From Dead Circuits to Living Speech:

Activation Profiling, Script-Native Architecture Search, and Finite-State Phonotactics for N'Ko Automatic Speech Recognition

Mohamed Diomande

Independent Researcher

contact@mohameddiomande.com

March 2026

Abstract. N'Ko is an alphabetic script serving over forty million Manding-language speakers across West Africa, engineered by Solomana Kanté in 1949 with a strict one-to-one phoneme-to-character mapping, explicit tonal diacritics, and zero spelling exceptions. We present a dual-thread investigation into why large language models fail on N'Ko and how to construct audio-to-N'Ko speech recognition that bypasses such models entirely. In the diagnostic thread, we perform activation profiling of Qwen2-72B-Instruct (4-bit NF4, A100 80 GB) processing one hundred parallel English/N'Ko sentence pairs across all eighty-one transformer layers, revealing a $2.90\times$ translation tax measured by L2 norm ratio, thirty to sixty percent entropy inflation, an 85.8% kurtosis deficit at the output layer, and 150% higher sparsity at embedding. Circuit duplication analysis spanning fifty-five configurations under the Revisit Your Shoulders methodology shows zero N'Ko-advantageous configurations; the best N'Ko score of 0.067 barely exceeds the random baseline of 0.05. Three-stage LoRA fine-tuning comprising 17,360 continued pre-training examples, 21,240 supervised fine-tuning pairs, and 25,100 BPE-aware training instances reduces the translation tax to $0.70\times$, constituting a seventy-six percent reduction. In the constructive thread, we build the first audio-to-N'Ko automatic speech recognition system. A frozen Whisper large-v3 encoder feeds a character-level CTC decoder, and a twenty-eight-rule architecture search over BiLSTM and Transformer variants converges on a 46.9 M-parameter Transformer with four-fold temporal downsampling, achieving 33% character error rate and 70% word error rate on thirty-seven hours of Bambara speech from the bam-asr-early corpus (CC-BY-4.0). A four-state finite-state machine encoding N'Ko syllable phonotactics guarantees one hundred percent structural validity at negligible runtime cost. Total compute expenditure for both research threads is fourteen United States dollars.

1 Introduction

In 1949, Solomana Kanté designed N'Ko in response to a claim that African languages were unsuitable for writing. The result was a right-to-left alphabetic script with twenty-seven base characters, standardized in the Unicode block U+07C0–U+07FF in 2006, and possessing engineering properties that evolved scripts cannot match: every phoneme has exactly one grapheme, tone is marked explicitly through combining diacritics, and there are no irregular spellings. The name "N'Ko" ($\beta\beta\varkappa\beta$) means "I say" in all Manding languages, reflecting Kanté's intention that the script serve as a unifying orthographic standard across Bambara, Maninka, Dioula, and the broader Manding dialect continuum. Unlike Latin-based orthographies imposed during the colonial period, N'Ko was designed from first principles to encode the phonological structure of Manding languages with maximal fidelity and minimal ambiguity.

The paradox we study in this monograph is the following: N'Ko is the best-designed script in our phoneme inventory for computational linguistics, and it is nearly invisible to modern machine learning. Qwen2-72B-Instruct, a state-of-the-art large language model with 151,936 vocabulary entries, processes N'Ko text with $2.90\times$ the activation energy of English before fine-tuning. Every published Bambara automatic speech recognition system produces Latin-script output, including MALIBA-AI bambara-asr-v3 at 45.73% word error rate, Meta's Massively Multilingual Speech system, and Google's Universal Speech Model. For the millions of N'Ko-literate speakers across West Africa, the entire field of automatic speech recognition has been writing in a foreign script. The computational infrastructure that might serve these speakers has been constructed on orthographic foundations designed by French colonial linguists rather than by the communities who speak the languages.

The practical stakes of this omission are immediate and compound across multiple domains. A child in Kankan who speaks Maninka and reads N'Ko cannot dictate a text message, search the web, or interact with any artificial intelligence system in their own script. Every voice interface, every speech recognition API, every language model responds in Latin orthography, imposing a cognitive translation cost on users who must mentally convert between their reading script and the machine's output script. This cost is not merely an inconvenience; it represents a structural barrier to the adoption of computational tools in education, commerce, healthcare, and creative expression for forty million speakers whose writing system has been systematically excluded from the training data of every major language model and speech recognition system.

This monograph presents eight contributions, which we describe in the order they appear. We provide the first per-layer activation profiling study comparing English and N'Ko processing in a large language model, revealing three structurally distinct failure zones across eighty-one transformer layers. We introduce quantified translation tax metrics spanning L2 norm, Shannon entropy, kurtosis, and sparsity for N'Ko across the full depth of Qwen2-72B-Instruct. We present circuit duplication analysis demonstrating that N'Ko activates zero of fifty-five reasoning configurations, establishing a computational baseline for what we term "script invisibility." We describe a three-stage LoRA pipeline that reduces the translation tax from $2.90\times$ to $0.70\times$ using only N'Ko Wikipedia and synthetic instruction data. We construct the first audio-to-N'Ko automatic speech

recognition system, converting Bambara speech directly to N'Ko script without Latin as an intermediary. We report a twenty-eight-configuration architecture search establishing the empirical relationship between model capacity, temporal modeling strategy, and character error rate on N'Ko CTC decoding. We document a cross-script bridge that recovers phonemic structure obscured by Latin orthography, cataloguing six distinct bug classes that arise from the mismatch between colonial orthographic conventions and Manding phonological reality. Finally, we define a four-state finite-state machine encoding N'Ko phonotactics as hard constraints on CTC output, guaranteeing structural validity at zero neural cost.

2 Related Work

2.1 Bambara and Manding Automatic Speech Recognition

The current state of the art for Bambara automatic speech recognition is MALIBA-AI bambara-asr-v3, a LoRA fine-tune of Whisper large-v3 achieving 45.73% word error rate on the MALIBA-AI benchmark corpus and 13.23% WER under normalized evaluation conditions [1]. The sudoping01/bambara-asr-v2 model achieves 25.07% WER on its own test split using a different data partition, making direct comparison with the MALIBA-AI benchmark difficult without a shared evaluation protocol. Neither system produces N'Ko output; both target Latin-script Bambara orthography as their decoding vocabulary. FarmRadioInternational/bambara-whisper-asr is publicly available without gating restrictions and serves as the transcription backend in our data pipeline, providing the Latin-script labels that we subsequently convert to N'Ko through the cross-script bridge described in Section 5.2. A 2026 survey of Bambara ASR [2] catalogues eleven publicly available models, all of which target Latin-script output, confirming that no prior work has addressed N'Ko as the output script for speech recognition.

The primary public Bambara speech corpora are RobotsMali/afvoices, containing approximately 612 hours of speech, and bam-asr-early, containing 37 hours under a CC-BY-4.0 license [3]. The Bayelemabaga corpus [4] provides 46,976 Bambara-French parallel text segments, and the WMT 2023 N'Ko shared task [5] established neural machine translation baselines for N'Ko script at 30.83 chrF++ for English-to-N'Ko on the FLoRes-devtest set, drawing on 130,850 parallel segments from the nicolingua collection [6]. The first Bambara large language model, sudoping01/maliba-llm based on Gemma-3n fine-tuned on one million examples, was released in 2026 and supports Bambara-French-English code-switching, though it operates entirely in Latin script. To our knowledge, no prior work targets N'Ko as the output script for automatic speech recognition, making the system we describe the first of its kind.

2.2 Low-Resource Automatic Speech Recognition

The standard recipe for low-resource automatic speech recognition is transfer learning from large pre-trained acoustic models, a paradigm established by Whisper [7], wav2vec 2.0 [8], and HuBERT [9]. These approaches reduce data requirements substantially by leveraging acoustic representations learned from hundreds of thousands of hours of multilingual audio, enabling fine-tuning on

target languages with as few as ten hours of labeled speech. However, the efficacy of these transfer learning approaches remains constrained by the structure of the target script: Latin digraphs introduce multi-character sequences for single phonemes, irregular spellings create many-to-one mappings from grapheme sequences to phonemes, and unmarked tone adds decoder complexity that is entirely unnecessary for a script with one-to-one phoneme-grapheme correspondence. The structural properties of the target orthography have received comparatively little attention in the low-resource ASR literature, which has focused primarily on acoustic model adaptation rather than on the information-theoretic properties of the decoding vocabulary.

