Phonology (2023), **xx**:xx 1–41
doi:10.1017/S0373463320000247

PHONOLOGY

ARTICLE

A Learning-Based Account of Local Phonological Processes

Caleb Belth

University of Michigan, Ann Arbor, MI, USA.

Received: 04 November 2021; **Accepted:** 01 March 2023

Keywords: Learning, Phonological Processes, Locality, Computational Modeling

Abstract
Phonological processes tend to involve local dependencies, an observation that has been expressed explicitly or implicitly in many phonological theories, such as the use of minimal symbols in SPE, and the inclusion of primarily strictly-local constraints in OT. We propose a learning-based account of local phonological processes, by providing an explicit computational model. The model is grounded in experimental results that suggest children are initially insensitive to long-distance dependencies, and as their ability to track non-adjacent dependencies grows, learners still prefer local generalizations to non-local ones. The model encodes these results by constructing phonological processes starting around an alternating segment, and expanding outward to incorporate more phonological environment only when the surface form cannot be sufficiently predicted. The model successfully constructs local phonological generalizations and exhibits the same preference for local patterns that humans do, suggesting that locality can emerge as a computational consequence of a simple learning procedure.

1. Introduction

Phonological processes tend overwhelmingly to involve dependencies between adjacent segments (Gafos, 2014; Chandlee *et al.*, 2014). For example, the English plural allomorph depends on the stem-final segment, to which it is adjacent, as in (1).

(1) /dɑg-z/ → [dɑgz]
/kæt-z/ → [kæts]
/hɔrs-z/ → [hɔrsəz]

Moreover, underlying forms are often considered to be minimally different from surface forms, only exhibiting abstractness when surface alternation necessitates it (Kiparsky, 1968; Peperkamp *et al.*, 2006; Ringe & Eska, 2013; Richter, 2021). This is supported by experimental findings, where children avoid introducing discrepancies between surface and underlying forms when there is little motivation for doing so (Jusczyk *et al.*, 2002; Coetzee, 2009; Kerkhoff, 2007; Van de Vijver & Baer-Henney, 2014).

When—and only when—concrete representations are abandoned in favor of (minimally) abstract underlying representations, a child must learn a phonological process to derive the surface form from the abstract underlying form. Experimental studies are revealing about the mechanism underlying sequence learning: humans show a strong proclivity for tracking adjacent dependencies, only beginning to track non-adjacent dependencies when the data overwhelmingly demands it (Saffran *et al.*, 1996, 1997; Aslin *et al.*, 1998; Santelmann & Jusczyk, 1998; Gómez, 2002; Newport & Aslin, 2004; Gómez & Maye, 2005). As Gómez & Maye (2005, p. 199) put it, 'It is as if learners are attracted by adjacent probabilities long past the point that such structure is useful.' Indeed, artificial language experiments have repeatedly demonstrated that learners more easily learn local phonological processes than non-local ones (Baer-Henney & van de Vijver, 2012) and, when multiple possible phonological generalizations are consistent with exposure data, learners systematically construct the most local generalization (Finley, 2011; White *et al.*, 2018; McMullin & Hansson, 2019).

In this paper, we hypothesize a mechanistic account of how learners construct phonological generalizations, modeling the learner's attention as initially fixed locally and expanding farther only when local dependencies do not suffice. Our proposed model incorporates the idea that the learning of a phonological process is triggered when, and only when, underlying abstraction introduces discrepancies between underlying and surface representations (Kiparsky, 1968). We view the model's locally-centered attention and default identity assumption as being computationally parsimonious, and thus call it the *Parsimonious Local Phonology* learner (PLP). When presented with small amounts of child-directed speech, PLP successfully learns local phonological generalizations. PLP's search strategy—starting as locally as possible—leads it to accurately exhibit the same preference for local patterns that humans do. Next we review experimental results on locality § 1.1, the view of learning that PLP adopts § 1.2, and how these reflect principles of efficient computation § 1.3.

1.1. Locality

Early studies of statistical sequence learning found infants to only be sensitive to dependencies between adjacent elements in a sequence. Saffran *et al.* (1996, 1997) and Aslin *et al.* (1998) found infants as young as 8-months old to be sensitive to dependencies between *adjacent* elements, but Santelmann & Jusczyk (1998) found that even at 15-months-old, children did not track dependencies between *non-adjacent* elements. Studies with older participants revealed that the ability to track non-adjacent dependencies does eventually emerge: adults show a sensitivity to dependencies between non-adjacent phonological segments (Newport & Aslin, 2004), and 18-month-old children can track dependencies between non-adjacent morphemes (Santelmann & Jusczyk, 1998). However, even as sensitivity to non-adjacent dependencies develops, learners still more readily track local dependencies. Gómez (2002) found that 18-month-olds could track non-adjacent dependencies, but that they only did so when adjacent dependencies were unavailable. Gómez & Maye (2005) replicated these results with 17-month-olds, and attempted to map the developmental trajectory of this ability to track non-adjacent dependencies, finding that it grew gradually with age. At

12 months, infants did not track non-adjacent dependencies, but they began to by 15 months, and showed further advancement at 17 months. These experiments involved a range of elements: words, syllables, morphemes, phonological segments. Moreover, similar results have been observed in different domains, such as vision (Fiser & Aslin, 2002). Together, these results suggest that learners might only discover local patterns at early stages in development, and that even after sensitivity to less-local patterns emerges, a preference for local patterns persists.

