# A Spectral Alphabet Wheel for Modeling Letter Transitions in German and English

Daniel Hommers
private research
daniel@hommers.de

**Abstract**

Modeling letter/digraph transitions can reveal phonotactic structure useful for computational linguistics and language modeling. We introduce the Alphabet Wheel, a data-driven representation that places letters and key digraphs as operators on a circle via spectral embedding; in real corpora, operator flows prefer small angular turns. For German (DE, N=14 sectors) and English (EN, N=16), the wheel's energy (mean angular turn) is significantly lower than two nulls—angle-shuffle (z ≈ -3.13 DE; -3.40 EN) and degree-preserving (z ≈ -2.51 DE; -3.79 EN)—and token placements are stable (DE: median circular SD ≈ 9.38°, 97.5% sector agreement; EN: SD ≈ 42.6° across domains). A shuffled-angle baseline shows $\sigma^\circ \approx 114^\circ$ and ≈9% sector agreement, underscoring the effect. High-PMI "chords" (e.g., QU→ECK, PMI=2.47) highlight language-specific constraints. The framework yields a compact, falsifiable phonotactic prior with immediate applications in NLP and linguistic visualization.

**Keywords:** phonotactics, spectral embedding, letter transitions, graph models, computational linguistics

## 1 Introduction

Local n-gram models capture short-range dependencies but miss higher-level structure. We propose the Alphabet Wheel: letters and fused digraphs act as operators placed on a circle so that typical flows take short angular steps. The wheel is learned from corpus bigrams and validated against null models. We present DE (single official corpus) and EN (news/web/wiki), with plans to extend to Indonesian.

## 2 Methods

### 2.1 Data & Tokens

Upper-case letters; DE diacritics normalized (Ä→AE, Ö→OE, Ü→UE, ß→SS). Fused tokens as single operators (DE: SCH, CH, ST, NG, PF, QU; EN: TH, SH, PH, GH, ST, PR, . . . ). Bigrams ($C_{ij}$) built over word-internal transitions (BOS/EOS optional).

### 2.2 Spectral Embedding

Compute a 2D spectral embedding (complex eigenvector of row-stochastic $P$, or first non-trivial pair of the normalized Laplacian). Angles $\theta_i = \operatorname{atan2}(\Im v_i, \Re v_i)$; rotate so the vowel centroid ≈ 100°. For EN splits, align news/web/wiki frames to NEWS by mean-phasor. For DE stability, re-embed from the symmetric graph $A = (C + C^\top)/2$.

### 2.3 Mod-N Selection

Per-edge turn $\Delta\theta_{ij} = \arccos(\cos(\theta_j - \theta_i))$. Energy $E = \sum p_{ij}\Delta\theta_{ij}$ with $p_{ij}$ proportional to counts. Quantize to $N \in \{13, 14, 15, 16\}$ sector centers and pick $N$ by the most negative z against two nulls: (1) angle-shuffle (permute angles, fixed counts); (2) degree-preserving (IPF/Sinkhorn; fixed row/col sums).

### 2.4 Viability Calculus & H-Adapter

Assign edge viability by PMI and turn: $\top$ if PMI $\geq 1.0$ and $\Delta\theta \leq \pi/4$; $\bot$ if PMI $< 0$ and $\Delta\theta > 2\pi/3$; else $\sim$. The optional H-adapter upgrades $\sim \rightarrow \top$ if both $s \rightarrow H$ and $H \rightarrow d$ exist—capturing H as a consonant bridge/hinge.

### 2.5 Stability Metrics

DE: Poisson bootstrap (B=200) on counts; symmetric re-embedding; circular SD $\sigma°$ and mod-14 sector agreement vs. baseline. EN: cross-domain circular SD after frame alignment; mod-16 sector agreement.

### 2.6 Artifacts

Reproducible CSVs (mod-N sweeps, per-token tables, stability, viability, chords): `alphabet_wheel_min_release_v01.zip` (available in the chat history above).

## 3 Results

### 3.1 Mod-N Selection

DE: N=14 (z ≈ -3.13 angle-shuffle; -2.51 degree-preserving). EN: N=16 (z ≈ -3.40; -3.79).

### 3.2 Stability (with context)

DE (B=200): 40 tokens, median $\sigma° \approx 9.38°$ (p90 ≈ 17.01°); 97.5% sector agreement (mod-14). Shuffled-angle null: $\sigma° \approx 114°$, agreement ≈ 9% (≈1/14). EN (news/web/wiki): 43 tokens, median $\sigma° \approx 42.6°$ (p90 ≈ 76.7°); 2/43 tokens fully stable (mod-16). Stable: OL (≈3.4°), ST (≈9.1°); less stable: O, B (≈78–88°).

### 3.3 Operator Cards (core)

Per-token fields: angle $\theta°$, sector (mod-N), $\sigma°$, onset/coda vowel shares, viability (out/in $\top/\sim/\bot$). DE narrative: QU acts as a clamp (very low out$\top$/in$\top$, resolves mainly before vowels, e.g., QU→ECK, PMI=2.47, Section 3.4); SCH/H show high outgoing viability with large H-upgrade shares (cluster bridges); ST is permissive (out$\top \approx 0.958$) and matches ST$\gg$TS (e.g., ST→ECK, PMI=2.37). See Figure 1 for a visualization of the DE wheel and its key chords.

### 3.4 Top Chords

PMI-ranked chords showcase constraints. We highlight QU, ST (DE) and TH, ST (EN) for their high mass and phonotactic significance in consonant clusters and vowel transitions. Examples: DE QU→ECK (PMI=2.47), ST→ECK (2.37); EN TH→E (1.54), Y→ST (1.84).

## 4 Discussion

The wheel captures robust phonotactic structure: lower energy than nulls, stable sectors (especially DE), and interpretable operators (QU clamp; SCH/H hinges; ST ratchet/pre-closure). The DE vs. EN stability contrast ($\sigma° \approx 9.38°$ vs. 42.6°) reflects corpus coherence: DE's single official corpus ensures tight clustering, while EN's news/web/wiki diversity increases angular variability. The vowel-free skeleton remains significant in DE, indicating a consonantal backbone. Applications include phonotactic priors for segmentation/decoding and error detection via large-$\Delta\theta$, low-PMI edges.

## References

[1] von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 395–416.

[2] Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed.). Draft. Retrieved from https://web.stanford.edu/~jurafsky/slp3/

`de_wheel.png`

Figure 1: DE Alphabet Wheel (N=14) showing key operators (QU: $\theta^\circ \approx 113.5$, SCH: $\theta^\circ \approx 126.5$, ST: $\theta^\circ \approx 129.8$) and top chords (e.g., ST→ECK, PMI=2.37, $\Delta\theta$=0.06).

[3] Hayes, B., & Wilson, C. (2008). A maximum entropy model of phonotactics. *Linguistic Inquiry*, 39(3), 379–440.

## 5 Conclusion

A compact, data-driven Alphabet Wheel summarizes operator flow in DE/EN, beats strong nulls, and offers practical priors and visualizations.