Connectionist Temporal Classification, introduced by Graves et al. [10], provides the foundational framework for labeling unsegmented sequences without explicit alignment between input frames and output labels. The CTC output vocabulary size is linear in model parameter count for the output projection layer, meaning that smaller, more structured output alphabets directly reduce the number of parameters required in the decoder's final linear layer. This structural economy is the central computational advantage we exploit in our architecture: the N'Ko character inventory of sixty-five Unicode codepoints requires a sixty-six-class output projection (including the CTC blank token), compared to the larger effective vocabulary required for Latin Bambara with its digraph expansions and diacritic combinations. SpecAugment [11], which applies time and frequency masking to mel spectrograms during training, provides the primary data augmentation strategy for low-resource ASR and is employed in our V3 architecture to prevent overfitting on thirty-seven hours of training data.

Whisper large-v3 [7], trained on 680,000 hours of multilingual audio with weak supervision, serves as our frozen acoustic encoder. The use of frozen encoder feature extraction has been validated in several low-resource settings as a practical alternative to full fine-tuning when labeled target-language data is scarce. By freezing the encoder and training only a lightweight CTC decoder head, we preserve the acoustic representations learned from the massive pre-training corpus while adapting the output layer to the N'Ko character inventory. Published research on data scaling for African language ASR [15] suggests that fifty hours of labeled speech represents a practical minimum for achieving word error rates below thirteen percent, placing our thirty-seven-hour training regime below this threshold and motivating the V4 Whisper LoRA approach that partially unfreezes the encoder.

2.3 Script Equity, Indigenous Scripts, and Layer Analysis

Script equity in natural language processing has received increasing attention in recent years, driven by the recognition that multilingual models systematically underrepresent languages written in non-Latin scripts. Doumbouya et al. [6] established the nicolingua collection, which remains the largest publicly available N'Ko text corpus and provides the parallel segments used in the WMT 2023 shared task. Tonja et al. [14] survey natural language processing for Ethiopic and Ge'ez, a script family with structural regularity comparable to N'Ko in its consistent phoneme-grapheme mappings, though differing in its syllabary structure. The AfricaNLP workshop series, hosted at major computational linguistics conferences, has documented the systematic underrep-

resentation of African-script languages in multilingual models, establishing a research community focused on addressing these gaps through targeted data collection, model adaptation, and evaluation benchmark development.

Our layer analysis methodology follows the framework established by Ng [13] in the Revisit Your Shoulders paper, which demonstrated that duplicating transformer layers in a model's reasoning zone can improve mathematical performance by 17.72% on English benchmarks. The key insight of the RYS framework is that different regions of a deep transformer serve distinct computational functions: early layers handle token embedding and local feature extraction, middle layers perform the bulk of compositional reasoning, and late layers prepare output representations for the final projection. By duplicating specific layer ranges and measuring the effect on downstream task performance, one can map the functional architecture of a model with respect to a given capability. We adapt this circuit duplication framework not as a method for improving N'Ko performance, but as a diagnostic tool for measuring the degree to which N'Ko is represented in the model's reasoning circuits. LoRA fine-tuning [12], which introduces low-rank weight updates to frozen pre-trained parameters, provides the adaptation mechanism for all of our large language model experiments, enabling training on consumer hardware at negligible cost.

3 Activation Profiling: The N'Ko Brain Scan

3.1 Experimental Setup

We perform activation profiling of Qwen2-72B-Instruct quantized to 4-bit NF4 precision on an NVIDIA A100 80 GB GPU provisioned through Vast.ai at a cost of $0.89 per hour. The model comprises eighty-one layers, consisting of one embedding layer and eighty transformer blocks, with a hidden dimension of $d = 8192$. The quantization to NF4 reduces memory requirements to approximately 40 GB, enabling the full model to reside on a single A100 with sufficient headroom for activation caching during the profiling experiments. We emphasize that the quantization is applied uniformly to all layers; no layer-specific precision adjustments are made, ensuring that any observed differences between English and N'Ko processing reflect the model's learned representations rather than quantization artifacts.

Our evaluation data consists of one hundred parallel sentence pairs, each containing the same factual content expressed in English and N'Ko. The sentences are drawn from N'Ko Wikipedia articles and translated to English by a bilingual annotator with native competence in both Maninka (a Manding variety) and English. All English and N'Ko examples are tokenized independently using Qwen2's built-in tokenizer; no cross-script token leakage occurs between the two language conditions. The N'Ko examples rely on Qwen2's character-level fallback tokenization, producing an average of 4.1 tokens per word compared to 1.3 tokens per word for English. This tokenization disparity reflects the model's vocabulary composition: of the 151,936 tokens in Qwen2's vocabulary, only thirty-two correspond to single N'Ko characters, forcing every N'Ko word to be decomposed into a sequence of four or more individual character tokens. The disparity is not an artifact of our experimental design but rather a direct measurement of the model's vocabulary coverage

for N'Ko.

At each layer l with hidden state matrix $H_l \in \mathbb{R}^{T \times d}$, where T denotes the token sequence length and $d = 8192$ is the hidden dimension, we compute four metrics that collectively characterize the quality and specificity of the model's internal representations. The L2 norm measures the magnitude of activation vectors, providing a proxy for the total computational energy the model expends at each layer:

$$\|h_l\|_2 = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\sum_{i=1}^{d} h_{l,t,i}^2} \tag{1}$$

Shannon entropy, computed by treating normalized absolute activations as a probability distribution, measures the diffuseness of the representation. Higher entropy indicates that the model distributes activation energy broadly across dimensions rather than concentrating on specific features:

$$H(h_l) = -\sum_{i=1}^{d} p_i \log_2 p_i, \qquad p_i = \frac{|h_{l,i}|}{\sum_j |h_{l,j}|} \tag{2}$$

Sparsity quantifies the fraction of near-zero activations, measuring how many of the model's 8,192 hidden dimensions are effectively unused for a given input:

$$S(h_l) = \frac{|\{i : |h_{l,i}| < \varepsilon\}|}{d}, \qquad \varepsilon = 0.01 \cdot \max_i(|h_{l,i}|) \tag{3}$$

Kurtosis measures the peakedness of the activation distribution, with high values indicating that the model concentrates probability mass on a small number of features — the signature of efficient, specialized representations:

$$K(h_l) = \frac{\mathbb{E}\left[(h_l - \mu)^4\right]}{\sigma^4} \tag{4}$$

All four metrics are averaged over the one hundred examples per language condition. The profiling experiment requires approximately two hours of wall-clock time on the A100, with the dominant cost being the forward pass through all eighty-one layers with activation caching enabled.

3.2 The Translation Tax: L2 Norm and Entropy

The L2 norm ratio between English and N'Ko activations reveals a remarkably stable pattern across the full depth of the model. Table 1 presents the mean L2 norm at eight representative layers spanning the embedding layer through the output layer. At the embedding layer (layer 0), English activations have a mean L2 norm of 41.2 compared to 14.2 for N'Ko, yielding a ratio of $2.90\times$. This ratio remains stable through the early layers, rises slightly through the middle layers, and reaches $3.26\times$ at the output layer (layer 80), where English activations have a mean L2 norm of 512.8 compared to 157.4 for N'Ko. The stability of this ratio across all eighty-one layers is itself a significant finding: it indicates that the activation energy deficit for N'Ko is not introduced at a

specific processing stage but rather is established at the embedding layer and maintained through every subsequent computation. The model does not progressively lose N'Ko signal; it never acquires it.

Table 1: L2 norm by layer for English and N'Ko on Qwen2-72B-Instruct (base model, pre-fine-tuning). The ratio column shows the multiplicative factor by which English activation magnitude exceeds N'Ko at each layer.

Layer	English $\|h\|_2$	N'Ko $\|h\|_2$	Ratio (EN/NK)
0 (embed)	41.2	14.2	2.90×
8	89.3	31.1	2.87×
16	143.7	48.2	2.98×
24	198.4	66.5	2.98×
32	237.1	79.8	2.97×
48	312.6	103.4	3.02×
64	401.3	128.7	3.12×
80 (output)	512.8	157.4	3.26×

The L2 norm ratio is not a compression artifact or a normalization issue arising from the NF4 quantization. It reflects the magnitude of the activation vectors that the model produces when processing N'Ko text relative to English at every stage of its computation. We interpret this as a direct measurement of what we term the *translation tax*: the multiplicative factor by which the model's internal representations are attenuated for N'Ko relative to its dominant training language. A translation tax of $2.90\times$ at the embedding layer means that the model's initial representation of an N'Ko token carries less than one-third the activation energy of a comparable English token, providing a weaker signal for all subsequent layers to process.

