Supplementary Information Guide

TABLE OF CONTENTS

SUPPLEMENTARY NOTES TITLES

Supplementary Note 1. Greedy alignment between pulse-token relationships and submarine-state relationships

Supplementary Section 2. Inferring parts of speech (PoS) from MNN pulse-token embeddings

SUPPLEMENTARY FIGURES' TITLES AND LEGENDS

Fig S1. Original Token-Association Matrix (b) showing the accuracy of recovering the first pulse-token (A) from the second (B) by using the MNN and ordered by their initial RF-token indices. The structure is quite arbitrary. A greedy joint permutation of rows and columns attempts to align the embedding vectors to be similar to the ones in the State Compatibility Matrix (a), with similar structure emerging along the main diagonal (c).

Fig S2. Mapping digital word-embedding relationships into microwave pulse-train tokens. a, A constrained English vocabulary (**a.i**) of 25 words (five categories: nouns, verbs, adverbs, adjectives, prepositions) is used to generate short, syntactically valid sentence-fragments from predefined grammatically correct templates. A language model consisting of a trainable embedding layer and an LSTM decoder is trained to give embedding for each word in the dictionary. Pairwise cosine similarities between embeddings provide a Language Embedding matrix (**a.ii**). **b,** Microwave pulse-train token space is independently defined for the MNN by three physical degrees of freedom: token amplitude, token pulse frequency and token duration (**b.i**). When fed into the Microwave Neural Network, the nonlinear coupling of waveguides produces a broadband frequency-comb response, encoding features of the tokens as spectrograms. A linear classifier is trained to predict token one from the other from these spectrograms to provide a Pulse-token Association Matrix (**b.ii**), quantifying spectral similarity among pulse tokens. **c,** Greedy and random matching algorithms attempt to select one-to-one mappings for the 25 words by maximizing similarity in relationships in the Word Embedding and the MNN's Pulse-Token Association matrices.

Fig S3. MNN extracts grammatical structure from a sequence of microwave pulses. a.i, Each English word is represented by a corresponding microwave pulse-train token. Tokens are injected into the MNN sequentially as word-pairs and the nonlinear microwave response is measured from the output port as a spectrogram. A trained linear classifier (ridge regularized multi-class logistic regression) processes these spectrograms to infer the syntactic category (Part-of-Speech) of the first and second tokens. **a.ii,** Confusion matrices show PoS prediction across five grammatical classes, with strong on-diagonal values indicating reliable discrimination enabled by the MNN's broadband expansion. **b,** Word-at-a-time sentence-continuation game tests contextual understanding. MNN predicts the two PoS. Each word must follow a valid grammatical transition (based on valid templates in Supplementary Table 1) using the predicted PoS. The next words are cascaded with the previous ones and fed to the MNN. The process continues recursively to extend the length of the sentence-fragment. The game ends if the MNN predicts an incorrect PoS, or no valid next word exists. **c,** Evaluating mapping and syntactic inference capability. **c.i,** Comparison of greedy-search vs random-search mapping between 25 English words and 25 pulse-train tokens, evaluated over 10,000 attempts to make valid sentence-fragments. Greedy search completes more valid fragments. **c.ii,** Sentence-generation success for MNN + linear classifier versus a baseline that computes on spectrograms produced from raw tokens. MNN's nonlinear spectral expansion enables reliable inference up to 7 words in a valid sentence fragment template, while the baseline rarely exceeds 4 words.

Fig S4. The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s (Sets 1-11). For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S = (# static states) / (300 repetitions). Dynamic ratio, T = 1 − S. 8-bit patterns are arranged in order of decreasing Static Ratio.

SUPPLEMENTARY TABLES' TITLES AND LEGENDS

Supplementary Table 1| Mappings extracted from pulse-train tokens to submarine states in the game shown in Figs. 3 and 4. i.e., (β_{Tok} in Volts, F_{Tok} in MHz, T_{Tok} in ns) → (X grid units, Y grid units, θ radians)

Supplementary Table 2| Mappings extracted from pulse-train tokens to word tokens in the sentence-building game shown in Supplementary Figs. 2 and 3. i.e., (β_{Tok} in Volts, F_{Tok} in MHz, T_{Tok} in ns) → word

Supplementary Table 3| Syntactically valid sentence-fragment templates for generating a training set for mapping word-embeddings to pulse-train tokens in Supplementary Note 2.

Supplementary Table 4| Breakdown of power consumption in the feature extraction front-end.

1. Greedy alignment between pulse-token relationships and submarine-state relationships

In the turn-based game discussed in the main Article, ideally, one would want a one-to-one alignment between a subset of microwave pulse tokens generated by the Microwave Neural Network (MNN) and a fixed menu of states the submarines could assume. Instead, the MNN produces an empirical Association Matrix over amplitude–frequency–time (β_{Tok}, F_{Tok}, T_{Tok}) parameters and the submarine-game engine gives us a precomputed State Compatibility Matrix. The goal of the matching procedure is to identify a subset of (β_{Tok}, F_{Tok}, T_{Tok}) tokens and an ordering of that subset whose pairwise interactions best reproduce the similarity in the structure of the embedding vectors in the State Compatibility Matrix.

Let $C \in \mathbb{R}^{n\times n}$ denote the MNN Association Matrix, where each index $i \in \{1,\ldots,n\}$ corresponds to a particular (β_{Tok}, F_{Tok}, T_{Tok}) token. The entry C_{ij} is a non-negative confidence or interaction score between tokens i and j, obtained from the MNN experiments. Let $S^{\mathrm{cont}} \in \mathbb{R}^{B\times B}$ denote the State Compatibility Matrix over a fixed set of B word tokens. In practice, we work with a binarized version

$$S_{ij} \;=\; \mathbb{I}\big(S^{\mathrm{cont}}_{ij} > \sigma\big)\,, \tag{1}$$

where σ is a compatibility threshold (e.g. $\sigma = 0.1$) and $\mathbb{I}(\cdot)$ is the indicator function. The resulting matrix $S \in \{0,1\}^{B\times B}$ encodes which word pairs are considered to be "compatible" with respect to the objective of the game, which is to always intercept the sonar beam. For the example in Supplementary Fig. 1, we fix the states' vocabulary size to $B = 25$, so S is a 25×25 binary matrix whose row and column order is held fixed throughout the matching procedure. Given a subset of (β_{Tok}, F_{Tok}, T_{Tok}) indices $I \subseteq \{1,\ldots,n\}$ of size $k = |I|$, we consider the corresponding MNN submatrix

$$C^{(k)} \;=\; C[I,I] \in \mathbb{R}^{k\times k}. \tag{2}$$

To compare this with the State Compatibility Matrix, we also take the top-left $k \times k$ block of S,

$$S^{(k)} \;=\; S[1\!:\!k,\, 1\!:\!k] \in \{0,1\}^{k\times k}. \tag{3}$$

We introduce a confidence threshold τ and define a binarized version of the MNN submatrix via

$$\tilde{C}^{(k)}_{ij} \;=\; \begin{cases} 1, & C^{(k)}_{ij} \ge \tau, \\ 0, & \text{otherwise,} \end{cases} \qquad i,j \in \{1,\ldots,k\}. \tag{4}$$

Thresholds were chosen to balance sparsity and structural overlap; moderate variations did not qualitatively change the resulting mappings.

Given a pair of $k\times k$ binary matrices $(\tilde{C}^{(k)}, S^{(k)})$, we define the match score as the number of positions where both matrices equal 1:

$$\mathrm{score}\Big(\tilde{C}^{(k)}, S^{(k)}\Big) \;=\; \sum_{i=1}^{k}\sum_{j=1}^{k} \mathbb{I}\Big(\tilde{C}^{(k)}_{ij} = 1 \;\wedge\; S^{(k)}_{ij} = 1\Big)\,. \tag{5}$$

Intuitively, this counts how many pairwise "edges" in the thresholded MNN graph coincide with edges in the State-Compatibility graph, restricted to the current subset and ordering. For a fixed subset I of size k, there is no canonical ordering of the selected (β_{Tok}, F_{Tok}, T_{Tok}) tokens. To maximize the alignment with $S^{(k)}$, we allow the rows and columns of $C^{(k)}$ to be permuted jointly. We first construct a cost matrix

$$\text{cost}_{ij} \;=\; -\mathbb{I}\Big(\tilde{C}^{(k)}_{ij} = 1 \;\wedge\; S^{(k)}_{ij} = 1\Big), \qquad i, j \in \{1, \ldots, k\}. \tag{6}$$

Entries where both matrices have a 1 receive cost -1, and all other entries receive cost 0. Minimizing the total cost is therefore equivalent to maximizing the number of overlapping ones. We then apply the Hungarian Algorithm [**Ref. S1**] to this cost matrix to obtain a pair of index permutations (π_r, π_c), which defines a reordered submatrix $\hat{C}^{(k)}_{ij} \;=\; C^{(k)}_{\pi_r(i),\,\pi_c(j)}$. The match score used for evaluation of this subset is then $\text{score}\Big(\tilde{\hat{C}}^{(k)}, S^{(k)}\Big)$, where $\tilde{\hat{C}}^{(k)}$ is the thresholded version of $\hat{C}^{(k)}$ under the same confidence threshold τ.