Further experiments targeted phonological learning in particular. Subjects in Finley (2011)'s artificial language experiments learned bounded (local) harmony patterns and did not extend them to non-local contexts when there is no evidence for it. However, when exposed to unbounded (non-local) harmony patterns, subjects readily extended them to local contexts. This asymmetry suggests that learners will not posit less-local generalizations until the evidence requires it. In a different study, McMullin & Hansson (2019) found that these results replicate with patterns involving liquids and with dissimilation. Baer-Henney & van de Vijver (2012) used an artificial language experiment to test the role of locality (as well as substance and amount of exposure) in learning contextually-determined allomorphs. They found that when the allomorph was determined by a segment two positions away, learners more easily acquired and extended the pattern than when the allomorph was determined by a segment three positions away. In short, these studies demonstrate that learners posit the most local generalization consistent with the data.

1.2. The Nature of the Learning Task

We adopt the view of others (e.g., Hale & Reiss 2008; Ringe & Eska 2013; Richter 2021) that children initially store words concretely, as accurately to what they perceive as their representational capacities allow. As their lexicon grows, surface alternations sometimes motivate the positing of abstract underlying forms, which introduce discrepancies between underlying and surface forms. For example, as Richter (2018, 2021) characterized in rigorous detail, alternations such as 'eat' [it] ∼ 'eating' [iɾɪŋ] lead to the flap [ɾ] and stop [t] being collapsed into allophones of underlying /T/. Similarly, a morphemic surface alternation such as 'cats' [kæts] ∼ 'dogs' [dɑgz] may motivate an abstract underlying plural suffix /-Z/ (or default /-z/) (Berko, 1958). This view is in the spirit of Kiparsky (1968)'s *alternation condition*, and has been termed *invariant transparency* (Ringe & Eska, 2013).

A consequence is that when, and only when, concrete segments are collapsed into abstract underlying representations, the need for a phonological grammar arises, to derive the surface form for abstract underlying forms. We will use the example of stops following nasals to exemplify two significant corollaries. Voiceless stops following nasals are often considered to be a marked sequence, because post-nasal articulation promotes voicing and post-nasal voicing is typologically pervasive (Locke, 1983; Rosenthall, 1989; Pater, 1999; Hayes & Stivers, 2000; Beguš, 2016, 2019). Nevertheless, many languages—e.g. English—tolerate post-nasal voiceless stops,[1] and

[1] We note that passive, phonetic post-nasal voicing still occurs in some such languages (Hayes & Stivers, 2000); we are referring here to phonological voicing.

a few even exhibit productive, phonological post-nasal *devoicing*. For example, Coetzee & Pretorius (2010) performed a detailed experimental study of Tswana speakers, finding that some extended post-nasal devoicing, as in (2; data from Coetzee & Pretorius 2010, p. 406), productively to nonce words.

(2)	a.	/m-batla/	→ [mpatla]	'want me'
		/m-botsa/	→ [mpotsa]	'ask me'
		/m-bulela/	→ [mpulela]	'open (for) me'
	b.	/re-batla/	→ [rebatla]	'want us'
		/re-botsa/	→ [rebotsa]	'ask us'
		/re-bulela/	→ [rebulela]	'open (for) us'

Beguš (2019, p. 699) found post-nasal devoicing to be reported as a sound change in thirteen languages and dialects, and argued that despite appearing to operate against phonetic motivation, it likely emerged in each case as the result of a sequence of individually phonetically-motivated sound changes.

Including a constraint to mark post-nasal voiceless stops in languages that tolerate them makes the learning task unnecessarily difficult, because the constraint must then be downranked despite there being no surface alternation present. Instead, under invariant transparency, children learning languages that tolerate post-nasal voiceless stops will simply not learn a phonological process regarding post-nasal stops, because there is nothing to learn. Moreover, when surface alternations that lack or operate in opposition to phonetic motivation (e.g., post-nasal devoicing) occur synchronically due to diachronic processes or other causes, no serious problem arises: the child simply learns a phonological process to account for the observed alternation, as has been observed in experiments (Seidl & Buckley, 2005; Beguš, 2018).

The view that children initially hypothesize identity between surface and underlying forms enjoys experimental support. Jusczyk *et al.* (2002) found that 10-month-old infants better recognize faithful word constructions than unfaithful ones. Van de Vijver & Baer-Henney (2014) found that both 5-7yr-olds and adults were reluctant to extend German alternations to nonce words, preferring instead to treat the nonce SRs as identical to their URs. Kerkhoff (2007) reports a consistent preference for non-alternation in Dutch children ages 3-7yrs. In an artificial language experiment, Coetzee (2009) found that learners more often extend non-alternation than alternation to test words, suggesting that this is learners' default.

Of course, children's initial productions are not faithful to adult productions (Smith, 1973; Fikkert, 1994; Grijzenhout & Joppen, 1998; Grijzenhout & Joppen-Hellwig, 2002; Freitas, 2003), but this is likely due to underdeveloped control of the child's articulatory system, rather than an early state of the adult grammar (see Hale & Reiss 2008, sec 3.1 for a detailed argument). For instance, children systematically fail to produce complex CC syllable onsets in early speech even in languages that allow complex onsets, like Dutch, German, Portuguese, and English (Fikkert, 1994; Grijzenhout & Joppen, 1998; Grijzenhout & Joppen-Hellwig, 2002; Freitas, 2003; Gnanadesikan, 2004). Clusters tend to be reduced by deleting a consonant, and development proceeds from a cluster reduction stage to a full CC production stage, suggesting the discrepancy may be due to limited articulatory control.

PLP is a model of how phonological processes are learned once underlying abstraction leads to discrepancies in (UR, SR) pairs, which constitute PLP's input. As some of our reviewers pointed out, the task of learning phonological processes to account for discrepancies between underlying and surface forms is intertwined with the task of figuring out when such abstract underlying representations are formed, and what they are like. This is evident when comparing the English PL voicing alternation (e.g. 'cats' [kæts] ~ 'dogs' [dɑgz]) to the Dutch PL voicing alternation (e.g. 'bed' [bɛt] ~ 'beds' [bɛdən]) (Kerkhoff, 2007, p. 1). English speakers show clear productive, rule-like behavior (Berko, 1958), while Dutch speakers' generalization is less-clearly rule-like (Ernestus & Baayen, 2003; Kerkhoff, 2007). The Dutch alternation is obfuscated by its interaction with other voicing alternations, such as assimilation (Buckler & Fikkert, 2016, sec. 2). Consequently, it may be that the English alternation is systematic enough to drive the learner to systematic underlying abstraction, while the Dutch alternation is not.

