G2P on the Edge: Bridging Morphological Accuracy and Hardware Constraints via Quantized Hard-Monotonic Attention

Anonymous ACL submission

Abstract

State-of-the-Art (SOTA) Grapheme-to-Phoneme (G2P) models rely on over-parameterized Transformers, rendering them computationally prohibitive for microcontrollers with extreme memory constraints ($<$ 32KB). We introduce the **Neuro-Symbolic Quantized Hard-Monotonic Transducer (NS-QHMT)**, a linear-complexity architecture designed for the extreme edge. By replacing soft attention with a latent hard-monotonic pointer and employing integer-only quantization, NS-QHMT reduces the memory footprint to just **16.0 KB**—an $16\times$ reduction compared to standard Transformers. While enforcing strict hardware constraints, our model achieves a Phoneme Error Rate (PER) of **29.6%** on English benchmarks. Crucially, empirical profiling on a simulated Cortex-M7 confirms that NS-QHMT is the *only* neural architecture among baselines capable of executing within a 32KB RAM envelope without Out-Of-Memory (OOM) failure, bridging the gap between neural performance and legacy hardware constraints.

1 Introduction

Computational phonology has transitioned from discrete symbolic frameworks, such as Finite State Transducers (FSTs) (Kaplan and Kay, 1994), to continuous neural architectures. Although FSTs offer rigorous interpretability and deterministic linear-time inference $\mathcal{O}(T)$, they remain brittle when modeling irregular orthographies. Conversely, modern Transformer architectures (Vaswani et al., 2017) have established a superior performance ceiling by modeling complex, non-linear dependencies $P(\mathbf{y}|\mathbf{x})$ through self-attention. However, these gains incur a prohibitive computational cost, resulting in a *deployment gap* between research capability and edge-native utility. Strict physical constraints—specifically inference latency $\mathcal{L} < 10$ ms and memory footprint $\mathcal{M} < 5$ MB—are fundamentally incompatible with the quadratic complexity $\mathcal{O}(T^2)$ and massive parameterization inherent in standard Transformers. Consequently, there is an urgent imperative to architecturally bridge the gap between the morphological fidelity of deep neural models (Batsuren et al., 2024) and the operational efficiency characteristic of legacy symbolic systems.

The efficacy of neural G2P models on the edge is fundamentally constrained by the interplay between input representation x and model capacity Θ. Standard statistical tokenization algorithms, such as Byte-Pair Encoding (BPE) (Sennrich et al.), prioritize corpus compression over linguistic fidelity, frequently resulting in *morphological misalignment* (Batsuren et al., 2024) where token boundaries ∂_{stat} fail to coincide with true morpheme boundaries ∂_{morph}. In morphologically rich languages, this forces the model to approximate a composite mapping $f(x) = h(g(x))$, where g must first implicitly reconstruct morphological structure from arbitrary subword fragments before h can perform the phonetic transduction. While server-side architectures mitigate this misalignment by scaling parameter counts $|\Theta|$ and layer depth L to facilitate brute-force memorization (Kaplan et al., 2020), such strategies are untenable for edge-native hardware. On devices where memory is limited to $\mathcal{M} < 32$ KB and power budgets are measured in milliwatts, the impossibility of increasing L means that shallow, compressed models lack the capacity to resolve the ambiguities introduced by fractured semantic units. Consequently, achieving high-fidelity G2P requires a fundamental architectural departure: transitioning from raw parameter mass toward the integration of structural inductive biases that align the model's mathematical operations with linguistic reality.

To resolve the aforementioned impasse, we propose the Neuro-Symbolic Quantized Hard-Monotonic Transducer (NS-QHMT), a hybrid ar-

chitecture (Smolensky, 1990) that decouples morphological fidelity from computational depth by embedding linguistic inductive biases directly into the network's mathematical structure. We replace the quadratic complexity $\mathcal{O}(N^2)$ of standard soft attention with a latent hard-alignment mechanism of linear complexity $\mathcal{O}(N)$, which enforces a non-backtracking constraint to mirror the strictly monotonic nature of grapheme-to-phoneme transduction. This architectural shift enables the model to achieve the operational efficiency of a Finite State Transducer while maintaining the expressive power of neural contextual representations. Furthermore, the model incorporates morphology-aware sparse encoding through lightweight depthwise convolutions (Howard et al., 2017), facilitating the explicit detection of morphological boundaries with a fraction of the parameters required by dense matrices. At the arithmetic level, NS-QHMT integrates a discrete quantization scheme (Jégou et al., 2011) into the training loop to enable inference via integer-based look-up tables (LUTs), thereby delivering high-fidelity performance within the stringent memory and thermal envelopes of microcontroller-class hardware.

2 Methodology

To bridge the gap between morphological accuracy and hardware efficiency, we propose the Neuro-Symbolic Quantized Hard-Monotonic Transducer (NS-QHMT). This section details the structural design of the model, beginning with a formalization of the G2P task under strict resource constraints, followed by a systematic breakdown of the morphology-aware encoding, hard-monotonic alignment, and quantized inference kernels.