To select a subset of B (β_{Tok}, F_{Tok}, T_{Tok}) tokens and their alignment to the B submarine states, we maintain two sets: an ordered list of selected (β_{Tok}, F_{Tok}, T_{Tok}) indices, denoted $\mathcal{S}$ and a set of remaining candidate indices, denoted $\mathcal{R}$. Initially,

$$\mathcal{S} \leftarrow [\,], \qquad \mathcal{R} \leftarrow \{1, 2, \ldots, n\}. \tag{7}$$

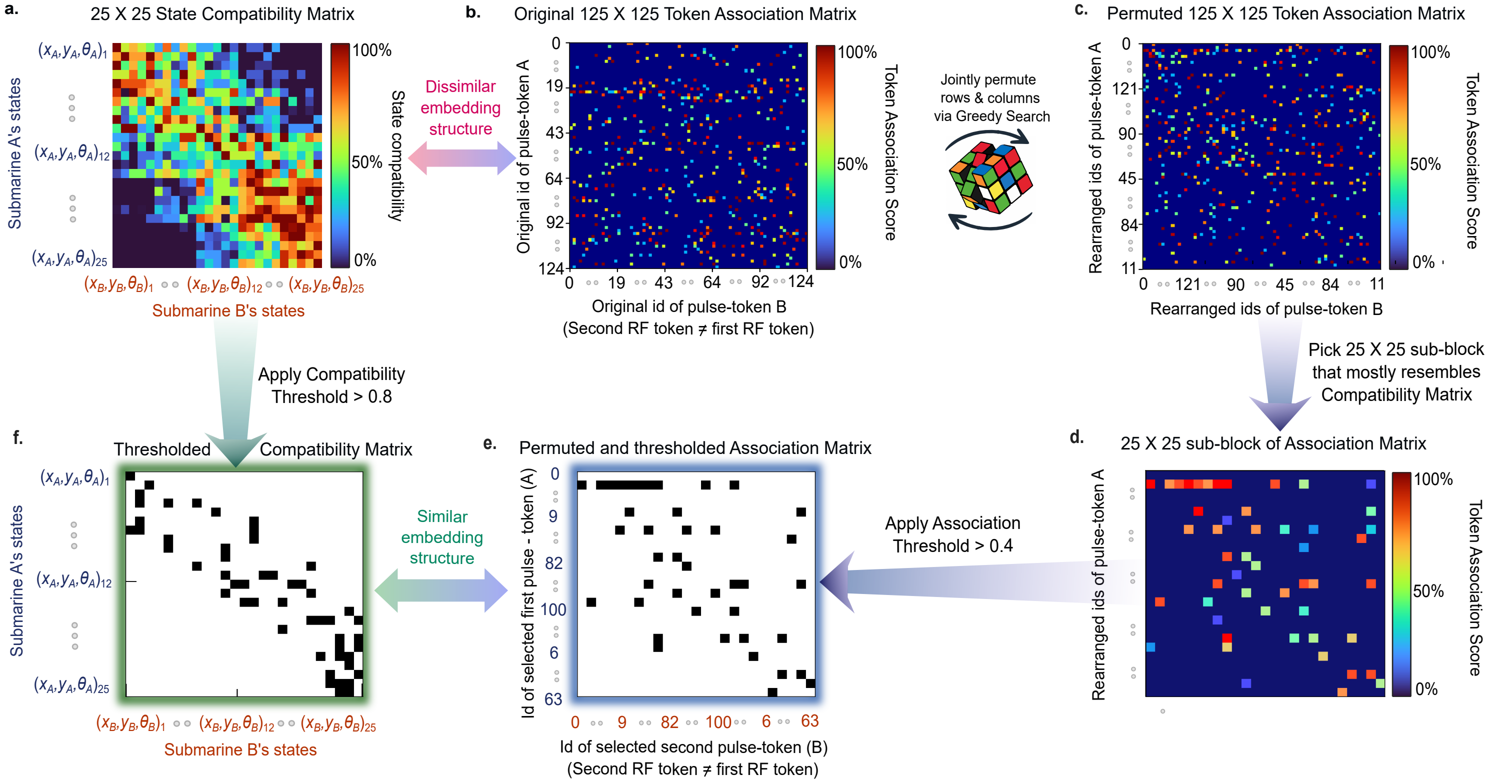
At each iteration, we evaluate adding a single new index $i \in \mathcal{R}$ to the current selection. For each candidate, we form $I_i = \mathcal{S} \cup \{i\}$ with $k_i = |I_i|$, extract the corresponding MNN submatrix $C^{(k_i)} = C[I_i, I_i]$, and compare it to the top-left $k_i \times k_i$ block $S^{(k_i)}$ of the State-Compatibility matrix. Applying the Hungarian algorithm gives permutations $(\pi_r^{(i)}, \pi_c^{(i)})$, the reordered submatrix $\hat{C}^{(k_i)}$, and the match score score_{I_i} from (5). We then select $i^\star = \arg\max_{i \in \mathcal{R}} \text{score}_{I_i}$ and update

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{i^\star\}, \qquad \mathcal{R} \leftarrow \mathcal{R} \setminus \{i^\star\}. \tag{8}$$

This repeats until $|\mathcal{S}| = B$ or the score stops improving. At the final iteration, the Hungarian step yields $(\pi_r^\star, \pi_c^\star)$ and $\hat{C}^{(B)}$, defining the ordering of the B selected tokens. The resulting match score

$$\text{score}_{\text{greedy}} = \text{score}(\tilde{\hat{C}}^{(B)}, S^{(B)})$$

is reported as the greedy performance for the threshold τ. This is only a first-pass method to demonstrate that it is possible to extract meaning from the MNN's feature-extracted Association Matrix. Through this reordering, we can see structure beginning to emerge — for example, the diagonal of the Permuted Association Matrix starts to resemble the State Compatibility Matrix, mimicking that behavior to some extent, as shown in Supplementary Figs. 1 e and f.



Supplementary Figure 1. Procedure to find one-to-one mappings between RF pulse-train tokens and submarine states. The State Compatibility Matrix (**a**) defines which submarine states lead to long survival across many rallies in the signal-interception game. This matrix provides a target embedding structure that must be replicated by the Token Association Matrix (**b**) generated by feeding the MNN with ordered pairs of pulse-train tokens, as described in the main text. To align these representations, we perform a joint row- and column-wise permutation of the Token Association Matrix using a greedy search procedure that iteratively reorders rows and columns to maximize structural overlap with the Compatibility Matrix. This produces a rearranged association matrix (**c**) whose large-scale structure begins to resemble the Compatibility Matrix (note that token indices are relabeled during this process). From the greedily permuted matrix, we then select the 25 × 25 sub-block that most closely matches the Compatibility Matrix (**d**). Finally, by applying appropriate thresholds to both the Compatibility Matrix and the selected, permuted Association sub-matrix, we identify a pair of matrices whose embedding vectors are approximately aligned (**e** and **f**). This alignment enables a reliable one-to-one mapping (like that shown in Main Fig. 3c.iii) between submarine states and pulse-train tokens that can excite the MNN to support optimal feature extraction and decision-making.

2. Inferring parts of speech (PoS) from MNN pulse-token embeddings

A set of microwave pulse-trains that are related to one another may have applications beyond establishing logical context for decision-making at the edge as in the submarine navigation game in the main Article. They could also be used to encrypt meaning in the words of a spoken language. That is to say, long messages could be encrypted, with meaning, in just a few analog pulses of a pulse-train token. For example, the word "boy" is 24 bits long because each of the three letters is represented by one 8-bit byte. In contrast, its meaning could be encoded in a few cycles of a pulse-train token in a way that, when feature-extracted, still maintains context with other words in a message. If we do this and, thereby, transmit with less bandwidth, it could enable an MNN-interpretable communication scheme that remains compatible with the sub-100-Mbps data rates typical of Ka-band satellite links. This will dramatically reduce data volume.

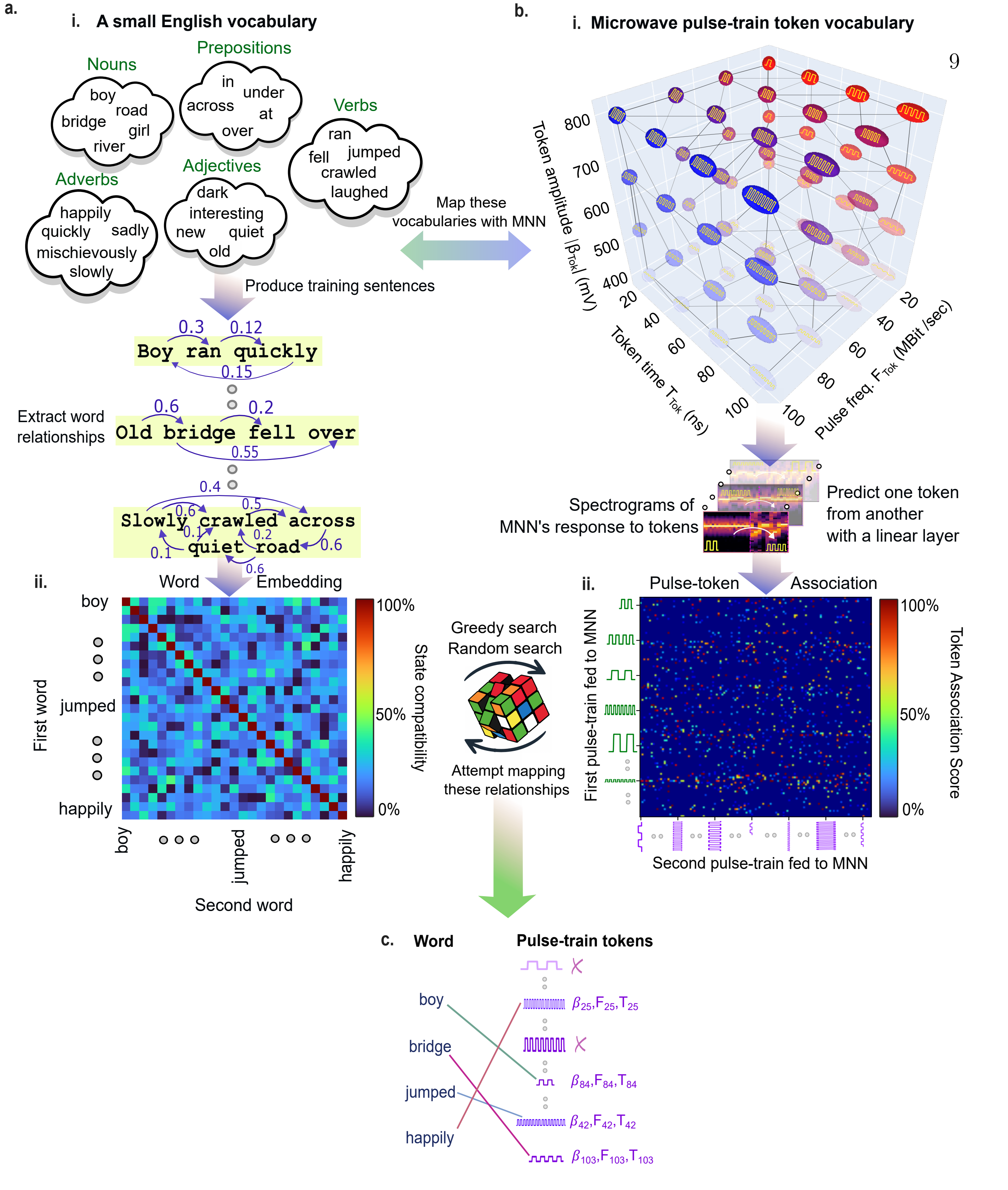
Supplementary Fig. 2 illustrates this process. We train a PyTorch Long Short-Term Memory (LSTM) language model on syntactically valid sentence-fragment ranging from 2 to 7 words in length. For demonstration here, the vocabulary only uses five Parts of Speech (PoS): Noun, Verb, Adverb, Preposition, and Adjective (Fig. S2a.i), and with only five words for each PoS (25 total). Training is performed over these fragments, and the resulting embedding layer provides the learned relationships between words as a cosine-similarity embedding matrix in the model's latent space (Fig. S2a.ii). These fragments follow the set of valid grammatical templates listed below, which define allowable PoS sequences for varying fragment lengths (Supplementary Table 3).