Thus, a complete theory of phonological learning must include, in addition to the mechanism by which processes are learned, a precise mechanism characterizing how and when abstract underlying forms are posited. For example, Richter (2018, 2021) has hypothesized a mechanism by which learners abandon the null hypothesis of concrete underlying forms in favor of abstraction, and applied it to the case of the English [t] / [ɾ] allophones. The results closely matched lexical studies of child utterances, including a U-shaped development curve. Thus, PLP is just one part of the story. However, we believe that this part of the story—learning phonological processes from (UR, SR) pairs—is nevertheless important, and in line with the vast majority of prior work on learning phonological grammars, which have likewise tended to presuppose abstract underlying forms for use in, for example, constraint ranking (Tesar & Smolensky, 1998; Legendre *et al.*, 1990; Boersma, 1997; Smolensky & Legendre, 2006; Boersma & Hayes, 2001; Boersma & Pater, 2008).[2]

1.3. Locality and Identity as Principles of Computational Efficiency

Locality and identity have natural interpretations as principles of computational efficiency, or "third factors" (Chomsky, 2005; Yang *et al.*, 2017). The more local the context around an underlying segment, the fewer segments the cognitive system must be sensitive to (Rogers *et al.*, 2013, p. 99) in determining its output. Moreover, it is computationally simpler to copy input segments *unaltered* to the output than to change them in the process.

We present our proposed model in § 2, discuss prior models in § 3, evaluate the model in § 4, and conclude with a discussion in § 5.

2. Model: PLP

Our proposed model is called PLP for *Parsimonius Local Phonology* learner. PLP learns from an input of (UR, SR) pairs, which may grow over time as the learner's

[2] One reviewer pointed out that the concept of underlying forms faces skepticism, and that many phonologists have rejected the concept all together. We acknowledge that the view of learning described here is not uncontroversial. Hyman (2018) provides a discussion of the merits of underlying representations.

vocabulary expands. It constructs the generalizations necessary to account for which segments surface unfaithfully in those pairs and in what phonological contexts that happens. These generalizations are placed in a grammar, for use in producing output SRs for input URs.

(3)

```
Input: (UR, SR) pairs
1. Initialize an empty grammar $G$ and empty vocabulary $\mathcal{V}$
2. While there are more pairs $(u, s)$ to learn from do
3.   – Update $\mathcal{V}$ with $(u, s)$
4.   – Use $G$ to predict surface representation $\hat{s}$ for underlying $u$        (§ 2.3.4)
5.   – For each discrepancy between $u$ and $s$ not accounted for in $\hat{s}$ do  (§ 2.1)
6.       — Construct a generalization $g$ for the discrepancy                       (§ 2.2)
7.       — Encode $g$ in $G$                                                         (§ 2.3)
8.   – Update any generalizations that now overextend due to $\mathcal{V}$ growth   (§ 2.4)
```

PLP assumes identity between URs and SRs by default via the fact that it only adds generalizations to G at steps 6-7, when discrepancies arise. A locality preference emerges via the generalization strategy it employs in steps 6 and 8: PLP starts with the narrowest context around an unfaithfully-surfacing segment and proceeds further from the segment only when an adequate generalization can not be found. Consequently, we consider steps 6 and 8, together with the addition of generalizations to the grammar only when motivated by discrepancies, to be PLP's main contributions. The code is available on Github.[3]

2.1. The Input

The input to PLP is set of (UR, SR) pairs, which may grow over time, simulating the learner's vocabulary growth. As discussed in § 1.2, discrepancies between a UR and its corresponding SR arise when a learner abandons concrete underlying representations in favor of underlying abstraction. A discrepancy can be an input segment that does not surface (deletion), an output segment that has no input correspondent (epenthesis), or an input segment with a non-identical output correspondant (segment change). In this work, we treat the (UR, SR) pairs, with discrepancies present, as PLP's input. Future work will combine this with the important problem of when abstract underlying forms are posited (e.g., Richter 2018). We also assume that the correspondence between input and output segments is known. The same assumption is tacit in constraint ranking models, which use the correspondence for computing faithfulness constraint violations.

The URs and SRs are sequences of segments, which we treat as sets of distinctive features (Jakobson & Halle, 1956; Chomsky & Halle, 1968). Thus, structuring sound into a phonological segment inventory organized by distinctive features is treated as a separate learning process (e.g. Mayer 2020). We use feature assignments from Mortensen *et al.* (2016).

We will use the English plural allomorph as a running example. Suppose that at an early stage in acquisition, a child has memorized some of the plural forms of nouns in their vocabulary, as shown in (4).

[3] https://github.com/cbelth/PLP

(4) /dɑgz/
/kæts/
/hɔrsəz/
⋮

At this stage, an empty grammar, which regurgitates each memorized word, will suffice. Moreover, since no discrepancies yet exist, PLP will be content with this empty grammar: the for loop (3; step 5) will not be entered. As the child begins to learn morphology, they may discover the morphological generalization that plurals tend to be formed by suffixing a /-z/. All of the child's plural URs will then, in effect, be reorganized as in (5).

