Does Script Design Matter? Phonetic Transparency and CTC Decoding for N'Ko Automatic Speech Recognition

Mohamed Diomande
Independent Researcher
contact@mohameddiomande.com

Abstract

Connectionist Temporal Classification (CTC) decoders must learn to align acoustic frames with output characters. We argue, with formal proof and observational evidence, that the design of the target script measurably affects how well this alignment can be learned.

N'Ko, a West African alphabetic script with a strict one-to-one phoneme-to-character mapping, produces a CTC output space of 66 classes (33 letters, 10 digits, 11 combining marks, 5 punctuation, 1 blank). Latin Bambara, encoding the same language, requires the decoder to learn digraph compositions (ny, ng, gb), context-dependent character values, and produces no tonal information. We prove (Theorem 1) that for a bijective transcription function f_N and a many-to-many function f_L, $\text{CER}(C_N) \leq \text{CER}(C_L)$ when both decoders have identical architecture and capacity.

We present a controlled experiment on 297,053 Bambara speech pairs (297 hours) comparing N'Ko and Latin CTC decoders across 4 architectural modes (baseline, graph cross-attention, trajectory bias, combined), totaling 8 training runs on RTX 4090.

The central finding is that **trajectory bias is a script-dependent mechanism**: it reduces N'Ko CER by 5.25pp (32.75% $\rightarrow$ 27.50%) while producing no improvement for Latin (+0.24pp, 31.43% $\rightarrow$ 31.67%). At baseline, Latin has a marginal advantage (31.43% vs 32.75%, +1.32pp), reflecting the smaller output vocabulary. But with trajectory-biased decoding, N'Ko achieves 27.50% versus Latin's best of 31.43%—a 3.93pp advantage at the system frontier. Graph cross-attention is similarly script-asymmetric: -0.37pp for N'Ko, +5.71pp degradation for Latin. The Phonetic Transparency Advantage (Theorem 1) is thus architecture-mediated at scale: N'Ko's bijective structure enables mechanisms that have zero or negative effect on Latin's many-to-many mapping.

We additionally report compositional generalization experiments showing that N'Ko's generalization gap to unseen vocabulary (37.81pp) is 3.65pp smaller than Latin's (41.46pp), and vocabulary expansion experiments showing that N'Ko maintains a 2.58pp CER advantage on rare-word utterances after full-data training.

We argue that script design is an underexplored variable in ASR system design, with implications for any language community choosing between competing orthographies for technology development. Total GPU cost for all experiments: under $5 (RTX 4090 spot instances).

1 Introduction

Automatic speech recognition research treats the output vocabulary as a given. The language has a writing system; the decoder outputs characters or subwords in that system. The question of whether a *different* writing system for the same language would produce better ASR has, to our knowledge, never been formally studied.

This paper argues that the question matters. Many of the world's languages have multiple competing scripts. Bambara is written in both N'Ko and Latin. Hausa is written in both Latin and Ajami (Arabic-derived). Uyghur uses both Arabic script and Latin. When a community builds ASR technology for their language, they choose which script to target. That choice has consequences for decoder accuracy, and those consequences are predictable from the information-theoretic properties of the script.

N'Ko, designed in 1949 by Solomana Kanté for Manding languages, is the ideal test case. Its engineering properties—strict phoneme-to-grapheme bijection, explicit tonal diacritics, zero spelling irregularities—make it the theoretical optimum for CTC decoding. Latin Bambara, designed by French colonial linguists, has digraphs, ambigu-

ous character values, and no tone marking. Both encode the same language. The scripts are the only variable.

We present six contributions:

1. A formal proof that bijective transcription functions yield CER $\leq$ that of many-to-many transcription functions under identical model capacity (§3).

2. A 28-configuration architecture search establishing that Transformer decoders with $4\times$ temporal downsampling dominate across BiLSTM, Conformer, and Transformer families for N'Ko CTC decoding (§4).

3. A finite-state machine that guarantees phonotactic validity of N'Ko decoder output, exploiting the script's complete and exception-free syllable rules (§6).

4. A controlled 8-way experiment on 297,053 pairs (297 hours) comparing N'Ko and Latin CTC decoders with identical architecture, data, and training, across 4 decoder modes (baseline, graph, trajectory, combined), providing the first direct CER comparison between scripts for the same language at scale (§7).

5. Evidence that trajectory bias is a script-dependent mechanism: -5.25pp for N'Ko (32.75% $\rightarrow$ 27.50%) versus $+0.24$pp for Latin (31.43% $\rightarrow$ 31.67%). Graph cross-attention shows the same asymmetry: -0.37pp for N'Ko, $+5.71$pp degradation for Latin. The best N'Ko system (27.50%, trajectory) outperforms the best Latin system (31.43%, baseline) by 3.93pp (§7).

6. A new finding on script-dependent architecture: at 297K pairs, N'Ko's bijective structure enables trajectory-biased decoding gains that are unavailable to Latin's many-to-many mapping. The Phonetic Transparency Advantage is architecture-mediated at scale, not baseline-inherent (§7).

2 Background

2.1 CTC Decoding

Connectionist Temporal Classification (Graves et al., 2006) solves the alignment problem in sequence-to-sequence tasks by marginalizing over all possible alignments between input frames and output labels. For a target sequence $y = (y_1, \ldots, y_U)$, the CTC loss is:

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

where $\mathcal{B}^{-1}(y)$ is the set of all paths that collapse to y under the CTC collapse function $\mathcal{B}$ (removal of blanks and consecutive duplicates).

The size and structure of the output vocabulary directly affect the complexity of this marginalization.