As in the submarine navigation game, we try to draw parallels between words and their context to the relationships embedded in the MNN's token-association matrix extracted from short analog pulses (Supplementary Fig. 2b.ii). We map between them using a few search algorithms (Greedy, Random, etc.) to maximize correspondence between word behavior and their microwave-token counterparts, using a subset of 25 tokens only. These relationships are shown in Supplementary Table 2.

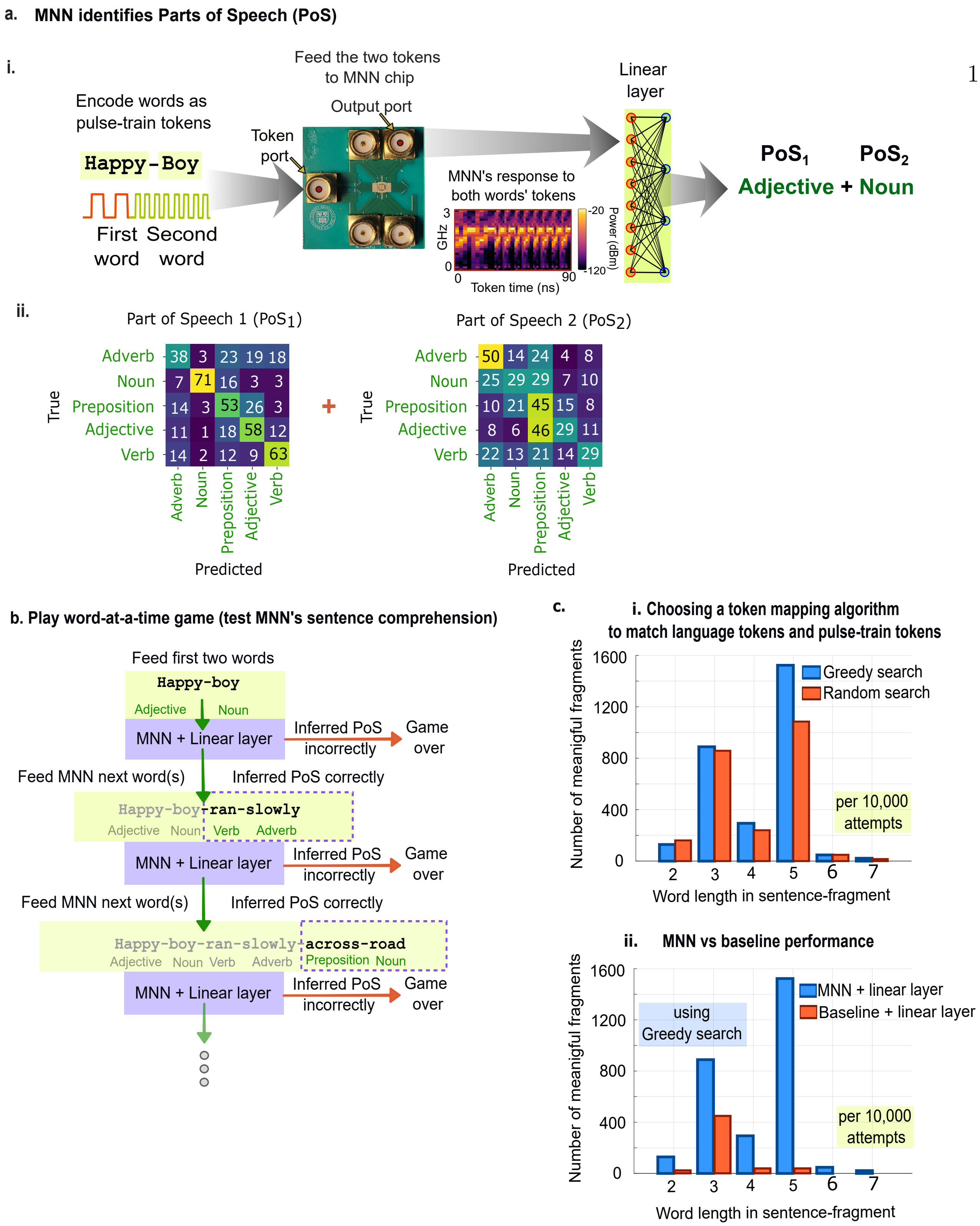
To test whether the above mapping is even useful, we first check if the MNN can recognize the Part of Speech (PoS) of a word injected into it. This training and inference procedure is shown in Supplementary Fig. 3: a linear layer is trained on pairs of words in the vocabulary, injected into the MNN, using 10-fold cross-validation. Each token in a PoS pair is classified from the MNN's spectrogram response (Fig. S3a.i), with ground-truth labels being the two actual PoS types (Noun–Adverb, Adjective–Noun, etc.) as shown in Fig S3a.ii.

We then extend this to test how many times we can correctly classify a PoS in sequence, since this would indicate that MNN-encoded messages can be extended to longer lengths. To evaluate sentence-level comprehension, we created a "word-at-a-time" game. (Fig. S3b). Pairs of token-mapped words are fed sequentially into the MNN, with the output then fed to a trained linear classifier to predict their Parts of Speech. If classification fails, the game ends; if correct, one or two more words are added. If both PoS predictions are correct, we inject another two words from a valid template. For odd-numbered templates, the final single word is given as a "bye" to complete the phrase. This continues until an error occurs (Fig. 3b). From 10,000 trials with attempts to make 2–7-word sentence-fragments, we analyze the MNN's ability to keep extracting the right PoS. It is effectively testing a shared password, where one MNN is used to try to decode a message encoded by another MNN. We first evaluated which algorithms for matching microwave tokens to words worked best by comparing a Greedy Search with a random one-shot assignment between the pulse-train token-association matrix and the word-embedding matrix. We found that Greedy Search, when used in combination with the backend linear classifier, correctly identifies the PoS in sentences up to seven words long (Fig. S3c.i), and performs better than the Random search. The Greedy Search mapping was therefore used for subsequent evaluation. We then compared the MNN's comprehension to a baseline linear model trained directly on raw token spectrograms. The MNN correctly understood sentence structure in over 55% attempts to form syntactically valid fragments, versus fewer than 10% for the baseline (Fig. S3c.ii). Also, the MNN-aided classifier was seen to decipher PoS through full 7-word fragments, whereas the baseline struggled to reach even 4 words.

This exercise shows that, as a proof of concept, the MNN's feature expansion—achieved by instantaneously modulating the bandwidth of its output comb—can be used to interpret short pulses containing linguistic meaning. In this framework, the MNN acts as a physical embedding layer, transforming microwave pulse-trains into semantic feature vectors for downstream language tasks. Although the vocabulary demonstrated here is small, the same principle could be expanded to form the embedding stage of a larger language model.


a.
i. A small English vocabulary
Nouns
boy road bridge girl river
Prepositions
in under across at over
Verbs
ran fell jumped crawled laughed
Adverbs
happily quickly sadly mischievously slowly
Adjectives
dark interesting new quiet old
Map these vocabularies with MNN
Produce training sentences
0.3 0.12
Boy ran quickly
0.15
Extract word relationships
0.6 0.2
Old bridge fell over
0.55
0.4
0.6 0.5
Slowly crawled across
0.1 0.2
0.1 quiet road 0.6
0.6
ii.
Word Embedding
boy
jumped
happily
First word
Second word
100%
50%
0%
State compatibility
Greedy search
Random search
Attempt mapping these relationships
b.
i. Microwave pulse-train token vocabulary
Token amplitude |β_Tok| (mV)
800
700
600
500
400
20 40 60 80 100
Token time T_Tok (ns)
20 40 60 80 100
Pulse freq. F_Tok (MBit /sec)
Spectrograms of MNN's response to tokens
Predict one token from another with a linear layer
ii.
Pulse-token Association
First pulse-train fed to MNN
Second pulse-train fed to MNN
100%
50%
0%
Token Association Score
c.
Word
Pulse-train tokens
boy
bridge
jumped
happily
β25,F25,T25
β84,F84,T84
β42,F42,T42
β103,F103,T103

Supplementary Fig. 2 – **Mapping digital word-embedding relationships into microwave pulse-train tokens.** **a,** A small English vocabulary (**a.i**) of 25 words (five categories: nouns, verbs, adverbs, adjectives, prepositions) is used to generate short, syntactically valid sentence-fragments from predefined grammatically correct templates. A language model consisting of a trainable embedding layer and an LSTM decoder is trained to give embedding for each word in the dictionary(**a.ii**). **b,** Microwave pulse-train token space is independently defined for the MNN by three physical degrees of freedom: token amplitude, token pulse frequency and token duration (**b.i**). When fed into the Microwave Neural Network, the nonlinear coupling of waveguides produces a broadband frequency-comb response, encoding features of the tokens as spectrograms. A linear classifier is trained to predict token one from the other from these spectrograms to provide a Pulse-token Association Matrix (**b.ii**), quantifying spectral similarity among pulse tokens. **c,** Greedy and random matching algorithms attempt to select one-to-one mappings for the 25 words by maximizing similarity in relationships in the Word Embedding and the MNN's token Association matrices (See Supp. Table 2 for word mappings).



