Hidden Tape: Addressable Communication Beneath Identical Text

Anonymous authors
Paper under double-blind review

Abstract

Identical text can conceal different messages in its token representation. Prior work uses this freedom to steer a model's response through retokenization, or hides payloads in generated text whose surface changes with the message. We instead hold one text fixed and ask whether its token order can store a message that a later query reads by address. We introduce Hidden Tape, an addressable communication channel that encodes messages through token order while preserving decoded bytes, token count, and the complete token histogram. A writer commits a token sequence before the query arrives, so one fixed sequence must support every address. We construct matched token cells that realize this channel and demonstrate reliable neural readout and learned writing. Three supervised-calibrated Qwen2.5-0.5B readers recover every bit of all 256 eight-bit messages after being frozen. An MLP writer trained with scalar correctness rewards achieves 99.98% query accuracy in the sixteen-bit setting. Targeted cell interventions verify address-specific readout. Decoding and canonically re-encoding the eight-bit tapes preserves their text while erasing the message, reducing the same readers from 100% to exactly 50% accuracy. These results establish token order as writable, addressable message state beneath a fixed textual surface and show how the transport interface preserves or removes that state.

1 Introduction

Language models consume token sequences (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023), whereas people usually inspect their decoded text. These two views can carry different information: distinct token sequences may display exactly the same bytes. Prior work shows that changing a prompt's tokenization can change model behavior while preserving its text (Geh et al., 2025). We ask whether this representational freedom can support an addressable communication channel: can a sender write a whole message beneath one fixed text, then let a receiver retrieve whichever bit is requested later?

Hidden Tape answers this question with a matched-cell construction (Figure 1). A writer observes a binary word and commits one token tape. Only afterward does a query specify which bit the reader should return. Different words change the order of tokens inside aligned cells while leaving the decoded text, total token count, and complete token histogram fixed. A reader with direct token-ID access can distinguish these choices. A person inspecting the decoded bytes receives the same text for every word.

The late query turns a token sequence into reusable message state. Committing before the address arrives requires every coordinate to remain recoverable from the same tape. Existing retokenization studies vary a prompt's representation to influence generation (Geh et al., 2025; Jain et al., 2026); linguistic steganography encodes payloads in generated texts and repairs extraction after those texts are retokenized (Ziegler et al., 2019; Yan & Murawaki, 2025). Hidden Tape holds one text fixed across the entire message family and tests whether its token representation supports independently chosen queries. The distinction is between changing a model's response and storing a message that the model can read by address.

The cells isolate the variable that carries the message. Two distinct token paths with equal expansions are concatenated in opposite orders. Each cell therefore contains exactly the same tokens under either

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.

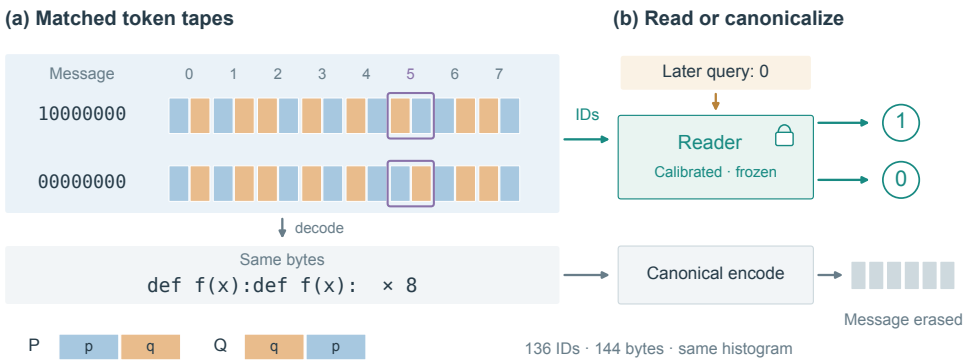


Figure 1: Matched eight-bit tapes beneath identical text. Each colored block is a token path; swapping paths in cell five changes the answer to query zero. Direct IDs preserve the message, while canonical encoding erases it.

choice, and changing one cell leaves the positions of all other cells unchanged. We calibrate Qwen2.5-0.5B readers to a fixed address convention and freeze them. Three readers retrieve every coordinate of the complete eight-bit family. In three exact writer–reader systems, changing the requested cell reverses the answer, while changing an irrelevant cell leaves it intact.

Canonicalization changes the entire communication problem. Decoding and canonically re-encoding all 256 eight-bit tapes preserves their bytes but maps every tape to one token sequence. The same three frozen readers move from exact answers to 50% accuracy at every query. A control that places the message in visible text retains 256 canonical sequences and supports exact readout after separate reader calibration. The intervention links neural readout to the decoding map: a text-dependent reader must answer identically on every tokenization of the same text.

The channel also supports learned writing. An MLP writer learns its cell choices through scalar correctness rewards from a frozen receiver. Three eight-bit writers achieve exact exhaustive retrieval and, in a separate split experiment, exact retrieval on odd-parity words after training on even-parity words. The selected sixteen-bit writer attains 99.98% all-query accuracy beneath 288 identical decoded bytes. Complementary readouts locate order information in frozen Qwen2.5-1.5B, Qwen2.5-3B, and GPT-2: first-block cell readouts exceed 95% at every coordinate of an eight-bit odd-parity test set, while final-query readouts depend strongly on source position.

Construction, neural readout, and reward-trained writing jointly establish an addressable message space beneath a fixed textual surface. The transport interface controls whether that space reaches the receiver: direct token IDs preserve the stored choices, and canonical re-encoding removes them while displaying exactly the same text.

2 Related work

Token representations. Subword methods include BPE (Sennrich et al., 2016), WordPiece (Schuster & Nakajima, 2012; Song et al., 2021) and SentencePiece (Kudo & Richardson, 2018), with segmentation choices affecting regularization (Kudo, 2018; Provilkov et al., 2020) and pretraining (Bostrom & Durrett, 2020). The foundations of tokenization distinguish string-level estimation from ambiguity of token representations (Gastaldi et al., 2025). Finite-state transduction characterizes valid segmentations and common tokenizers (Cognetta & Okazaki, 2024). We organize decoding preimages into message spaces: matched cells carry independent bits at fixed token and byte offsets.

Noncanonical inputs. Adversarial Tokenization studies semantic retention and behavioral changes across tokenizations of a fixed prompt (Geh et al., 2025). Retokenization symmetry studies how this sensitivity evolves with training (Jain et al., 2026). Related attacks use adversarial triggers or substitutions (Wallace et al., 2019; Jin et al., 2020; Zou et al., 2023), imperceptible characters (Boucher et al., 2022), indirect prompt injection (Abdelnabi et al., 2023), and trained backdoors (Cao et al., 2024). Our task asks whether one representative, selected before the query, supports retrieval of

every coordinate of an external word. Matching length and the full token histogram isolates order as the message-bearing variable.