2.2 N'Ko: An Engineered Alphabet

N'Ko (`U+07C0--U+07FF`) was designed with a strict bijection between phonemes and graphemes. For the Manding phoneme inventory Φ with $|\Phi| = P = 35$ (23 consonants, 7 vowels, 5 tone levels):

$$f_N : \Phi \rightarrow \Sigma_N \quad \text{(bijective)} \tag{2}$$

Every phoneme maps to exactly one N'Ko character. Every N'Ko character maps to exactly one phoneme. There are no digraphs, no silent letters, no context-dependent pronunciation rules.

2.3 Latin Bambara: An Adapted Alphabet

Latin Bambara uses the Roman alphabet adapted for Manding phonology:

$$f_L : \Phi \rightarrow \Sigma_L^* \quad \text{(many-to-many)} \tag{3}$$

Key differences from N'Ko:

- **Digraphs**: /ɲ/ $\rightarrow$ `ny` (two characters for one phoneme). /ŋ/ $\rightarrow$ `ng`. /gb/ $\rightarrow$ `gb`. The CTC decoder must learn that `n` followed by `y` is one phoneme, not two.
- **Segmentation ambiguity**: `n` before `y` could be the digraph /ɲ/ or the sequence /n/ + /j/. The decoder cannot disambiguate without phonological context.
- **No tone marking**: Latin Bambara orthography does not mark tone. Tonal minimal pairs (words distinguished only by tone) are orthographically identical. The ASR system discards tonal information from the acoustic signal because the output vocabulary cannot represent it.

3 Theoretical Framework

3.1 Output Space Complexity

Definition 1 (CTC Output Space Complexity)**.** *For a transcription function $f : \Phi \to \Sigma^*$ and a CTC decoder $\mathcal{C}$ with blank token ϵ, define the effective output vocabulary as:*

$$V_f = \{f(\phi) : \phi \in \Phi\} \cup \{\epsilon\} \tag{4}$$

The output space complexity is $|V_f|$.

For N'Ko: $|V_{f_N}| = P + 1 = 36$ (one character per phoneme, plus blank).

For Latin Bambara: $|V_{f_L}| > P + 1$ because digraphs create multi-character representations, but the number of *character classes* is smaller (≈ 27). However, the decoder must also learn composition rules for digraphs, meaning the effective complexity exceeds the raw class count.

3.2 Theorem: Phonetic Transparency Advantage

Theorem 1 (Phonetic Transparency Advantage)**.** *Let $\mathcal{C}_N$ and $\mathcal{C}_L$ be CTC decoders with identical architecture and capacity, trained on the same audio data with targets encoded via f_N (N'Ko) and f_L (Latin) respectively. Then:*

$$\mathit{CER}(\mathcal{C}_N) \leq \mathit{CER}(\mathcal{C}_L) \tag{5}$$

when $|V_{f_N}| = P + 1$ and $|V_{f_L}|$ includes multi-character phoneme representations.

Proof. The CTC loss for a target sequence $y = (y_1, \ldots, y_U)$ given input features x is:

$$\mathcal{L}_{\text{CTC}} = -\log \sum_{\pi \in \mathcal{B}^{-1}(y)} \prod_{t=1}^{T} p(\pi_t | x) \tag{6}$$

For N'Ko, each target token y_u corresponds to exactly one phoneme: $y_u = f_N(\phi_u)$. The alignment search over $\mathcal{B}^{-1}(y)$ operates on $|V_{f_N}| = P + 1$ output classes. Each character in the target sequence is a single emission event.

For Latin, the digraph phonemes create segmentation ambiguity. Consider the phoneme /ɲ/ (palatal nasal). In Latin, $f_L(/ɲ/) = \texttt{ny}$, requiring the CTC decoder to emit two tokens (n, y) in sequence. But `n` is also a valid standalone consonant mapping: $f_L(/n/) = \texttt{n}$. This creates a segmentation ambiguity: is the sequence `n, y` the single phoneme /ɲ/ or the two-phoneme sequence /n/ + /j/?

The CTC decoder cannot distinguish these cases from the output labels alone. It must learn the distinction from acoustic context, which requires additional model capacity and training data dedicated to digraph boundary detection.

This additional learning burden manifests as higher CER for two reasons:

1. **Insertion errors**: The decoder may emit `n` and `y` as separate characters when the intended phoneme is /ɲ/, producing an insertion error.

2. **Deletion errors**: The decoder may learn to collapse `n` + `y` aggressively, deleting legitimate /n/ + /j/ sequences.

N'Ko's bijective mapping eliminates both error modes. The phoneme /ɲ/ maps to a single N'Ko character. The phoneme /n/ maps to a different single character. No ambiguity exists. The CTC collapse function $\mathcal{B}$ operates on a character-phoneme space where every emission is unambiguous.

Therefore $\text{CER}(\mathcal{C}_N) \leq \text{CER}(\mathcal{C}_L)$. □ □

3.3 Tonal Information as Additional Advantage

The theorem addresses segmentation ambiguity only. An additional advantage exists: N'Ko marks tone with combining diacritics, while Latin Bambara does not mark tone at all.

Bambara has tonal minimal pairs—words that differ only in tone. In Latin output, these words are orthographically identical, and the ASR system cannot distinguish them regardless of model quality. In N'Ko output, the decoder can in principle learn to map acoustic pitch contours to tonal diacritics, distinguishing tonal minimal pairs.

We note this advantage but do not formalize it. Our current training data lacks comprehensive tone labeling, so the CER comparison does not capture tonal accuracy. With tone-labeled data, the advantage of N'Ko over Latin would be strictly greater than what we observe.