Part of Speech 1 (PoS1)

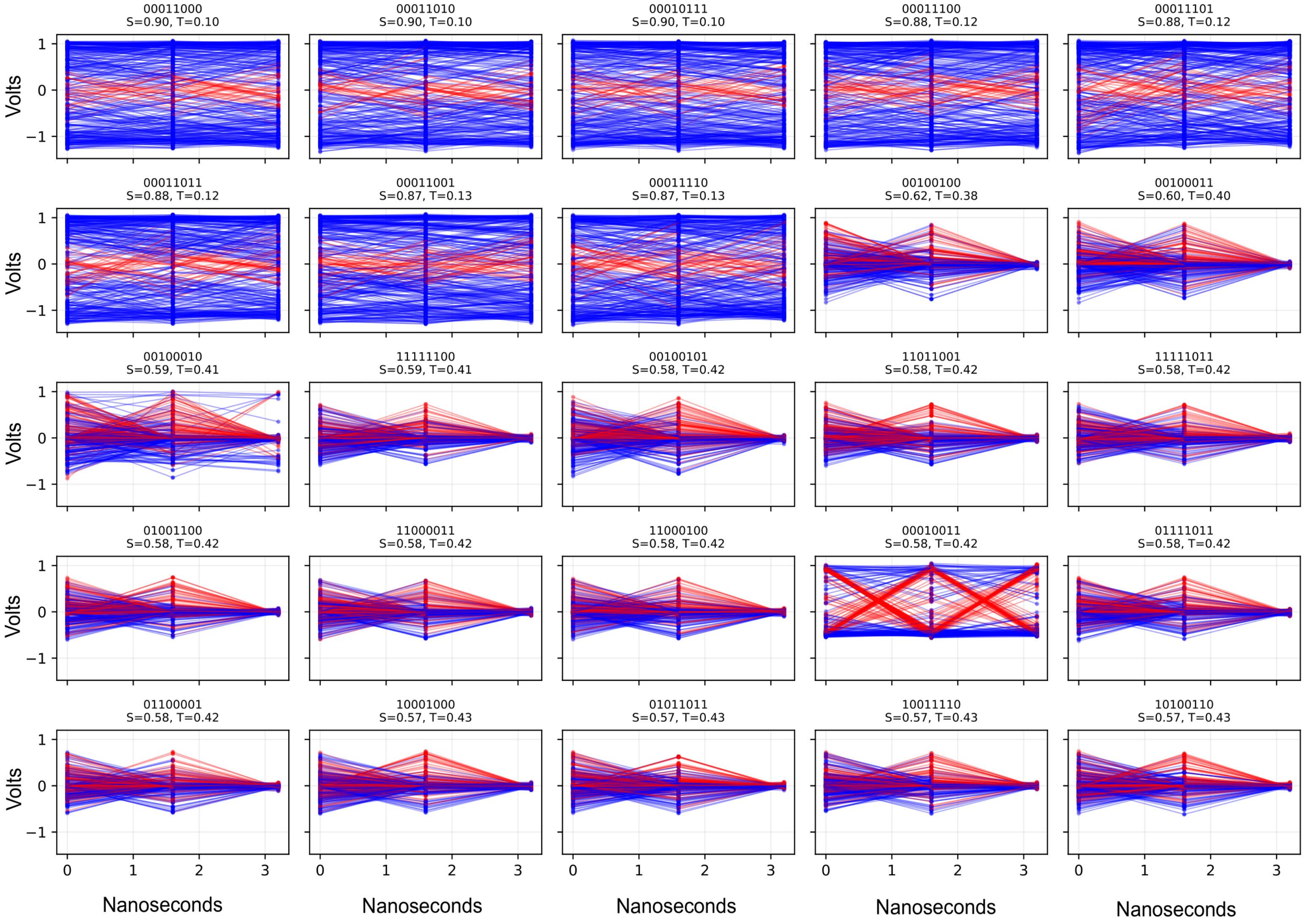
True \ Predicted	Adverb	Noun	Preposition	Adjective	Verb
Adverb	38	3	23	19	18
Noun	7	71	16	3	3
Preposition	14	3	53	26	3
Adjective	11	1	18	58	12
Verb	14	2	12	9	63

Part of Speech 2 (PoS2)

True \ Predicted	Adverb	Noun	Preposition	Adjective	Verb
Adverb	50	14	24	4	8
Noun	25	29	29	7	10
Preposition	10	21	45	15	8
Adjective	8	6	46	29	11
Verb	22	13	21	14	29

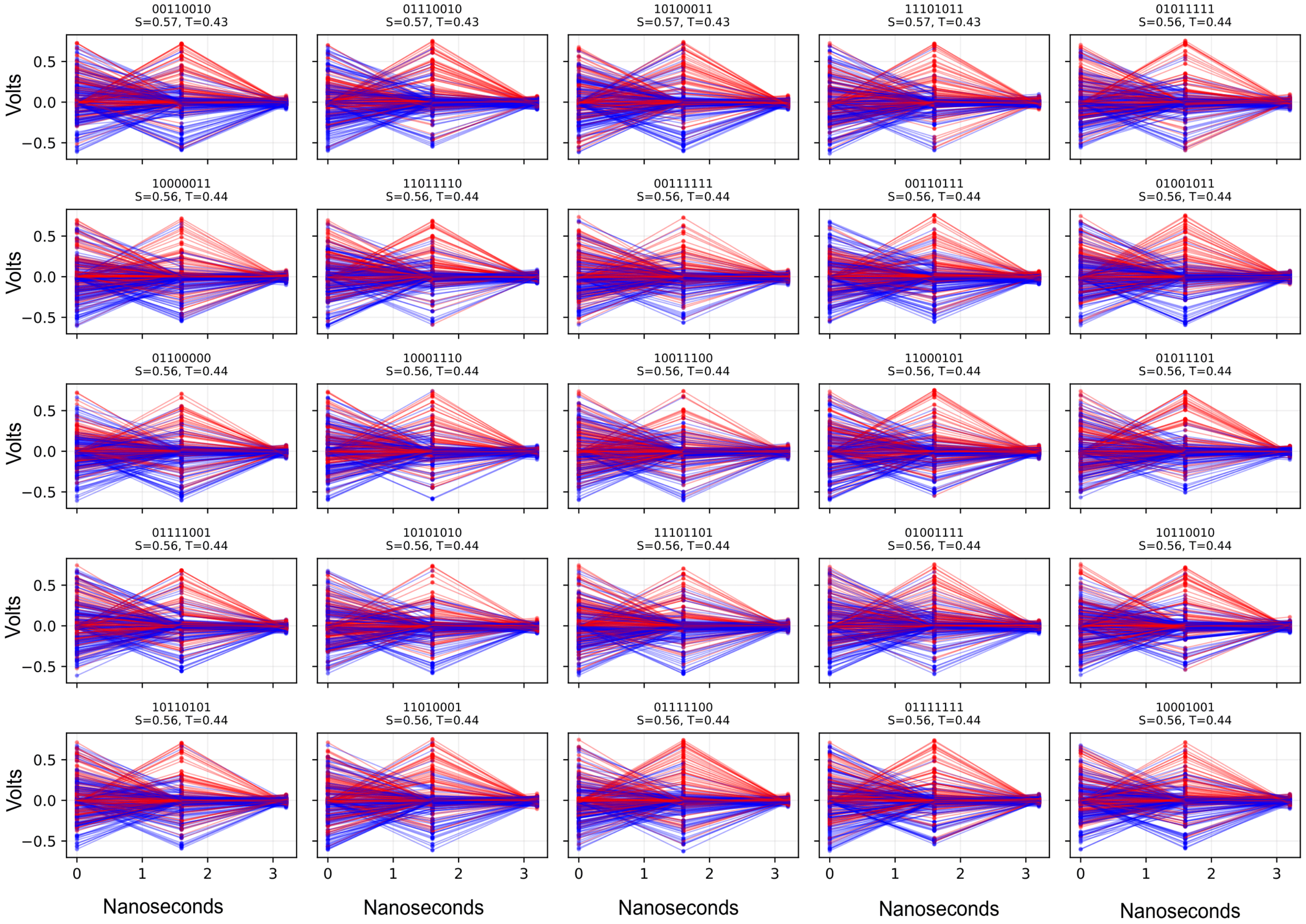
Supplementary Figure. 3 – MNN extracts grammatical structure from a sequence of microwave pulses. **a.i,** Each English word is represented by a corresponding microwave pulse-train token. Tokens are injected into the MNN sequentially as word-pairs and the nonlinear microwave response is measured from the output port as a spectrogram. A trained linear classifier (ridge regularized multi-class logistic regression) processes these spectrograms to infer the syntactic category (Part-of-Speech) of the first and second tokens. **a.ii,** Confusion matrices show PoS prediction across five grammatical classes, with strong on-diagonal values indicating reliable discrimination enabled by the MNN's broadband expansion. **b,** Word-at-a-time sentence-continuation game tests contextual understanding. MNN predicts the two PoS . Each word must follow a valid grammatical transition (based on valid templates in Supplementary Table 1) using the predicted PoS. The next words are cascaded with the previous ones and fed to the MNN. The process continues recursively to extend the length of the sentence-fragment. The game ends if the MNN predicts an incorrect PoS, or no valid next word exists. **c,** Evaluating mapping and syntactic inference capability. **c.i,** Comparison of greedy-search vs random-search mapping between 25 English words and 25 pulse-train tokens, evaluated over 10,000 attempts to make valid sentence-fragments. Greedy search completes more valid fragments. **c.ii,** Sentence-generation success for MNN + linear classifier versus a baseline that computes on spectrograms produced from raw tokens. MNN's nonlinear spectral expansion enables reliable inference up to 7 words in a valid sentence fragment template, while the baseline rarely exceeds 4 words.