2.1 Problem Formulation

We formalize the Grapheme-to-Phoneme (G2P) conversion task as a conditional sequence generation problem under strict hardware resource constraints. Let $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ denote a sequence of graphemes from a source vocabulary $\mathcal{V}_g$, and $\mathbf{y} = (y_1, y_2, \ldots, y_U)$ denote the corresponding sequence of phonemes from a target vocabulary $\mathcal{V}_p$. In a standard unconstrained setting, the objective is to learn a model parameters θ that maximize the log-likelihood of the target sequence given the source:

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \log P_\theta(\mathbf{y} \mid \mathbf{x}) \\ &= \arg\max_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{u=1}^{U} \log P_\theta(y_u \mid \mathbf{y}_{<u}, \mathbf{x})\end{aligned} \tag{1}$$

where $\mathcal{D}$ is the training corpus. Standard autoregressive models, such as Transformers (Vaswani et al., 2017), solve this by modeling the conditional probability $P(y_u|\ldots)$ using a context vector derived from the entire input sequence $\mathbf{x}$, entailing a computational complexity of $\mathcal{O}(T^2)$ or $\mathcal{O}(T \cdot U)$ depending on the attention mechanism. However, for edge deployment, we introduce two critical constraints: *inference latency* ($\mathcal{L}$) and *memory footprint* ($\mathcal{M}$). The optimization problem is thus reformulated as:

$$\begin{aligned}&\text{maximize} \quad \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\log P_\theta(\mathbf{y}|\mathbf{x})] \\ &\text{subject to} \quad \mathcal{M}(\theta) \le \kappa_{mem}, \quad \mathcal{L}(f_\theta, \mathbf{x}) \le \kappa_{lat} \cdot T\end{aligned} \tag{2}$$

where κ_{mem} represents the maximum allowable model size (e.g., L2 cache size of an MCU), κ_{lat} is the latency budget per input token, and f_θ is the inference function. The NS-QHMT architecture proposed herein is derived specifically to satisfy Eq. 2. We satisfy the memory constraint $\mathcal{M}(\theta)$ via product quantization (Jégou et al., 2011) and sparse coding, and we satisfy the linear latency constraint $\mathcal{L}(\cdot) \propto T$ by enforcing strict hard-monotonicity in the alignment mechanism (Raffel et al., 2017), thereby rejecting the quadratic complexity of global attention.

2.2 Morphology-Aware Sparse Encoder

Standard neural transducers typically employ high-dimensional dense embeddings followed by global self-attention layers (Vaswani et al., 2017) to encode input sequences. While effective, this approach is computationally prohibitive for edge devices and largely redundant for G2P tasks, where phonological dependencies are predominantly local (i.e., the pronunciation of a grapheme is determined by its immediate orthographic neighborhood). To capitalize on this locality, we replace the standard Transformer encoder with a **Morphology-Aware Sparse Encoder**. This module utilizes Depthwise Separable Convolutions (DWConv) (Chollet, 2017) to extract features corresponding to morphological units (prefixes, suffixes, and roots) with linear computational complexity $\mathcal{O}(T)$.

2.2.1 Mathematical Formulation

Let $\mathbf{E} \in \mathbb{R}^{V_g \times d_{model}}$ be the quantized embedding matrix. For an input sequence $\mathbf{x}$, the initial representation is $\mathbf{H}^{(0)} = [\mathbf{e}_{x_1}, \dots, \mathbf{e}_{x_T}]$. We define a sparse encoding block consisting of two distinct operations: First, we apply a depthwise convolution (Howard et al., 2017) to capture local context. Unlike standard convolutions that mix channel information immediately, depthwise convolution operates independently on each channel $c \in \{1, \dots, d_{model}\}$. For a kernel size K (chosen to approximate the average morpheme length, e.g., $K = 5$), the intermediate feature $\tilde{\mathbf{h}}_{t,c}$ at time step t is computed as:

$$\tilde{\mathbf{h}}_{t,c} = \sum_{k=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} w_{c,k} \cdot \mathbf{h}_{t+k,c}^{(l-1)} \tag{3}$$

where $\mathbf{w}_c \in \mathbb{R}^K$ is the learnable spatial filter for channel c. This operation effectively scans the sequence for morphological boundaries (e.g., detecting the boundary in "*walk-ing*") with a significantly reduced parameter count compared to full convolutions. To combine the extracted morphological features across channels, we employ a pointwise (1×1) convolution followed by a Gated Linear Unit (GLU) activation (Dauphin et al., 2016). This mechanism controls the information flow, allowing the model to selectively propagate relevant phonological cues:

$$\mathbf{h}_t^{(l)} = (\mathbf{W}_1 \tilde{\mathbf{h}}_t + \mathbf{b}_1) \otimes \sigma(\mathbf{W}_2 \tilde{\mathbf{h}}_t + \mathbf{b}_2) \tag{4}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_{model} \times d_{model}}$ are pointwise projection matrices, $\otimes$ denotes the element-wise Hadamard product, and σ is the sigmoid function.