Communication and retrieval. Linguistic steganography embeds messages through language-model generation (Ziegler et al., 2019; Fang et al., 2017; Yang et al., 2019; Kaptchuk et al., 2021); watermarks encode provenance in generated outputs (Kirchenbauer et al., 2023; Kuditipudi et al., 2024). Tokenization-consistent extraction uses verification, rollback or synchronization (Yan & Murawaki, 2025; Wang et al., 2026). Our fixed-surface construction isolates the complementary channel: direct token transport preserves the payload, while canonicalization maps all messages to one sequence. Subliminal learning concerns trait transfer through training data (Schrodi et al., 2025); our frozen receiver retrieves an externally selected binary word. Circuit and intervention studies trace Transformer retrieval (Olsson et al., 2022; Wang et al., 2023; Geva et al., 2023; Meng et al., 2022). Mixing Mechanisms distinguishes stored information from requested retrieval (Gur-Arieh et al., 2025), motivating our local and answer-site probes.

3 Method: constructing and addressing the channel

3.1 Interfaces and the order of information

Let Γ be a finite token-ID alphabet, Σ a byte alphabet, and $h : \Gamma^* \to \Sigma^*$ a non-erasing expansion homomorphism. Each ordinary token expands to a nonempty byte string, and sequence expansion concatenates those strings. For a surface s, its *detokenization fiber* is

$$F_h(s) = \{z \in \Gamma^* : h(z) = s\}. \tag{1}$$

The writer observes $x \in \{0,1\}^R$ and selects $z = W(x) \in F_h(s_R)$. The reader receives (z, q) through a direct token-ID interface, where $q \in \{0, \dots, R-1\}$ is supplied after the write. Its target is x_q. A human text observer receives $h(z) = s_R$. At a fixed width, this observation is constant over all messages; the appended query is also the same across messages for a fixed address. Our experimental transport concatenates the tape IDs with the query IDs and passes them directly to the model.

Querying after the write gives an exact criterion. If two distinct messages share a tape, choose a coordinate on which they differ. A deterministic reader presented with that tape and coordinate produces the same answer for both messages. Therefore, exact all-query retrieval requires an injective write map and at least 2^R distinct tapes (Shannon, 1948; Cover & Thomas, 2006). Our evaluation writes once per word and reuses that representative for all R queries. Figure 2 instantiates this write-before-query interface.

3.2 Matched cells from tokenization ambiguity

Suppose distinct token words $p, q \in \Gamma^+$ expand to the same nonempty byte string u. Define $P = pq$ and $Q = qp$. These cells satisfy

$$P \neq Q, \qquad h(P) = h(Q) = u^2, \qquad |P| = |Q|, \qquad \text{hist}(P) = \text{hist}(Q). \tag{2}$$

Here hist counts each token ID. The inequality follows from a free-monoid fact: commuting words are powers of a common word. If $pq = qp$, write $p = v^\alpha, q = v^\beta$. Their equal, nonempty expansions imply $\alpha = \beta$ and hence $p = q$, a contradiction. Appendix A supplies a self-contained argument.

For a code $b \in \{0,1\}^R$, concatenate one cell per coordinate:

$$\tau(b) = C_{b_0} C_{b_1} \cdots C_{b_{R-1}}, \qquad C_0 = P, \quad C_1 = Q. \tag{3}$$

Equal block lengths make the 2^R tapes distinct, and every one expands to u^{2R}. Their total length is $R(|p| + |q|)$ and their histogram is $R(\text{hist}(p) + \text{hist}(q))$. Thus any non-erasing ambiguous vocabulary supplies a family with positive message rate, one bit per $2|u|$ decoded bytes. The construction enlarges the carrier with width while keeping each width's surface fixed across all codewords.

Equal lengths keep every cell at fixed token and byte offsets. Flipping one logical bit changes one cell's order while preserving its token inventory and all neighboring cells. Fixed spans thus act as addresses, with order within each span supplying the stored value.

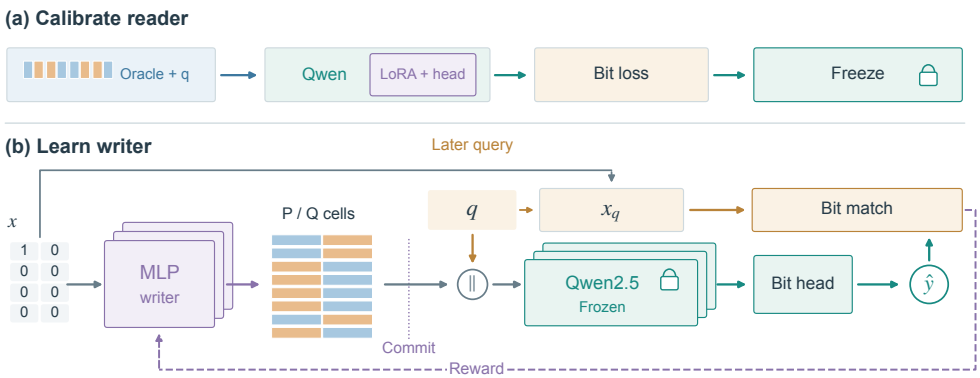


Figure 2: The writer commits before the query. A calibrated, frozen Qwen reader returns one bit; comparison with the requested input bit supplies the writer a scalar correctness reward.

The Qwen experiments use two paths for `def f(x):`, with nine and eight tokens. A matched cell has 17 tokens and expands to 18 bytes. Eight cells contain 136 tokens and 144 decoded bytes; sixteen cells contain 272 tokens and 288 bytes. The eight-bit audit enumerates 256 distinct token sequences, one decoded surface, one token count, and one histogram. Appendix B gives the exact IDs. A GPT-2 witness uses two paths for `hello world` and a 21-token matched cell.

The accompanying tokenizer census searches explicitly generated candidate byte strings. In a Qwen expansion table covering the first 50,000 token IDs, 43,918 of 51,149 candidates admit multiple paths. In the GPT-2 byte vocabulary, the corresponding counts are 115,403 of 123,293. Candidates combine corpus templates, printable short token expansions, and concatenated expansion pairs; Appendix B.1 specifies this population. Equation 2 converts each valid ambiguity in this candidate set into matched cells.

These invariants give the neural experiments a controlled carrier: local interventions change a cell's order at a fixed span, and wider tapes add independently addressable spans using the same two cell types.

More generally, an audit with at most M possible outputs partitions the 2^R tapes into at most M classes. One class therefore contains at least $2^R/M$ representatives (Appendix A). For the joint text, length and histogram audit, $M = 1$: the complete constructed codebook shares one report. Direct token transport supplies the reader with the representative inside that class, which is the information used for addressable retrieval.