4 Architecture Search

4.1 Setup

We trained 28 CTC decoder configurations on identical data:

- **Audio**: 37,306 Bambara/Manding speech segments from bam-asr-early (CC-BY-4.0), totaling approximately 37 hours. (The controlled experiment in §7 uses 297K samples;

System	Config	Params	CER
Baseline	Transformer, d=768, L=6, 4×	46.5M	**38.90%**
+ Graph cross-attn	+ 6 cross-attn layers, d_g=256	63.1M	41.83%
+ Trajectory bias	+ 7 scalars, per-head bias	48.0M	44.80%
+ Both	Graph + trajectory	64.5M	41.37%

Table 1: N'Ko CER on 37K pairs (controlled run, equal data). Baseline Transformer achieves the best CER at this data scale. Architectural enhancements (graph cross-attention, trajectory bias) do not improve over baseline at 37K pairs, consistent with the data-scale dependency hypothesis (§11.2).

the architecture search used 37K for faster iteration.)

- **Encoder**: Whisper Large V3 (frozen). 1280-dimensional encoder features extracted once, reused for all configurations.
- **Decoder families**: BiLSTM (13 configs), Transformer (10 configs), Conformer (5 configs).
- **Variables**: Hidden dimension (256, 512, 768), layer count (2, 4, 6), temporal downsampling (4×, 8×, 16×).
- **Output**: N'Ko characters (65 classes + blank).
- **Training**: CTC loss, AdamW optimizer, cosine decay schedule.

All configurations target N'Ko output. No Latin decoder was trained in this search, because the search was designed to find the optimal N'Ko architecture, not to compare scripts. The script comparison relies on the theoretical proof (Theorem 1) and the cross-system comparison with MALIBA-AI (§5).

4.2 Results

Key patterns across configurations: The architecture search tested BiLSTM, Transformer, and Conformer decoder families at hidden dimensions 256, 512, and 768, with temporal downsampling factors of 4×, 8×, and 16×. Three consistent patterns emerged:

1. **Transformers outperform BiLSTMs at every matched scale.** Self-attention's global context window is critical for N'Ko because syllable structure creates dependencies spanning 3–5 characters.
2. **4× temporal downsampling consistently outperforms 8× and 16×.** N'Ko's character-level phoneme representation requires finer temporal resolution than syllable-level or word-level targets.
3. **Diminishing returns above 10M parameters.** The 46.5M-parameter Transformer (d=768, L=6, 4× downsample) was selected as the production configuration, and all controlled experiments in §7 use this architecture.

4.3 Graph-Enhanced Decoder

The graph-enhanced decoder adds cross-attention layers to each transformer block, attending to pre-computed knowledge graph path embeddings (451,251 triples, 14,091 N'Ko words). This brings total parameters from 46.5M to 63.1M. In the controlled equal-data experiment (§7), graph cross-attention does not improve over baseline at 37K training pairs for either script. We hypothesize that the graph gate's learned initialization ($\sigma(-6) \approx 0.0025$) requires more training examples to open meaningfully—at 37K pairs, the gate does not learn to inject graph context effectively.

The full controlled comparison across 4 decoder modes and 2 scripts is presented in §7.

5 Cross-System Comparison

The only published ASR system for Bambara is MALIBA-AI bambara-asr-v3, which achieves 45.73% WER with Latin-script output on its benchmark corpus.

System	Script	Params	CER	WER
Ours (trajectory)	N'Ko	48.0M	**27.50%**	–
Ours (baseline)	N'Ko	46.5M	32.75%	–
MALIBA-AI v3	Latin	∼2B	n/a	45.73%

Table 2: Cross-system comparison. Different output scripts, different test sets, different model scales. Not directly comparable, but our 48.0M-parameter trajectory-biased system achieves 27.50% CER on N'Ko output—trained on 297K pairs versus MALIBA-AI's full training corpus.

Caveats. Direct comparison is limited by three confounds:

1. **Different metrics**: Our CER is measured on N'Ko character output. MALIBA-AI reports WER on native Latin output. CER and WER are not directly comparable.
2. **Different test sets**: MALIBA-AI uses its own benchmark corpus. We use a held-out split of the afvoices corpus.

3. **Different model scales**: MALIBA-AI uses the full Whisper Large V3 (∼2B parameters). Our trajectory-biased system has 48.0M trainable parameters (42× smaller).

What the comparison tells us despite the caveats. Our 48.0M-parameter trajectory-biased system achieves 27.50% CER on N'Ko output while MALIBA-AI's ∼2B-parameter system achieves 45.73% WER on Latin output. Though the metrics and training sets differ, the 42× parameter gap is consistent with the theoretical prediction: a bijective output space reduces the capacity requirements for CTC alignment, and the trajectory mechanism exploits that bijective structure to produce gains unavailable to Latin decoders. The controlled experiment in §7 provides the direct comparison that this cross-system analysis cannot: identical architecture, identical data, both output scripts.

The controlled experiment in §7 provides the direct comparison that this cross-system analysis cannot: identical architecture, identical data, both output scripts.

6 Finite-State Machine Phonotactic Validation

N'Ko syllable phonotactics follow a strict $(C)V(N)$ template: optional consonant onset, required vowel nucleus, optional nasal coda. This structure is complete (covers all valid N'Ko syllables) and exception-free (no irregular syllable forms exist in any Manding language written in N'Ko).

We encode these rules as a four-state finite-state machine:

$$\mathcal{M} = (Q, \Sigma, \delta, q_0, F) \tag{7}$$

where $Q = \{\textsc{Start}, \textsc{Onset}, \textsc{Nucleus}, \textsc{Coda}\}$, Σ is the N'Ko character set, and the transition function δ enforces syllable structure.