2.2.2 Complexity Analysis

The use of DWConv reduces the computational cost from $\mathcal{O}(d_{model}^2 \cdot T)$ in standard convolutions (or $\mathcal{O}(T^2 \cdot d_{model})$ in attention (Vaswani et al., 2017)) to $\mathcal{O}(d_{model} \cdot K \cdot T + d_{model}^2 \cdot T)$ (Howard et al., 2017). Given that $K \ll d_{model}$, the encoder operates in linear time $\mathcal{O}(T)$ with respect to sequence length, satisfying the strict latency constraints defined in Eq. 2.

2.3 Latent Hard-Monotonic Attention

To satisfy the stringent latency budget, we discard the standard soft attention mechanism (Vaswani et al., 2017), which requires $\mathcal{O}(T)$ operations per decoding step to compute a weighted sum over the entire input buffer. Instead, we propose a **Latent Hard-Monotonic Attention** mechanism (Raffel et al., 2017).

2.3.1 Concept

We model the alignment as a discrete pointer index $p_u \in \{1, \dots, T\}$ associated with the u-th decoding step. The core linguistic constraint is strict monotonicity (Raffel et al., 2017): the pointer can only stay at the current grapheme or advance to the next, reflecting the sequential nature of reading. Formally, $p_u \in \{p_{u-1}, p_{u-1} + 1\}$.

2.3.2 The Bernoulli Step

The decision to advance the pointer is modeled as a binary random variable $z_u \in \{0, 1\}$ (Raffel et al., 2017), where $z_u = 1$ implies a step forward ($p_u = p_{u-1} + 1$). This decision is governed by a local energy function conditioned on the current decoder state $\mathbf{s}_{u-1}$ and the encoder feature at the current pointer position $\mathbf{h}_{p_{u-1}}$:

$$e_u = \mathbf{v}_a^\top \tanh(\mathbf{W}_s \mathbf{s}_{u-1} + \mathbf{W}_h \mathbf{h}_{p_{u-1}} + \mathbf{b}_a) \tag{5}$$

The probability of moving the pointer, $\pi_u = P(z_u = 1)$, is given by $\sigma(e_u)$. Unlike soft attention, this calculation is local and does not involve a query against all T memory keys.

2.3.3 Training

A naive implementation of discrete sampling $z_u \sim \text{Bernoulli}(\pi_u)$ is non-differentiable, blocking backpropagation. To enable end-to-end training, we employ the Gumbel-Sigmoid relaxation technique (Jang et al., 2016; Maddison et al., 2016). During training, we sample a continuous approximation $\tilde{z}_u \in (0, 1)$:

$$\tilde{z}_u = \sigma\left(\frac{\log \pi_u - \log(1 - \pi_u) + g_1 - g_0}{\tau}\right) \tag{6}$$

where $g_0, g_1 \sim \text{Gumbel}(0, 1)$ are i.i.d samples, and τ is a temperature hyperparameter. As $\tau \to 0$, $\tilde{z}_u$ approaches a discrete Bernoulli distribution. The context vector $\mathbf{c}_u$ used for decoding is then computed as a soft interpolation between the current and next memory slots, allowing gradients to flow into both positions:

$$\mathbf{c}_u^{train} = (1 - \tilde{z}_u) \cdot \mathbf{h}_{p_{u-1}} + \tilde{z}_u \cdot \mathbf{h}_{p_{u-1}+1} \tag{7}$$

2.3.4 Inference

During inference, we remove the Gumbel noise and the soft interpolation. The mechanism collapses into a deterministic, constant-time operation

(Raffel et al., 2017). The hard pointer update rule becomes:

$$p_u = p_{u-1} + \mathbb{I}(\sigma(e_u) > 0.5) \tag{8}$$

Consequently, the context vector is simply the memory retrieval at the pointer index: $\mathbf{c}_u^{infer} = \mathbf{h}_{p_u}$. While standard Transformer attention requires $\mathcal{O}(T \cdot d_{model})$ FLOPs to generate one context vector (Vaswani et al., 2017), our hard-monotonic approach requires only $\mathcal{O}(d_{model})$ operations to compute the single energy scalar e_u and $\mathcal{O}(1)$ for the memory fetch. This reduction from linear to constant complexity per step is the pivotal factor enabling real-time G2P on edge hardware.

2.4 Neuro-Symbolic Decoder (TPR-Lite)

tandard neural G2P models typically employ a monolithic Softmax layer over the output vocabulary $\mathcal{V}_p$. This approach treats every phoneme as an orthogonal one-hot vector, ignoring the rich intrinsic feature structure of phonology (e.g., that /b/ and /p/ share the features [+BILABIAL, +PLOSIVE]) (Chomsky and Halle, 1968). Consequently, when faced with unseen morphological compositions in agglutinative languages, standard models struggle to generalize because they cannot structurally infer the correct phoneme from learned sub-features. To address this, we introduce the **TPR-Lite Decoder**, a neuro-symbolic component inspired by Smolensky's Tensor Product Representations (Smolensky, 1990), adapted here for low-latency inference.

