Dead Circuits: Script Invisibility and Representation Failure for N'Ko in Large Language Models

Mohamed Diomande

Final paper series, May 2026

Abstract

This paper studies *script invisibility*: the condition in which a large language model accepts a writing system as valid Unicode while allocating little functional internal representation to it. The test case is N'Ko, the script designed by Solomana Kante for Manding languages. N'Ko is not a noisy informal encoding of Bambara, Maninka, or Dioula. It is a dedicated alphabetic system in the Unicode block U+07C0–U+07FF, with a close mapping between Manding phonology and written symbols, explicit diacritic machinery, and an active literacy tradition. For computational linguistics, it should be unusually favorable: it is more phonemically transparent than Latin Bambara, it avoids many digraph ambiguities, and it preserves distinctions that standard Latin transcriptions often hide.

The empirical problem is that current LLMs do not receive N'Ko that way. Across the project papers, activation profiling found a repeated failure signature: reduced hidden-state norms, higher entropy, elevated sparsity, weaker output-layer kurtosis, and little evidence of reusable reasoning circuits for N'Ko strings. In one Qwen3-8B protocol, N'Ko incurred an average representation tax of about 2.94x, a 1.2–1.7 bit entropy gap, approximately 2.2x higher embedding sparsity, a 78.1% output-layer kurtosis deficit, and no N'Ko-advantageous configuration in 45 layer-duplication probes. In a cross-model protocol, the average translation tax was 3.30x for Qwen3-8B, 3.59x for Qwen2.5-7B, and 2.67x for Mistral-7B; N'Ko activations were roughly 66–72% weaker than English, and output-layer kurtosis deficits ranged from 64.6% to 93.5%.

The main conclusion is not that N'Ko is intrinsically difficult. The opposite is more plausible: N'Ko is computationally regular, but invisible in the data and tokenizer regimes from which general LLM capability emerges. Arabic provides the control: another right-to-left script can be handled competently when it receives large-scale pretraining exposure and substantial tokenizer allocation. The failure is therefore structural and historical, not a rendering artifact. The paper argues that script invisibility should be measured directly in hidden-state geometry before downstream claims about translation, ASR correction, or language support are trusted.

1 Introduction

Multilingual model evaluation normally treats script as a surface channel. A model may be tested on English, Arabic, Hindi, Bambara, or another language, but the benchmark rarely asks whether the writing system itself has usable internal support. That habit is unsafe for scripts that exist in Unicode but not in the training distribution. For these scripts, a model can tokenize the characters, pass them through the network, and emit an output, while never forming the kinds of specialized internal circuits that make a script computationally available.

N'Ko exposes this failure cleanly. The script is historically and technically specific: it was created for Manding languages, it is written right-to-left, it is encoded in Unicode, and it supports

a community of readers, teachers, publishers, religious writers, and digital users [5, 1]. Its linguistic design also matters for NLP. Compared with Latin Bambara, N'Ko keeps the representation closer to the target phonological system. A language model that cannot represent N'Ko is not merely missing a font; it is missing access to a writing system that carries distinctions relevant to language technology.

This paper consolidates the representation side of the project. The later papers in the series ask how N'Ko should be evaluated for speech recognition, how a script-native ASR system reached an archived 20.57% character error rate anchor, and how row-level governance should work for deployment. Those papers depend on the diagnostic claim established here: general-purpose LLMs should not be assumed to understand N'Ko simply because they accept N'Ko codepoints. The script must be measured as an internal representation.

The central research question is whether a general LLM allocates functional hidden-state geometry to N'Ko, or merely routes unsupported Unicode through fallback pathways. The working hypothesis is that N'Ko is structurally underrepresented in the tested LLMs because it is absent or nearly absent from training data and poorly supported by tokenizers. If this is correct, the deficit should appear as lower activation energy, more diffuse hidden states, elevated sparsity, reduced output specialization, and weak response to circuit amplification. It should also persist across model families unless the model family explicitly acquired N'Ko data.