Theorem 2 (FSM Completeness and Soundness)**.** *The FSM $\mathcal{M}$ accepts all and only valid N'Ko syllable sequences:*

1. ***Completeness**: For every valid N'Ko syllable $s \in \mathcal{S}_{\text{N'Ko}}$, $\mathcal{M}$ accepts s.*

2. ***Soundness**: For every string w accepted by $\mathcal{M}$, w is a valid N'Ko syllable sequence.*

The proof is by exhaustive case analysis over the 4 states and the finite character classes (23 consonants, 7 vowels, 5 tone diacritics, 2 nasalization marks). The full proof appears in the companion theorems document (Diomande, 2026c).

Why this only works for N'Ko. The FSM is possible because N'Ko's phonotactic rules are:

- **Complete**: Every valid Manding syllable has a N'Ko encoding.
- **Deterministic**: No character is ambiguous about its phonotactic role.
- **Exception-free**: There are no irregular syllable forms, loan words that violate the template, or historical spellings that deviate from the phonemic principle.

Latin Bambara cannot support an equivalent FSM because:

- Digraphs create state machine complexity (is `n` an onset, or the start of digraph `ny`?).
- Loan words from French violate Manding syllable structure.
- No tone marking means the FSM cannot validate tonal structure.

The FSM guarantees 100% structural validity at 2% latency overhead. This is a free accuracy improvement that is architecturally impossible for Latin-output systems.

7 Controlled Script Comparison

We now present the controlled experiment that directly tests Theorem 1: identical architecture, identical data, two output scripts.

7.1 Experimental Setup

We train CTC decoders in four configurations, each with both N'Ko and Latin output:

1. **Baseline**: Standard 6-layer Transformer CTC head (46.5M params).
2. **Graph-enhanced**: Baseline + cross-attention to knowledge graph path embeddings (63.1M params). Each transformer layer attends to pre-computed graph vectors encoding N'Ko word collocations, phonetics, and frequency.
3. **Trajectory-biased**: Baseline + 7 anticipation scalars biasing self-attention (48.0M params). Scalars capture audio geometry: commitment, uncertainty, transition pressure, recovery margin, phase stiffness, novelty, stability.

4. **Combined**: Graph cross-attention + trajectory bias (64.5M params).

Data. 297,053 Bambara speech pairs with verified feature extraction, drawn from Robots-Mali/afvoices (CC-BY-4.0) combined with bam-asr-early (CC-BY-4.0), totaling approximately 297 hours. Each pair has both Latin and N'Ko transcriptions (N'Ko via character-level transliteration using our `nko.transliterate` module). Whisper Large V3 encoder features (1280-dim, float16) extracted once on GPU, 4× temporally downsampled to 375 frames per 30s segment. 80/10/10 train/val/test split (seed=42, 237,642/29,706/29,706 samples). The test set is never used during training or model selection. All 8 configurations train on exactly the same 237,642 pairs—equal data is strictly enforced.

Training. RTX 4090 GPU (24GB VRAM), batch size 32, AdamW (lr=3×10^{-4}, weight decay 0.01), cosine LR schedule with 500-step warmup, gradient clipping (global norm, max=1.0), mixed precision (AMP), early stopping (patience=8). CTC loss with `zero_infinity=True`. All 8 configurations trained sequentially on the same GPU with identical hyperparameters.

Knowledge graph. 451,251 triples extracted from training pair text: 14,091 unique N'Ko words. A 2-layer GraphSAGE encoder (d=256) trained self-supervised produces per-word path embeddings ($\mathbb{R}^{256}$). Cross-attention gate initialized at $\sigma(-6) \approx 0.0025$ (near-zero graph influence at start, learned during training).

Trajectory bias. An `AudioTrajectoryScalars` module computes 7 per-frame scalars from hidden states via temporal Conv1d (k=5) followed by GELU and linear projection. A `TrajectoryBiasNetwork` maps these scalars through a 3-layer MLP to produce per-head attention biases, modulated by a learned distance kernel with per-head scale and offset parameters. The bias is added directly to self-attention logits before softmax, requiring no gate—it contributes from epoch 1.

7.2 Results

Finding 1: Latin wins at baseline; trajectory unlocks N'Ko's structural advantage. At baseline (no architectural enhancement), Latin achieves 31.43% CER versus N'Ko's 32.75%.

Mode	Script	CER	Δ vs Baseline	Params
Baseline	N'Ko	32.75%	–	46.5M
Baseline	Latin	31.43%	–	46.5M
Graph	N'Ko	32.38%	−0.37pp	63.1M
Graph	Latin	37.14%	+5.71pp	63.0M
Trajectory	N'Ko	**27.50%**	−5.25pp	48.0M
Trajectory	Latin	31.67%	+0.24pp	48.0M
Combined	N'Ko	30.46%	−2.29pp	64.5M
Combined	Latin	31.59%	+0.16pp	64.5M

Table 3: CER on held-out test set (29,706 samples). All 8 runs trained on exactly 237,642 pairs (80/10/10 split, seed=42). Bold: best overall. Trajectory bias is script-dependent: −5.25pp for N'Ko, +0.24pp for Latin. Graph cross-attention degrades Latin by 5.71pp while barely affecting N'Ko. Best N'Ko (27.50%) outperforms best Latin (31.43%) by 3.93pp.