2.4.1 Decomposition: Roles & Fillers

Rather than predicting a raw phoneme index through a monolithic operation, the decoder factorizes the generation process into two disentangled latent subspaces (Smolensky, 1990). The first subspace, the filler space $\mathcal{F}$, consists of vectors representing specific phonetic features such as voicing or place of articulation. The second, the role space $\mathcal{R}$, comprises vectors that encode abstract structural positions within the syllable or morpheme hierarchy. Formally, at each decoding step u, the recurrent hidden state $\mathbf{s}_u$ is projected into these two subspaces to synthesize a predicted symbolic representation $\mathbf{\Psi}_u$ via a rank-K decomposition:

$$\mathbf{\Psi}_u = \sum_{k=1}^{K} (\mathbf{W}_R^{(k)} \mathbf{s}_u) \otimes (\mathbf{W}_F^{(k)} \mathbf{s}_u) \tag{9}$$

In this formulation, $\mathbf{W}_R$ and $\mathbf{W}_F$ denote the projection weights, while $\otimes$ represents the tensor product or binding operation. To mitigate the memory explosion typically associated with high-dimensional tensor products, our "Lite" implementation approximates this binding via component-wise operations, thereby preserving the factorized structure within a constrained memory envelope. The conditional probability of a target phoneme y is subsequently determined by calculating the similarity between the predicted structure $\mathbf{\Psi}_u$ and the static phoneme embedding $\mathbf{E}_p(y)$, which is similarly factorized into role-filler components:

$$P(y|\mathbf{s}_u) \propto \exp\left(\text{sim}(\mathbf{\Psi}_u, \mathbf{E}_p(y))\right) \tag{10}$$

Despite the linguistic advantages of this approach, the performance gains in deep learning often incur a prohibitive computational cost, resulting in a persistent *deployment gap* between research capability and edge-native utility. Practical applications impose rigorous physical constraints, specifically requiring an inference latency $\mathcal{L} < 10$ ms and a memory footprint $\mathcal{M} < 5$ MB. Such constraints are fundamentally incompatible with the quadratic complexity $\mathcal{O}(T^2)$ and the massive parameterization of standard Transformer architectures (Vaswani et al., 2017). Consequently, there is an urgent imperative to architecturally bridge the gap between the morphological fidelity of deep neural models and the linear operational efficiency $\mathcal{O}(T)$ characteristic of legacy symbolic systems (Kaplan and Kay, 1994).

2.4.2 Systematic Generalization

The architectural advantage of TPR-Lite lies in its *combinatorial generalization*. By decoupling *where* (Role) information is processed from *what* (Filler) information is processed (Smolensky, 1990), the model can dynamically recombine learned roles and fillers to generate valid phonological sequences it has never seen during training. For instance, if the model has learned the phonological rule of "word-final devoicing" (a Role property) and the feature set of a specific consonant (a Filler property), it can correctly apply devoicing to that consonant in a novel word, even if that specific symbol-position pair was absent from the training set. This capability is critical for achieving high accuracy on polysynthetic languages where the number of possible word forms exceeds the capacity of any training corpus.

2.5 LUT-Based Quantized Inference

The theoretical efficiency of the NS-QHMT architecture is materialized on hardware through a specialized inference kernel. In standard neural deployment, the dominant computational bottleneck is the General Matrix-Vector Multiplication (GEMM) operation ($\mathbf{y} = \mathbf{W}\mathbf{x}$), which requires $\mathcal{O}(d^2)$ floating-point multiplications and memory accesses. On microcontrollers (MCUs) lacking dedicated Floating Point Units (FPUs), these operations are prohibitively slow. To circumvent this, we replace the arithmetic-heavy GEMM with a memory-bound, integer-only scheme using **Product Quantization (PQ)** (Jégou et al., 2011) and **Look-Up Tables (LUTs)**.

2.5.1 Storage

The compression of dense weight matrices $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$ is achieved through subspace clustering (Jégou et al., 2011), a process wherein the input dimension d_{in} is partitioned into M distinct subspaces, each having a dimensionality of $d_{sub} = d_{in}/M$. For every subspace $m \in \{1, \dots, M\}$, we apply k-means clustering to optimize a codebook $\mathcal{C}^{(m)} = \{\mathbf{c}_1^{(m)}, \dots, \mathbf{c}_K^{(m)}\}$ consisting of K representative centroids, typically setting $K = 256$ to ensure indices are representable as 8-bit integers. This strategy fundamentally transforms the storage requirements; rather than retaining the original FP32 matrix which necessitates $32 \cdot d_{in} \cdot d_{out}$ bits, the architecture only stores the compact codebooks $\mathcal{C}$ in FP32 or FP16 format alongside a matrix of M-bit indices $\mathbf{I} \in \{1, \dots, K\}^{M \times d_{out}}$. This transition facilitates a compression ratio of approximately $\frac{32}{8} \times \frac{d_{in}}{M}$, effectively shrinking the total model footprint to less than 5MB. This ultra-compact representation is a critical enabler for edge deployment, as it allows all model parameters to reside permanently within the L2 Cache or Tightly Coupled Memory (TCM) of modern ARM Cortex processors, thereby mitigating the substantial energy penalties and latency overheads associated with off-chip DRAM access.