2 Research Design and Claim Tests

The paper treats script support as an empirical property of a trained model, not as a label inherited from Unicode coverage. A model may be technically capable of reading the bytes for a script while still lacking the representational geometry required for reliable reasoning, correction, or generation. The research design is therefore built around observable internal states. It asks whether N'Ko induces the same kind of structured hidden-state behavior as better-supported scripts.

This structure is important for publication because the claim is not simply that models "perform badly" on N'Ko. Performance can fail for many reasons: prompt format, evaluation data, decoding parameters, translation ambiguity, or reference quality. Script invisibility is narrower. It predicts a measurable representation gap before any downstream benchmark is scored. The claim is strongest when several independent diagnostics point in the same direction and when a plausible control, such as Arabic, shows that the failure is not caused merely by directionality.

2.1 Operational Claims

The project uses four levels of claim strength. A *rendering claim* says only that the software stack can display and pass N'Ko codepoints. A *tokenization claim* says that the tokenizer allocates usable subword or character structure to N'Ko. A *representation claim* says that hidden states show stable, script-sensitive organization. A *task claim* says that the model performs a downstream task, such as correction or translation, with accuracy that is grounded in the script rather than in a fallback paraphrase. This paper concerns the third level. It does not infer representation from display, and it does not infer downstream competence from representation alone.

Table 1: Research questions, hypotheses, and falsification criteria for the script invisibility claim.

ID	Question	Working hypothesis	What would falsify it
RQ1	Does N'Ko receive comparable hidden-state energy?	N'Ko induces lower layer-wise norms than supported comparison scripts.	Norm ratios approach parity under the same prompts and length controls.
RQ2	Are N'Ko states specialized or diffuse?	Entropy is higher and output kurtosis is lower for N'Ko.	N'Ko exhibits equal or stronger specialization without task-specific finetuning.
RQ3	Is the deficit a one-model artifact?	The deficit appears across unrelated open model families.	A broad model sample with adequate tokenizer controls shows no recurring N'Ko deficit.
RQ4	Is right-to-left direction sufficient to explain failure?	Arabic acts as a control showing that direction is not enough.	Right-to-left scripts with strong data exposure fail in the same way while low-exposure left-to-right scripts do not.
RQ5	Can hidden circuits be amplified?	Duplication or amplification does not recover useful N'Ko behavior when the circuit is absent.	Targeted layer duplication reliably improves N'Ko without additional N'Ko training data.

3 Background: Script Invisibility

3.1 Definition

Let M be a language model, s a script, and $h_l(x_s)$ the hidden state at layer l for an input x written in script s. A script is *visible* to M when its inputs induce stable, specialized, task-relevant hidden-state trajectories comparable to those induced by scripts that the model demonstrably supports. A script is *invisible* when the model accepts its characters but processes them through weak, diffuse, or fallback geometry.

This definition separates three questions that are often collapsed:

$$\text{Unicode acceptance} \neq \text{tokenizer support} \neq \text{functional internal representation.}$$

Unicode acceptance only means the input string can enter the program. Tokenizer support means the script receives meaningful units rather than only byte, character, or fallback fragments. Functional representation means the network has learned circuits that transform those units into useful linguistic and semantic states.

Table 2: Evidence ladder for script support. Higher levels require the lower levels but are not implied by them.

Level	Evidence	Insufficient substitute
Rendering	Codepoints survive input, display, logging, and bidirectional layout.	Screenshots or fonts alone.
Tokenization	Stable script-aware units, low fallback fragmentation, reasonable token-per-word burden.	Unicode support in the tokenizer library.
Representation	Norms, entropy, sparsity, and kurtosis resemble supported scripts under controls.	A fluent answer in English about the script.
Task competence	Translation, correction, retrieval, or generation works on held-out N'Ko material.	One prompt demo or self-reported model confidence.