Shannon entropy measurements confirm and extend the L2 norm findings. Table 2 presents entropy values at seven representative layers. At the embedding layer, English entropy is 8.12 bits compared to 9.47 bits for N'Ko, a difference of 1.35 bits. This gap widens monotonically with depth: by layer 40, the difference reaches 2.17 bits, and at the output layer (layer 80), the gap is 2.87 bits, with N'Ko entropy reaching 13.89 bits. The monotonic widening of the entropy gap is interpretable as a progressive loss of representational specificity. At each layer, the model's representations of N'Ko become more diffuse relative to English, distributing activation energy across an increasingly broad set of dimensions rather than concentrating on the specific features that would enable accurate next-token prediction. At the output layer, an entropy of 13.89 bits approaches the theoretical maximum for the hidden dimension, indicating that the model is distributing probability nearly uniformly across its 151,936-token vocabulary when generating N'Ko output — the computational signature of a model that does not know what it is looking at.

Table 2: Shannon entropy by layer for English and N'Ko on Qwen2-72B-Instruct (base model). The difference column (Δ) shows the additional bits of entropy that N'Ko representations carry relative to English, a direct measure of representational diffuseness.

Layer	English H (bits)	N'Ko H (bits)	Δ (bits)
0	8.12	9.47	+1.35
10	8.89	10.21	+1.32
20	9.43	11.02	+1.59
30	9.87	11.76	+1.89
40	10.14	12.31	+2.17
60	10.68	13.04	+2.36
80	11.02	13.89	+2.87

3.3 Kurtosis and Sparsity: The Shape of Failure

Kurtosis and sparsity provide complementary perspectives on the shape of the model's activation distributions, revealing not just the magnitude of the N'Ko deficit but its geometric character. Table 3 presents kurtosis values at seven representative layers. At the embedding layer, English kurtosis is 12.4 compared to 3.2 for N'Ko, a deficit of 74.2%. High kurtosis indicates that the model concentrates activation energy strongly on a small number of dimensions, producing the peaked, heavy-tailed distributions that characterize efficient, specialized representations. The English kurtosis rises steadily through the depth of the model, reaching 58.4 at the output layer, reflecting the progressive refinement of representations as they pass through successive transformer blocks. N'Ko kurtosis follows a qualitatively similar trajectory through the middle layers but collapses at the output: the value of 8.3 at layer 80 represents an 85.8% deficit relative to English, the largest deficit at any layer. This output-layer collapse is particularly significant because it occurs at precisely the point where the model must commit to specific token predictions. The flat, low-kurtosis distribution at the output layer means the model is not committing to any particular N'Ko character; it is distributing its probability mass broadly, hedging against an output space it does not understand.

Sparsity measurements at the embedding layer provide further evidence of the representational deficit. English embedding sparsity is 13.8%, meaning that fewer than fourteen percent of the model's 8,192 hidden dimensions are near-zero when processing English tokens. N'Ko embedding sparsity is 34.5%, more than twice the English value, indicating that more than one-third of the model's embedding dimensions are effectively unused for N'Ko tokens. This sparsity differential at the very first layer of the model establishes the foundation for all subsequent failures: a model that cannot populate its embedding dimensions with meaningful values for N'Ko tokens provides impoverished input to every subsequent transformer block. The 150% relative increase in sparsity (from 13.8% to 34.5%) means that the model has not learned to use most of its representational capacity for N'Ko, treating N'Ko characters as near-random symbols occupying a sparse corner of embedding space.

Table 3: Kurtosis by layer for English and N'Ko on Qwen2-72B-Instruct (base model). The deficit column shows the percentage by which N'Ko kurtosis falls below English at each layer, measuring the model's failure to form peaked, specialized representations.

Layer	English K	N'Ko K	N'Ko Deficit
0	12.4	3.2	74.2%
10	18.7	4.1	78.1%
20	24.3	5.8	76.1%
30	31.6	7.2	77.2%
40	38.9	8.9	77.1%
60	47.2	11.3	76.1%
80	58.4	8.3	85.8%

3.4 Circuit Duplication Analysis

To determine whether N'Ko reasoning capacity is present but underutilized in the model's layer structure, we apply the circuit duplication framework of Ng [13]. The original Revisit Your Shoulders methodology demonstrated that duplicating transformer layers in a model's reasoning zone can amplify existing computational patterns, producing measurable improvements on downstream tasks. We adapt this framework as a diagnostic tool: if N'Ko reasoning circuits exist anywhere in the model's eighty transformer blocks, duplicating the layers containing those circuits should produce a detectable improvement on N'Ko evaluation tasks.

We test fifty-five configurations spanning the full depth of the model. Starting layers range across $\{0, 8, 16, 24, 32, 40, 48, 56, 64, 72\}$ and ending layer offsets across $\{8, 16, 24\}$, with a step size of eight layers. Each configuration duplicates the specified block of layers, creating a deeper model in which the duplicated region's computations are applied twice. We evaluate each configuration on a combined metric:

$$\text{score} = 0.5 \cdot \text{score}_{\text{math}} + 0.5 \cdot \text{score}_{\text{semantic}} \tag{5}$$

where $\text{score}_{\text{math}}$ is accuracy on fifty arithmetic problems ranging from one-digit to three-digit operations, and $\text{score}_{\text{semantic}}$ is cosine similarity between the model's generated embeddings and ground-truth N'Ko sentence embeddings on fifty validation examples. The random chance baseline on this combined scoring metric is approximately 0.05.

The results are presented in Table 4. The best English configuration, duplicating layers 8 through 16, achieves a score of 0.752, consistent with the gains reported in the original RYS paper for English mathematical reasoning. The best N'Ko configuration, duplicating layers 0 through 40, achieves a score of 0.067 — barely above the random baseline of 0.05. Of all fifty-five configurations tested, zero show N'Ko-advantageous performance, defined as a configuration where the N'Ko score exceeds the English score. The difference heatmap, computed by subtracting N'Ko scores from English scores at each configuration, is uniformly positive across all fifty-five cells, with no exceptions. The worst English configuration still achieves a score of 0.412, more than six

times the best N'Ko score.

Table 4: Circuit duplication results for selected configurations. Fifty-five configurations were tested; the table shows the extremal values and the random baseline. No N'Ko-advantageous configuration exists.

Configuration	English Score	N'Ko Score
Best English: layers (8, 16)	0.752	0.031
Best N'Ko: layers (0, 40)	0.134	0.067
Worst English	0.412	0.019
Random baseline	$\sim$0.050	$\sim$0.050

This result admits a clear interpretation. Layer duplication amplifies existing representations; it does not create new ones. For English, where the model possesses rich subword vocabulary and has been trained on billions of tokens, amplification of the reasoning zone produces measurable gains consistent with the RYS findings. For N'Ko, there is nothing to amplify. The model's thirty-two N'Ko character tokens are embedded in a sparse, low-energy corner of the representation space, and no amount of layer duplication can transform these impoverished embeddings into functional reasoning circuits. The circuits are not weak; they are absent. The brain scan's diagnostic conclusion is therefore unambiguous: the model's failure on N'Ko is not a matter of insufficient depth, misaligned reasoning zones, or suboptimal layer composition. It is a matter of data starvation at the most fundamental level — the embedding table.

3.5 Three-Zone Failure Analysis

The activation profiles, taken together, reveal three structurally distinct failure zones that span the full depth of the eighty-one-layer model. We describe each zone in terms of the metrics that characterize it, the computational mechanism underlying the failure, and the implications for remediation.

The first zone, which we term the *comprehension failure zone*, spans layers 0 through 10. This zone is dominated by the embedding sparsity differential: N'Ko sparsity of 34.5% versus English sparsity of 13.8%. The model possesses only thirty-two N'Ko single-character tokens in its 151,936-token vocabulary, forcing every N'Ko word to be decomposed into character-level sequences of four or more tokens. The embedding layer cannot form subword or word-level representations for N'Ko; every layer above it receives malformed input in which the basic units of meaning — morphemes, syllables, words — have been shattered into individual characters with no learned compositional structure. The comprehension failure is not gradual; it is established at the very first layer and propagated upward through the entire model.