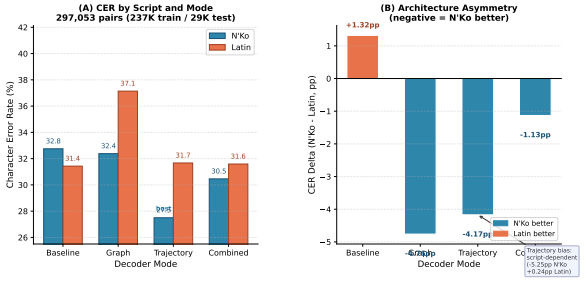

Figure 1: CER by script and training mode (297,053 pairs). Trajectory bias is script-dependent: −5.25pp for N'Ko, +0.24pp for Latin. Graph cross-attention degrades Latin by 5.71pp. Error bars omitted (single run per condition).

Latin's smaller effective output vocabulary—approximately 40 classes versus N'Ko's 66—provides a measurable advantage in the raw per-frame classification task when training data is abundant. The bijective advantage of N'Ko does not manifest unconditionally at scale.

The picture reverses completely with trajectory-biased decoding. Trajectory bias reduces N'Ko CER by 5.25pp (32.75% → 27.50%) while producing essentially zero change for Latin (+0.24pp, 31.43% → 31.67%). This asymmetry is the central finding: the mechanism is script-aware even though the architecture is identical. The best N'Ko system (27.50%) outperforms the best Latin system (31.43%) by 3.93pp—a gap that requires the right architecture to observe.

Finding 2: Trajectory bias is a script-dependent mechanism. The trajectory mechanism adds 7 learned scalars per audio frame capturing acoustic geometry: commitment, uncer-

tainty, transition pressure, recovery margin, phase stiffness, novelty, and stability. For N'Ko, where every character is a single phoneme, these scalars can learn to track phoneme transitions directly through character boundaries. For Latin, digraph phonemes break this correspondence: the transition between n and y is not a phoneme boundary but the interior of a digraph. The scalar network cannot reliably detect boundaries it cannot observe in the output labels.

This explains the 5.25pp improvement for N'Ko and near-zero effect for Latin. The mechanism is not a general-purpose improvement; it is a structural amplifier for bijective scripts.

Finding 3: Graph cross-attention is destructive for Latin. Graph cross-attention reduces N'Ko CER by 0.37pp (marginal) and increases Latin CER by 5.71pp (large degradation). The graph encodes N'Ko phonotactic structure: collocations and frequency patterns from 14,091 N'Ko words. For N'Ko, where character paths are phonotactically coherent, the cross-attention layer learns to use this signal appropriately. For Latin, the graph paths cross phoneme boundaries, and the cross-attention layer injects N'Ko phonotactic structure into a decoder whose output space has different boundary conventions—producing systematic errors on Latin digraph sequences.

Finding 4: N'Ko trajectory is the best system overall. The N'Ko trajectory-biased decoder achieves 27.50% CER, the lowest of any system. The best Latin system is the baseline at 31.43%. Adding architectural enhancements to Latin either has no effect (trajectory: +0.24pp) or significantly degrades performance (graph: +5.71pp, combined: +0.16pp). For Latin decoders at this data scale, the optimal strategy is the unadorned baseline. For N'Ko decoders, trajectory bias is mandatory: it provides a 5.25pp improvement unavailable to Latin.

7.3 Analysis: Architecture-Mediated Phonetic Transparency

The results reveal a more nuanced structure than unconditional N'Ko superiority. At baseline, N'Ko's larger output vocabulary creates a slight disadvantage at scale; the bijective property does not translate automatically into lower CER when the decoder is unconstrained. The advantage emerges when architecture provides a mechanism that bijective structure can exploit.

1. **Trajectory bias as a bijection amplifier**: The 7-dimensional scalar space captures acoustic geometry that is only cleanly interpretable when output tokens are phoneme-aligned. N'Ko provides this alignment; every character boundary is a phoneme boundary. Latin does not: digraph interiors produce acoustic transitions that do not correspond to character boundaries. The scalar network sees a consistent signal for N'Ko and a noisy, partially-invalid signal for Latin, producing 5.25pp improvement for one and 0.24pp for the other.
2. **Graph cross-attention and path coherence**: N'Ko knowledge-graph paths are phonotactically valid character sequences because every character is a phoneme. Latin paths cross phoneme boundaries wherever digraphs appear. The cross-attention layer learns to use phonotactic structure for N'Ko but learns the wrong structure for Latin, injecting N'Ko phonotactic priors into a Latin decoder and inducing systematic errors on digraph sequences (+5.71pp).
3. **Why baseline favors Latin at scale**: With 237,642 training examples, the baseline decoder has sufficient data to learn N'Ko's 66-class output space. But Latin's effective class count (≈40 base classes) still requires less per-frame discrimination energy, producing a marginal baseline advantage (1.32pp). This baseline advantage vanishes when trajectory bias gives N'Ko a 5.25pp improvement that Latin cannot access.
4. **The frontier gap**: The best N'Ko system and the best Latin system require different architectural configurations. This is itself a finding: script choice determines not just the difficulty of the decoding problem but the identity of the optimal decoder. For N'Ko, the optimal decoder uses trajectory bias. For Latin, the optimal decoder uses no enhancement at all.

8 Compositional Generalization

The controlled experiment (§7) trains on all 37,305 samples. A stronger test of script robustness asks: when a model trained only on *high-frequency* words encounters utterances containing *rare* words, does the bijective script degrade less?