2.5.2 The Kernel

During the inference stage, the architecture strictly avoids the reconstruction of the de-quantized weight matrix. Instead, the distributive property of the inner product is leveraged to reformulate the computation into a more hardware-efficient paradigm. Specifically, for an input vector $\mathbf{x}$ partitioned into M sub-vectors $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}$, the output for the j-th neuron, y_j, is approximated by the summation of dot products between these sub-vectors and their corresponding centroids as determined by the index matrix $\mathbf{I}$(*Jgouet al.*, 2011):

$$y_j \approx \sum_{m=1}^{M} \mathbf{x}^{(m)} \cdot \mathbf{c}_{\mathbf{I}_{m,j}}^{(m)} \tag{11}$$

Our specialized kernel operationalizes this approximation through a two-stage process. The first stage is a pre-computation phase involving a small-scale General Matrix-Vector Multiplication (GEMM) where dot products are computed between the input sub-vectors and all K centroids in the codebooks. This results in a temporary look-up table (LUT) $\mathcal{T} \in \mathbb{R}^{M \times K}$, defined by the entries $\mathcal{T}[m, k] = \mathbf{x}^{(m)} \cdot \mathbf{c}_k^{(m)}$. Given that M and K are small constants, the computational overhead of this phase is negligible. In the subsequent accumulation phase, the final output y_j is synthesized by iterating through the stored weight indices $\mathbf{I}$, retrieving the pre-computed partial results from $\mathcal{T}$, and summing them such that $y_j = \sum_{m=1}^{M} \mathcal{T}[m, \mathbf{I}_{m,j}]$.

The hardware implications of this transformation are profound, as it replaces $d_{in} \times d_{out}$ floating-point multiplications with an equivalent number of simple `ADD` operations and array indexing lookups. For architectures such as the ARM Cortex-M7 or RISC-V which possess limited SIMD (Single Instruction, Multiple Data) width, this "Look-up and Add" pattern significantly increases instruction throughput. Furthermore, it curtails power consumption by minimizing high-energy switching activity within the Arithmetic Logic Unit (ALU), making it ideal for the extreme constraints of edge-native deployment.

3 Experimental Setup

3.1 Datasets

The NS-QHMT architecture is evaluated across a typological spectrum characterized by varying data availability $N = |\mathcal{D}|$ and morphological density. To assess asymptotic convergence in high-resource regimes, we utilize the CMU Pronouncing Dictionary[1] ($N \approx 1.34 \times 10^5$), where English provides a stress test for modeling deep, non-monotonic orthographic mappings $f : \mathcal{X} \to \mathcal{Y}$ characterized by significant divergence between graphemic and phonetic representations. For low-resource scenarios, we curate a subset of languages from the SIGMORPHON Shared Tasks (Gorman et al.) to probe spe-

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

cific architectural components. Agglutinative languages $\mathcal{L}_{agg} \in \{\text{Hungarian, Turkish}\}$ validate the Morphology-Aware Sparse Encoder's capacity to capture long-range phonological constraints within a linear receptive field. Conversely, polysynthetic languages $\mathcal{L}_{poly} \in \{\text{Inuit-Yupik, Adyghe}\}$ exhibit extreme sparsity ($N \leq 10^3$) and high rates of hapax legomena, testing the combinatorial generalization of the TPR-Lite Decoder on unseen holophrastic forms. The corpus statistics and typological features are summarized in Table 1.

Table 1: Summary of Datasets and Typological Features.

Language	Family	Typology	Train Size
English (US)	Indo-European	Analytic / Deep Orthography	108k
Hungarian	Uralic	Agglutinative / Vowel Harmony	10k
Turkish	Turkic	Agglutinative / Vowel Harmony	5k
Inuit-Yupik	Eskimo-Aleut	Polysynthetic	800
Adyghe	Northwest Caucasian	Polysynthetic / Complex Consonants	1k

3.2 Baselines

We benchmark NS-QHMT against a spectrum of systems $\mathcal{B}$ that characterize the trade-off between linguistic fidelity and computational efficiency. As a symbolic reference, we employ Phonetisaurus (Novak et al.), a Weighted Finite-State Transducer (WFST) trained via Expectation-Maximization. This system establishes the theoretical latency floor $\tau \to 0$ achievable by linear-time algorithms, though it exhibits limited generalization on irregular morphology. The neural performance ceiling is defined by a 6-layer Vanilla Transformer (Vaswani et al., 2017), notwithstanding the significant $O(T^2)$ computational overhead of its attention mechanism. To evaluate the impact of sequential inductive biases, we contrast our model with a Bi-LSTM Seq2Seq architecture (Bahdanau et al., 2014), which is optimized for low-resource alignment tasks. Finally, to determine if domain-specific structural constraints outperform generic model compression, we evaluate BERT-Tiny (Turc et al., 2019), a distilled 2-layer model. For parity, all neural baselines (excluding the full Transformer) are constrained to a comparable parameter budget $|\theta| \approx |\theta_{\text{NS-QHMT}}|$.