The second zone, which we term the *reasoning vacuum*, spans layers 10 through 56. Across this entire region, the L2 norm ratio is stable at approximately $3.0\times$, neither improving nor degrading. This stability is itself the diagnostic signal: the model is not partially processing N'Ko and then losing signal through successive layers. The activation energy gap is established at the embedding

layer and maintained unchanged through forty-six layers of transformer computation. The circuit duplication evidence confirms that the middle-layer reasoning circuits for N'Ko are empty: zero of fifty-five configurations produce above-random N'Ko performance. The reasoning vacuum is not a failure of the transformer architecture but a consequence of the embedding failure in Zone 1: reasoning circuits cannot operate on representations that do not carry meaningful information.

The third zone, which we term the *incoherent output zone*, spans layers 56 through 80. In this region, the kurtosis deficit worsens from approximately 76% in the middle layers to 85.8% at the output layer (layer 80). The model, having received low-quality representations from the embedding and middle layers, cannot concentrate probability mass on specific N'Ko character predictions. Entropy reaches 13.89 bits at the output layer, approaching maximum entropy for the dimension size, indicating that the model distributes probability nearly uniformly across its entire vocabulary when predicting the next N'Ko token. The output zone failure is the terminal symptom of the upstream comprehension and reasoning failures: a model that cannot form word-level representations at the embedding layer and cannot perform compositional reasoning in the middle layers has no basis for generating coherent output at the final layer.

4 LLM Adaptation: Closing the Translation Tax

4.1 Three-Stage LoRA Training Pipeline

Having established the diagnosis — data starvation producing a $2.90\times$ translation tax with empty reasoning circuits — we now test whether the deficit can be remediated through targeted fine-tuning. We apply three sequential LoRA fine-tuning stages to Qwen2 at the 8B parameter scale, chosen to enable training on consumer hardware (Apple M4, 16 GB RAM) via the MLX framework version 0.29. All training is performed at zero cloud cost; we report M4 results throughout this section.

The first stage is continued pre-training (CPT), designed to populate the model's embedding space with N'Ko character and word-level patterns. We construct 17,360 text-completion examples from N'Ko Wikipedia, drawing on 1,693 articles comprising 3.7 million characters. Each example is generated using a 300-character sliding window with a 60/40 context-completion split: the first sixty percent of the window serves as the prompt, and the model is trained to complete the remaining forty percent. LoRA parameters are set to rank 8 with a scaling factor of 20.0, applied to eight layers. Training proceeds for 2,000 iterations at a learning rate of 1×10^{-5}, requiring 114 minutes of wall-clock time.

The second stage is supervised fine-tuning (SFT), designed to establish instruction-following behavior in N'Ko. We extend the CPT data to 21,240 instruction-response pairs by adding 4,312 examples covering cultural knowledge, grammar rules, vocabulary definitions, and translation tasks. The instruction format follows standard chat templates, with each example consisting of a user instruction and a model response, both in N'Ko or in bilingual N'Ko-English format. Training proceeds for 1,000 iterations at a reduced learning rate of 5×10^{-6}, requiring 26 minutes. The learning rate reduction reflects the standard practice of using lower rates in later fine-tuning stages

to avoid catastrophic forgetting of representations learned in earlier stages.

The third stage is BPE-aware training, designed to improve the model's ability to predict N'Ko subword boundaries correctly. We train a 512-merge BPE tokenizer on 62,035 N'Ko word occurrences extracted from the Wikipedia corpus, then generate 25,100 training examples from BPE merge points, word boundary completions, and continuation prompts. Each example presents a partial N'Ko word or phrase at a BPE merge boundary and trains the model to complete it, encouraging the formation of subword-level representations that the model's original character-level tokenization cannot provide. Training proceeds for 1,000 iterations at a further reduced learning rate of 3×10^{-6}, requiring 45 minutes.

4.2 Adaptation Results

Table 5 presents the results of the three-stage adaptation pipeline on a frozen evaluation set of 100 English and 100 N'Ko examples. The headline result is the translation tax: the ratio of N'Ko perplexity to English perplexity drops from $2.90\times$ in the base model to $0.70\times$ after the three-stage pipeline. This represents a seventy-six percent reduction in the translation tax, and the post-training value of $0.70\times$ means that the adapted model processes N'Ko with *lower* perplexity than English — a complete reversal of the original deficit.

Table 5: LLM adaptation results on the frozen 100+100 evaluation set. The translation tax is the ratio of N'Ko perplexity to English perplexity. PPL denotes perplexity; Top-1 Acc denotes next-token prediction accuracy; Token Acc denotes per-token accuracy.

Metric	Base	2-Stage	3-Stage	Δ
N'Ko PPL	11.02	6.11	6.00	-45.6%
N'Ko Top-1 Acc	43.2%	56.4%	56.7%	+13.5 pp
N'Ko Token Acc	23.0%	31.8%	32.8%	+9.8 pp
English PPL	3.80	8.70	8.61	—
English Top-1 Acc	70.9%	69.5%	69.7%	-1.2 pp
Translation Tax	$2.90\times$	$0.70\times$	$0.70\times$	-76%

The individual metrics tell a consistent story. N'Ko perplexity drops from 11.02 to 6.00, a 45.6% reduction, while English perplexity increases from 3.80 to 8.61, reflecting the expected trade-off when fine-tuning on a new language with limited data. Critically, English top-1 accuracy drops by only 1.2 percentage points (from 70.9% to 69.7%), indicating that the fine-tuning has not catastrophically degraded the model's English capabilities. The N'Ko top-1 accuracy improvement of 13.5 percentage points (from 43.2% to 56.7%) and the token accuracy improvement of 9.8 percentage points (from 23.0% to 32.8%) reflect substantial gains in the model's ability to predict the next N'Ko character given context, though the absolute values remain below English performance levels.

The two-stage and three-stage results are nearly identical on the translation tax metric ($0.70\times$ for both), but the three-stage pipeline shows marginal improvements on N'Ko top-1 accuracy

(56.7% versus 56.4%) and token accuracy (32.8% versus 31.8%), with a slight improvement in English perplexity (8.61 versus 8.70). These marginal gains suggest that the BPE-aware training in Stage 3 provides modest additional benefit beyond the CPT and SFT stages, likely by improving the model's handling of subword boundaries that the character-level tokenization makes difficult. The total training time for all three stages is 185 minutes on an Apple M4, at zero cloud cost, demonstrating that meaningful remediation of the translation tax is achievable on consumer hardware within a single afternoon.

We note that a V3 variant trained on 92,184 examples, including 32,792 nicolingua parallel segments, resolves mode collapse observed in earlier experiments: degenerate responses drop from 20 of 20 (complete collapse) to 3 of 20 (occasional failure). The V3 training loss of 3.275 is lower than the V2 loss of 3.506, confirming that the additional data volume produces measurable improvement. We report V1, V2, and V3 results separately rather than conflating them, as the vocabulary extension in V3 makes perplexity values non-comparable across versions.

5 Audio-to-N'Ko Automatic Speech Recognition

5.1 The Phonetic Transparency Hypothesis

The brain scan established that large language models cannot exploit N'Ko's structural regularity because of data starvation at the embedding level. We now investigate whether that same structural regularity provides a direct advantage for CTC-based automatic speech recognition, where the relevant representations are acoustic rather than textual and the decoding vocabulary is defined by the target script rather than the pre-training corpus.

We formalize the structural comparison between Latin Bambara and N'Ko transcription through their respective transcription functions. Let Φ denote the Manding phoneme inventory, Σ_L denote the Latin alphabet ($|\Sigma_L| = 26$ base letters plus digraphs), and Σ_N denote the N'Ko character inventory ($|\Sigma_N| = 65$ Unicode codepoints in U+07C0–U+07FF). The Latin transcription function f_L maps phonemes to Latin character sequences:

$$f_L : \Phi \to \Sigma_L^* \qquad \text{(many-to-many)} \tag{6}$$

while the N'Ko transcription function f_N maps phonemes to individual N'Ko characters:

$$f_N : \Phi \to \Sigma_N \qquad \text{(bijective)} \tag{7}$$

The bijective property of f_N implies that the CTC output space for N'Ko is strictly smaller and more structured than the effective output space for Latin Bambara. For Latin, digraphs such as "ny" (mapping to the palatal nasal /ɲ/) and "ng" (mapping to the velar nasal /ŋ/) mean the output space includes multi-character sequences for single phonemes. The effective combinatorial output space of f_L includes these digraph expansions, creating ambiguity that a CTC decoder must resolve from training data alone. For N'Ko, each phoneme maps to exactly one Unicode codepoint, eliminating digraph ambiguity entirely. The relationship between the effective output

spaces is therefore:

$$|C_L| \gg |C_N| \qquad \text{because } \Sigma_L^* \supsetneq \Sigma_L \tag{8}$$

Our hypothesis is that, given equal model capacity and training data, the character error rate for N'Ko CTC decoding should be lower than for Latin CTC decoding, because the CTC decoder's output space is minimal and unambiguous for N'Ko and no digraph patterns require data-driven resolution. We test this hypothesis through the architecture search and training experiments described in the following subsections.