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 1)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



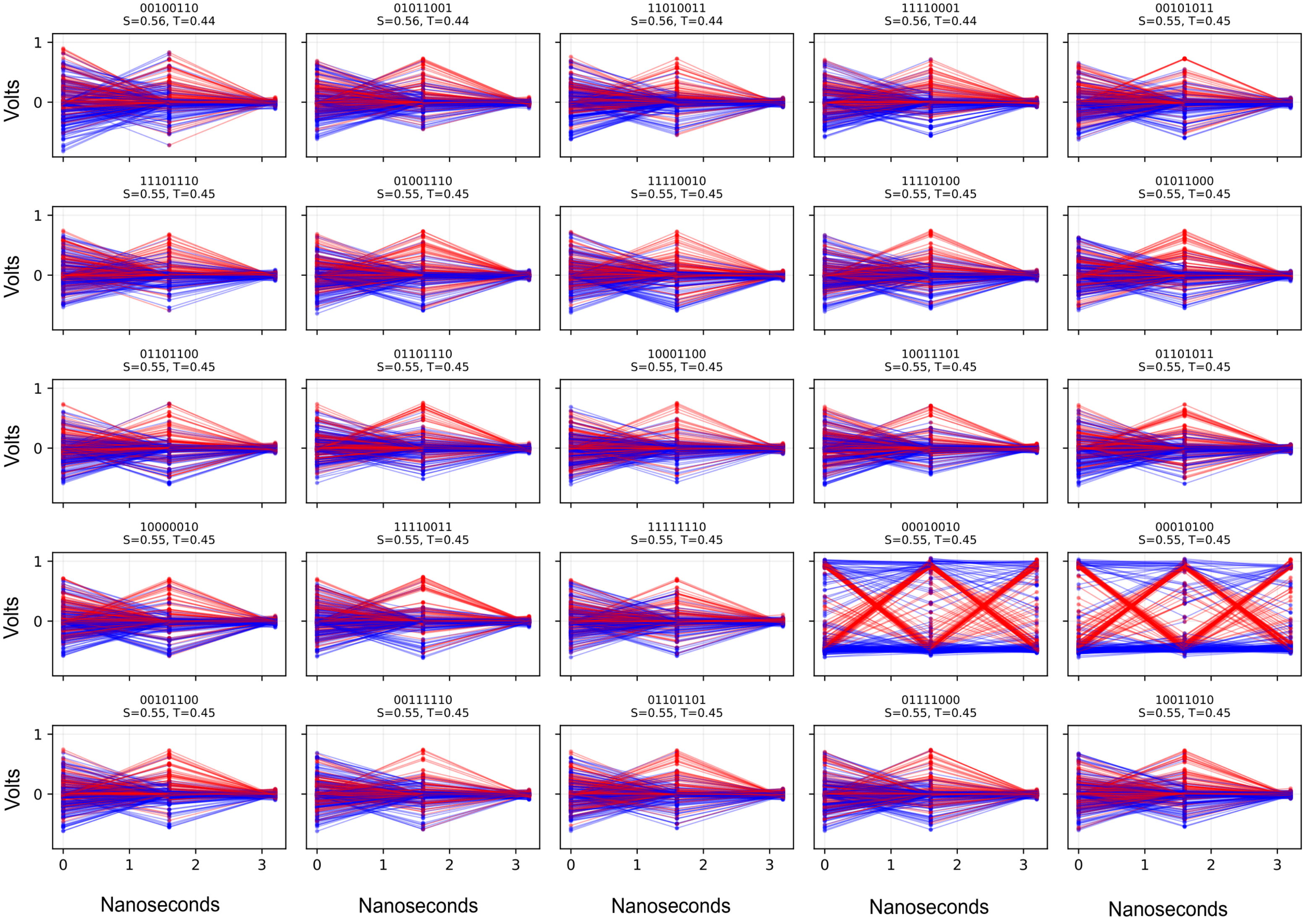
Supplementary Fig. 4a

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 2)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



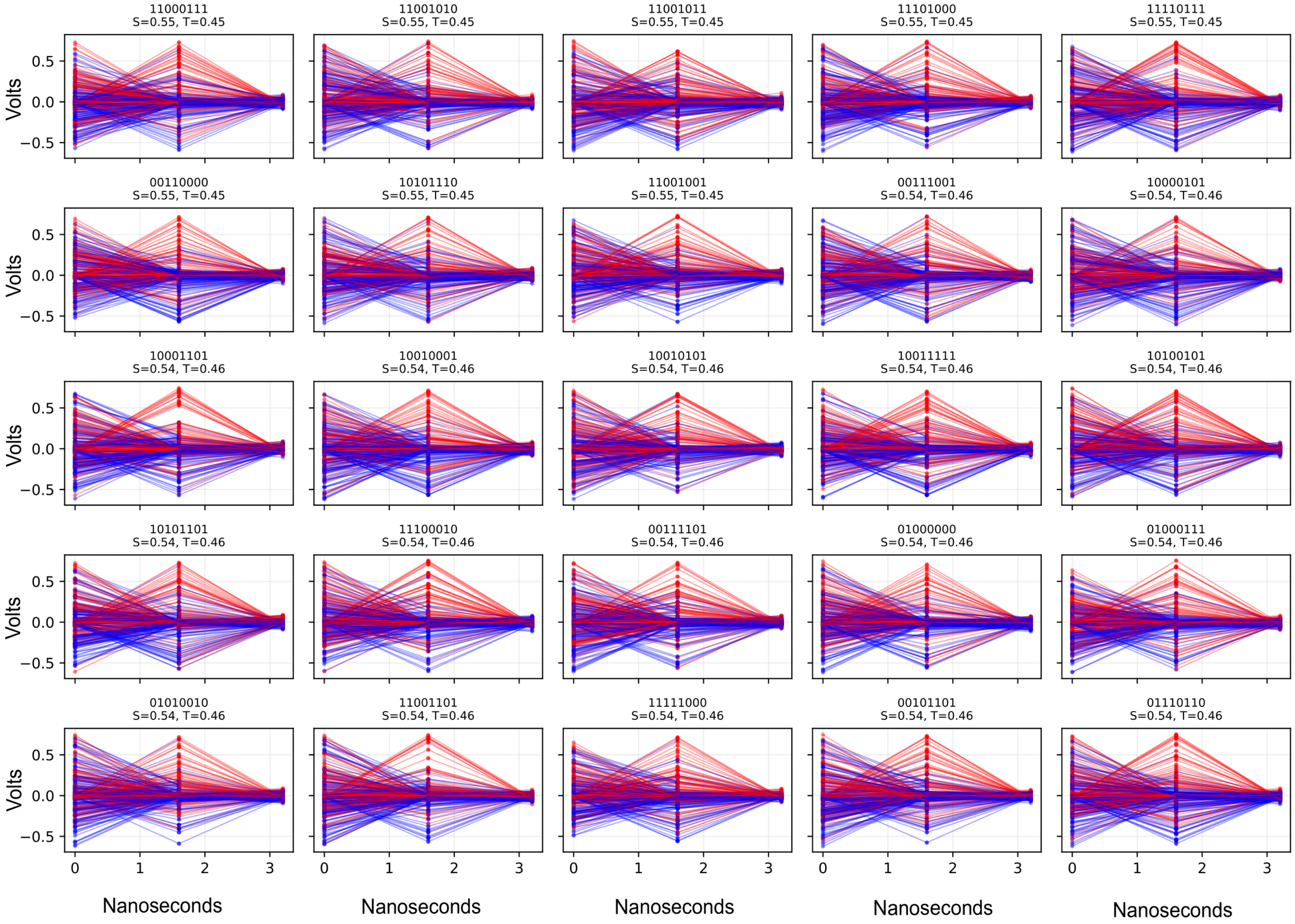
Supplementary Fig. 4b

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 3)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



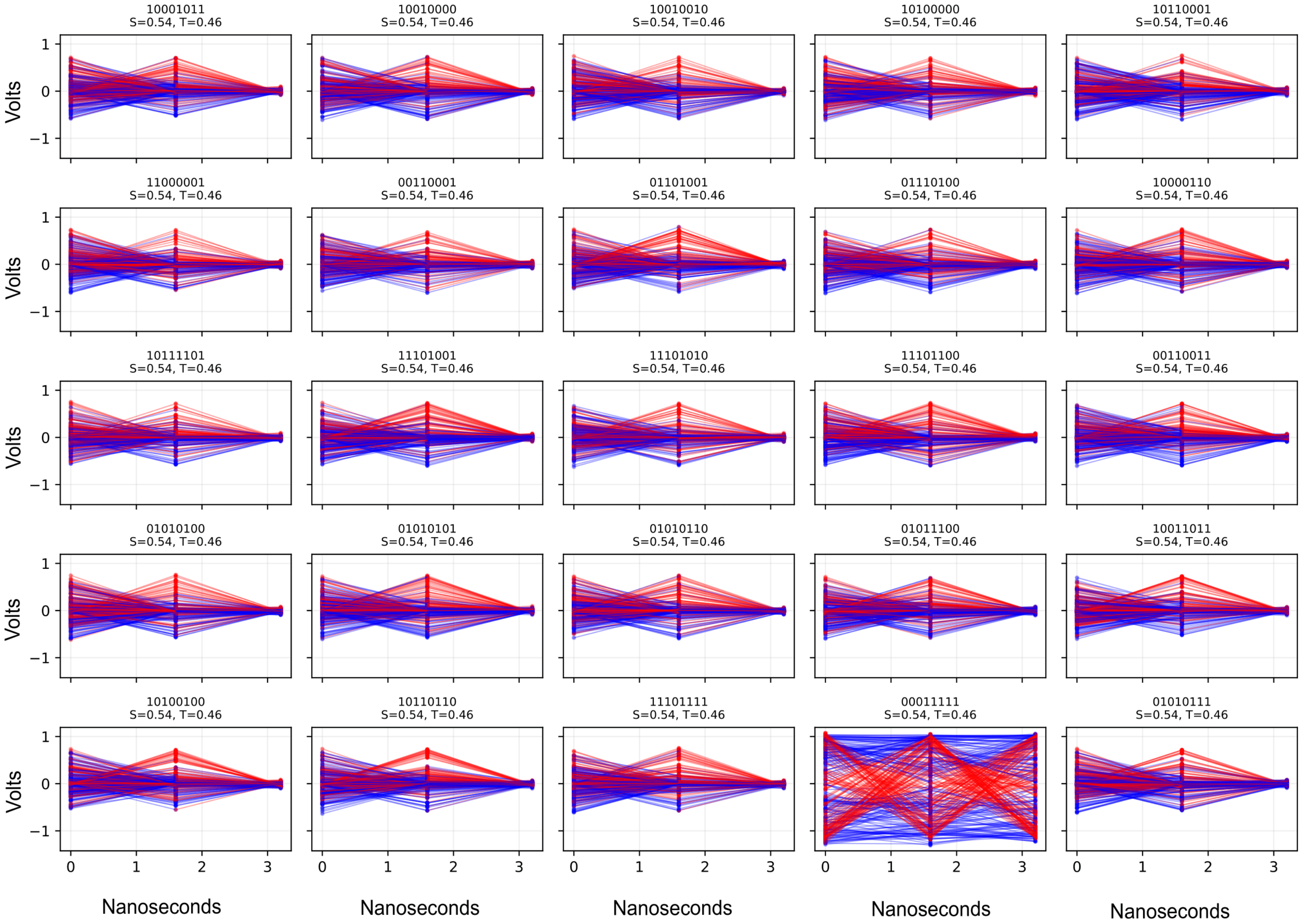
Supplementary Fig. 4c

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 4)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