3.3 Reader calibration and the address convention

The logical code uses a fixed permutation π and mask m:

$$b_j = x_{\pi(j)} \oplus m_j. \tag{4}$$

We generate the permutation and mask with protocol seed 20260815 and use them for all reader seeds at each width. Reader training pairs oracle tapes $\tau(b)$ and queries with the target bit. Qwen2.5-0.5B-Instruct (Yang et al., 2024) receives rank-eight LoRA adapters (Hu et al., 2022) on attention projections and a binary linear head on the last query token's final hidden state. After calibration, we freeze the reader and head. The writer chooses P or Q at each position, and its R choices materialize the complete tape through Eq. 3. It learns these choices from scalar correctness rewards, as described in Section 4.3.

The convention separates a logical address from its physical location. To answer query k, an exact reader locates cell $j = \pi^{-1}(k)$ and returns $b_j \oplus m_j$. At eight bits, for example, logical query zero addresses physical cell five. The relevant cell has the same expansion in either orientation, but its orientation determines the requested answer after applying the mask. The reader therefore has two coupled tasks: distinguish the local order pattern and route the requested cell's value to the answer position. This decomposition motivates the separate local-source and answer-site measurements in Section 4.2.

(a) Which cell changes the answer?

	Seed 0	Seed 1	Seed 2
Other cell	0%	0%	0%
Queried cell	**100%**	**100%**	**100%**
All cells	100%	100%	100%

256 words · 2,048 query checks / reader

(b) Readout after transport

	Seed 0	Seed 1	Seed 2
Direct IDs	**100%**	**100%**	**100%**
Canonical	50%	50%	50%
Visible control	100%	100%	100%

Distinct tapes: 256 direct → 1 canonical · 1 text

Figure 3: Exact eight-bit interventions across three readers. (a) Answers changed by cell swaps. (b) Query accuracy after transport. Each reader receives 2,048 checks; canonicalization merges 256 tapes into one.

For a concrete write, take the eight-bit message $x = (1, 0, 0, 0, 0, 0, 0, 0)$. With the permutation from Appendix C and mask $m = (0, 0, 1, 1, 1, 0, 0, 1)$, this gives $b = (0, 0, 1, 1, 1, 1, 0, 1)$, hence tape $PPQQQQPQ$. Once this tape is committed, query zero reads cell five and returns one; query one reads cell seven and removes its mask bit to return zero. Both answers use the same 136 token IDs. Replacing the input by the all-zero word changes only cell five, giving $PPQQQPPQ$. Both tapes expand to eight copies of `def f(x):def f(x):`.

4 Experiments

4.1 Address access and canonical erasure

Coordinate access. The exhaustive eight-bit oracle family supplies every code under Eq. 4. Each of three supervised Qwen readers answers all 2,048 word–query checks correctly with its parameters frozen. For every word, its 136-token representative is reused across all eight queries.

We intervene on individual cells in the three exact exhaustive writer–reader systems from Section 4.3. Reversing the path order in the requested cell reverses the prediction on every word–query pair; swapping an irrelevant cell preserves every prediction (Figure 3a). Complementing the entire code reverses every answer. Each intervention preserves the text, length, and histogram. Its effect follows the relation between the changed physical cell and the logical address, showing that the reader uses the appropriate coordinate of the committed code.

Canonical erasure. For the erasure intervention, we enumerate oracle codes directly and use the same three readers. Let $T : \Sigma^* \to \Gamma^*$ be a deterministic canonical encoder satisfying $h(T(s)) = s$ on the carrier domain. Define the canonicalization operator $\mathcal{C} = T \circ h$. Every tape in a fixed fiber has the same image:

$$z, z' \in F_h(s) \quad \Longrightarrow \quad \mathcal{C}(z) = \mathcal{C}(z') = T(s). \tag{5}$$

For a uniform binary word and an independently chosen query, a reader given $(\mathcal{C}(\tau(b)), q)$ receives one fixed tape for each address. Its target bit is balanced at every address. Consequently its accuracy is exactly $1/2$ for any deterministic prediction, and its expected accuracy is $1/2$ for randomized predictions. Canonicalization removes the message variable while preserving the decoded bytes.

We apply this operator to all 256 oracle-code tapes in the eight-bit fiber and evaluate the same three frozen readers before and after the transformation. Before canonicalization there are 256 distinct 136-token tapes; afterward there is one 64-token tape. The decoded 144-byte string is identical throughout. Direct readout is 100% for every reader and query. Canonical readout is 50% for every reader and query, each evaluated on 256 words times eight addresses. Figure 3b shows this paired intervention.

The control stores the message in visible text while retaining the original cell length and histogram. Its two cells are the original 17-token cell P and a one-token cyclic rotation of that cell. They expand to `def f(x):def f(x):` and `ef f(x):def f(x):d`. Combining eight cells produces 256 distinct surfaces and 256 distinct canonical sequences. Their canonical lengths range from 64 to 68 tokens. Three readers calibrated on those canonical tapes achieve 100% all-query and worst-query accuracy (Figure 3b). Canonicalization therefore collapses the fixed-surface codebook while retaining the visible-text codebook.

The decoding map also characterizes a reader whose answers are invariant to tokenization. For each query, a deterministic reader is constant on every fiber precisely when it factors through decoded text:

$$D(z, q) = f(h(z), q) \tag{6}$$

for some text-level function f. One direction follows by substitution. For the other, define $f(s, q)$ as the common reader value on the fiber of s. Composing any reader with $T \circ h$ enforces this factorization and the chance-accuracy result above.

The paired codebooks show the action of this operator. In the hidden codebook, all message distinctions lie within one fiber, and canonicalization sends that entire fiber to one representative. In the visible control, the messages occupy distinct fibers, so their canonical representatives remain distinct. The readout results follow this difference in codebook geometry: the hidden family loses its message variable, while separately calibrated readers recover every coordinate of the visible family. Decoded content therefore determines which message distinctions survive canonical transport.

4.2 Where frozen models retain the code

The matched tape stores its physical code in token order. We locate a readable representation of that code in frozen Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and GPT-2 (124M) (Radford et al., 2019). Each model receives an eight-cell matched tape: the Qwen carrier above, or the GPT-2 `hello world` carrier. We measure two sites. The *local source* is the mean hidden vector across the cell carrying a particular coordinate, after the first transformer block. The *answer site* is the last query token's final hidden vector. Model weights stay fixed; for each bit, a mean-difference linear readout and its midpoint threshold are fitted on the 128 even-parity words and tested on the 128 odd-parity words.

Figure 4a shows mean local accuracies of 99.71%, 99.51%, and 99.41% for Qwen 1.5B, Qwen 3B, and GPT-2. Every local coordinate exceeds 95% in all three models. The corresponding even-split class-mean separation norms range from 0.26 to 0.39, 0.22 to 0.38, and 0.55 to 0.85. Together with the held-out accuracies, these separations locate a concrete representation of order within the contextualized cell vectors.

Mean answer-site accuracy is 71.09%, 71.88%, and 58.01%, respectively. The query-by-source matrix in Figure 4c shows the source of this distribution in Qwen 3B: information about the first and last physical cells is strongly available across multiple query contexts. The requested source, outlined in orange, moves with the query according to the query-to-cell mapping. The matrix therefore distinguishes information tied to source location from information aligned with a requested address. Panels (b–d) show all three models, enlarged in Figure 7.