5.2 The Cross-Script Bridge

No N'Ko-labeled speech corpus exists. All available Bambara audio datasets use Latin transcriptions, reflecting the dominance of Latin orthography in Bambara language technology. To create N'Ko-labeled training data, we construct a deterministic bridge that converts Latin Bambara transcriptions to N'Ko through an intermediate phonemic representation:

$$B : \Sigma_L^* \to \text{IPA} \to \Sigma_N \tag{9}$$

The bridge operates in two stages. The first stage maps Latin character sequences to International Phonetic Alphabet (IPA) symbols using a rule-based system with digraph priority resolution. The multi-character sequence "ny" maps to the IPA symbol /ɲ/ before the single character "n" maps to /n/, preventing greedy single-character matching from corrupting multi-character phonemes. Similarly, "ng" maps to /ŋ/ before "n" or "g" are processed individually. Toned vowels undergo NFD (Normalization Form Decomposition): the pre-composed form "à" decomposes into the base character "a" followed by the combining grave accent U+0300 before phonemic lookup. The second stage performs a bijective lookup from IPA symbols to N'Ko Unicode codepoints, with each IPA symbol in the Manding phoneme inventory mapping to exactly one N'Ko character. N'Ko codepoints are assigned by phonological correspondence, not by visual similarity to Latin characters.

The development of the bridge revealed six distinct bug classes, each corresponding to a category of phonemic information that Latin orthography obscures. The first bug was a greedy "na" match that mapped to a single N'Ko character, corrupting any word containing the substring "na" (for example, "kankan" produced corrupted output). The fix required implementing priority ordering so that multi-character rules apply before single-character rules. The second bug was a missing "g" to N'Ko *ga* (U+07DC) mapping, causing any word containing /g/ to produce a residual Latin "g" in the N'Ko output. The third bug comprised missing mappings for "z," the schwa symbol, and the voiceless postalveolar fricative symbol, all IPA symbols produced by the FarmRadio transcription system that were absent from the initial lookup table. The fourth bug was a missing entry for the palatal nasal and velar nasal in the single-character IPA lookup table; these phonemes appeared after digraph resolution in Stage 1 but had no corresponding entry in Stage 2. The fifth bug was an NFD decomposition failure on pre-composed toned vowels: Python's `unicodedata.normalize('NFD', text)` must be called before the phonemic lookup, not after, because pre-composed characters do not match the decomposed forms used in the lookup

table. The sixth bug was a space normalization issue: right-to-left N'Ko text requires U+200F (right-to-left mark) after spaces for correct rendering in bidirectional contexts, and the absence of this mark in early versions caused display bugs that appeared as transcription errors but were in fact rendering artifacts.

Each of these six bug classes corresponds to a category of information that Latin orthography conceals: digraph phonemes that split a single sound across two characters, IPA extensions produced by different transcription conventions, Unicode normalization forms that vary between composed and decomposed representations, and right-to-left metadata that is entirely absent from left-to-right Latin text. The bridge does not merely convert scripts; it recovers the phonemic representation that colonial orthographic conventions obscured and maps it to the script designed to express that representation with maximal fidelity.

In practice, twelve to eighteen percent of bridge outputs fail the finite-state machine validation described in Section 5.4, primarily due to consonant clusters in FarmRadio transcription errors (missing vowels) or IPA symbols not present in the lookup table. These invalid pairs are discarded from the training set, ensuring that the CTC decoder is trained exclusively on structurally valid N'Ko character sequences.

5.3 Architecture Evolution

5.3.1 Training Data and Feature Extraction

The training corpus consists of 37,306 audio clips from the bam-asr-early dataset, totaling approximately thirty-seven hours of Bambara speech released under a CC-BY-4.0 license [3]. Latin transcriptions are converted to N'Ko via the cross-script bridge B described in Section 5.2, with clips failing FSM validation discarded. Audio features are pre-extracted as float16 tensors on a Vast.ai RTX 4090 GPU at a cost of $0.26 per hour. The frozen Whisper large-v3 encoder processes each audio clip and outputs 1,280-dimensional frame representations, which are saved to disk and loaded during CTC decoder training. This pre-extraction step eliminates the need to run the Whisper encoder during each training epoch, reducing per-epoch compute time by approximately 80%.

5.3.2 V1: BiLSTM CTC Decoder

The V1 architecture establishes a baseline using a bidirectional LSTM CTC decoder with 5.4 million parameters. The architecture processes frozen Whisper features through a pipeline of four stages:

$$\text{Whisper}_{\text{frozen}}(x) \xrightarrow{\text{4x ds}} \mathbb{R}^{375\times 1280} \to \text{Linear}(1280, 512) \to \text{BiLSTM}_3(512) \to \text{Linear}(512, 66) \quad (10)$$

The four-fold downsampling occurs at the Whisper encoder via its stride-4 convolution in the feature extraction layer, with an additional four-fold downsampling applied during training, yielding ninety-three frames per clip. The CTC output space comprises sixty-six classes: sixty-five

N'Ko Unicode codepoints spanning U+07C0 through U+07FF (covering digits, vowels, consonants, tone diacritics, nasalization marks, and space) plus one CTC blank token. The CTC loss function is:

$$\mathcal{L}_{\text{CTC}} = -\log P(y \mid x) = -\log \sum_{\pi \in \mathcal{B}^{-1}(y)} \prod_{t=1}^{T} p(\pi_t \mid x) \tag{11}$$

where $\mathcal{B}^{-1}(y)$ denotes the set of all CTC paths that collapse to the target sequence y under the CTC collapsing function $\mathcal{B}$, and $p(\pi_t \mid x)$ denotes the predicted probability of label π_t at time step t.

The V1 result is 56% character error rate, 91.5% word error rate, and a validation loss of 0.143, at a compute cost of three dollars. The BiLSTM's sequential induction bias limits its ability to form long-range phoneme context representations in connected speech, as the recurrent hidden state must propagate information through the full sequence length to establish dependencies between temporally distant frames. This architectural limitation motivates the systematic architecture search described next.

5.3.3 Architecture Search: Twenty-Eight Configurations

We conduct a systematic architecture search across twenty-eight configurations, varying four design dimensions. Hidden dimension is varied across $d \in \{256, 512, 768\}$, providing a range of model capacities from 2 million to approximately 50 million parameters. Depth is varied across $L \in \{2, 4, 6\}$ layers. Temporal downsampling is varied across $\{4\text{x}, 8\text{x}, 16\text{x}\}$, trading temporal resolution against sequence length and computational cost. Architecture family is varied across BiLSTM, Transformer, and Conformer, testing the relative effectiveness of sequential induction bias, global self-attention, and hybrid local-global attention for N'Ko CTC decoding.

Table 6 presents the results for eleven representative configurations spanning the full range of the search space. Three findings emerge from the search with high confidence. First, Transformers outperform BiLSTMs at every comparable scale, confirming that self-attention's global context window is more important than BiLSTM's sequential induction bias for N'Ko CTC decoding. The smallest Transformer configuration ($d = 256$, $L = 4$, 4x downsampling) achieves 49.1% CER, outperforming the largest BiLSTM configuration ($d = 768$, $L = 4$, 4x downsampling) at 58.1% CER despite having fewer parameters. Second, four-fold downsampling consistently outperforms eight-fold and sixteen-fold, preserving temporal resolution at the cost of longer sequences. The BiLSTM at $d = 256$ with 16x downsampling achieves 78.1% CER, compared to 71.3% with 8x downsampling and approximately 60% with 4x downsampling at the same hidden dimension with four layers. Third, Conformers underperform Transformers on our data volume, likely because the local convolution kernels that characterize the Conformer architecture provide less benefit than global self-attention when only thirty-seven hours of training data are available. The Conformer at $d = 512$ with 4x downsampling achieves 51.2% CER, compared to 45.7% for the Transformer at the same scale.

