OVERRIDE_PRIORITY, EMERGENCY, LOCKDOWN |
| 8 | TEAM | 4 | RED_TEAM, BLUE_TEAM, GREEN_TEAM, GOLD_TEAM |
| 9 | REGION | 4 | NORTH, SOUTH, EAST, WEST |
| 10 | PHASE | 4 | ALPHA, BETA, GAMMA, DELTA |
| 11 | COMM | 3 | OPEN, CLOSED, RESTRICTED |
| 12 | ASSET | 4 | VEHICLE, AIRCRAFT, DRONE, BOAT |
| 13 | WINDOW | 4 | DAWN, MIDDAY, DUSK, NIGHT |
| 14 | COVER | 3 | DEEP, SHALLOW, NONE |
| 15 | SUPPORT | 3 | ACTIVE, STANDBY, OFFLINE |
| 16 | THREAT | 4 | LOW, MEDIUM, HIGH, CRITICAL |
| 17 | WEATHER | 3 | CLEAR, STORM, FOG |
| 18 | TERRAIN | 4 | URBAN, RURAL, COASTAL, MOUNTAIN |
| 19 | EXTRACT | 3 | READY, DELAYED, UNAVAILABLE |
| 20 | CIPHER | 4 | AES, RSA, BLOWFISH, TWOFISH |
| 21 | FREQ | 4 | HF, VHF, UHF, SHF |
| 22 | PAYLOAD | 4 | LIGHT, MEDIUM, HEAVY, CRITICAL |
| 23 | ROUTE | 4 | ALPHA, BRAVO, CHARLIE, DELTA |
| 24 | DURATION | 4 | SHORT, MEDIUM, LONG, EXTENDED |
| 25 | CONTACT | 4 | FRIENDLY, NEUTRAL, HOSTILE, UNKNOWN |
| 26 | FUEL | 4 | FULL, HALF, LOW, CRITICAL |
| 27 | ALTITUDE | 3 | LOW, MEDIUM, HIGH |
| 28 | VISIBILITY | 3 | CLEAR, REDUCED, ZERO |
| 29 | NOISE | 4 | SILENT, QUIET, MODERATE, LOUD |
| 30 | FORMATION | 4 | SINGLE, PAIR, SQUAD, PLATOON |
| 31 | ARMOR | 4 | NONE, LIGHT, MEDIUM, HEAVY |
| 32 | SIGNAL | 4 | STRONG, WEAK, JAMMED, LOST |
| 33 | MORALE | 3 | HIGH, MEDIUM, LOW |
| 34 | SUPPLY | 4 | ABUNDANT, ADEQUATE, SCARCE, DEPLETED |
| 35 | INTEL | 4 | CONFIRMED, PROBABLE, UNCERTAIN, NONE |
| 36 | EVAC | 4 | STANDING, PREPPED, LAUNCHED, ABORTED |
| 37 | WEATHER2 | 4 | SUNNY, OVERCAST, RAIN, SNOW |
| 38 | DOCTRINE | 4 | OFFENSIVE, DEFENSIVE, RECON, SUPPORT |
| 39 | COMMS | 4 | SECURE, OPEN, COMPROMISED, SILENT |

Table 9: N-Hop Reasoning Templates. Each arrow denotes a necessary compositional dependency: the output of each channel read conditions the next branching decision, rather than parallel independent reads. Channel indices refer to the 40-channel FDM memory schema.

| Hops | Question Prompt | Channels | Channel Names |
|---|---|---|---|
| 1 | Report current status. | 3 | STATUS |
| 1 | Report threat level. | 16 | THREAT |
| 1 | Report active rule. | 6 | RULE |
| 1 | Report current location. | 1 | LOCATION |
| 2 | Evaluate status under current rule. | 6, 3 | RULE → STATUS |
| 2 | Assess threat in current terrain. | 16, 18 | THREAT → TERRAIN |
| 2 | Report agent deployment location. | 2, 1 | AGENT → LOCATION |
| 3 | Should we proceed given rule, status, and backup? | 6, 3, 5 | RULE → STATUS → BACKUP |
| 3 | Evaluate threat exposure with cover and signal status. | 16, 14, 32 | THREAT → COVER → SIGNAL |
| 4 | Full override check: meta, rule, status, priority. | 7, 6, 3, 4 | META → RULE → STATUS → PRIORITY |
| 4 | Tactical assessment: threat, terrain, weather, asset. | 16, 18, 17, 12 | THREAT → TERRAIN → WEATHER → ASSET |
| 5 | Complete chain: meta override, rule, status, backup, threat. | 7, 6, 3, 5, 16 | META → RULE → STATUS → BACKUP → THREAT |
| 5 | Environmental chain: weather, visibility, window, terrain, extract. | 17, 28, 13, 18, 19 | WEATHER → VISIBILITY → WINDOW → TERRAIN → EXTRACT |

## B NON-LINGUISTIC ENCODED SEQUENCE

The FDM-encoded token sequence the model receives for storage and retrieval bears no surface relationship to the channel values it encodes. For an input containing `STATUS=CLEAR`, `RULE=SAFETY_FIRST`, and 38 other channel values, the FDM-encoded prompt delivered to Qwen3 (first 25 of 512 tokens) reads as:

```
[MEMORY] silhouetteyna counterparts neighborhoodidepress
TerritBone Recall substantiveabilia neighborhood
neighborhood substantive TerritRepe Territ neighborhood
neighborhood counterparts silhouette neighborhood...
[/MEMORY]
```

## C THEORETICAL CAPACITY TABLE

Table 10: Theoretical FDM capacity at matched frequency resolution ($\Delta f \approx 0.39\,\text{Hz}$). $C = K_{\max} \times b$.

| Model | $W$ | $K_{\max}$ | $C$ ($b = 1$) | $C$ ($b = 3$) | $C$ ($b = 6$) |
|---|---|---|---|---|---|
| GPT-2 Medium | 1,024 | 49 | 49 bits | 147 bits | 294 bits |
| Qwen3-0.6B | 32,768 | 3,199 | 3,199 bits | 9,597 bits | 19,194 bits |
| LFM2.5-1.2B | 32,768 | 3,199 | 3,199 bits | 9,597 bits | 19,194 bits |
| Hermes3-3B | 131,072 | 12,799 | 12,799 bits | 38,397 bits | 76,794 bits |