The matched construction sharpens this interpretation. Before contextual processing, a mean of token embeddings over a cell is invariant to cell order because its histogram is fixed. For GPT-2, the added absolute-position embeddings also have a fixed mean on the same span. After a transformer block, the cell vectors support the observed local readouts. Thus contextual processing exposes order information across both model families (Appendix C).

The query/source matrices measure the next step: transporting a stored coordinate to the answer position. Each row holds the query context fixed while testing the availability of every source bit. The outlined entry follows the prescribed address permutation, whereas a vertical pattern follows one physical location across queries. This separates two spatial relationships in the same representation. In the calibrated 0.5B readers, the cell interventions of Section 4.1 test the requested relationship directly by changing its source cell and observing the resulting answer.

4.3 Learning to write with scalar rewards

The writer is an MLP with two 128-unit ReLU hidden layers and R output logits. Given the word alone, it defines a factorized Bernoulli distribution $p_\theta(b \mid x)$. Training samples b, materializes its hard token tape, and then samples q uniformly. The environment evaluates the frozen reader D_ϕ and returns

$$r = \mathbf{1}\{D_\phi(\tau(b), q) = x_q\}, \qquad \widehat{g} = (r - c)\nabla_\theta \log p_\theta(b \mid x), \tag{7}$$

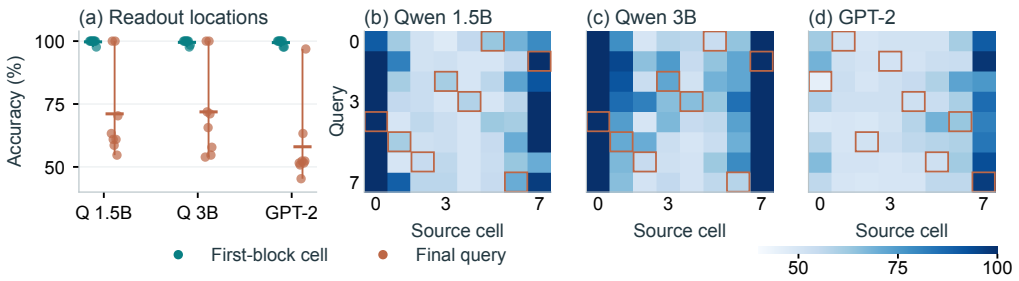


Figure 4: Frozen-model readouts on 128 odd-parity words. Left: every coordinate and mean at cell versus query locations. Right: all query/source matrices; orange outlines identify the requested source cell.

with a moving-average baseline c. This score-function estimator (Williams, 1992) uses the input word and binary outcome, with reader execution frozen and detached. At evaluation, thresholding the writer logits selects one deterministic tape for each word.

Training uses AdamW, batch size eight, learning rate 10^{-3} for the writer, and baseline update rate 0.05. Each training word receives one sampled query and one correctness reward, so reward counts equal eight times writer update counts. Reader calibration uses learning rate 2×10^{-4}. At widths four, eight, and twelve it samples word–query pairs; the sixteen-bit reader sees all sixteen queries for each sampled word during calibration. Queries have fixed token length within a task, using `Q0`–`Q7` at eight bits and zero-padded addresses at the wider widths. Model and protocol details appear in Appendix C.

At sixteen bits, the 16,000 reader updates contain 256,000 supervised word–query presentations. We then optimize the writer while holding the reader fixed. Acquisition curves track how scalar feedback changes the mapping from input words to token tapes.

Evaluation protocol. We measure mean all-query accuracy, the minimum accuracy over query coordinates, exact agreement with the affine code, and the empirical Shannon entropy of the writer's output code distribution. At widths four, eight, and twelve, evaluation enumerates all 2^R words and every query. At sixteen bits, accuracy uses 8,192 sampled word instances arranged in pairwise-balanced affine blocks. Within each 32-word block, each bit takes each value sixteen times and each bit pair takes each joint value eight times, and each word's tape is tested on all sixteen queries. A separate sample of 262,144 antithetic word instances estimates code entropy. Samples may repeat words; all denominators count instances. Appendix C.1 gives the deterministic sampling procedure and seeds.

For the width study we select the first recorded evaluation at which exact-code agreement, worst-query accuracy, and entropy divided by width are all at least 0.95. Figure 5 shows acquisition prefixes ending at those evaluations. We report every writer trained against a reader meeting the calibration criterion (mean and worst-query accuracy at least 0.95): three seeds each at four and eight bits, two at twelve bits (0 and 2), and one at sixteen bits (0).

The three acquisition metrics describe complementary properties of the learned write map. Query accuracy measures retrieval from the committed tape, and its minimum over addresses tracks access across the whole memory. Exact-code agreement asks whether the complete tape follows the receiver's permutation and mask. Entropy measures how many of the available code choices the writer uses across input words.

Sixteen bits beneath 288 identical bytes. The sixteen-bit reader reaches exact oracle accuracy at calibration update 16,000 on its 8,192-instance all-query evaluation and is then frozen. The writer reaches the selected checkpoint at update 64,000, corresponding to 512,000 correctness rewards. At this checkpoint, it answers 131,047 of 131,072 word–query checks correctly: 99.98% mean accuracy. The hardest query answers 8,177 of 8,192 instances correctly, or 99.82%. The entire predicted code agrees with the affine convention on 8,167 of 8,192 instances, or 99.69%. All tapes have 272 token IDs and decode to the same 288-byte string.

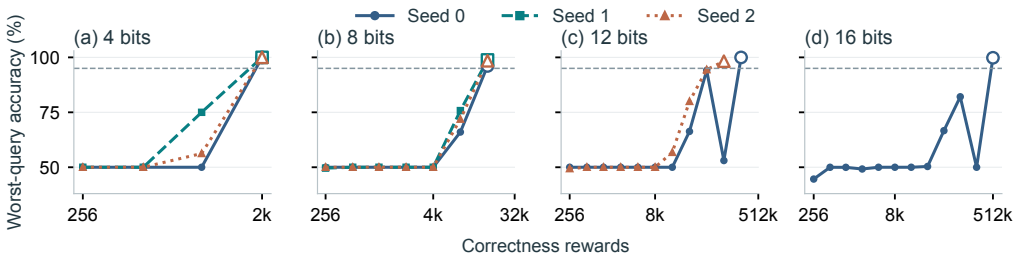


Figure 5: Writer acquisition across four widths and all calibrated seeds. Open markers mark the selected checkpoint (first joint 95% accuracy/code/entropy). Widths 4/8/12 enumerate all words; width 16 uses 8,192 balanced instances.

Table 1: Selected checkpoints: percentage accuracies and entropy in bits. Bold/underline denote best/second distinct values within each width.