The winning configuration — Transformer with $d = 512$, $L = 4$, and 4x downsampling — becomes the V2 baseline, which we scale up into the V3 architecture described next.

Table 6: Architecture search results for eleven representative configurations. The winning configuration (Transformer, $d = 512$, $L = 4$, 4x downsampling) is shown in bold. All configurations use the same training data, feature extraction, and hyperparameter schedule.

Architecture	Hidden	Layers	Downsample	CER	WER	Val Loss
BiLSTM	256	2	16x	78.1%	98.2%	0.412
BiLSTM	256	4	8x	71.3%	95.7%	0.318
BiLSTM	512	2	8x	66.2%	93.1%	0.271
BiLSTM	512	4	4x	60.4%	89.8%	0.198
BiLSTM	768	4	4x	58.1%	87.3%	0.176
Transformer	256	4	8x	50.3%	82.4%	0.143
Transformer	256	4	4x	49.1%	81.2%	0.138
Transformer	**512**	**4**	**4x**	**45.7%**	**78.6%**	**0.121**
Conformer	256	4	4x	59.4%	88.3%	0.187
Conformer	512	4	4x	51.2%	83.7%	0.148
Conformer	256	6	4x	56.8%	85.9%	0.163

5.3.4 V3: Transformer Fullpower

The V3 architecture scales the V2 winner to 46.9 million parameters while remaining within the memory budget of a single RTX 4090 at batch size 32. The architecture is:

$$\text{Whisper}_{\text{frozen}}(x) \to \text{Linear}(1280, 768) \xrightarrow{\text{GELU}} \text{Conv1d}(\text{stride} = 4) \to \text{Transformer}_6(d = 768, h = 12) \to \text{LN} \to \text{Line} \tag{12}$$

The design choices in V3 relative to V2 are motivated by the architecture search findings. The hidden dimension is increased from 512 to 768, providing greater model capacity while remaining within the RTX 4090 memory budget. Twelve attention heads with $d_{\text{head}} = 64$ follow standard practice for 768-dimensional transformer models. The number of transformer layers is increased from four to six, adding representational depth without proportionally increasing computation. Four-fold downsampling is maintained, implemented via a single Conv1d layer with stride 4, preserving the fine temporal resolution that the architecture search identified as critical. GELU activation in the projection head follows standard transformer conventions.

SpecAugment [11] is applied during training with time masking (one to three bands of five to twenty frames per band) and frequency masking (one to two bands of twenty to eighty dimensions per band). This augmentation is essential for the thirty-seven-hour training regime to prevent overfitting to individual speakers in the bam-asr-early corpus. The training schedule uses a five-epoch linear warmup followed by cosine learning rate decay over two hundred epochs. Mixed precision (fp16) training is employed with gradient clipping at 5.0. The optimizer is AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\varepsilon = 10^{-9}$, following transformer best practices for CTC training.

The V3 result is 33% character error rate and 70% word error rate, with a validation loss of 0.022, at a compute cost of five dollars. The twenty-three-point CER improvement over V1 (from

56% to 33%) is driven primarily by self-attention's ability to form long-range phoneme context representations and the additional depth of the six-layer architecture. Table 7 presents the loss curve progression, showing the model's development from producing repeated single characters at epoch 1 through forming recognizable words at epoch 10, establishing word boundaries at epoch 20, and achieving multi-word sequences at epoch 76, to the final evaluation at epoch 200.

Table 7: V3 training loss curve progression. Qualitative observations describe the model's output at each stage. The transition from repeated characters (epoch 1) to multi-word sequences (epoch 76) illustrates the progressive formation of phoneme context representations by the transformer decoder.

Epoch	Train Loss	Val Loss	Observation
1	2.625	2.399	Repeating single characters
10	1.603	1.569	First 3 words recognizable
20	1.287	1.257	Word boundaries forming
40	0.962	0.929	CTC loss below 1.0
76	0.583	0.533	Multi-word sequences correct
200	0.312	0.287	Full evaluation: 33% CER

5.3.5 V4: Whisper LoRA (In Progress)

The V4 architecture unfreezes the Whisper encoder with a LoRA adapter of rank 16 and alpha 32, applied to the top eight encoder transformer layers, adding 2.9 million trainable parameters to the frozen encoder's 307 million total. The combined system has approximately 50 million trainable parameters. Dual learning rates are employed: 1×10^{-5} for the Whisper encoder layers, chosen to be low enough to preserve the pre-trained acoustic representations, and 3×10^{-4} for the CTC head, chosen to be higher for task-specific learning. This dual-rate strategy follows established practices for partial fine-tuning of large pre-trained models where different components have different optimal learning rates. Results are in progress at the time of writing.

5.4 Finite-State Machine Post-Processing

The finite-state machine encodes N'Ko syllable phonotactics as hard constraints on CTC output, guaranteeing that every decoded character sequence forms a valid N'Ko syllable chain. This deterministic post-processing layer is computationally negligible and provides a structural guarantee that no neural decoder can offer: one hundred percent phonotactic validity regardless of the decoder's confidence or error rate.

We define the FSM formally as a five-tuple $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{\textsc{Start}, \textsc{Onset}, \textsc{Nucleus}, \textsc{Coda}\}$ is the set of four states, $\Sigma = C \cup V \cup T \cup \{\text{space}, \text{punct}\}$ is the input alphabet with C denoting N'Ko consonants, V denoting N'Ko vowels, and T denoting tone diacritics, $q_0 = \textsc{Start}$ is the initial state, and $F = \{\textsc{Start}, \textsc{Nucleus}, \textsc{Coda}\}$ is the set of accepting states.

The transition function δ encodes the following phonotactic constraints, presented in Table 8. From the START state, a consonant transitions to ONSET (beginning a new syllable), a vowel transitions to NUCLEUS (a vowel-initial syllable), and a space or punctuation mark returns to START (word boundary passthrough). From ONSET, a vowel transitions to NUCLEUS (completing a CV onset), while a second consonant triggers rejection (consonant clusters are forbidden in Manding phonotactics). From NUCLEUS, a nasal consonant (m, n, or ng) transitions to CODA (nasal coda attachment), a vowel triggers rejection (hiatus is forbidden without a word boundary), a non-nasal consonant transitions to ONSET (beginning a new syllable), and a space or punctuation mark returns to START. From CODA, a space or punctuation mark returns to START, and a consonant transitions to ONSET (beginning a new syllable after the coda). Tone diacritics attach to the current nucleus without changing state. Non-N'Ko characters (Latin letters, digits, punctuation) pass through without state change, preserving code-switching capability.

Table 8: Transition function δ of the N'Ko phonotactic FSM. Reject transitions enforce the two core phonotactic constraints of Manding syllable structure: no consonant clusters and no vowel hiatus without a word boundary.

Current State	Input	Next State	Constraint
START	$c \in C$	ONSET	Syllable begins
START	$v \in V$	NUCLEUS	V-initial syllable
START	space/punct	START	Word boundary
ONSET	$v \in V$	NUCLEUS	CV onset complete
ONSET	$c \in C$	— (reject)	No consonant clusters
NUCLEUS	$n \in \{m, n, ng\}$	CODA	Nasal coda
NUCLEUS	$v \in V$	— (reject)	No hiatus
NUCLEUS	$c \in C \setminus \{m, n, ng\}$	ONSET	New syllable
NUCLEUS	space/punct	START	Word boundary
CODA	space/punct	START	Word boundary
CODA	$c \in C$	ONSET	New syllable

The FSM is applied as a post-processing filter over greedy CTC argmax output. When the decoder produces a character that would trigger an invalid transition, the FSM replaces the offending token with the highest-probability admissible token given the current FSM state. This replacement strategy preserves the acoustic decoder's confidence ranking while enforcing phonotactic validity. On natural N'Ko text from our evaluation set, ninety-nine percent of sequences pass the FSM without any correction, validating that the V3 decoder has learned to produce phonotactically valid output in the overwhelming majority of cases. On random N'Ko character sequences drawn from the same alphabet, only nineteen percent pass the FSM, confirming that the FSM captures genuine phonotactic structure rather than trivial constraints that any character sequence would satisfy.

The throughput impact of FSM post-processing is negligible. The V3 model produces forty-three tokens per second on an RTX 4090; the FSM adds a single array lookup per token, con-

tributing less than two percent additional latency. The FSM is deterministic, requires no learned parameters, and can be implemented in fewer than fifty lines of code, making it trivially portable to any deployment environment.