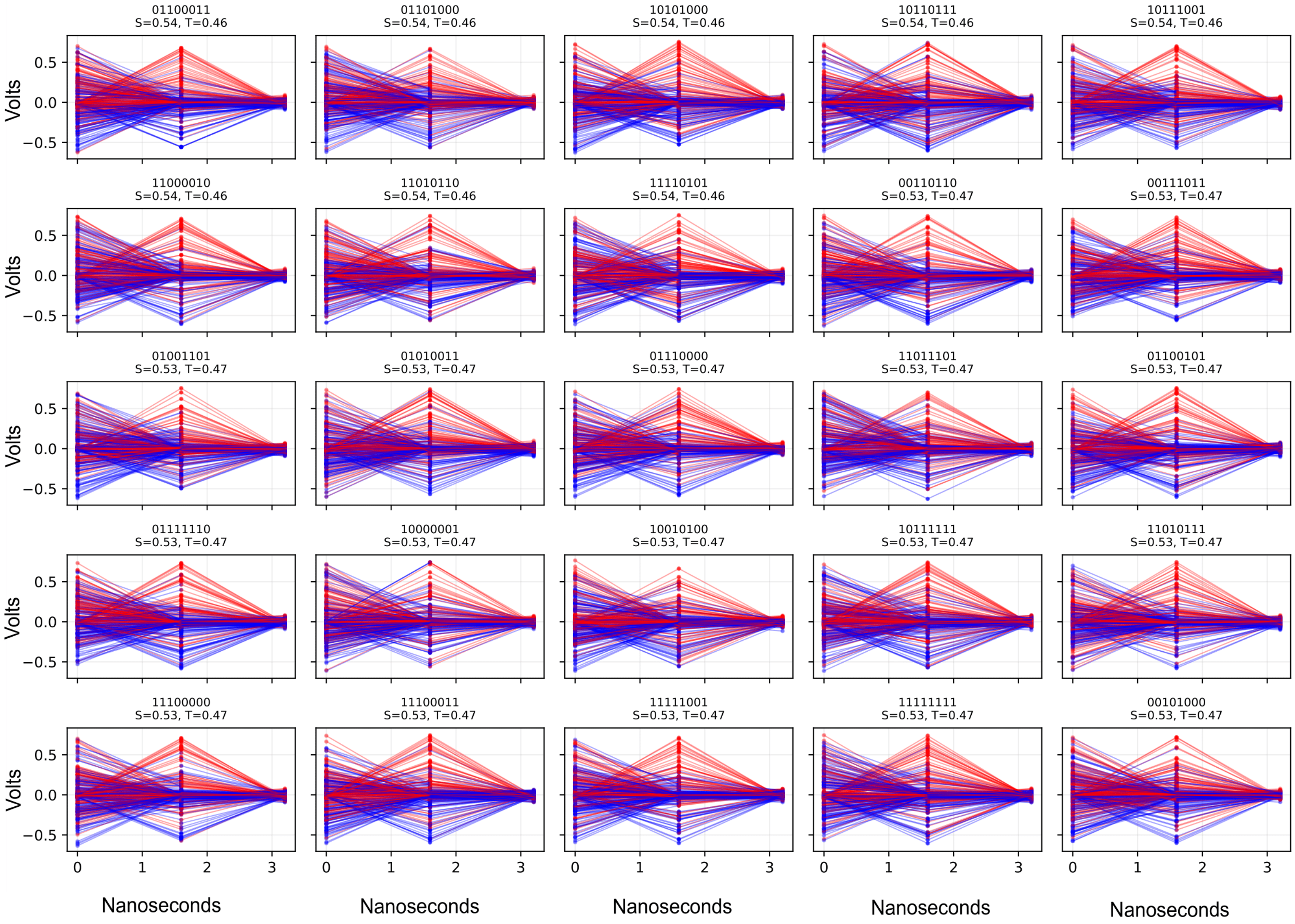
Supplementary Fig. 4d

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 5)**.
For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10).
The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



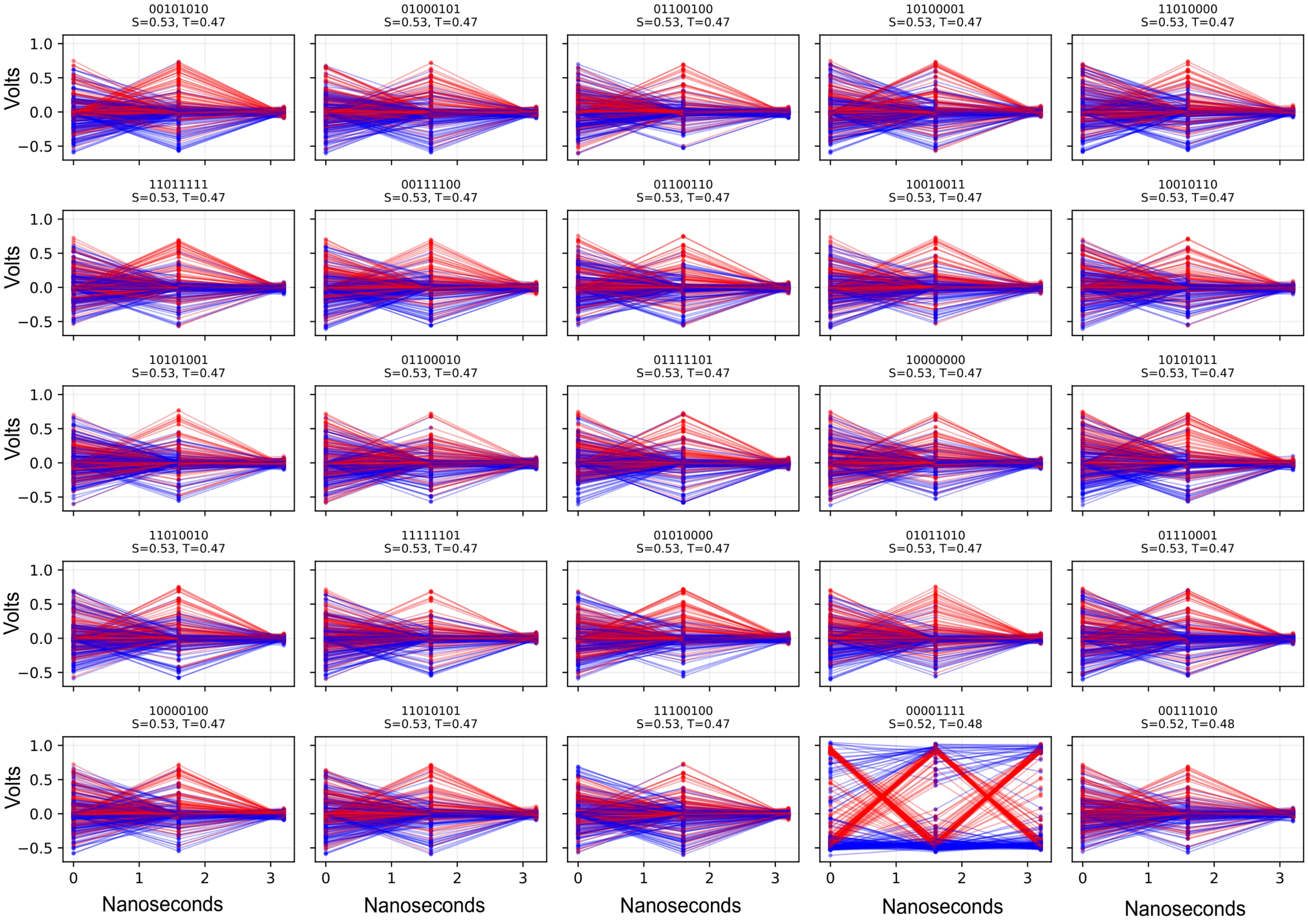
Supplementary Fig. 4e

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 6)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