Bits	Seed	Rewards	Mean (%)	Worst (%)	Code (%)	Entropy
4	0,1,2	2,000	100	100	100	4
8	0	16,000	<u>99.51</u>	96.09	<u>96.09</u>	<u>7.922</u>
8	1	16,000	**99.80**	**98.83**	**98.44**	**7.969**
8	2	16,000	**99.80**	<u>98.44</u>	**98.44**	**7.969**
12	0	256,000	**100.0**	**99.98**	**99.98**	**12.00**
12	2	128,000	<u>99.86</u>	<u>98.27</u>	<u>98.27</u>	<u>11.97</u>
16	0	512,000	99.98	99.82	99.69	15.80

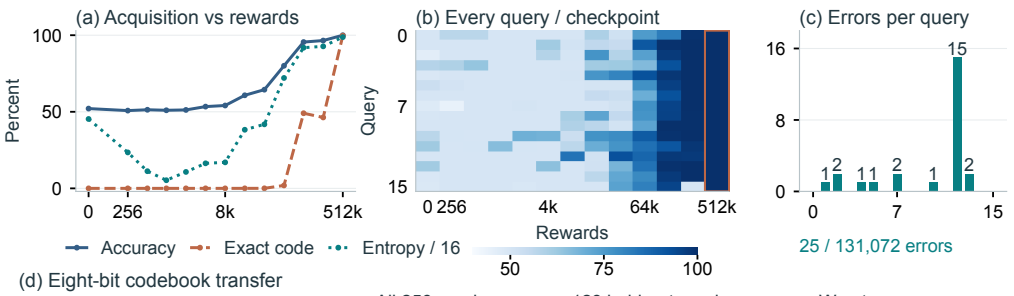
Empirical code entropy is 15.80 bits on the separate 262,144-instance sample, or 98.77% of the sixteen-bit maximum. The writer uses a broad codebook while closely matching the reader's address mapping. Figure 6 follows code acquisition and readout at each address through this checkpoint.

The address-resolved view shows how this accuracy is distributed through the tape. Eight addresses have exact retrieval on the evaluated instances, and every address reaches at least 99.82%. The same committed representative supports all sixteen answers, linking the complete-code agreement to coordinate-level access. The acquisition heatmap follows these addresses through training, alongside the growth of code agreement and entropy.

Exhaustive acquisition and a held-out codebook split. All three eight-bit writers reach exact answers over the 2,048 word–query checks and the full eight-bit codebook at 32,000 rewards. Figure 5 also shows acquisition through the selected checkpoint at four and twelve bits. Table 1 brings the three selection metrics together with each seed's reward count; Appendix D gives the evaluation populations.

The complete eight-bit study also tests writer-side generalization. Writers trained on the 128 even-parity words are evaluated on all 128 odd-parity words, paired with every query. Each of the three seeds answers all 1,024 held-out checks correctly and exactly recovers the affine code on all 128 held-out words. The reader is fixed during this split experiment. Because the convention is a permutation plus a fixed mask, every odd-parity word maps to a code of the opposite parity from all training codes; exact odd-half retrieval therefore establishes that the writer acquired a rule extending beyond its even-parity training support. Appendix D reports both support halves and full-cube values per seed.

The parity split retains balanced values at every individual coordinate while separating complete messages into disjoint sets. Training therefore exposes the writer to both bit values at each address, and evaluation asks it to compose those choices into previously unseen words. Exact held-out code agreement means that this composition reproduces the complete address convention.



(d) Eight-bit codebook transfer

	All 256 words	128 held-out words	Worst query
Seed 0	2,048/2,048	1,024/1,024	**100.0%**
Seed 1	2,048/2,048	1,024/1,024	**100.0%**
Seed 2	2,048/2,048	1,024/1,024	**100.0%**

Figure 6: Sixteen-bit acquisition and query errors (a–c); exhaustive and held-out eight-bit retrieval (d). Counts record correct/total answers. At the selected sixteen-bit checkpoint, the writer makes 25 errors in 131,072 checks.

5 Discussion

A detokenization fiber can act as memory. Each matched cell stores one independent value, and the reader can access a requested value after the write is complete. The matched count and histogram make the state specifically sequential: the token inventory stays the same as the stored word changes.

The interface determines whether this state persists. A direct token-ID handoff preserves the chosen representative. Decode–canonical-encode identifies all representatives with the same expansion, and the full-family intervention measures the resulting erasure with the reader held fixed. The visible control shows the complementary operation: distinct text states survive canonicalization and remain readable after calibration. Thus preserving displayed content and preserving the state used by a token-level reader are separate properties of the handoff.

The factorization criterion makes this distinction useful beyond the particular cell layout. A reader that factors through decoded text gives one answer per surface and query; canonicalization enforces this property by construction. Conversely, answering different message bits from the same surface requires access to the representative. For a system carrying intentional token-level metadata, that choice is part of the transmitted state. For a boundary specified to carry decoded content alone, canonical encoding selects a reproducible text-level input. The relevant interface specification therefore includes the representation delivered to the receiver as well as the string displayed to a person.

This distinction also determines what a reproducible communication record contains. A decoded-text record identifies the surface. A token-ID record additionally identifies the representative selected by the writer. Replaying the latter preserves the message-bearing cell choices; canonical encoding of the former selects the common representative. Hidden Tape makes these two handoffs explicit through a message that can be queried repeatedly, giving a concrete test of which representation an interface actually transmits.

6 Conclusion

Hidden Tape establishes writable, addressable message state beneath identical decoded text. Matched token-order cells preserve length and histogram while carrying bits that a trained reader retrieves after commitment. Scalar correctness rewards train a writer to use this channel, reaching 99.98% query accuracy at sixteen bits. Targeted cell changes control the corresponding answers, and canonical re-encoding erases the message while preserving every displayed byte. The communication interface therefore determines whether the receiver obtains the stored token-level message or its shared textual surface.

Reproducibility statement

Appendices specify the constructive proofs, exact carrier IDs, the writer's reward boundary, calibration and optimization settings, evaluation populations, and checkpoint-selection rule. Quantitative figures are generated from the released evaluation files using the accompanying plotting sources.

Ethics statement

The experiments use public models and synthetic messages. Token-preserving communication is dual use: the same representative can carry intentional metadata or information absent from the displayed text. Deterministic decode–canonical-encode removes the fixed-surface message variable studied here.

AI use statement

Generative AI assisted development of the theoretical framework, mathematical claims and proofs, synthetic experimental design, implementation, and interpretation in the source research. It also assisted evidence synthesis, literature review, proof exposition, figure code, and manuscript drafting in preparing this version.

References

Sahar Abdelnabi, Kai Greshake, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In Maura Pintor, Xinyun Chen, and Florian Tramèr (eds.), *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, AISec 2023, Copenhagen, Denmark, 30 November 2023*, pp. 79–90. ACM, 2023. doi: 10.1145/3605764.3623985. URL `https://doi.org/10.1145/3605764.3623985`.