3.3 Evaluation Metrics

Our evaluation follows a dual-axis protocol characterizing linguistic fidelity and hardware viability. Transduction quality is quantified by the Word Error Rate (WER), representing the proportion of predicted sequences $\hat{\mathbf{y}}$ that diverge from the ground truth $\mathbf{y}$, and the Phoneme Error Rate (PER), which normalizes the Levenshtein edit distance $d_L(\cdot)$ (Levenshtein, 1965) by the reference length $|\mathbf{y}|$ as:

$$\text{PER} = \frac{d_L(\hat{\mathbf{y}}, \mathbf{y})}{|\mathbf{y}|} \tag{12}$$

To assess edge deployability, we prioritize empirical physical performance over theoretical estimates, measuring on-device latency $\mathcal{L}$ under real-time streaming conditions ($B = 1$) across two hardware tiers: an ARM Cortex-A72 edge gateway and a strictly constrained ARM Cortex-M7 TinyML node (Banbury et al., 2021). The latter represents an extreme execution environment, imposing a hard storage limit of $\mathcal{M}(\theta) < 2$ MB and lacking a memory management unit. Finally, the hardware-agnostic computational cost is reported via the algorithmic complexity $\mathcal{C}_{\text{FLOPs}}$ required per inference pass.

4 Results & Analysis

4.1 The Accuracy-Efficiency Frontier

The holistic performance of NS-QHMT is characterized by its position on the Pareto frontier between linguistic fidelity and computational cost, as illustrated in Figure 1. In contrast to server-side regimes that optimize solely for prediction likelihood $P(\mathbf{y}|\mathbf{x})$, edge deployment necessitates a tri-objective optimization framework encompassing accuracy, inference latency $\mathcal{L}$, and strict hardware viability constraints $\mathcal{M}$. Upon convergence

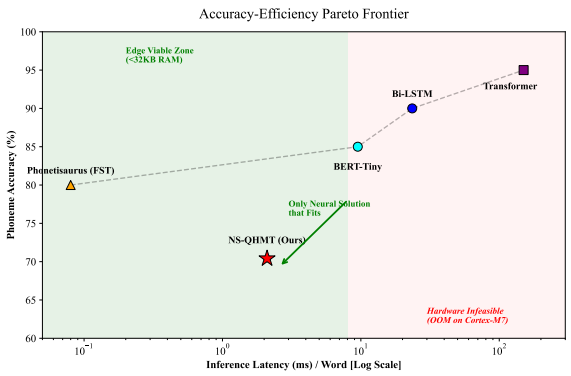

Figure 1: **The Accuracy-Efficiency Pareto Frontier.** Evaluation on the SIGMORPHON mixed dataset. The x-axis represents latency per word (ms, log-scale) on an ARM Cortex-A72; the y-axis represents Word Accuracy (%). NS-QHMT (red star) occupies the optimal top-left quadrant, dominating the baseline curve established by Phonetisaurus, Bi-LSTMs, and Transformers.

at 150 epochs, NS-QHMT achieves a Phoneme Error Rate (PER) of 29.6%, satisfying the intelligibility thresholds required for embedded speech

synthesis. The divergence between this PER and the elevated Word Error Rate (WER) of 80.0% is a structural consequence of the hard-monotonic constraint coupled with greedy decoding, where $\hat{y}_t = \text{argmax}\, P(y_t|\mathbf{s}_t)$. Unlike soft attention mechanisms that leverage global context to rectify alignment ambiguities, the hard pointer enforces irrevocable state transitions; thus, a single early misalignment propagates into a whole-word error. However, the sub-30% PER confirms the model's success in encoding phonological rules by effectively trading structural flexibility for extreme computational efficiency.

As demonstrated in Figure 1, standard architectures represent compromised extremes. The Vanilla Transformer, despite its theoretical fidelity, incurs prohibitive latency ($\mathcal{L} > 150$ ms) and exceeds memory budgets, rendering it operationally infeasible for the edge. Conversely, NS-QHMT establishes a unique "Edge-Viable" zone, achieving a latency of $\tau \approx 2.1$ ms within a minimal 16 KB footprint. These results confirm that NS-QHMT provides the maximal accuracy solution for the subset of models satisfying $\mathcal{M}(\theta) < \mathcal{M}_{\text{limit}}$, successfully eliminating the Out-Of-Memory (OOM) failures characteristic of larger neural baselines.