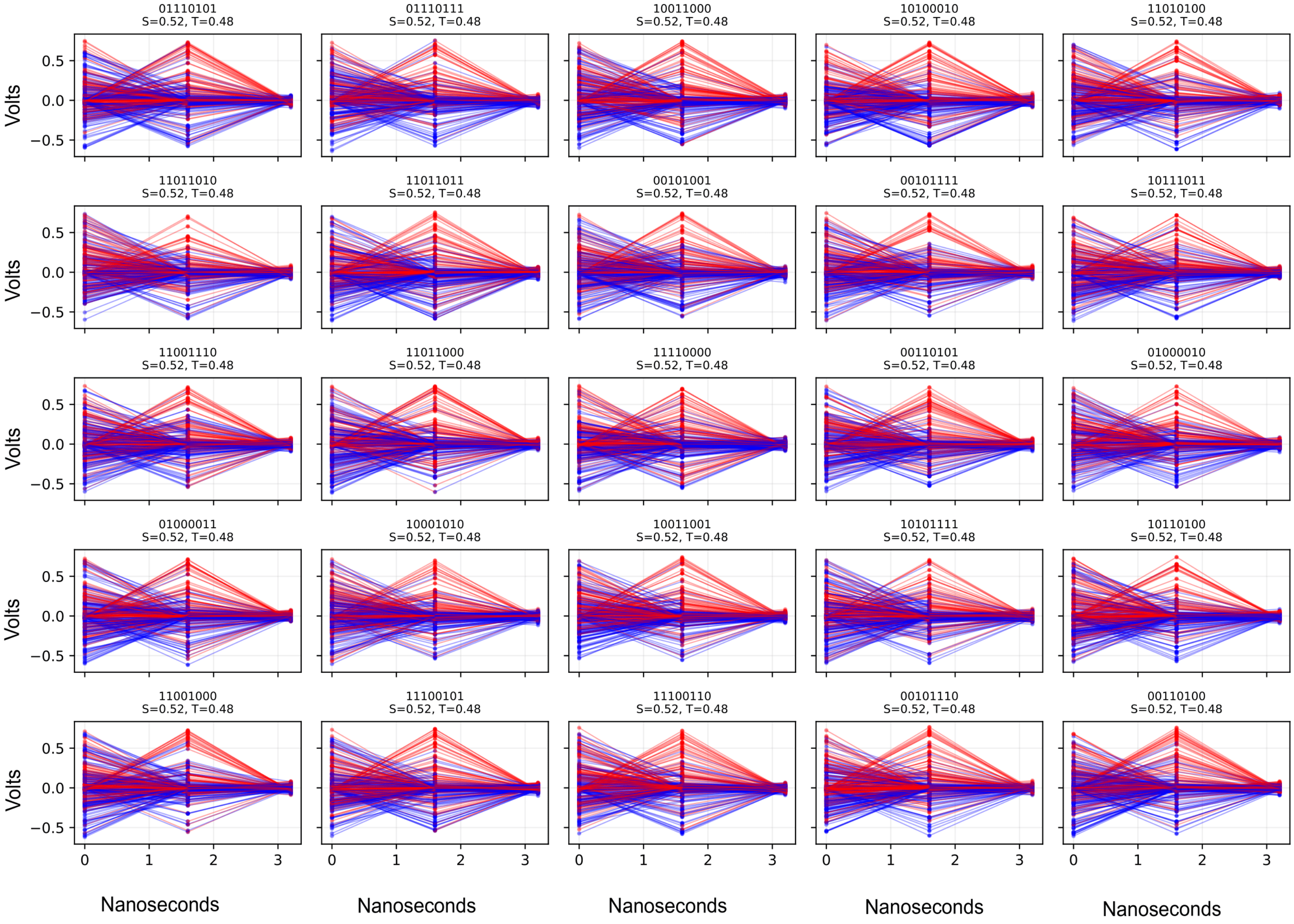
Supplementary Fig. 4f

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 7)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



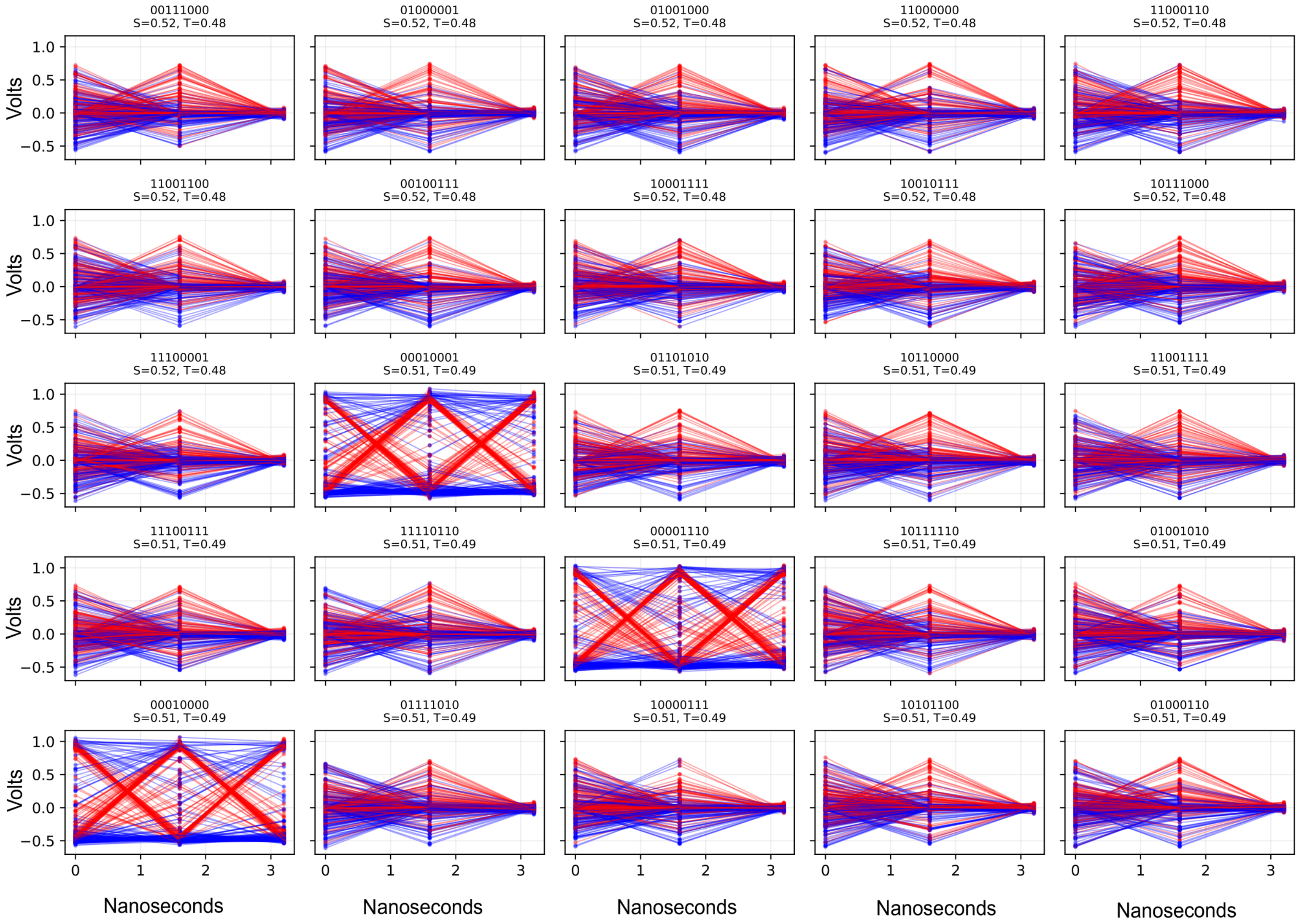
Supplementary Fig. 4g

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 8)**.
For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10).
The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



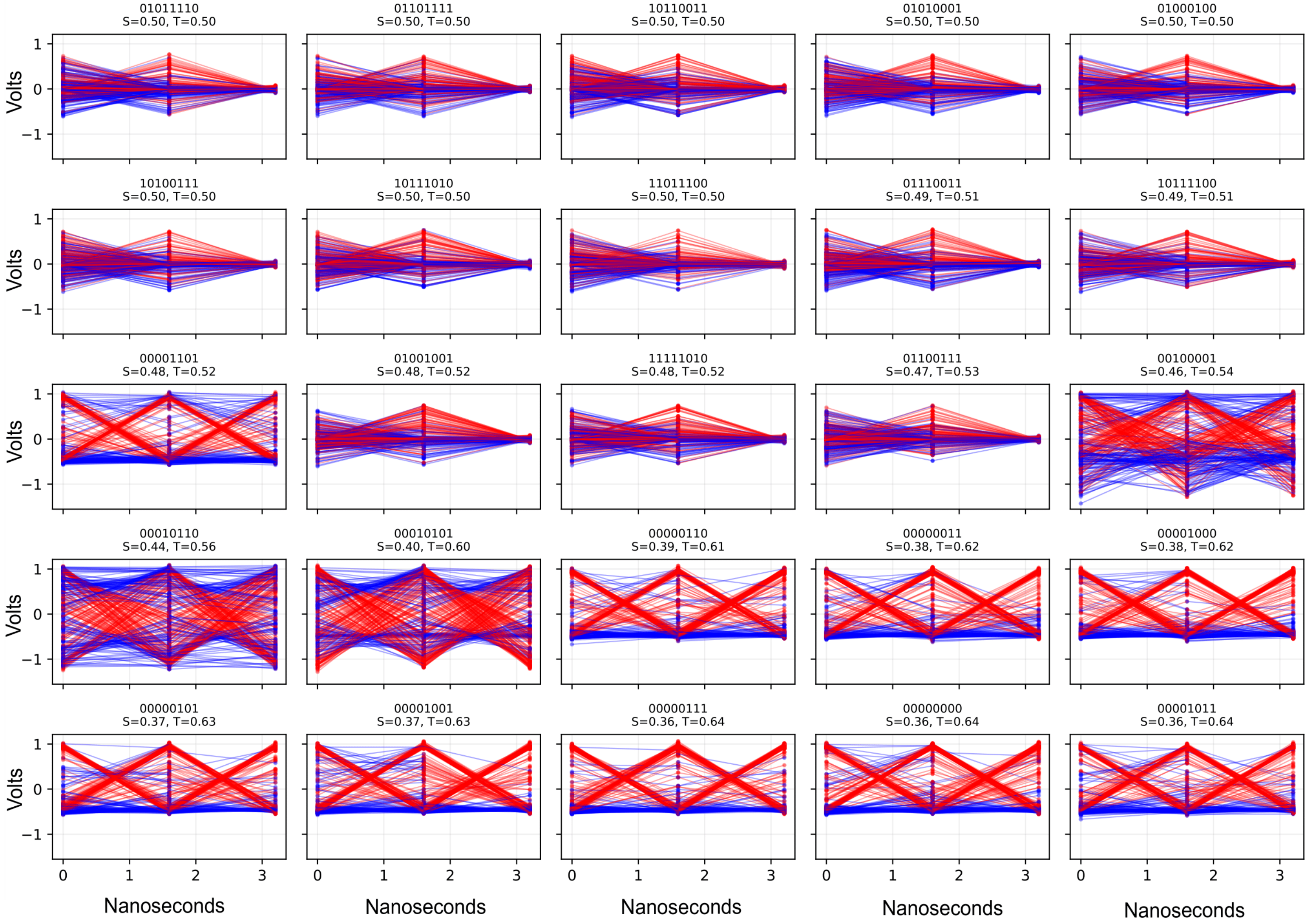
Supplementary Fig. 4h

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 9)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