Fredrik Andersson and Marcus Carlsson. On the structure of positive semi-definite finite rank general domain hankel and toeplitz operators in several variables. *Complex Analysis and Operator Theory*, 11(4):755–784, 2017.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 4617–4624. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.414. URL `https://doi.org/10.18653/v1/2020.findings-emnlp.414`.

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible NLP attacks. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pp. 1987–2004. IEEE, 2022. doi: 10.1109/SP46214.2022.9833641. URL `https://doi.org/10.1109/SP46214.2022.9833641`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Yuanpu Cao, Bochuan Cao, and Jinghui Chen. Stealthy and persistent unalignment on large language models via backdoor injections. In Kevin Duh, Helena Gómez-Adorno, and Steven

Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 4920–4935. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.276. URL https://doi.org/10.18653/v1/2024.naacl-long.276.

Marco Cognetta and Naoaki Okazaki. Tokenization as finite-state transduction. *CoRR*, abs/2410.15696, 2024. doi: 10.48550/ARXIV.2410.15696. URL https://doi.org/10.48550/arXiv.2410.15696.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. ISBN 978-0-471-24195-9. URL http://www.elementsofinformationtheory.com/.

Martin Engert. Finite dimensional translation invariant subspaces. *Pacific Journal of Mathematics*, 32(2):333–343, 1970.

Tina Fang, Martin Jaggi, and Katerina J. Argyraki. Generating steganographic text with lstms. In Allyson Ettinger, Spandana Gella, Matthieu Labeau, Cecilia Ovesdotter Alm, Marine Carpuat, and Mark Dredze (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Student Research Workshop*, pp. 100–106. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-3017. URL https://doi.org/10.18653/v1/P17-3017.

Juan Luis Gastaldi, John Terilla, Luca Malagutti, Brian DuSell, Tim Vieira, and Ryan Cotterell. The foundations of tokenization: Statistical and computational concerns. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=B5iOSxM2I0.

Renato Lui Geh, Zilei Shao, and Guy Van den Broeck. Adversarial tokenization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 20738–20765. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1012. URL https://doi.org/10.18653/v1/2025.acl-long.1012.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.751. URL https://doi.org/10.18653/v1/2023.emnlp-main.751.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. *CoRR*, abs/2510.06182, 2025. doi: 10.48550/ARXIV.2510.06182. URL https://doi.org/10.48550/arXiv.2510.06182.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kanishk Jain, Matthew Day, and Tankut Can. Emergent retokenization symmetry in large language models: phenomenology and applications. *CoRR*, abs/2606.15521, 2026. doi: 10.48550/ARXIV.2606.15521. URL https://doi.org/10.48550/arXiv.2606.15521.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8018–8025. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6311. URL https://doi.org/10.1609/aaai.v34i05.6311.

Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. Meteor: Cryptographically secure steganography for realistic distributions. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (eds.), *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pp. 1529–1548. ACM, 2021. doi: 10.1145/3460120.3484550. URL https://doi.org/10.1145/3460120.3484550.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=FpaCL1MO2C.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 66–75. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-1007. URL https://aclanthology.org/P18-1007/.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *CoRR*, abs/2209.11895, 2022. doi: 10.48550/ARXIV.2209.11895. URL https://doi.org/10.48550/arXiv.2209.11895.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. Bpe-dropout: Simple and effective subword regularization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 1882–1892. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.170. URL https://doi.org/10.18653/v1/2020.acl-main.170.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer. *CoRR*, abs/2509.23886, 2025. doi: 10.48550/ARXIV.2509.23886. URL https://doi.org/10.48550/arXiv.2509.23886.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pp. 5149–5152. IEEE, 2012. doi: 10.1109/ICASSP.2012.6289079. URL https://doi.org/10.1109/ICASSP.2012.6289079.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1162. URL https://doi.org/10.18653/v1/p16-1162.

Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948. doi: 10.1002/J.1538-7305.1948.TB01338.X. URL https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 2089–2103. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.160. URL https://doi.org/10.18653/v1/2021.emnlp-main.160.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2153–2162. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1221. URL https://doi.org/10.18653/v1/D19-1221.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Yaofei Wang, Rui Wang, Weilong Pang, JiaLiang Han, Yuan Qi, Donghui Hu, and Kejiang Chen. Retoksync: Self-synchronizing tokenization disambiguation for generative linguistic steganography. *CoRR*, abs/2604.25486, 2026. doi: 10.48550/ARXIV.2604.25486. URL https://doi.org/10.48550/arXiv.2604.25486.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Ruiyi Yan and Yugo Murawaki. Addressing tokenization inconsistency in steganography and watermarking based on large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 7076–7098. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.361. URL https://doi.org/10.18653/v1/2025.emnlp-main.361.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin

Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Zhongliang Yang, Xiaoqing Guo, Zi-Ming Chen, Yongfeng Huang, and Yu-Jin Zhang. Rnn-stega: Linguistic steganography based on recurrent neural networks. *IEEE Trans. Inf. Forensics Secur.*, 14(5):1280–1295, 2019. doi: 10.1109/TIFS.2018.2871746. URL https://doi.org/10.1109/TIFS.2018.2871746.

Zachary M. Ziegler, Yuntian Deng, and Alexander M. Rush. Neural linguistic steganography. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 1210–1215. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1115. URL https://doi.org/10.18653/v1/D19-1115.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023. doi: 10.48550/ARXIV.2307.15043. URL https://doi.org/10.48550/arXiv.2307.15043.

A Constructive proofs

Theorem 1 (Matched binary tape)**.** *Let $h : \Gamma^* \to \Sigma^*$ be a non-erasing expansion homomorphism. If distinct nonempty token words p, q satisfy $h(p) = h(q) = u$, then $P = pq$ and $Q = qp$ are distinct equal-length token words with equal histograms and expansion u^2. For every integer $R \geq 1$, the family $\{P, Q\}^R$ contains 2^R distinct token words sharing expansion u^{2R}, length $R(|p| + |q|)$, and histogram $R(\text{hist}(p) + \text{hist}(q))$.*

Proof. We first show that commuting nonempty words are powers of a common nonempty word. Induct on their total length. If $w_1 w_2 = w_2 w_1$ and $|w_1| = |w_2|$, their prefixes of length $|w_1|$ give $w_1 = w_2$. If $|w_1| > |w_2|$, equality implies $w_1 = w_2 c$ for a nonempty c. Substitution and cancellation yield $cw_2 = w_2 c$. Induction expresses c and w_2 as powers of one word, and hence so does $w_1 = w_2 c$. The case $|w_2| > |w_1|$ is symmetric.