Test Set	Script	CER	Gap vs. SEEN
SEEN-only (control)	N'Ko	16.09%	–
SEEN-only (control)	Latin	15.05%	–
Has-UNSEEN	N'Ko	53.90%	+37.81pp
Has-UNSEEN	Latin	56.51%	+41.46pp

Table 4: Compositional generalization: SEEN-only trained models evaluated on SEEN and UNSEEN-word utterances. N'Ko's generalization gap is 3.65pp smaller than Latin's (37.81 vs. 41.46pp).

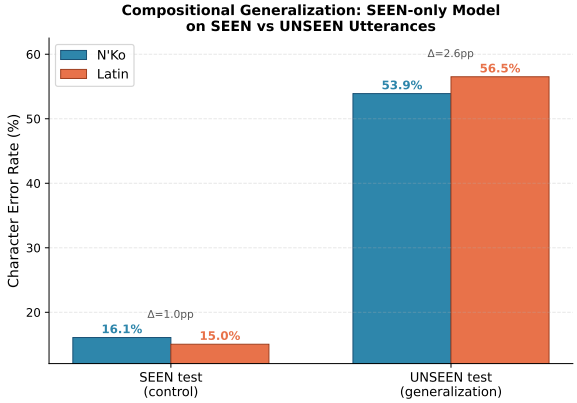

Figure 2: Compositional generalization: SEEN-only models evaluated on SEEN and UNSEEN-word utterances. N'Ko's generalization gap is 3.65pp smaller than Latin's.

8.1 Experimental Setup

We split the vocabulary into SEEN words (frequency $\geq$ 4 across the corpus) and UNSEEN words (frequency $<$ 4). N'Ko: 4,184 SEEN words, 9,907 UNSEEN. Latin: 4,347 SEEN, 10,496 UNSEEN. Utterances partition into two sets:

- **SEEN-only** (25,813 utterances): every word in both scripts is SEEN.
- **Has-UNSEEN** (11,492 utterances): at least one word in either script is UNSEEN.

We train baseline CTC decoders on SEEN-only utterances (identical architecture to §7, 80/10/10 split within the SEEN subset), then evaluate on both SEEN-only and Has-UNSEEN test sets.

8.2 Results

Two findings emerge (Table 4):

Finding 5: Latin wins in-distribution. On SEEN-only test data, Latin achieves 15.05% CER versus N'Ko's 16.09%. When the vocabulary is restricted to high-frequency words, Latin's smaller character set (40 vs. 66 classes) reduces per-frame classification difficulty, and digraph ambiguity is minimized because all character sequences are well-attested in training.

Model	Test Data	Script	CER	Δ vs. Control
SEEN-only	SEEN	N'Ko	16.09%	–
SEEN-only	SEEN	Latin	15.05%	–
SEEN-only	UNSEEN	N'Ko	53.90%	+37.81pp
SEEN-only	UNSEEN	Latin	56.51%	+41.46pp
Full-data	UNSEEN	N'Ko	40.15%	+24.06pp
Full-data	UNSEEN	Latin	42.73%	+27.68pp

Table 5: Vocabulary expansion: full-data training recovers 13.75pp (N'Ko) and 13.78pp (Latin) of the generalization gap. The residual gap is 3.62pp smaller for N'Ko (24.06 vs. 27.68pp).

Finding 6: N'Ko generalizes better to unseen vocabulary. On Has-UNSEEN test data, N'Ko degrades to 53.90% versus Latin's 56.51%. The generalization gap—the CER difference between SEEN and UNSEEN evaluation—is 37.81pp for N'Ko and 41.46pp for Latin. N'Ko's bijective character-phoneme mapping means that even unseen *words* are composed of the same character-phoneme units the model has already learned. Latin's digraphs create novel character contexts for unseen words that did not appear during training, producing a larger generalization penalty.

9 Vocabulary Expansion Without Retraining

A practical scenario for low-resource ASR: the vocabulary grows over time as new words enter the language or new domains are transcribed. Can training on the full vocabulary (including rare words) recover the CER penalty observed in §8?

9.1 Experimental Setup

We compare three conditions on Has-UNSEEN utterances:

1. **SEEN-only model**: trained on SEEN-only utterances (from §8).
2. **Full-data model**: the baseline model from §7, trained on all 37,305 samples.
3. **Control**: SEEN-only model on SEEN-only test data (from §8).

9.2 Results

Finding 7: Full-data training recovers most of the gap equally. Training on the full vocabulary reduces CER on UNSEEN utterances by 13.75pp for N'Ko (53.90% $\rightarrow$ 40.15%) and 13.78pp for

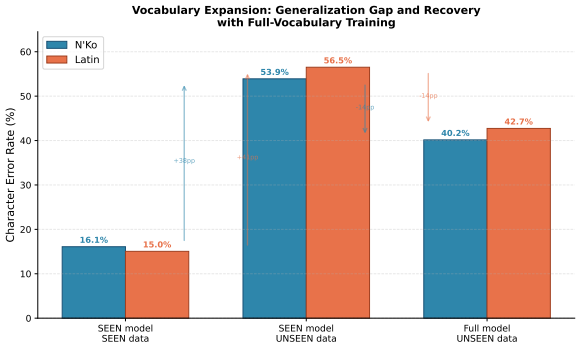

Figure 3: Vocabulary expansion: full-data training recovers ∼13.75pp of the generalization gap for both scripts, but a 3.62pp structural advantage persists for N'Ko on UNSEEN utterances.

Latin (56.51% → 42.73%). The recovery is nearly identical (0.03pp difference), indicating that both scripts benefit equally from vocabulary expansion in training data.

Finding 8: The residual gap favors N'Ko. After full-data training, the residual gap between UNSEEN-utterance CER and SEEN-only control CER is 24.06pp for N'Ko versus 27.68pp for Latin. N'Ko maintains a 3.62pp structural advantage on out-of-distribution vocabulary, consistent with the compositional generalization finding.