(5) /dɑg-z/
/kæt-z/
/hɔrs-z/
⋮

At this point, when the child goes to use their grammar (3; step 4), they will discover that it now predicts *[kætz] and *[hɔrsz], inconsistent with their expectation based on prior experience with the words. The newly-introduced discrepancies trigger the for loop (3; step 5) and require PLP to provide an explanation for them. Suppose the first word to trigger this is /kæt-z/, erroneously predicted as *[kætz] instead of the expected [kæts]. PLP then constructs a generalization to capture the phonological context in which /z/ surfaces as [s] (3; step 6).

2.2. Constructing Generalizations

The core component of PLP is its component for constructing generalizations (3; step 6).

2.2.1. The Structure of Generalizations

The generalizations that PLP constructs are pairs $g = (\bar{s}, a) \in \mathcal{S} \times \mathcal{A}$, where $\bar{s} \in \mathcal{S}$ (6) is a sequence and $a \in \mathcal{A}$ (7) is an action carried out at a particular position in the sequence. Each element in a sequence is a set of segments from the learner's segment inventory, Σ (6).[4]

(6) $\mathcal{S} \triangleq \bigcup_{k=1}^{\infty} \{s_1 s_2 \ldots s_k : s_i \subset \Sigma\}$

A set of segments may be extensional, e.g., $s_i = \{$s, ʃ, z, ʒ$\}$, or a natural class—e.g., $s_i =$ [+sib]. An action can be any in (7): deletion of the ith segment, insertion of new segment(s) to the right of the ith segment,[5] or setting the ith segment's feature f to '+' or '−'.[6]

[4] We allow these elements to also contain syllable/word-boundary information, which we implement following Chomsky & Halle (1968); Hayes & Wilson (2008) by introducing a [±segment] feature and corresponding −segment element in Σ to mark boundaries.

[5] Insertion in initial position is achieved with $i = 0$.

[6] More generally, we can treat the first parameter as a vector of features and the second as a vector of ± values to capture multiple feature changes, but for simplicity we only describe the case of a single feature change.

(7) $\mathcal{A} \triangleq \{\textsc{Del}(i), \textsc{Ins}(s_{\text{new}}, i), \textsc{Set}(f, \pm, i)\}$

For example, the generalization (8a) states that a consonant is deleted when it follows and precedes other consonants, (8b) states that a ‘ə’ is inserted to the right of any sibilant that precedes another sibilant, and (8c) states that the voicing feature of voiced obstruents in syllable final position is set to ‘−’ (we use ‘$]_\sigma$’ $\in \Sigma$ to mark syllable boundary).

(8) a. ([+cons][+cons][+cons], $\textsc{Del}(2)$)

b. ([+sib][+sib], $\textsc{Ins}$(‘ə’, 1))

c. ([+voi,−son]$]_\sigma$, $\textsc{Set}$(voi, ‘−’, 1))

Any grammatical formalism capable of encoding these generalizations could be used, but in this paper we chose a rule-based grammar. The specified set of possible actions is meant to cover a majority of phonological processes, but more could be added if necessary (e.g. metathesis).

The part of the sequence picked out by the action’s index i determines the rule’s target, and the part of the sequence to the left and right of i determine the rule’s left and right contexts. Each type of action (7) can be encoded in one of the rule-schemas in (9), where $k = |\bar{s}|$.

(9)

$\textsc{Del}(i)$	$s_i \rightarrow \emptyset \;/\; s_1 \ldots s_{i-1} \_\_ \; s_{i+1} \ldots s_k$
$\textsc{Ins}(s_{\text{new}}, i)$	$\emptyset \rightarrow s_{\text{new}} \;/\; s_1 \ldots s_i \_\_ \; s_{i+1} \ldots s_k$
$\textsc{Set}(f, \text{‘+’}, i)$	$s_i \rightarrow [+f] \;/\; s_1 \ldots s_{i-1} \_\_ \; s_{i+1} \ldots s_k$
$\textsc{Set}(f, \text{‘−’}, i)$	$s_i \rightarrow [-f] \;/\; s_1 \ldots s_{i-1} \_\_ \; s_{i+1} \ldots s_k$

Thus, the generalizations in (8) are encoded as the rules in (10).

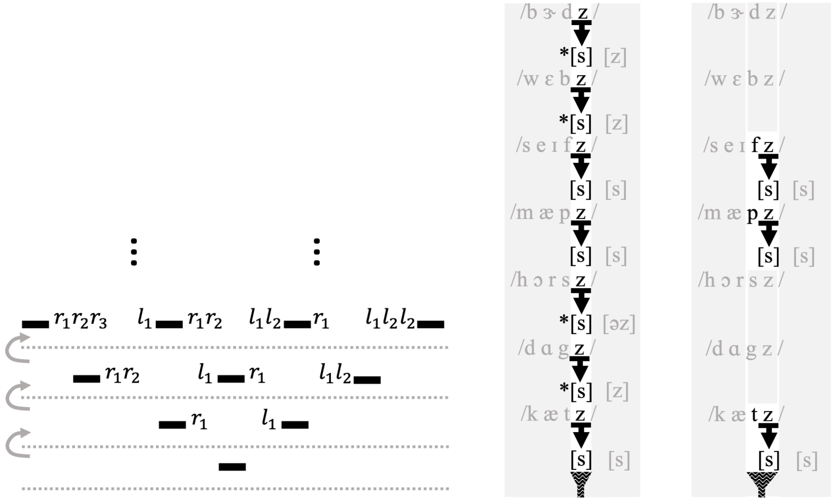
(10) a. [+cons] → $\emptyset$ / [+cons] __ [+cons]

b. $\emptyset$ → ə / [+sib] __ [+sib]

c. [+voi, −son] → [−voi] / __ $]_\sigma$

Each sequence in $\mathcal{S}$ is *strictly local* (McNaughton & Papert, 1971)—describing a contiguous sequence of segments (cf. § A.1)—and has the same structure as the ‘sequence of feature matrices’ constraints from Hayes & Wilson (2008, p. 391). Moreover, the input-output relations described by each generalization are probably[7] *input strictly local* maps (Chandlee, 2014). These structures are not necessarily capable of capturing *all* phonological generalizations, and intentionally so. Typological considerations point to strict locality as a central property of generalizations, due to its prevalence (Chandlee, 2014) and repeated occurrence across representations (Heinz *et al.*, 2011). This paper is intentionally targeting precisely those generalizations, and we discuss principled extensions for non-local generalizations in § 5.2.