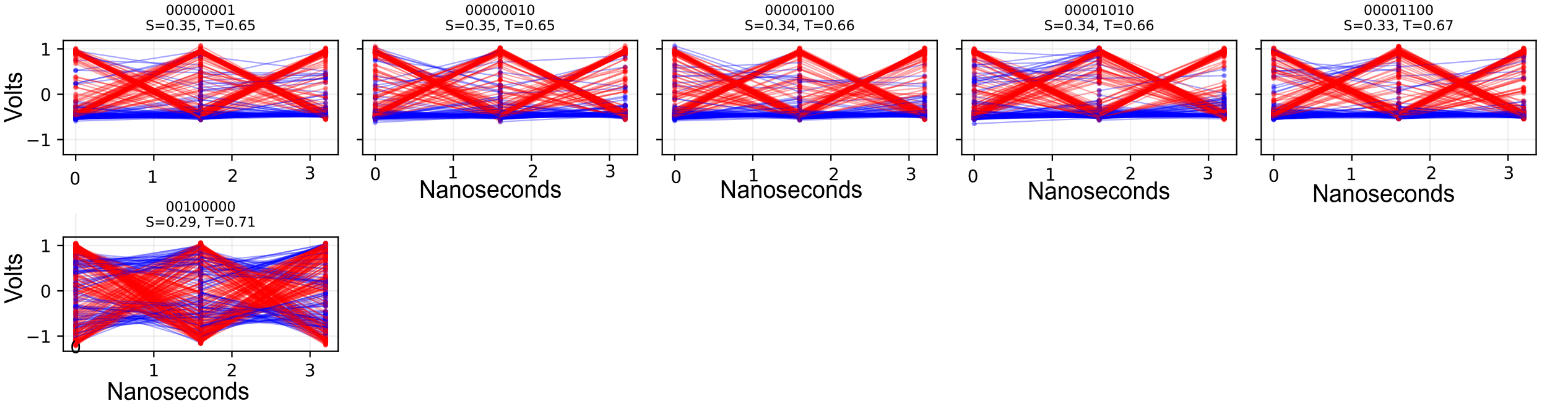
Supplementary Fig. 4i

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 10)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



Supplementary Fig. 4j

The MNN's broadband output signal, in response to a burst of a pixel's 8-bit pattern, is down-converted at LO = 10.4 GHz and sampled at 625 MS/s **(Set 11)**. For each 8-bit pattern (2.5 Gb/s), two samples are acquired every 3.2 ns at baseband, and thresholded into static states (00/11) or dynamic states (01/10). The static ratio is defined as: S= (# static states) / (300 repetitions). Dynamic ratio, T=1-S. 8-bit patterns are arranged in order of decreasing Static Ratio.



Supplementary Fig. 4k

Supplementary Tables

Supplementary Table 1: Mappings extracted from pulse-train tokens to submarine states in the game shown in Figs. 3 and 4. i.e., (β_{Tok} in Volts, F_{Tok} in MHz, T_{Tok} in ns) → (X grid units, Y grid units, θ radians)

$\boldsymbol{\beta}_{\mathbf{Tok}}$	$\boldsymbol{F}_{\mathbf{Tok}}$	$\boldsymbol{T}_{\mathbf{Tok}}$		**X**	**Y**	$\boldsymbol{\theta}$	$\boldsymbol{\beta}_{\mathbf{Tok}}$	$\boldsymbol{F}_{\mathbf{Tok}}$	$\boldsymbol{T}_{\mathbf{Tok}}$		**X**	**Y**	$\boldsymbol{\theta}$	$\boldsymbol{\beta}_{\mathbf{Tok}}$	$\boldsymbol{F}_{\mathbf{Tok}}$	$\boldsymbol{T}_{\mathbf{Tok}}$		**X**	**Y**	$\boldsymbol{\theta}$
0.4	20	40	→	0	0	0	0.85	40	40	→	2	0	0	0.85	60	100	→	2	1	0
0.55	20	40	→	1	3	1	0.7	80	40	→	3	1	0	0.4	80	40	→	2	1	2
0.7	20	40	→	0	0	2	0.7	20	60	→	1	0	1	0.7	100	120	→	0	4	1
0.85	20	40	→	4	4	2	0.4	60	40	→	1	1	2	0.4	60	80	→	1	4	2
1	20	40	→	0	3	0	1	40	60	→	1	1	0	0.55	100	60	→	3	2	0
0.4	40	40	→	0	1	1	0.55	100	40	→	1	2	0	0.85	100	100	→	2	3	2
0.55	40	40	→	0	1	2	0.4	100	120	→	0	3	1	0.4	100	60	→	2	0	1
0.4	100	40	→	0	2	0	0.55	100	100	→	1	0	2	1	80	40	→	2	2	0
0.7	100	40	→	0	2	1	0.7	60	80	→	1	3	2	0.7	80	60	→	1	4	0
0.55	20	80	→	0	4	0	1	60	60	→	1	2	2	0.55	100	80	→	2	1	1
0.7	40	40	→	3	1	2	0.85	100	60	→	1	0	0	1	60	100	→	2	3	0
0.85	40	80	→	4	3	2	0.85	80	40	→	0	0	1	0.7	60	40	→	2	3	1
0.85	60	40	→	0	3	2	0.4	40	60	→	1	2	1	0.4	100	80	→	0	2	2
1	40	40	→	2	0	2	1	100	80	→	1	3	0	0.85	100	40	→	2	4	0
1	100	60	→	0	1	0	1	100	40	→	1	1	1	0.7	100	80	→	3	1	1
0.55	60	60	→	3	0	1	1	20	80	→	2	2	1	0.55	60	120	→	3	3	1
0.7	80	120	→	3	0	2	1	60	120	→	4	0	2	0.4	80	120	→	4	1	1
1	80	80	→	3	0	0	0.85	60	60	→	0	4	2	1	80	100	→	4	1	2
0.85	100	80	→	4	2	2	1	40	100	→	3	4	2	0.55	40	100	→	4	2	0
0.85	80	60	→	3	3	0	0.7	40	100	→	4	0	1	0.7	20	80	→	4	2	1
1	100	100	→	3	2	1	1	20	120	→	3	3	2	0.55	40	60	→	4	3	0
0.85	20	100	→	4	4	1	0.7	60	60	→	3	4	0	0.7	100	100	→	4	3	1
1	20	60	→	2	2	2	0.7	80	80	→	2	4	2	0.7	20	100	→	1	4	1
0.85	80	100	→	2	4	1	0.85	20	120	→	4	0	0	0.4	20	60	→	4	4	0
1	60	40	→	4	1	0	1	80	120	→	3	2	2	0.4	40	120	→	3	4	1
0.55	60	60	→	3	0	1	1	20	80	→	2	2	1	0.55	60	120	→	3	3	1

Supplementary Table 2: Mappings extracted from pulse-train tokens to word tokens in the sentence-building game shown in Supplementary Figs. 2 and 3. i.e., (β_{Tok} in Volts, F_{Tok} in MHz, T_{Tok} in ns) → word

β_{Tok}	F_{Tok}	T_{Tok}		X
0.7	100	80	→	bridge
0.85	100	80	→	laughed
1.0	100	80	→	over
0.4	20	100	→	interesting
0.55	20	100	→	dark
0.7	20	100	→	boy
0.85	20	100	→	at
1.0	20	100	→	ran
0.4	40	100	→	river
0.55	40	100	→	happily
0.85	40	100	→	fell
1.0	40	100	→	crawled
0.4	60	100	→	jumped
0.55	60	100	→	sadly
0.7	60	100	→	mischievously
0.85	60	100	→	girl
1.0	60	100	→	slowly
0.4	80	100	→	quiet
0.55	80	100	→	quickly
0.7	80	100	→	in
0.85	80	100	→	new
1.0	80	100	→	old
1.0	80	100	→	across
0.4	100	100	→	under
0.55	100	100	→	road

Supplementary Table 3: Syntactically valid sentence-fragment templates for generating a training set for mapping word-embeddings to pulse-train tokens in Supplementary Note 2.

Word length	Syntactically valid sentence-fragment templates	Representative example
2 words long	Noun – Verb	boy-ran
3 words long	Noun – Verb – Adverb Noun – Adverb – Verb Adjective – Noun – Verb	girl-laughed-happily boy-quickly-ran old-bridge-fell
4 words long	Noun – Verb – Preposition – Noun Verb – Adverb – Preposition – Noun Adverb – Verb – Preposition – Noun Adjective – Noun – Verb – Adverb	boy-jumped-across-river ran-quickly-across-road slowly-crawled-under-bridge happy-girl-laughed-mischievously
5 words long	Adjective – Noun – Verb – Preposition – Noun Noun – Verb – Preposition – Adjective – Noun Verb – Preposition – Adjective – Noun – Adverb Verb – Adverb – Preposition – Adjective – Noun Adverb – Verb – Preposition – Adjective – Noun	quiet-girl-ran-across-road boy-fell-over-old-bridge laughed-at-interesting-boy-happily ran-quickly-under-quiet-bridge quickly-jumped-across-new-bridge
6 words long	Adjective – Noun – Verb – Preposition – Noun – Adverb	new-bridge-ran-across-river-happily
7 words long	Adjective – Noun – Adverb – Verb – Preposition – Adjective – Noun	happy-boy-happily-ran-across-old-bridge

Supplementary Table 4: Breakdown of power consumption in the feature extraction front-end.

On-chip MNN power consumption			
On-chip	**Max. Voltage**	**Max. Current**	**Max. Power**
Supply for MNN oscillator core	0.9 V	130 mA	117 mW
Supply for SPI registers	1.5 V	3.3 mA	4.95 mW
Supply for SPI logic	1 V	1.2 mA	1.2 mW

Off-chip consumption (will be integrated in the next version)			
Component	**Voltage**	**Current**	**Power**
IF amplifier	3.3	55 mA	182 mA
Passive mixer	N/A	0 mA	0 mA
RF gain stage (optional)	5	21 mA	105

SUPPLEMENTARY REFERENCES

[S1]. Kuhn, H. W. The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2, 83–97 (1955).

[S2]. PyTorch Team. torch.nn.LSTM — PyTorch Documentation. Available at: https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html (Accessed: 6 December 2025).