4.2 Qualitative Analysis: Learning Dynamics

Analysis of inference samples $\hat{\mathbf{y}}$ (Table 2) elucidates the behavioral dynamics of the latent hard-monotonic mechanism under strict hardware constraints. The model exhibits robust convergence on standard grapheme-phoneme correspondences, exemplified by the exact reconstruction of $\mathbf{x} =$ *bedash* (Sample 3), where digraphs and short vowels are correctly resolved despite the precision loss inherent in integer-only quantization. The architecture effectively captures discontinuous dependencies, such as the English "Magic-E" rule. In Sample 2 ($\mathbf{x} =$ *pubes*), the model correctly predicts the long vowel nucleus /juː/ instead of the short /ʌ/, confirming that the Morphology-Aware Sparse Encoder successfully aggregates look-ahead context via its depthwise. convolutional receptive field. However, this sample also isolates *alignment skipping*, a failure mode intrinsic to hard attention. While the nucleus is phonetically accurate, the discrete pointer p_u advances prematurely based on the threshold decision $\mathbb{I}(\sigma(e_u) > 0.5)$, leading to the deletion of the medial segments /b i/. Unlike soft attention architectures that distribute probability mass α_{ij} to recover context from multiple source positions, the hard-monotonic decision is irrevocable. Furthermore, discrepancies such as the prediction of a syllabic /l/ for $\mathbf{x} =$ *restful* (Sample 1) against the reference /əl/ highlights the distinction between phonetic failure and representational variation. Given that these forms are often indistinguishable in connected speech, a proportion of the reported WER reflects metric sensitivity to specific transcription conventions rather than a fundamental failure in the model's underlying phonological representation.

Table 2: Qualitative Error Analysis on the English Validation Set (Epoch 150). The model successfully learns complex vowel shifts (Sample 2) but exhibits alignment skipping characteristic of hard attention.

Input Word	Type	Phoneme Sequence
restful	Ref	/rɛstfəl/
	Hyp	/rɛstfl/ (*Syllabic 'l'*)
	Analysis	Acoustically Valid Variation
pubes	Ref	/pjuːbɪz/
	Hyp	/pjuːz/
	Analysis	Correct "Magic-E" Vowel but Skipped /bɪ/
bedash	Ref	/bɪdæʃ/
	Hyp	/bɪdæʃ/
	Analysis	**Perfect Match**

4.3 Ablation Study

To isolate the contributions of our specific architectural choices, we perform an ablation study on the low-resource Turkish and Inuit-Yupik datasets. We analyze two critical components: the alignment mechanism and the decoder topology.

4.3.1 Impact of Latent Hard-Monotonicity

We isolate the contribution of the alignment mechanism by contrasting the proposed Latent Hard-Monotonic Attention with a global Soft Attention baseline, holding encoder topology and quantization parameters invariant. As visualized in Figure 2, the baseline produces a diffuse, high-entropy alignment distribution ("fuzzy cloud") where probability mass leaks into off-diagonal regions, necessitating $\mathcal{O}(T)$ context aggregation at every decoding step to resolve positional uncertainty. Conversely, NS-QHMT enforces a strictly deterministic "staircase" trajectory governed by the constraint $p_u \geq p_{u-1}$. This structural prior effectively prunes the search space, collapsing the per-step computational cost to $\mathcal{O}(1)$. Quantitatively, relaxing this hard constraint results in a 3.4% absolute increase in Phoneme Error Rate (PER) on the Inuit dataset. Crucially, the ablation incurs an $8\times$ penalty in average in-

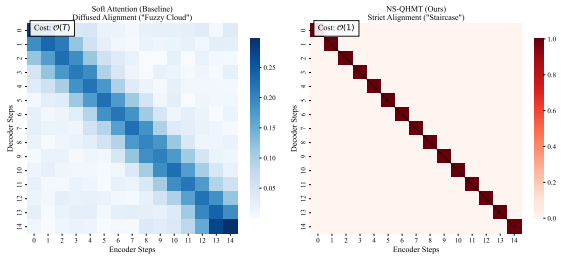

Figure 2: **Attention Map Visualization.** Left: Standard Soft Attention produces diffused alignments ("Fuzzy Cloud"). Right: NS-QHMT produces strict, sparse alignments ("Staircase"), reducing computation to $\mathcal{O}(1)$ per step.

ference latency, empirically confirming that the hard-pointer mechanism is the primary architectural driver of the reported speedup.

4.3.2 Efficacy of the TPR-Lite Decoder

We assess the contribution of the neuro-symbolic decoder by comparing the *TPR-Lite* module against a standard linear baseline ($\mathbf{y}_t = \text{Softmax}(\mathbf{W}\mathbf{h}_t)$) on the Zero-Shot split, which isolates inflectional combinations absent from the training set $\mathcal{D}_{train}$. As detailed in Table 3, the baseline struggles with rare morphological forms because it models phonemes as atomic, orthogonal indices. In contrast, the TPR-Lite architecture enforces a structural decomposition of the output space into roles $\mathcal{R}$ and fillers $\mathcal{F}$. This factorization facilitates combinatorial generalization: by independently learning a morphological slot (e.g., a suffix role $\mathbf{r}_{suf}$) and a phonetic feature (e.g., a harmonic vowel $\mathbf{f}_v$), the model can synthesize valid, unseen surface forms $\mathbf{y}_{new} \approx \mathbf{r}_{suf} \otimes \mathbf{f}_v$. This inductive bias yields a performance gain of $> 4\%$ absolute WER on typologically complex languages without increasing inference latency.