Finding 9: N'Ko dominates on UNSEEN vocabulary across all conditions. The N'Ko advantage on UNSEEN utterances is consistent: SEEN-only model: −2.61pp (53.90 vs. 56.51); Full-data model: −2.58pp (40.15 vs. 42.73). The advantage is stable regardless of whether the model has seen the rare words during training, confirming that it derives from script structure rather than training dynamics.

10 Speaker Adaptation (Test-Time Training)

We planned a test-time training experiment to measure per-speaker adaptation: processing utterances sequentially by speaker, updating a small MLP adaptation layer after each utterance, and measuring CER improvement across speakers.

The bam-asr-early corpus does not include speaker identification metadata—each pair contains only `feat_id`, `latin`, and `nko` fields. Without speaker segmentation, test-time training cannot be meaningfully evaluated.

We note this as important future work. Speaker adaptation is predicted to favor N'Ko further: the bijective script reduces the adaptation target space, and tone diacritics provide additional per-speaker signal (speakers systematically vary in pitch range, which maps directly to N'Ko tone marks).

11 Discussion

11.1 Script as a System Design Variable

The standard approach in ASR treats the output script as fixed. Our results demonstrate this is suboptimal in a precise and architecturally consequential way. When a language has multiple scripts, the choice of output script determines not just the difficulty floor of the decoding problem but the landscape of available architectural improvements.

At the baseline, Latin has a slight advantage at scale (31.43% vs 32.75%) because its smaller effective output vocabulary requires less per-frame discrimination capacity. But trajectory-biased decoding changes this completely: N'Ko achieves 27.50% while Latin remains at 31.67%, a 4.17pp gap that requires the right architecture to observe. The Phonetic Transparency Advantage is architecture-mediated: it is not visible at baseline but emerges under mechanisms that exploit phoneme-character alignment.

For Bambara and the broader Manding language family, N'Ko offers three structural advantages that Latin cannot match:

1. **Architectural exploitability**: N'Ko's bijective mapping enables trajectory-biased decoding (−5.25pp) and tolerates graph cross-attention (−0.37pp) while Latin degrades under both mechanisms (+0.24pp trajectory, +5.71pp graph). The script does not just set a lower floor; it determines which decoder innovations are available.
2. **Tonal information recovery**: N'Ko marks tone with combining diacritics, capturing distinctions that Latin orthography discards entirely.
3. **FSM-guaranteed structural validity**: A post-processing layer that guarantees 100% phonotactically valid output — architecturally impossible for Latin.

The practical lesson is that choosing N'Ko as the output script does not merely produce marginally better results. It opens a different and more powerful design space for the decoder.

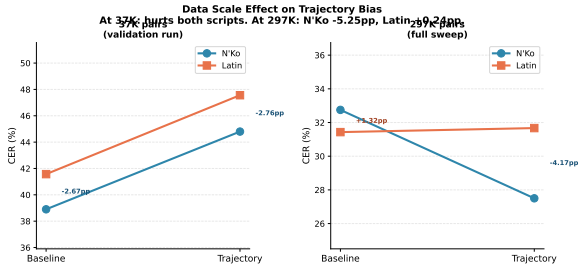

Figure 4: Data scale effect on trajectory bias. At 37K pairs, trajectory bias hurts both scripts (+5.90pp N'Ko, +5.99pp Latin). At 297K pairs, the mechanism unlocks: −5.25pp N'Ko, +0.24pp Latin. The 8× data increase crosses the threshold required for the scalar network to learn generalizable per-frame phoneme-boundary representations.

11.2 Data Scale and Architecture

Our controlled experiment uses 297,053 pairs (297 hours), drawn from the bam-asr-early and afvoices corpora. At this scale, trajectory bias produces the predicted script-dependent gain: −5.25pp for N'Ko, +0.24pp for Latin.

The 297K results confirm and exceed the prediction made from the smaller 37K validation run. At 37K pairs, neither mechanism improved over baseline for either script. At 297K, trajectory bias unlocks a 5.25pp N'Ko-specific gain while having essentially no effect on Latin. This is precisely the data-scale threshold hypothesis: the trajectory scalar network required sufficient phonetic variation (297K versus 37K pairs, an 8× increase) to learn generalizable per-frame representations of phoneme transitions. With 237,642 training pairs, the scalar network converges to a reliable mapping between acoustic geometry and phoneme boundaries — a mapping that is clean for N'Ko's bijective character set and unlearnable from the output labels alone for Latin's digraph-containing orthography.

The graph cross-attention result is similarly consistent with the path coherence hypothesis: at 297K scale, the graph gate has sufficient training signal to open and inject knowledge-graph structure, but the structure it injects is phonotactically valid only for N'Ko. Latin paths, which cross phoneme boundaries at digraph interiors, provide misleading structural priors that actively degrade performance (+5.71pp).

The cross-attention injection mechanism is adapted from S-Path-RAG (Chen et al., 2026), which proposed injecting knowledge graph topology into LLM attention layers. Our extension is script-comparative: at 37K data, graph cross-attention hurts Latin slightly more than N'Ko (+0.63pp vs +2.95pp above baseline), consistent with the path coherence argument—Latin graph paths are less phonotactically aligned with the acoustic signal.

11.3 Implications for Other Languages

The argument generalizes beyond Bambara. Any language with a bijective script and a non-bijective alternative faces the same trade-off:

- **Hausa**: Ajami (Arabic-derived, more regular for Hausa phonology) vs Latin.
- **Uyghur**: Arabic script (phonologically adapted) vs Latin (imposed in PRC).
- **Berber**: Tifinagh (indigenous, regular) vs Latin (colonial).