Now suppose $pq = qp$. Write $p = v^\alpha$ and $q = v^\beta$ for positive integers α, β. The non-erasing hypothesis gives $|h(v)| > 0$, and $h(p) = h(q)$ implies $\alpha|h(v)| = \beta|h(v)|$. Therefore $\alpha = \beta$ and $p = q$, a contradiction. Thus $P \neq Q$. Their expansions, lengths, and histograms agree by concatenation. Each member of $\{P, Q\}^R$ can be parsed into blocks of common token length $|P|$. Distinct binary choices differ in at least one such block, so there are exactly 2^R distinct members. Their shared invariants follow by adding the per-block counts. □

Proposition 1 (Exact reusable retrieval requires distinct tapes)**.** *Let $W : \{0,1\}^R \to \Gamma^*$ and $D : \Gamma^* \times \{0, \ldots, R-1\} \to \{0,1\}$ be deterministic. If $D(W(x), q) = x_q$ for every x, q, then W is injective.*

Proof. For $x \neq x'$, choose q with $x_q \neq x'_q$. If $W(x) = W(x')$, the single value $D(W(x), q)$ would equal both distinct targets. Hence $W(x) \neq W(x')$. □

Proposition 2 (Canonical reset)**.** *Let T be a deterministic exact encoder on the carrier domain. If $Z \in F_h(s)$ for one fixed surface s, then $T(h(Z)) = T(s)$ surely. For a uniform binary word X and independent query K, the conditional law of X_K given $T(h(Z))$ and K is Bernoulli$(1/2)$.*

Proof. The first identity follows from the definition of the fiber. At any fixed query, the canonical tape is a constant and the target coordinate is balanced. A deterministic reader therefore answers exactly half of an exhaustive balanced population correctly. Averaging over a randomized reader's prediction gives the same expectation. □

These arguments identify the precise invariant used by the experiment. A text observer receives a constant at a fixed width, while the token observer can receive a distinct representative for every word. A deterministic audit A with at most M outputs on a finite tape family F also has an output class of size at least $|F|/M$, by partitioning F into its nonempty preimages. In particular, the length-and-histogram audit is constant on the constructed family. With $M = 1$, the audit places the entire constructed codebook in a single class.

B Carrier identities and census population

The Qwen carrier paths, in token-ID notation, are

$$p = (67, 68, 69, 220, 69, 7, 87, 8, 25),$$
$$q = (67, 68, 69, 220, 69, 7, 87, 1648).$$

Both expand to `def f(x):`. The two cells concatenate these paths in opposite order. Their common histogram has multiplicities $67 : 2$, $68 : 2$, $69 : 4$, $220 : 2$, $7 : 2$, $87 : 2$, $8 : 1$, $25 : 1$, and $1648 : 1$. Multiplying these counts by R gives every tape's histogram.

For GPT-2 (Radford et al., 2019), the two paths are

$$p = (71, 68, 75, 75, 78, 220, 86, 78, 81, 75, 67),$$
$$q = (71, 68, 75, 75, 78, 220, 86, 78, 81, 335).$$

Both expand to `hello world`. Their matched cell has 21 tokens and expands to 22 bytes. Eight cells contain 168 tokens and 176 bytes. In both families, direct input-ID concatenation preserves the chosen noncanonical sequence. With the GPT-2 cell, a wrapper example surrounds a matched `hello world` cell with fixed prose, producing `Note that hello worldhello world is already defined.` with 28 tokens in either representative. It illustrates composition with fixed surrounding text.

B.1 Enumerating ambiguous candidate strings

The census builds a byte-string candidate set from twenty corpus strings, five templates, printable short token expansions, and ordered concatenations of short expansions. It adds the first 4,000 qualifying individual expansions of length at most twelve and all ordered pairs from the first 350, then deduplicates byte strings and keeps lengths from one to 96. For each candidate, dynamic programming counts token paths by total token length. A string is counted as ambiguous when it admits at least two distinct non-erasing paths.

The Qwen table uses decoded UTF-8 expansions of the first 50,000 token IDs; 128 of its first 350 expansions are lone bytes that decode to the same replacement character, so deduplication leaves fewer Qwen candidates than GPT-2 candidates. The GPT-2 table uses the vocabulary's byte mapping, which keeps all 350 distinct. Table 2 counts candidate strings and ambiguous strings from this procedure. Path enumeration saves up to eight witnesses per string. We also check whether the first two enumerated paths already match in length and histogram; applying Eq. (2) makes any pair matched. Applying the matched-cell construction gives the Qwen and GPT-2 cells above, each with equal bytes, token count, and histogram in both orientations.

Table 2: Candidate-string census. An ambiguous string has multiple token paths.

Expansion table	Candidate byte strings	Ambiguous strings
Qwen, first 50,000 token IDs	51,149	43,918
GPT-2 byte vocabulary	123,293	115,403

C Calibration, reward optimization, and evaluation

Models and readout. The learned communication backbone is Qwen/Qwen2.5-0.5B-Instruct (Yang et al., 2024), with source snapshot identifier `7ae557604adf67be50417f59c2c2f167def9a775`. Rank-eight LoRA (Hu et al., 2022) targets `q_proj`, `k_proj`, `v_proj`, and `o_proj`, with alpha 16 and dropout 0.05. Backbone execution uses bfloat16; the binary head is float32. The head maps the last token's final hidden state to one logit. Reader training minimizes binary cross-entropy, uses gradient clipping at norm one, and optimizes the adapters and head with AdamW at learning rate 2×10^{-4}. Direct IDs are passed without a chat wrapper.

Reader calibration uses 2,000 updates at four bits, 4,000 at eight bits, and 6,000 at twelve bits, with eight sampled word–query pairs per update. The sixteen-bit calibration uses one sampled word expanded to all sixteen queries per update. Evaluations are scheduled at updates 8,000, 16,000, 32,000, and 64,000, with early stopping when both mean and worst-query accuracy reach 0.999. The reported reader stops at 16,000 updates with exact sampled all-query accuracy. Reader qualification requires mean and worst-query accuracy at least 0.95; writer training starts from the restored reader. The width-study writer population is seeds 0/1/2 at four and eight bits, 0/2 at twelve, and 0 at sixteen.

Writer and reward boundary. The writer's three linear layers have dimensions $R \to 128 \to 128 \to R$, with ReLU after each hidden layer. Its input is the binary word. Bernoulli sampling produces one hard code, after which the environment draws the query and calls the frozen reader. The environment exposes scalar correctness to the score-function update. The baseline uses its previous value when forming the advantage and is updated with exponential rate 0.05 after the batch. Writer AdamW uses learning rate 10^{-3}, weight decay zero, batch size eight, and gradient clipping at norm one. The writer random generator is reset with training seed plus 1,000,003 after reader loading. Periodic evaluation saves and restores the random state.

Protocol and query format. The fixed permutation and mask are generated at protocol seed 20260815. At eight bits the physical-cell-to-logical-coordinate permutation is $(4, 5, 6, 2, 3, 0, 7, 1)$, whose inverse maps queries to cells $(5, 7, 3, 4, 0, 1, 2, 6)$. Qwen queries use `Q0` through `Q7`; at twelve and sixteen bits they use zero-padded addresses `Q00` onward. The complete query table is verified to have common token length within each width. The frozen GPT-2 readout study uses openai-community/gpt2 (Radford et al., 2019), its matched `hello world` carrier, protocol seed 20260816, and digit queries zero through seven.