3.2 Why N'Ko is a strong test case

N'Ko is useful for this diagnostic because the usual excuses are weak. The script is not visually ambiguous in the way OCR noise might be. It is not newly invented for an artificial benchmark. It is not merely an idiosyncratic transliteration. It is a standardized script for a major West African language cluster. Its right-to-left direction does create rendering and bidirectional-text engineering requirements, but modern LLMs already handle other right-to-left scripts when those scripts are present in data. Arabic is the relevant comparison: direction alone does not explain failure.

The key contrast is data and tokenizer allocation. Arabic receives substantial representation in public web corpora, Wikipedia, religious texts, news, education content, and multilingual benchmarks. It also receives thousands of tokenizer entries in major model families. N'Ko receives little of either. If the same network family handles Arabic but fails on N'Ko, the likely mechanism is exposure, not direction.

4 Methods

4.1 Activation profiling

The project uses layerwise activation profiling. For each model and input pair, the hidden state tensor at layer l is reduced to several scalar diagnostics. Let $H_l \in \mathbb{R}^{T \times d}$ be the hidden-state matrix for a sequence of length T and hidden dimension d.

The average L2 norm measures activation energy:

$$A_l = \frac{1}{T} \sum_{t=1}^{T} \|H_{l,t,:}\|_2.$$

The representation tax for N'Ko relative to a comparison script c is:

$$\tau(M, N'Ko, c) = \frac{\frac{1}{L} \sum_{l=1}^{L} A_l(x_c)}{\frac{1}{L} \sum_{l=1}^{L} A_l(x_{N'Ko})}.$$

Values above 1 mean the comparison script receives more activation energy. This is not a runtime cost; it is a hidden-state energy ratio.

Entropy measures whether activation mass is concentrated or diffuse. For a flattened or pooled layer vector v_l, define

$$p_i = \frac{|v_{l,i}|}{\sum_j |v_{l,j}| + \epsilon}, \qquad H(v_l) = -\sum_i p_i \log_2 p_i.$$

Higher entropy in this setting indicates less specialization: the model spreads activation across many dimensions rather than routing to a smaller set of learned features.

Sparsity measures the share of dimensions that are effectively inactive:

$$S(v_l) = \frac{|\{i : |v_{l,i}| < \eta\}|}{d}.$$

Kurtosis measures peakedness:

$$K(v_l) = \frac{\mathbb{E}[(v_l - \mu)^4]}{\sigma^4}.$$

High output-layer kurtosis is interpreted as evidence of specialized circuits; low kurtosis is evidence that the model has no strong output-side direction for the input.

4.2 Circuit duplication as a diagnostic

The project also adapts layer-duplication ideas from reasoning-yield work [3]. The logic is diagnostic rather than purely performance-seeking. If a model has functional circuits for a script, then duplicating or amplifying layers in the relevant reasoning band may improve or at least change behavior in structured ways. If the script has no useful circuits, duplication will mostly amplify noise. In the Qwen3-8B N'Ko protocol, 45 configurations were inspected, and no configuration was N'Ko-advantageous. The best N'Ko score was close to random-chance behavior in that setup.

4.3 Comparison scope

The project contains two closely related activation protocols. The first focuses on Qwen3-8B and reports a 2.94x average tax with additional entropy, sparsity, and kurtosis diagnostics. The second repeats the brain scan across Qwen3-8B, Qwen2.5-7B, and Mistral-7B and reports taxes of 3.30x, 3.59x, and 2.67x. These numbers should not be pooled into a single meta-analysis because the exact protocol and model set differ. Their value is convergent: both protocols show that N'Ko receives weak internal support.

4.4 Tokenizer burden

Tokenizer burden is the first measurable site where script invisibility can enter. Let $w(x)$ be the number of whitespace-delimited words or reference lexical units in a string and $t_M(x)$ be the number of tokenizer units produced by model M. The script burden for script s can be summarized as

$$B_M(s) = \mathbb{E}_{x_s}\left[\frac{t_M(x_s)}{w(x_s) + \epsilon}\right].$$

High burden does not prove failure by itself, because character-level tokenization can still be usable when a model is trained for it. The burden matters when it co-occurs with weak early-layer norms and elevated sparsity. In that setting, the model is not merely using smaller units; it is receiving the script through a poor lexical interface. This is why tokenizer reports should be published alongside activation diagnostics in future N'Ko model studies.