[7] It is generally believed that processes describeable with the types of rules that PLP constructs are input strictly local maps (Chandlee, 2014), but—to our knowledge—there does not exist a published proof of this fact. See § A.1 for more.



(a) (b) (c)

Figure 1: **(a)**: The width of PLP's search is expanded (upward arrows) when and only when an adequate generalization cannot be found in virtue of a less-wide context. **(b)**-**(c)**: An example of PLP's search: the first generalization (13) fails because it makes too many wrong predictions, but the second (15a) allows the /z/ ⟶ [s] instances to be isolated.

2.2.2. Searching Generalizations

When PLP encounters a discrepancy—an input segment surfacing unfaithfully—it uses algorithm (11) to construct a generalization $g = (\bar{s}, a)$. We refer to the discrepancy as $x \longrightarrow y$, where x is the input segment and $y \neq x$ is what it surfaced as.

```
(11)  Input: A discrepancy, $x \longrightarrow y$, and the current training vocabulary $\mathcal{V}$
      1. Initialize a window $\bar{w} = [\{x\}]$ of width one
      2. Infer $a$ from $x \longrightarrow y$ and initialize a generalization $g = (\bar{w}, a)$
      3. While $g$ is insufficiently accurate over $\mathcal{V}$ do[8]                (§ 2.2.3)
      4.   – Expand the width of the window by length one                      (§ 2.2.4)
      5.   – Set $g$'s sequence $\bar{s}$ to the most accurate context around $x$ that fits in $\bar{w}$
                                                                                (§ 2.2.4)
```

PLP uses a window, $\bar{w}$, to control the breadth of its search. The window is a sequence of cells that can be filled in to create g's sequence (6). The window starts with only one cell, filled with $s_0 = \{x\}$ (11; step 1). PLP then infers the type of change from $x \longrightarrow y$ (12).

[8] The loop also exits if the search runs out of context, in which case no sufficiently accurate generalization is possible.

(12)

$$a = \begin{cases} \textsc{Del} & \text{if } x \rightarrow \emptyset \ \ (y = \emptyset) \\ \textsc{Ins} & \text{if } \emptyset \rightarrow y \ \ (x = \emptyset) \\ \textsc{Set} & \text{otherwise} \end{cases}$$

For Ins, the value inserted (s_{new}) is y; for Set, the featured changed (f) and its value ($\pm$) are inferred from the difference between x and y. The index, i, specifies where x falls in g's sequence, $\bar{s}$; initially since $\bar{s} = \bar{w} = [\{x\}]$, $i = 1$ (11; step 2).

As Fig. 1a visualizes, PLP starts with the most local generalization, which makes no reference to the segment's context: the segment always surfaces unfaithfully. In the running example, PLP first posits (13), which predicts that /z/ always surfaces as [s] (Fig. 1b).

(13) /z/ → [−voi] / __

This, however, is contradicted by other words in the vocabulary: /z/ surfaces faithfully as [z] in words like [dɑgz] and with an epenthetic vowel in words like [hɔrsəz], which suggests that this initial generalization is wrong (11; step 3) and that the breadth of the search must be expanded (Fig. 1a).

2.2.3. When to Expand Breadth of Search

To come to such a verdict, PLP computes the number of predictions the rule makes over the current vocabulary and how many of those were correct. The number of predictions (13) makes is the number of times /z/ appears in the learner's vocabulary, and those that surface as [s] are the correct predictions. There are a number of options for determining the adequacy of the generalization. We could require a perfect prediction record, but this may be too rigid due to the near inevitability of exceptions in naturalistic data. More generally we could place a threshold on the number or fraction of errors that the generalization can make. The choice of criterion does not substantially change PLP: proceeding from local generalizations to less-local generalizations proceeds in the same way regardless of the quality criterion, which simply determines the rate at which the more local generalizations are abandoned. In this work, PLP uses the Tolerance Principle (Yang, 2016) as the threshold, which states that a generalization making N predictions about what an underlying segment surfaces as is productive—and hence the while loop (11; step 3) can be exited—if and only if the number of incorrect predictions it makes (called e for *exceptions*) satisfies (14).

(14)

$$e \leq \frac{N}{\ln N}$$

The threshold is cognitively motivated, predicting that children accept a linguistic generalization when it is cognitively more efficient to do so (see Yang 2016, ch. 3 for the threshold's derivation). Since the threshold is based on cognitive considerations and has had success in prior work (e.g. Schuler *et al.* 2016; Koulaguina & Shi 2019; Emond & Shi 2021; Richter 2021; Belth *et al.* 2021), it is a reasonable choice for this paper. In the current example, (13) has $N = 7$ and $e = 4$, which fails to pass (14): $4 > 7/\ln 7$. Thus, the while loop (11; step 3) is entered.

2.2.4. Expanding Breadth of Search

Once the initial hypothesis that /z/ always surfaces as [s] is ruled out as too errant, PLP adds one cell to the window (11; step 4). PLP fills the window with the sequence that matches the fewest of the sequences where /z/ does not surface as [s]. In other words, it chooses the context that better separates words like /kæt/ from words like /dɑg/ and /hɔrs/. Thus, for the vocabulary in Fig. 1b-1c, PLP prefers (15a) over (15b)[9] because a left context of {t, p, f} better separates the places where /z/ does indeed surface as [s] from those where it does not, than does a right context of {#}. That is, PLP chooses the rule with the most accurate context fitting in the current window, where accuracy is measured as the fraction of the rule's predictions over the training URs that match the corresponding training SRs. In our example then, PLP's second hypothesis is that /z/ surfaces as [s] whenever it follows a /t/, /p/, or /f/.