Frozen-model probes. For each physical cell, hidden states are averaged over its token positions. The reported local comparison uses hidden-state index one, after the first transformer block. Given even-parity class means μ_0, μ_1, the readout direction is $v = \mu_1 - \mu_0$ and the threshold is $v^\top(\mu_0 + \mu_1)/2$. Classification uses this direction and midpoint on the odd-parity vectors. Answer-site probes use the same fitting rule on the last query token's final hidden vector. The query-by-source matrix fits one classifier for each query context and source bit. All fits use 128 even words and all tests use 128 odd words, with 64 examples of each label in each half.

Where order enters in the frozen Qwen models. A cell's mean token embedding cannot separate its two orientations, and normalization and MLP layers act position-wise, so in the Qwen2.5 probes the order signal enters through attention. Qwen2.5 applies rotary position encodings to queries and keys (Yang et al., 2024): position m rotates each two-dimensional block by angle $m\theta_\ell$. For fixed query and key vectors, the attention logit is therefore a function of the offset $n - m$ lying in the finite-dimensional translation-invariant span of the exponentials $e^{\pm \mathrm{i}\theta_\ell(n-m)}$; such spaces consist of exponential polynomials (Engert, 1970). When the query equals the key, the logits $\sum_\ell \|\kappa_\ell\|^2 \cos((n-m)\theta_\ell)$ form a positive semidefinite Toeplitz matrix of rank at most the head dimension, with entries a nonnegative combination of these exponentials, the one-variable case of the finite-rank structure in Andersson & Carlsson (2017). Swapping the two paths of a cell preserves its histogram but in general changes which token pairs meet at each offset and, under the causal mask, which tokens each position attends to. The first block can therefore separate the two orientations, although the embedding mean cannot.

C.1 Sixteen-bit sampling and selection

For accuracy, the sampler builds 256 blocks of 32 words each. Each block enumerates all five-bit vectors and maps them through sixteen distinct nonzero five-bit linear forms, adding a random binary mask. Every form is balanced, and each pair of distinct forms is linearly independent over the binary field. Thus every output coordinate is balanced and every coordinate pair has each binary combination equally often in a block. Independently randomized forms and masks are drawn for successive blocks, and the resulting instances are shuffled. The accuracy seed is 20260814. All sixteen queries are evaluated for each of the 8,192 word instances, giving 131,072 checks.

Entropy uses seed 20260816 and 262,144 antithetic instances sampled with replacement: each randomly drawn word is paired with its bitwise complement. The empirical entropy is $-\sum_c \hat{p}(c) \log_2 \hat{p}(c)$, where $\hat{p}(c)$ is the output-code frequency on that sample. Accuracy-set distinct codewords and entropy-sample frequencies have different denominators. In particular, the selected sixteen-bit accuracy set produces 7,701 distinct codes; its empirical entropy is computed on the separate larger sample.

The joint selection rule uses the same scheduled evaluation population at each stored point. It selects the first point with exact-code agreement, worst-query accuracy, and normalized empirical entropy all at least 0.95. The sixteen-bit writer is training seed zero at update 64,000. Figure 5 displays each run's positive-reward records through this rule's endpoint on a logarithmic reward axis.

D Selected per-seed and per-coordinate values

Table 1 brings readout accuracy, code agreement, and entropy together at the same selected checkpoints. Accuracy evaluates 16, 256, 4,096, and 8,192 word instances at widths four, eight, twelve, and sixteen, respectively. Testing every query gives 64, 2,048, 49,152, and 131,072 checks. Entropy uses the exhaustive populations through twelve bits and a separate 262,144-instance sample at sixteen

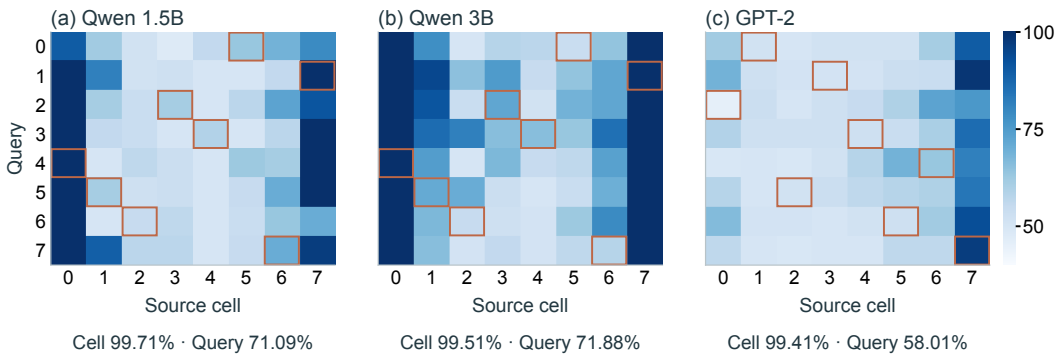


Figure 7: Query/source accuracy on 128 held-out odd-parity words after fitting on 128 even-parity words. Orange outlines mark requested sources; bottom labels summarize local-cell and final-query means.

bits. Code agreement compares the complete writer output with Eq. 4; entropy measures diversity across output tapes.

Table 3 retains the exact sixteen-bit counts by address. Their sum is 131,047 and their minimum is 8,177, while 8,167 of the 8,192 predicted complete codes agree with the convention. The query and code measurements use the same selected set of sampled word instances.

Table 3: Correct sixteen-bit answers by query, each out of 8,192 word instances.

Query	Correct	Query	Correct	Query	Correct	Query	Correct
0	8,192	4	8,191	8	8,192	12	8,177
1	8,191	5	8,191	9	8,192	13	8,190
2	8,190	6	8,192	10	8,191	14	8,192
3	8,192	7	8,190	11	8,192	15	8,192

The even-to-odd experiment uses 32,000 rewards for each writer. Table 4 reports the training-support and held-out halves as well as their union. All three writers retrieve the held-out half exactly. Its 128 distinct codes have seven bits of empirical entropy, matching the size of that half-cube.

Table 4: Eight-bit parity split: correct answers / all-query checks.

Seed	Even support	Odd held out	Full cube	Odd exact codes
0	1,024/1,024	1,024/1,024	2,048/2,048	128/128
1	1,023/1,024	1,024/1,024	2,047/2,048	128/128
2	1,024/1,024	1,024/1,024	2,048/2,048	128/128

Canonicalization geometry. The fixed-surface intervention has 256 direct tapes of length 136 and one canonical tape of length 64. The visible control has 256 direct tapes of length 136 and 256 canonical tapes. Its canonical-length histogram is 64:1, 65:36, 66:126, 67:84, and 68:9. Visible-control readers are separately trained for 4,000 updates on canonical tapes, with the same rank-eight adapter configuration, query convention, binary objective, and sampled-pair calibration. Both experiments evaluate all 256 words and eight queries for each of three reader seeds.