Table 3: Minimum artifact contract for a publishable script-invisibility scan.

Artifact	Purpose
Prompt manifest	Shows semantic pairing, script condition, length controls, and normalization.
Tokenizer report	Counts fallback units, token-per-word burden, and script-specific vocabulary allocation.
Layer metric table	Stores per-layer norms, entropy, sparsity, kurtosis, and pooled tax values.
Model manifest	Records model revision, tokenizer revision, precision, extraction library, and hardware.
Analysis script	Recomputes all tables and figures from raw rows without manual copying.
Control script set	Includes Arabic or another high-exposure right-to-left control and at least one low-exposure non-N'Ko script where possible.

4.5 Artifact and replication protocol

A replication-quality brain scan should preserve the model identifier, tokenizer identifier, prompt set, script pairs, normalization policy, sequence lengths, layer-extraction code, random seeds where applicable, and the raw per-layer metric rows. The publication claim should not depend on plotted averages alone. The raw rows make it possible to recompute taxes, inspect layer bands, remove length outliers, and test whether the same conclusion holds under alternative pooling rules.

5 Results

5.1 Qwen3 diagnostic profile

The single-model scan found four aligned signals. First, activation energy was lower for N'Ko than for comparison text, producing an average tax of about 2.94x. Second, the entropy gap was approximately 1.2–1.7 bits, indicating that N'Ko representations were more diffuse. Third, embedding-layer sparsity was about 2.2x higher for N'Ko, consistent with early failure at the token-to-state interface. Fourth, output-layer kurtosis was 78.1% lower, consistent with missing specialized prediction circuits.

5.2 Cross-model profile

The cross-model scan asks whether the pattern is one-model-specific. It is not. The same broad geometry appears across Qwen-family and Mistral-family models.

The important feature of Table 4 is that model quality and recency do not erase the deficit. A newer or larger general model does not automatically discover a script that is absent from its training distribution. Table 5 adds a more mechanistic interpretation: even where activation energy exists, the output layers do not look specialized for N'Ko.

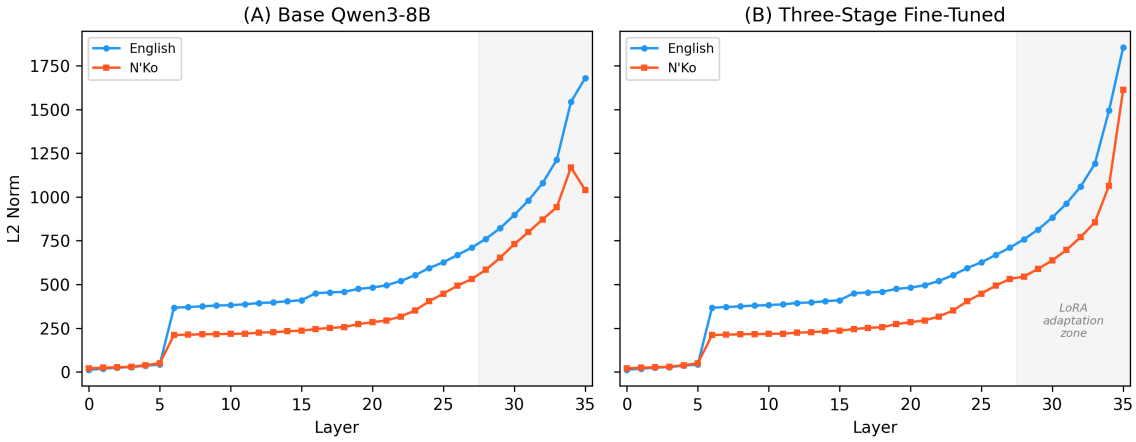

Figure 1: Layerwise activation-norm comparison from the brain-scan experiments. The main visual result is not an isolated outlier layer; the deficit extends through the depth of the model.