(15) a. /z/ → [−voi] / {t, p, f} __
b. /z/ → [−voi] / __ {#}

Figure 1a visualizes PLP's search: it hypothesizes a context where an underlying segment surfaces as some particular segment other than itself, checking whether the hypothesis is satisfactorily accurate, and expanding the breadth of its search if not. This process halts once a sufficiently accurate hypothesis has been discovered.

2.3. Encoding Generalizations in a Grammar

The generalizations that PLP constructs are encoded in a grammar to be used in producing an SR for an input UR. The grammar, *G*, consists of a list of rules. Each time PLP constructs a generalization (3; step 6), it is placed in the appropriate rule schema (9) and added to the list of rules. If PLP replaces a generalization due to underextension or overextension (3; step 8), as described in § 2.4, the old, offending rule is removed and a new one added. § 2.3.1 discusses how rules that carry out the same action are combined, § 2.3.3 discusses how the list of rules is ordered, § 2.3.2 discusses how natural classes are induced, and § 2.3.4 discusses how *G* produces outputs from inputs.

2.3.1. Combining Generalizations

Generalizations that carry out the same change over different segments are combined in the grammar, so long as the resulting rule is satisfactorily accurate, via (14). For instance, the three rules in (16a) would be grouped into the single rule (16b).

(16) a. /d/ → [−voi] / __ $]_\sigma$
/v/ → [−voi] / __ $]_\sigma$
/g/ → [−voi] / __ $]_\sigma$
b. {d, v, g} → [−voi] / __ $]_\sigma$

2.3.2. Inducing Natural Classes

Up to this point, PLP's generalizations have been over sets of particular segments. Humans appear to generalize from individual segments to natural classes, as has been

[9] The symbol '#' denotes a word boundary.

recognized by theory (Chomsky & Halle, 1965; Halle, 1978; Albright, 2009) and evidenced by experiment (Berent & Lennertz, 2007; Finley & Badecker, 2009; Berent, 2013).

PLP thus attempts to generalize to natural classes for each set of segments in a generalization's sequence $\bar{s}$, in terms of shared distinctive features (Jakobson & Halle, 1956; Chomsky & Halle, 1968). The procedure can be thought of as retaining only the features shared by segments in $\bar{s}$ needed to keep the rule satisfactorily accurate. To exemplify this part of the model, we will assume PLP has constructed the epenthesis rule (17)—e.g., /hɔrsz/ → [hɔrsəz].

(17) $\emptyset \rightarrow$ ə / {s, ʃ, z} __ {z}

The procedure, outlined in (18), starts with a new length-$|\bar{s}|$ sequence $\bar{n}$, with each element an empty natural class (18; step 1).

```
(18)  Input: A generalization $g = (\bar{s}, a)$
      1. Initialize a new generalization $g_{\text{nc}} = (\bar{n}, a)$ with empty natural classes, $\bar{n}$
      2. Initialize feature options for natural classes
      3. While $g_{\text{nc}}$ is insufficiently accurate over $\mathcal{V}$ do
      4. – Add to $\bar{n}$ the feature that best narrows $\bar{n}$'s extension down to $\bar{s}$'s
      5. Replace $g$ with $g_{\text{nc}}$
```

For generalization (17), the sequence $\bar{s}$ is (19a) and the (empty) initial natural class sequence is (19b). Each element of $\bar{n}$ can take any feature shared by the corresponding segments in $\bar{s}$, so the set of feature options is (19c), which includes elements like (+sib, 1) because {s, ʃ, z} share '+sib' as a feature and (+voi, 2) because {z} has '+voi' as a feature, but it does not include (+voi, 1) because {s, ʃ, z} do not agree on this feature.

(19) a. $\bar{s}$ = {s, ʃ, z}{z}

b. $\bar{n}$ = [] []

c. {(+cons, 1), (+sib, 1), (−son, 1), ..., } ∪ {(+sib, 2), (+voi, 2) ...}

Inside the while loop (18; step 3), features are added one at a time to $\bar{n}$, choosing at each step the feature from (19c) that best narrows the extension of $\bar{n}$ (initially all length-$|\bar{s}|$ sequences) to those in the extension of $\bar{s}$ (which is {sz, ʃz, zz}). Thus, adding the feature '+sib' to the first natural class (20a) will narrow $\bar{n}$'s extension towards $\bar{s}$'s better than '+cons.' The new generalization, g_{nc} is evaluated as before with Tolerance Principle, via (14). In the current example, $\bar{n}$ (20a) will still have sequences like {st, zi, ʃu, ...} in its extension, so '+sib' will then be added to the second natural class (20b).

(20) a. $\bar{n}$ = [+sib][]

b. $\bar{n}$ = [+sib][+sib]

This new sequence, $\bar{n}$, still has an extension greater than the original $\bar{s}$. However, because adjacent sibilants are indeed disallowed in English, this inductive leap is possible, and thus (17) will be replaced with (21) in the grammar.

(21) $\emptyset \rightarrow$ ə / [+sib] __ [+sib]

This differs from the natural class induction in Albright & Hayes (2002, 2003), which generalizes as conservatively as possible by retaining all shared features (see § A.2.4).

It may be possible for natural class induction to influence rule-ordering, so PLP induces them prior to rule ordering. Specifically, natural classes are induced with rules temporarily ordered by scope (narrowest first), before the final ordering is computed as in § 2.3.3.