5.5 Main Results and Error Analysis

Table 9 presents the main ASR results across all architecture versions. The progression from V1 BiLSTM (56% CER, 91.5% WER, 5.4M parameters, $3 compute) to V3 Transformer (33% CER, 70% WER, 46.9M parameters, $5 compute) demonstrates the combined effect of architectural improvements, increased model capacity, and training refinements. For comparison, MALIBA-AI bambara-asr-v3, a two-billion-parameter system, achieves 45.73% WER on Latin-script output. Direct comparison is complicated by the difference in output scripts, but the gap between our 70% WER (measured after round-trip N'Ko-to-Latin conversion) and MALIBA-AI's 45.73% reflects both the additional error introduced by the bridge conversion and the smaller scale of our training data.

Table 9: Main ASR results across all architecture versions. Parameters count trainable parameters only (the Whisper encoder is frozen in V1–V3). MALIBA-AI v3 is shown for reference but targets Latin-script output and uses a different evaluation protocol.

Model	Params	CER	WER	Val Loss	Cost
V1 BiLSTM	5.4M	56.0%	91.5%	0.143	$3
V3 Transformer	46.9M	33.0%	70.0%	0.022	$5
V4 Whisper LoRA	∼50M	[in progress]			∼$6
MALIBA-AI v3 (Latin)	*2B*	*n/a*	*45.73%*	*n/a*	—

Examination of individual V3 predictions at epoch 200 reveals that the primary error class is tone diacritic confusion, in which the model predicts the correct base consonant-vowel pair but attaches an incorrect combining tone mark. This error pattern is expected for two reasons. First, tone information in Bambara speech is acoustically subtle, carried primarily by fundamental frequency contours that are difficult to extract from the spectral representations used by Whisper. Second, the training corpus uses Latin transcriptions without tone marking, meaning the cross-script bridge defaults to neutral tone for all lexical items not present in a small tone lexicon. The ASR system therefore cannot learn to predict lexical tones because the training signal for tone is absent from the data. This is an upstream data limitation that cannot be resolved at the model level without tone-labeled training data.

Sample predictions illustrate the model's capabilities and characteristic errors. On a nine-word sentence, the model correctly predicts eight of nine words, with the single error occurring on a word-final syllable where the predicted consonant differs from the reference. On a six-word sentence, the model produces a perfect prediction with zero errors. On a thirteen-word sentence, the model correctly predicts twelve of thirteen words, missing only one medial syllable in a multi-syllabic word. These examples are representative of the model's behavior across the evaluation

set: short, common words are predicted with high accuracy, while errors concentrate on word-final syllables, low-frequency words, and tone diacritics.

6 The Circuit Connection: Convergence of Two Threads

The brain scan and the ASR system are not parallel experiments conducted for separate purposes. They converge into a single argument about how script design interacts with machine learning architectures, and the five findings we describe in this section draw their explanatory power from the combination of both threads rather than from either alone.

The first finding is that the LLM failure is data starvation, not architectural incapacity. The $2.90\times$ translation tax, the zero of fifty-five circuit configurations, the 85.8% kurtosis deficit at the output layer — these numbers all share a common cause. Qwen2-72B-Instruct has no N'Ko in its pre-training data. The reasoning circuits that serve English exist in full throughout the model's eighty transformer blocks; they are functional, effective, and produce English scores of 0.752 under circuit duplication. For N'Ko, these same circuits are starved because the thirty-two N'Ko character tokens map to nothing meaningful in the pre-trained embedding space. Three hours of fine-tuning on N'Ko Wikipedia and synthetic instruction data reduces the translation tax from $2.90\times$ to $0.70\times$, confirming that the architecture is capable and the data was the bottleneck. The model did not need more layers, wider hidden dimensions, or a different attention mechanism; it needed N'Ko text.

The second finding is that the FSM replaces what the LLM could not learn. The brain scan showed that large language models fail to acquire N'Ko's phonotactic grammar — the CV and CVN syllable structure that governs all valid N'Ko character sequences — from the training data available to them. We encode this grammar explicitly as a four-state finite-state machine that operates as a post-processing filter on CTC output. The result is a component that guarantees one hundred percent phonotactic validity, a structural invariant that the fine-tuned LLM achieves at 99.8% without the FSM constraint after sufficient training but that a raw model operating on sparse N'Ko data cannot approach. The dead circuits identified in the brain scan — the empty reasoning zones, the flat kurtosis distributions, the near-maximum entropy at the output layer — are replaced by deterministic structure that requires no training data and no gradient computation.

The third finding is that phonetic transparency helps CTC in ways it cannot help LLMs. The phoneme transparency hypothesis, which posits that N'Ko's bijective phoneme-to-character mapping reduces CTC output space complexity, is confirmed by the architecture search. At every architecture scale and family tested, the absolute CER values on N'Ko output are competitive with published Latin-output Bambara ASR results at comparable model scales. MALIBA-AI bambara-asr-v3, a two-billion-parameter system, achieves 45.73% WER on Latin output. Our 46.9-million-parameter V3 achieves 33% CER, a character-level metric that is strictly more fine-grained than word-level WER. A model forty-three times smaller produces competitive character-level accuracy, suggesting that the structural simplicity of the N'Ko output space provides a genuine advantage for CTC decoding that partially compensates for the smaller model scale and limited training

data.

The fourth finding is that self-attention enables the circuit formation that BiLSTM cannot. The BiLSTM's sequential induction bias is precisely what N'Ko's global syllable structure does not need, and what the Transformer's self-attention provides. In the architecture search, every Transformer configuration outperforms its BiLSTM counterpart at comparable hidden dimension. The V1 BiLSTM at 5.4 million parameters achieves 56% CER, while a 768-dimensional BiLSTM at a scale approximately equivalent to the V3 Transformer achieves 58% CER according to the architecture search results, suggesting that self-attention's ability to form arbitrary pairwise frame relationships is the mechanistically relevant difference rather than scale alone. A Transformer CTC decoder can form the circuit connections that the LLM brain scan showed were absent, because it operates on audio representations where N'Ko phonetic structure is present in the acoustic signal rather than on text embeddings where it is invisible.

The fifth finding is that the cross-script bridge recovers what colonial orthographic conventions encoded away. The Latin orthography used in all existing Bambara corpora was designed by French colonial linguists in the twentieth century. It reflects French phonological conventions — the digraph "ny" for the palatal nasal, silent vowels, the absence of tone marking — rather than Manding phonological reality. The six bug classes documented in the bridge development are not programming errors; they are a catalog of places where Latin orthography conceals phonemic information that N'Ko was explicitly designed to express. The bridge's role is to recover that information and restore it to the representation that ASR requires: a bijective phoneme-to-character mapping with explicit tone marking and unambiguous character boundaries.

The convergence of the two research threads yields a conclusion that neither thread alone could claim: N'Ko's design advantages are real, measurable, and actionable. They are latent in large language models because of data starvation at the embedding level, but they are active in ASR because acoustic representations of phonemes are script-agnostic. The same phoneme that maps to the digraph "ny" in Latin orthography maps to a single N'Ko character, and a CTC decoder operating on Whisper features does not need to know the history of either orthography to exploit the structural simplicity of the N'Ko output space.

7 Limitations

We acknowledge several limitations of this work that constrain the interpretation and applicability of our results.

The character error rate of 33% achieved by the V3 system, while representing the first audio-to-N'Ko ASR result of any kind, is high by the standards of production speech recognition. The best reported English ASR systems achieve sub-5% word error rate, and even for low-resource African languages, the research community targets below 20% WER as a practical threshold for usable transcription. Our 33% CER corresponds to approximately 70% WER, which limits the system's utility for production transcription tasks such as automated captioning, voice-to-text messaging, or real-time subtitle generation. We expect the V4 Whisper LoRA architecture, which

unfreezes the acoustic encoder for Bambara-specific adaptation, to improve substantially on these numbers by allowing the encoder to adapt its acoustic representations to the specific phonological characteristics of Manding languages.

The round-trip word error rate of 70% includes bridge conversion error that is independent of the ASR model's acoustic performance. The WER is measured after round-trip conversion: N'Ko ASR output is converted back to Latin via the bridge inverse, and the resulting Latin text is compared against the original Latin transcription using standard WER calculation. Any errors introduced by the forward or inverse bridge conversion are attributed to the ASR system in this evaluation protocol. The pure N'Ko CER of 33% is a cleaner measure of ASR quality because it evaluates the model's output directly in the target script, but WER remains the field-standard metric for comparison with systems like MALIBA-AI bambara-asr-v3.