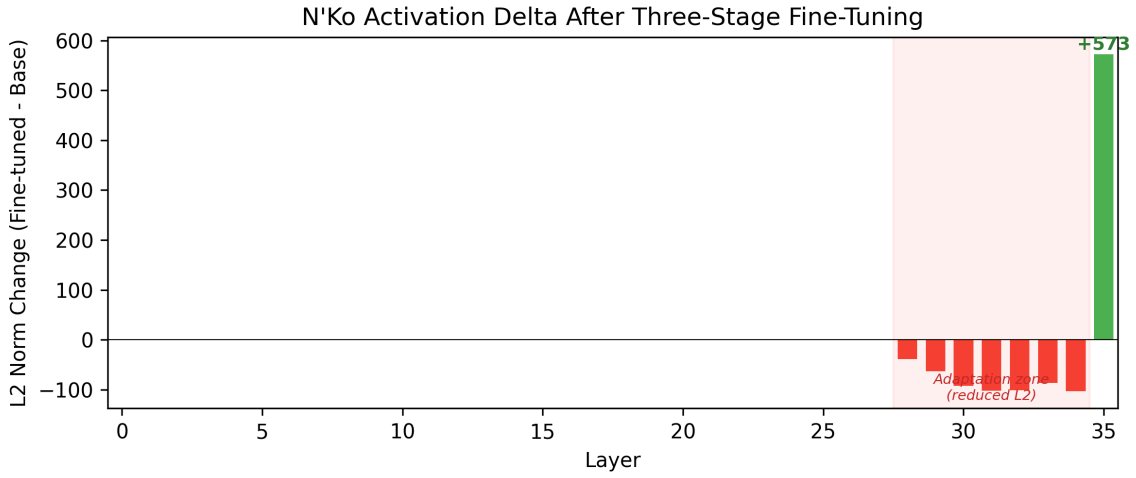

Figure 2: Activation delta across layers. Delta plots make the representation gap visible as a trajectory rather than as a single averaged score.

5.3 Three-zone failure model

The observed failure can be organized into three zones. The first is a tokenizer and embedding zone. If N'Ko is decomposed into isolated codepoints or fallback units, the model begins with weak lexical structure. The second is a middle-layer geometry zone. Information propagates, but it is diffuse: norms are lower, entropy is higher, and activations lack the compressed directions seen for better-supported scripts. The third is an output specialization zone. The model may still generate text, but the low-kurtosis output state suggests that generation is not grounded in stable N'Ko circuits.

6 Arabic as the Control

Right-to-left direction is a tempting explanation for N'Ko failure, but it is the wrong one. Arabic is also right-to-left, and modern multilingual models often have usable Arabic representations. The

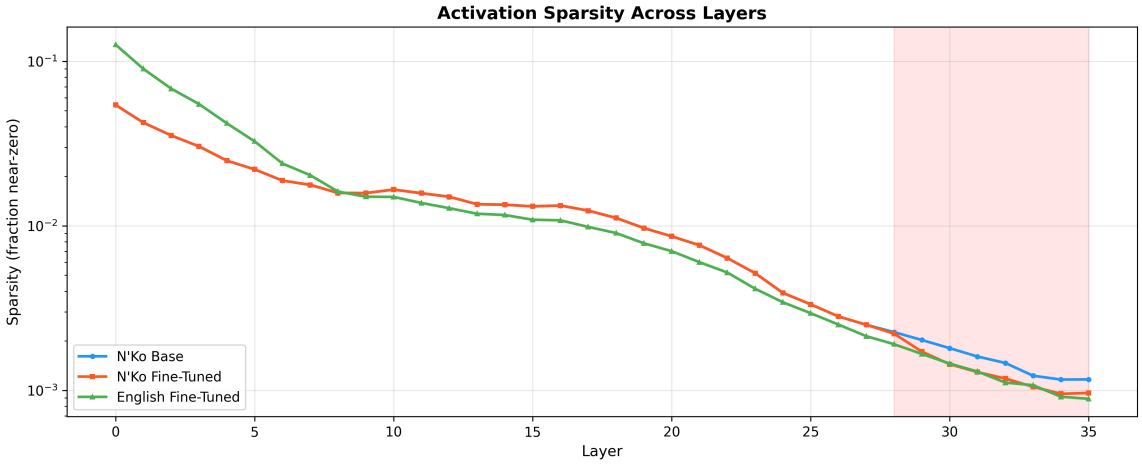

Figure 3: Sparsity comparison. Elevated sparsity in early processing is consistent with a tokenizer and embedding bottleneck for N'Ko inputs.