2.3.3. Rule Ordering

In some cases, phonological processes may interact, in which case the interacting rules may need to be ordered. The topic of rule interaction and ordering has received immense attention in the literature—especially in discussions of opacity—and is well-beyond the scope of the current paper to fully take up here. However, we will summarize PLP's approach to rule ordering, and characterize the path to a more systematic study of PLP's handling of complex rule interactions.

The standard rule interactions discussed in the literature are Feeding, Bleeding, Counterfeeding, and Counterbleeding, described in (22) following McCarthy (2007); Baković (2011).

(22) Given two rules r_i and r_j, where r_i precedes r_j,
   a. r_i Feeds r_j iff r_i creates additional inputs to r_j
   b. r_i Bleeds r_j iff r_i destroys potential inputs to r_j
   c. r_j Counterfeeds r_i iff r_j creates additional inputs to r_i
   d. r_j Counterbleeds r_i iff r_j destroys additional inputs to r_i

Counterfeeding and Counterbleeding are *counterfactual inverses* of Feeding and Bleeding: if r_j Counterfeeds (resp. Counterbleeds) r_i, it would Feed (resp. Bleed) r_i if it preceded r_i. McCarthy (2007, sec. 5.3)'s example of Feeding, reproduced in (23), comes from Classical Arabic, where vowel epenthesis before word-initial consonant clusters (r_i) feeds [ʔ] epenthesis before syllable-initial vowels (r_j).

(23) /dˤrib/ (underlying) →
idˤrib (vowel epenthesis) →
ʔidˤrib ([ʔ] epenthesis) →
[ʔidˤrib] (surface) 'beat! Masc.Sg.'

McCarthy (2007, sec 5.4) also provides an example of Counterfeeding. In Bedouin Arabic, short high vowels are deleted in non-final open syllables, and /a/ is raised in the same environment. However, as (24) shows, because deletion precedes raising, the raising of the short vowel /a/ to [i] does not feed deletion.

(24) /dafaʔ/ (underlying) →
dafaʔ (no deletion) →
difaʔ (raising) →
[difaʔ] (surface) 'he pushed'

Examples of Bleeding and Counterbleeding come from dialects of English where /t/ and /d/ are flapped—[ɾ]—between stressed and unstressed vowels, while

/aɪ/ and /aʊ/ raise to [ʌɪ] and [ʌʊ] before voiceless segments. The canonical case is Counterbleeding order, where raising occurs before underlying /t/ even when it surfaces as voiced [ɾ] on the surface (25).

(25) /raɪtɚ/ (underlying) →
rʌɪtɚ (raising) →
rʌɪɾɚ (flapping) →
[rʌɪɾɚ] (surface)

In lesser-discussed dialects of English in Ontario, Canada (Joos, 1942) and in Fort Wayne, IN (Berkson *et al.*, 2017), the flapping of voiceless /t/ as voiced [ɾ] bleeds raising (26).

(26) /raɪtɚ/ (underlying) →
raɪɾɚ (flapping) →
raɪɾɚ (/aɪ/ raising does not apply due to voiced 'ɾ') →
[raɪɾɚ] (surface)

Given two interacting rules r_i and r_j, it is straight-forward to order them by following standard arguments. Specifically, ordering r_i before r_j (Feeding/Bleeding order), will produce errors on data from a language where r_j in fact precedes r_i (Counterfeeding/Counterbleeding) and vice versa. For example, if we call English dialects where flapping counterbleeds raising (25) 'Dialect A' and the dialects with bleeding (26) 'Dialect B', ordering flapping before raising in 'Dialect A' will erroneously cause /raɪtɚ/ to surface as [raɪɾɚ] instead of [rʌɪɾɚ]. Consequently, the correct Counterfeeding order will yield higher accuracy than Feeding order for a leaner exposed to 'Dialect A.' A symmetrical argument holds for ordering in 'Dialect B.'

Thus, for each pair of learned rules, PLP chooses the pairwise ordering with higher accuracy. To yield a full ordering of the rules, PLP constructs a directed graph where each rule in $\mathcal{R}$ forms a node. PLP considers each pair of rules $(r_i, r_j) \in \mathcal{R} \times \mathcal{R}$ and places a directed edge from r_i to r_j iff the accuracy of $r_j \circ r_i$ (i.e., applying r_i first and r_j to its output) is greater than the reverse, $r_i \circ r_j$. The directed graph is then topologically sorted[10] to yield a full ordering. In such an ordering, the ordering between any pair of rules that do not interact is arbitrary, while that between any pair that do interact is the order that achieves higher accuracy.

The bigger challenge is the possibility that the interactions between r_i and r_j obfuscate the independent existence of the rules, thereby making it difficult for them to be discovered in the first place. Counterfeeding and Counterbleeding present no issues, because applying each rule independently, directly over the UR, produces the same SR as applying them sequentially in Counterfeeding/Counterbleeding order. For example, in McCarthy (2007)'s Bedouin Arabic example (24) /a/ → [i] is accounted for by the raising rule, and there is no deletion in /dafaʔ/ → [difaʔ] to hinder the discovery of the deletion rule. Similarly, the /a/ → [ʌ] discrepancy in (25) can be accounted for by raising, without reference to flapping, and the /t/ → [ɾ] discrepancy can be accounted for by flapping without reference to raising. We give an empirical demonstration of PLP learning rules in Counterbleeding order in § 4.3.4.

[10] A *topological sort* of a directed graph is a linear ordering of its nodes such that every ordering requirement encoded in its edges is preserved (Cormen *et al.*, 2009, p. 612).

Since Bleeding destroys contexts where a rule would have applied, it can cause overextensions. For example, when PLP is attempting to construct a raising rule for (26), rule (27) (treating the diphthong as a single segment) would overextend to /raɪtɚ/.