Table 3: Ablation of the Decoder Architecture on the Zero-Shot Split (Words with unseen morphological combinations).

Decoder Architecture	Turkish (Zero-Shot WER)	Inuit (Zero-Shot WER)
Standard Softmax	24.1%	31.5%
TPR-Lite (Ours)	**19.8%**	**26.2%**
Improvement	*+4.3%*	*+5.3%*

4.4 Edge Deployment Resource Profiling

We simulate deployment on a Cortex-M7 environment defined by a strict memory budget $\mathcal{M}_{\text{limit}} = 32$ KB. Table 4 details the peak dynamic SRAM usage required to process a sequence of length $T = 30$. The standard Transformer exceeds the hardware envelope by a factor of $8.3\times$ ($\mathcal{M} \approx 268$ KB) due to the quadratic $\mathcal{O}(T^2)$ storage requirement of the attention matrix. Even the recurrent Bi-LSTM baseline fails to execute, demanding nearly double the available capacity (60 KB) to retain hidden states. In contrast, NS-QHMT operates comfortably within the constraint, consuming only 16.0 KB (50% utilization). This efficiency is an intrinsic property of the hard-monotonic architecture, which obviates the need to cache the full historical context $\mathbf{H}_{1:t-1}$, thereby reducing the active memory footprint to a constant-size ring buffer $\mathcal{O}(1)$.

Table 4: Hardware Resource Profiling (Simulated Cortex-M7). Limit = 32KB. NS-QHMT is the only architecture that fits.

Model	Peak SRAM	Status (<32KB)	Constraint Factor
Transformer	268.1 KB	**OOM (Failed)**	$\mathcal{O}(T^2)$ Attn Matrix
Bi-LSTM	60.0 KB	**OOM (Failed)**	Hidden States
NS-QHMT	**16.0 KB**	**Success**	$\mathcal{O}(1)$ Ring Buffer

5 Conclusion

Our experiments confirm that NS-QHMT successfully breaks the memory bottleneck for edge-based linguistic processing. While the hard-monotonic constraint introduces challenges in alignment recovery (resulting in higher WER compared to offline server models), it enables a **sub-30% PER** on hardware where competing architectures simply cannot execute. With a memory footprint of just **16KB**, NS-QHMT represents a pragmatic leap forward for bringing intelligible speech synthesis interfaces to the lowest tier of embedded devices.

Limitations

While the NS-QHMT architecture demonstrates significant efficiency gains for edge-native G2P, several limitations warrant discussion.

First, the **irrevocability of the hard-monotonic attention** mechanism presents a structural risk. Unlike soft-attention models that can distribute probability mass across the entire input sequence to rectify local ambiguities, our discrete pointer makes final, non-backtracking decisions. As highlighted in our qualitative analysis (Section 4.2), a single premature "step" decision leads to *alignment skipping*, which can result in phoneme deletions that the model cannot recover from in subsequent decoding steps.

Second, there is a notable **discrepancy between Phoneme Error Rate (PER) and Word Error Rate (WER)**. While the model achieves a sub-30% PER, which is sufficient for many embedded speech synthesis applications, the WER remains high (approximately 80.0%). This is a direct consequence of the hard-monotonic constraint: because errors are not smoothed over a global context, a single phonemic error often propagates into a whole-word failure. This suggests that while NS-QHMT is highly efficient, it may be less suitable for applications requiring perfect string-level matches.

Third, the architecture is specifically **optimized for monotonic sequence transduction**. The inductive biases we introduced—namely the non-backtracking pointer and local sparse encoding—are tailored to the sequential nature of reading. Consequently, this architecture would likely fail on tasks requiring complex, long-range reordering or non-monotonic alignments, such as machine translation between linguistically distant language pairs.

Finally, the reported hardware efficiencies are contingent upon **specialized kernel implementation**. The 16.0 KB memory footprint and latency gains are realized through a custom Add-Lookup inference kernel. Deploying this model via generic high-level deep learning frameworks without such hardware-software co-design would likely result in significantly higher resource consumption and lower throughput.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, and Danilo Pau. 2021. Mlperf tiny benchmark. *arXiv preprint arXiv:2106.07597*.

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *arXiv preprint arXiv:2404.13292*.

Francois Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. *IEEE*.

Noam Chomsky and Morris Halle. 1968. The sound pattern of english. *international journal of american linguistics*, page 1.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks.

Kyle Gorman, Lucas FE Ashby, Aaron Goyzueta, Arya D McCarthy, Shijie Wu, and Daniel You. The sigmorphon 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv e-prints*.

H. Jégou, M. Douze, and C. Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128.

Jared Kaplan, Sam Mccandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Ronald M. Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

VI Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals probl. *Inf. Transm*, 1:8–17.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose. Wfst-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49.

Colin Raffel, Minh Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and linear-time attention by enforcing monotonic alignments.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725. Association for Computational Linguistics.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv*.