Table 4: Cross-model representation tax. Values come from the cross-model protocol and should be read as protocol-internal ratios, not as directly pooled with the separate 2.94x Qwen3 protocol.

Model	Comparison avg L2	N'Ko avg L2	Tax
Qwen3-8B	2,908.6	880.2	3.30x
Qwen2.5-7B	3,642.2	1,014.9	3.59x
Mistral-7B	47.9	17.9	2.67x

difference is exposure. Arabic has substantial text in pretraining corpora, dedicated benchmarks, large web presence, and thousands of tokenizer entries. N'Ko has dramatically less data and far less tokenizer support.

This comparison matters because it changes the intervention. If direction were the problem, the solution would be better rendering, bidirectional-text handling, or positional encoding. Those may still matter for software quality, but they do not explain the hidden-state collapse. If representation exposure is the problem, the solution is script-aware data, tokenizer allocation, continued pretraining, adapters, and downstream systems that do not rely on generic LLM script competence.

7 Adaptation and Remediation Agenda

The diagnosis implies a specific adaptation agenda. The first intervention is not prompt engineering; it is script data. A model cannot be expected to develop reusable N'Ko circuits from isolated examples if its pretraining mixture contains little N'Ko text. Continued pretraining on normalized N'Ko corpora is therefore the cleanest test of whether the hidden-state gap is data-limited. The second intervention is tokenizer allocation. A tokenizer that forces N'Ko through fallback fragments can preserve Unicode while still destroying lexical structure. The third intervention is parameter-efficient adaptation, such as adapters or LoRA [4, 2], but such adaptation should be evaluated with the same representation metrics rather than only with downstream prompt demos.

Table 5: Output-layer specialization. A high kurtosis deficit means the output layer is much less specialized for N'Ko than for comparison text.

Model	Comparison output kurtosis	N'Ko output kurtosis	Deficit
Qwen3-8B	601.5	131.9	78.1%
Qwen2.5-7B	644.7	41.7	93.5%
Mistral-7B	168.0	59.5	64.6%

Table 6: Failure zones and their downstream consequences.

Zone	Diagnostic signature	Consequence
Tokenizer	fallback tokenization, elevated sparsity, low early-layer energy	The script enters the model without lexical or subword structure.
Middle geometry	high entropy, low norms, weak trajectories	The model carries a string forward without forming reliable linguistic state.
Output circuits	low kurtosis and poor circuit-duplication yield	Generation or translation can appear fluent in another language while failing the script.

8 Implications for N'Ko ASR

The speech-recognition papers in this series do not begin from a blank slate. They inherit the diagnostic result that general LLMs are weak N'Ko processors. This has three consequences.

First, ASR should not decode through Latin and expect a generic model to restore N'Ko distinctions after the fact. A Latin bridge can discard tone, merge digraph structure, and introduce orthographic convention into what should be an acoustic measurement.

Second, language-model correction for N'Ko ASR must be treated with suspicion unless the model has script-native competence. A correction layer can improve surface fluency while corrupting acoustic evidence. This is why the fourth paper in the series introduces AGP as a conservative row-level governance layer.

Third, evaluation must separate model output from script support. A poor N'Ko LLM translation result does not prove N'Ko is a poor computational script; it proves that the model did not learn the script. Conversely, a useful N'Ko ASR artifact does not mean all LLMs understand N'Ko. Representation, ASR decoding, and correction are distinct layers.