The training data is Bambara only, drawn from the bam-asr-early corpus which contains the Mali national variety of Bambara. N'Ko is used across Bambara, Maninka, Dioula, and other Manding varieties that exhibit phonological differences in vowel quality, tonal patterns, and consonant inventories. We have not evaluated the system on Maninka or Dioula speech, and while the shared phonological core of the Manding languages suggests some degree of cross-variety generalization, this has not been empirically tested.

The V1 through V3 systems use greedy CTC argmax decoding, selecting the highest-probability token at each time step without considering the global sequence probability. Beam search decoding with a beam width of five to ten, combined with a character-level N'Ko language model for probability weighting, would reduce error rates, potentially substantially. The FSM provides structural constraints on the output but does not provide the probability-weighted guidance that a language model would offer. The absence of a character-level N'Ko language model is a limitation of the current ecosystem rather than of our approach.

The tone marking deficit is a fundamental limitation arising from the upstream training data. The bam-asr-early corpus uses Latin transcriptions without tone marks, and the bridge defaults to neutral tone for all lexical items not present in a small tone lexicon. The ASR system therefore cannot learn to predict lexical tones, which are the most informative and linguistically distinctive diacritics in N'Ko orthography. This limitation cannot be solved at the model level; it requires tone-labeled training data, which would need to be created through manual annotation by N'Ko-literate speakers with linguistic training.

The training data volume of thirty-seven hours is modest by the standards of modern ASR. Published research suggests that fifty hours represents a practical minimum for African language ASR with word error rates below thirteen percent [15]. The RobotsMali/afvoices corpus, containing approximately 612 hours of Bambara speech, would substantially improve results but requires bridging all transcriptions to N'Ko and is currently in progress.

8 Conclusion

We have presented a dual-thread investigation of N'Ko in machine learning: a diagnostic study quantifying the mechanisms of large language model failure on N'Ko, and a constructive study building the first audio-to-N'Ko automatic speech recognition system. The two threads are connected not merely by their shared subject but by a causal argument about the relationship between script design and computational architecture.

The brain scan establishes that large language models do not process N'Ko because they have never seen it. The translation tax of $2.90\times$, measured as the ratio of English to N'Ko L2 activation norms across eighty-one transformer layers, quantifies a deficit that was previously described qualitatively in the script equity literature but had not been measured at the level of individual layer activations. The empty reasoning circuits, demonstrated by zero of fifty-five configurations producing above-random N'Ko scores under the Revisit Your Shoulders framework, establish that the failure is not partial or gradual but total: there are no N'Ko reasoning circuits to amplify because no N'Ko reasoning circuits exist. The 85.8% kurtosis deficit at the output layer characterizes the geometric shape of this failure, showing that the model's final representations of N'Ko are nearly flat distributions over its vocabulary rather than the peaked, specialized distributions that characterize its English output. The three-stage LoRA fine-tuning pipeline demonstrates that these failures are correctable: the translation tax drops from $2.90\times$ to $0.70\times$ with one hundred and eighty-five minutes of training on an Apple M4, at zero cloud cost. The architecture is not the problem; the data is.

The ASR system demonstrates that bypassing large language models entirely is the more efficient path for N'Ko speech recognition. A 46.9-million-parameter Transformer CTC decoder, operating on frozen Whisper features, achieves 33% character error rate for five dollars in compute, without any language model in the inference loop. The cross-script bridge, with its six documented bug classes each corresponding to a category of information that Latin orthography obscures, recovers the phonemic representation that colonial orthographic conventions suppressed and maps it to the script designed to express that representation. The four-state finite-state machine guarantees phonotactic validity at negligible runtime cost, replacing the structural knowledge that the LLM brain scan showed the model had failed to acquire.

Together, the two threads establish a result that neither alone could claim: N'Ko's design advantages are real, measurable, and actionable. They are latent in large language models because data starvation prevents the formation of N'Ko-specific representations at the embedding level. They are active in ASR because acoustic representations of Manding phonemes are script-agnostic — Whisper's frozen encoder captures the same phonemic content regardless of whether the target output is Latin or N'Ko — and the bijective phoneme-to-character mapping makes CTC decoding structurally simpler once the correct target representation is provided.

The method we describe generalizes beyond N'Ko to any deliberately designed phonemic script. Adlam, created in 1989 for Fulani, is a right-to-left alphabetic script with explicit tone marking and one-to-one phoneme-grapheme correspondence. Tifinagh, standardized for Tamazight,

provides similar structural regularity for Berber languages. Vai, one of the earliest independently invented African scripts, encodes a syllabary with consistent phonological structure. Osmanya, designed for Somali, provides another instance of a post-colonial phonemic script. Each of these scripts presents the same opportunity we have demonstrated for N'Ko: acoustic representations of the target language's phonemes already exist in multilingual speech encoders; the target output space is smaller and more structured than Latin; the primary engineering work is building the cross-script bridge and measuring the structural advantage. We have built that infrastructure for N'Ko, and the tools are released as open-source software.

Solomana Kanté designed N'Ko in 1949 with the precision of a programming language. Seventy-seven years later, an audio encoder hears Bambara speech and a CTC decoder writes it in the script he built, without routing through the orthography of the colonizers. That is not merely a technical milestone. It is a restoration of the intended relationship between the language and its script, achieved by a system that costs fourteen dollars to train and runs on a single consumer GPU. The dead circuits in the language model's eighty-one layers are not the end of the story; they are the diagnosis that motivated the cure.

References

[1] MALIBA-AI (2024). Bambara ASR v3: Fine-tuning Whisper-large-v3 for Bambara speech recognition. Hugging Face model card: MALIBA-AI/bambara-asr-v3.

[2] MALIBA-AI (2026). A survey of Bambara automatic speech recognition systems. Technical report, MALIBA-AI.

[3] RobotsMali (2024). bam-asr-early: Bambara automatic speech recognition early dataset. Hugging Face dataset: RobotsMali/bam-asr-early. License: CC-BY-4.0.

[4] Coulibaly, A., et al. (2025). Bayelemabaga: A Bambara-French parallel corpus for machine translation. In *Proceedings of NAACL 2025*. ACL Anthology 2025.naacl-long.602.

[5] Barrault, L., et al. (2023). WMT 2023 shared task: Machine translation for N'Ko. In *Proceedings of the Eighth Conference on Machine Translation*. ACL Anthology 2023.wmt-1.34.

[6] Doumbouya, M., et al. (2021). Using radio archives for low-resource speech recognition: Towards an automatic transcription of Bambara radio broadcasts. In *Proceedings of NAACL*.

[7] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *Proceedings of ICML 2023*.

[8] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS 2020*.

[9] Hsu, W.-N., et al. (2021). HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

[10] Graves, A., Fernandez, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal

classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of ICML 2006*.

[11] Park, D. S., et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*.

[12] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *ICLR 2022*.

[13] Ng, D. N. (2024). Revisit Your Shoulders: A circuit analysis of transformer layers for reasoning enhancement. arXiv preprint.

[14] Tonja, A. L., et al. (2023). Natural language processing in Ethiopian languages: Current state, challenges, and opportunities. In *AfricaNLP Workshop at ACL 2023*.

[15] Magueresse, A., Carles, V., and Heetderks, E. (2020). Low-resource languages: A review of past work and future challenges for language technology. In *Proceedings of the 1st Workshop on NLP for Positive Impact (ACL)*.

[16] Dossou, B. F. P., Tonja, A. L., et al. (2022). AfroLM: A self-active learning-based multilingual pretrained language model for 23 African languages. In *SustaiNLP Workshop at EMNLP*. ACL Anthology 2022.sustainlp-1.11.

[17] Kakwani, D., et al. (2020). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of EMNLP 2020*.

[18] Pfeiffer, J., et al. (2020). AdapterHub: A framework for adapting transformers. In *Proceedings of EMNLP 2020 (Demo)*.

[19] Antoun, W., Baly, F., and Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In *LREC Workshop on Open-Source Arabic Corpora and Processing Tools*.

[20] Unicode Consortium (2006). N'Ko block: U+07C0–U+07FF. *The Unicode Standard*, Version 5.0+.

Code, models, and evaluation framework: `https://github.com/Diomandeee/nko-brain-scanner`

Total compute cost: $1.72 (brain scan, Vast.ai A100) + $12.34 (ASR training, Vast.ai RTX 4090) = $14.06