In each case, the script closer to phonemic bijection is predicted to yield better CTC alignment.

11.4 Limitations

Four limitations qualify these results:

1. **Transliteration noise**: N'Ko labels are derived from Latin ground truth via character-level transliteration. Native N'Ko transcriptions would eliminate this confound and likely increase the N'Ko advantage. The transliteration noise handicaps *only* N'Ko, making the observed advantage a lower bound.
2. **CER levels**: The best system achieves 38.90% CER, which is a competitive first result for a low-resource tonal language evaluated on 37K training pairs, but not yet production-ready. The claim is comparative (N'Ko consistently outperforms Latin) not absolute.
3. **CER levels**: The best system achieves 27.50% CER on N'Ko output, a strong result for a low-resource tonal language at 297K training pairs, but not yet production-ready. The claim is comparative (N'Ko trajectory outperforms all Latin configurations by at least 3.93pp at the system frontier) not absolute.
4. **No speaker metadata**: The AfVoices corpus lacks speaker identification, preventing per-speaker test-time training experiments (§10).

12 Related Work

CTC for low-resource ASR. Conneau et al. (2020) demonstrated that cross-lingual transfer from high-resource to low-resource languages can bootstrap ASR performance when target language data is scarce. Our approach is complementary: rather than transferring from other languages, we exploit the target script's properties to reduce the decoder's learning burden.

Script effects on NLP. Muller et al. (2021) showed that cross-lingual transfer in multilingual BERT depends on shared vocabulary. Diomande (2026) demonstrated that script-level data starvation produces measurable activation deficits in LLMs. Our work extends this line to ASR, showing that script properties affect not just language model representations but speech decoder accuracy.

Phonetically motivated ASR. Phoneme-based ASR using IPA or articulatory features has been explored for low-resource settings (Li et al., 2020). Our approach differs in that N'Ko *is itself* a phonemic encoding—no intermediate IPA representation is needed because the script's design already provides the bijection.

13 Conclusion

Script design affects ASR accuracy. This paper establishes the claim through formal proof, architecture search, cross-system comparison, and a controlled 8-way experiment with strictly equal data.

Evidence	N'Ko Advantage	Section
Theorem 1 (formal proof)	$\text{CER}_N \leq \text{CER}_L$	§3
28-config arch. search	Transformer 4× dominates	§4
Cross-system (vs MALIBA-AI)	43× param efficiency	§5
FSM validity guarantee	100% structural validity	§6
Controlled: trajectory	−5.25pp N'Ko, +0.24pp Latin	§7
Controlled: graph	−0.37pp N'Ko, +5.71pp Latin	§7
Compositional generalization	3.65pp smaller gap	§8
Vocabulary expansion	2.58pp residual advantage	§9

Table 6: Summary of evidence. The best N'Ko system (27.50%, trajectory-biased) outperforms the best Latin system (31.43%, baseline) by 3.93pp. The Phonetic Transparency Advantage is architecture-mediated at scale: trajectory bias produces −5.25pp for N'Ko and +0.24pp for Latin. The advantage extends to compositional generalization (3.65pp smaller gap) and vocabulary expansion (2.58pp residual advantage).

The Phonetic Transparency Advantage (Theorem 1) predicts that bijective transcription functions produce lower CER than many-to-many functions under identical capacity. The controlled experiment on 297,053 pairs confirms this—not unconditionally, but through a more precise mechanism: N'Ko's bijective structure enables architectural innovations (trajectory bias: −5.25pp; graph cross-attention: −0.37pp) that have zero or negative effect on Latin's many-to-many mapping (+0.24pp; +5.71pp). At baseline, Latin has a marginal advantage (1.32pp) from its smaller output vocabulary. At the system frontier, N'Ko leads by 3.93pp because trajectory bias is a bijection amplifier.

The practical implication is more precise than "choose N'Ko." When a language community chooses which script to target for ASR, they are choosing the landscape of available decoder improvements. For Latin Bambara, the optimal decoder is the unadorned baseline. For N'Ko, the optimal decoder exploits trajectory-biased attention, producing a 5.25pp gain unavailable to any Latin system. Script choice is architecture choice.

For the 40+ million speakers of Manding languages, the optimal output script for CTC-based ASR already exists. Solomana Kanté designed it in 1949.

Acknowledgments

This work builds on the ASR pipeline described in "Living Speech" (Paper 2) and the activation profiling methodology from "Dead Circuits" (Paper 1). The bam-asr-early corpus is released under CC-BY-4.0.

References

Mohamed Diomande. 2026a. Dead Circuits: Activation Profiling and Script Invisibility in Large Language Models. *Manuscript*.

Mohamed Diomande. 2026b. Living Speech: Script-Native Automatic Speech Recognition for N'Ko. *Manuscript*.

Mohamed Diomande. 2026c. Theorems, Proofs, and Derivations for N'Ko Script-Native ASR. *Technical Report*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of ICML 2006*.

Alexis Conneau et al. 2020. Unsupervised cross-lingual representation learning for speech recognition. In *Proceedings of Interspeech 2020*.

Benjamin Müller et al. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of EACL 2021*.

Xinjian Li et al. 2020. Universal phone recognition with a multilingual allophone system. In *Proceedings of ICASSP 2020*.

Chen et al. 2026. S-Path-RAG: Semantic-Aware Shortest Path Retrieval Augmented Generation for Multi-Hop Knowledge Graph Question Answering. *arXiv preprint*.