9 Reviewer Checklist

For this line of work to be reviewable, a reader should be able to answer five questions from the paper and artifacts alone. First, what exact model and tokenizer were scanned? Second, what script-normalization policy was used before tokenization? Third, were comparison prompts matched in meaning and approximate length? Fourth, does the result survive more than one internal diagnostic rather than depending on a single plotted norm? Fifth, are downstream claims separated from representation claims? If any answer is missing, the correct response is not to reject the entire script-invisibility concept, but to lower the claim strength until the artifact chain is complete.

Table 7: Remediation hypotheses implied by script invisibility.

Intervention	Predicted internal change	Required evaluation
N'Ko continued pretraining	Higher middle-layer norms, lower entropy, stronger output kurtosis.	Held-out language modeling plus layerwise brain scan.
Tokenizer reallocation	Lower token burden and lower early-layer sparsity.	Tokenizer report before and after adaptation.
LoRA or adapters	Local recovery of task-relevant circuits without full pretraining.	Task metrics plus activation comparison to the base model.
Retrieval augmentation	Better factual answers without necessarily repairing representation.	Separate retrieval accuracy from hidden-state support.
ASR-specific correction model	Better post-ASR corrections only if trained on script-native rows.	Row-level correction benchmark with accepted-regression rate.

10 Threats to Validity

The primary construct-validity risk is that hidden-state metrics are proxies. Norms, entropy, sparsity, and kurtosis are not direct measures of linguistic competence. They become meaningful when their pattern is coherent and when controls rule out simpler explanations. The primary internal-validity risk is prompt mismatch: if N'Ko prompts differ semantically or in length from comparison prompts, the tax can be inflated. The primary external-validity risk is model selection: an untested model with substantial N'Ko data could behave differently. The primary conclusion-validity risk is overgeneralization: evidence of weak general LLM support does not imply that N'Ko is computationally weak, and it does not imply that a script-native ASR system cannot perform well.

11 Limitations

This paper is a diagnostic synthesis, not a universal survey of every model family. The tested models are informative but finite. A future study should include more open and closed models, explicitly count tokenizer entries under a standardized definition, and test N'Ko alongside Adlam, Vai, Tifinagh, Osmanya, Ethiopic, and other scripts whose Unicode presence exceeds their model presence.

The activation metrics are also indirect. Norms, entropy, sparsity, and kurtosis do not replace task performance. Their value is mechanistic: they show why task performance is likely to fail and where adaptation should intervene. A full evaluation should connect these metrics to controlled translation, ASR correction, retrieval, and language-modeling tasks.

Finally, the paper does not claim that N'Ko invisibility is permanent. The LoRA and adaptation results in the project notes suggest that script representations can be improved with targeted training [2, 4]. The claim is that current general-purpose models should not be credited with N'Ko competence unless they demonstrate it.

12 Conclusion

N'Ko reveals a gap between Unicode inclusion and computational inclusion. The script exists in standards, in community practice, and in digital text, but that is not enough for LLMs to represent it. The activation evidence shows a consistent failure geometry: lower energy, higher entropy, more sparsity, weaker kurtosis, and dead or absent circuits.

The practical conclusion is straightforward. Any serious N'Ko language technology stack must measure script support directly. It cannot assume that a multilingual LLM will repair the script after ASR, translation, or retrieval. The following papers therefore build on this result: they argue for phonemically meaningful N'Ko evaluation, preserve the archived 20.57% CER ASR anchor with claim boundaries, and define a conservative deployment layer for correction and corpus governance.

References

[1] Coleman Donaldson. *Clear Language: Script, Register and the N'ko Movement of Manding-Speaking West Africa*. PhD thesis, University of Pennsylvania, 2017.

[2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *ICLR*, 2022.

[3] David Noel Ng. Reasoning yield from stacking, 2024. Preprint.

[4] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In *EMNLP: System Demonstrations*, 2020.

[5] Unicode Consortium. N'ko block: U+07C0–U+07FF, 2006. The Unicode Standard, Version 5.0+.