(27) aɪ → ʌɪ / __ [−voi]

However, since PLP allows some exceptions via the Tolerance Principle, this will only matter if the bled cases are pervasive enough to push the rule over the Tolerance Principle threshold (Eq. 14). Whether this happens must be determined on a case-by-case basis by the learner's lexicon. If the threshold of exceptions is crossed, PLP will simply expand the width of its search. When flapping bleeds raising (26), raising occurs distributionally before underlying voiceless segments *that are not between a stressed and an unstressed vowel*. The latter condition describes the contexts where raising is not bled, and still falls within a fixed-size window of the raising target, as shown with underlines in /raɪtɚ/. The general point here is that if two rules interact extensively, there is likely to still be a fixed-length context—possibly a slightly larger context—that accounts for the processes. In fact, Chandlee *et al.* (2018) showed that a wide-range of phonological generalizations that have been characterized as opaque in the literature can be characterized as Input Strictly Local maps. In the appendix, we show that the rules PLP learns correspond to Input Strictly Local maps. Thus, we are optimistic that PLP can succeed even with instances of opaque rule interactions. § 4.4 provides an empirical demonstration of PLP learning rules in Bleeding order.

Feeding may require small adaptions to PLP. In (23), no issue arises for the vowel-epenthesis rule, which does the feeding. The search for a rule to account for epenthetic [ʔ] will proceed analogously to the Bleeding case. There are two underlying environments where epenthetic [ʔ] surfaces: before underlyingly initial vowels (# __ V) and before underlying initial consonant clusters (i.e. where raising feeds it, # __ CC). These are disjoint contexts, so it may be appropriate to adapt PLP to allow it to return two disjoint rules from its search (11) to account for a discrepancy. In that case, the rules in (28) account for [ʔ]-epenthesis directly from URs.

(28) ∅ → ʔ / # __ CC (‘fed’ [ʔ]-epenthesis cases)
∅ → ʔ / # __ V

Alternatively, PLP could be adapted such that the search for new generalizations (3; step 6) operates over intermediate representations—specifically those derived by existing rules—instead of underlying representations. In that case, the [ʔ]-epenthesis rule could be directly learned over the intermediate forms derived by the vowel-epenthesis rule.

In summary, this paper is not an attempt to provide a complete account of rule ordering, which is beyond its scope. The results in § 4.3.4 and § 4.4 provide empirical demonstration of PLP learning some interacting rules, and the above discussion provides an outline of how PLP approaches rule interaction and what extensions may be necessary.

2.3.4. Production

The rules are applied one after the other in the order produced by the procedure in § 2.3.3. Each individual rule is interpreted under *simultaneous application* (Chomsky

& Halle, 1968), which means that when matching the rule's target and context, only the input is accessible, not the result of previous applications of the rule. Thus, following the example from Chandlee (2014, p. 37), the rule (29) applied simultaneously to the input string *aaaa* yields the output *abba* rather than *abaa*, because the second application's context is not obscured by the the first application.

(29) $a \to b \,/\, a \_\_\; a$

Simultaneous application is the interpretation of rules that corresponds to input-strictly local maps, as we discuss in § A.1.2. Other types of rule application, such as iterative or directional (e.g., Howard 1972; Kenstowicz & Kisseberth 1979), could be used in future work.

Thus, for an input u and ordered list of rules $\mathcal{R} = r_1, r_2, ..., r_{|\mathcal{R}|}$, the grammar's output $\hat{s}$ is given by the composition of rules in (30).

(30) $\hat{s} = G(u) = r_{|\mathcal{R}|} \circ r_{|\mathcal{R}|-1} \circ ... \circ r_1(u) = r_{|\mathcal{R}|}(r_{|\mathcal{R}|-1}(...r_1(u)))$

2.4. Updating Incrementally

As PLP proceeds, vocabulary growth may cause the grammar to become stale and underextend or overextend, at which point PLP updates any problematic generalizations (3; step 8).

Denoting the discrepancies between the input u and the predicted output $\hat{s}$ as $d(u, \hat{s})$, and those between u and s as $d(u, s)$, underextensions are defined in (31a) as discrepancies between the input and expected output that are not accounted for in PLP's prediction $\hat{s}$, and overextensions are defined in (31b) as discrepancies in the predicted output that should not be there. Here '\' denotes set difference, and '$\triangleq$' means 'equal by definition.'

(31) a. $U \triangleq d(u, s) \setminus d(u, \hat{s})$
b. $O \triangleq d(u, \hat{s}) \setminus d(u, s)$

Underextensions are handled by the for loop (3; step 5). Inside the loop, a new generalization is created (3; step 6). This is encoded in the grammar (3; step 7) by adding it to this list of rules. If a prior generalization for the discrepancy exists, it is deleted from the list. An example of this is (32), where the word /mæp-z/ (32b) freshly enters the vocabulary.

(32) a. /dɑg-z/ → [dɑgz]
/kæt-z/ → [kæts]
/hɔrs-z/ → [hɔrsəz]
b. /mæp-z/ → [mæps]

Prior to its arrival, the rule (33a) was sufficient to explain when /z/ surfaces as [s]. This, however, fails to account for the new word, which ends in /p/ not /t/. PLP handles this by discarding the old rule and replacing it with a fresh one, such as (33b), derived by the exact same process described above in § 2.2.

(33) a. /z/ → [−voi] / {t} __

b. /z/ → [−voi] / {t, p} __

Overextension—a discrepancy between the input u and PLP's prediction $\hat{s}$ that did not exist between u and the expected output s—is handled by (3; step 8). An example is (34), where (34b) enters the learner's vocabulary after (34a).

(34) a. /kæt-z/ → [kæts]

b. /dɑg-z/ → [dɑgz]

In such a case, the rule (35) will have been sufficient to explain (34a), but will result in an erroneous *[dɑgs] for (34b).

(35) /z/ → [−voi] / __

PLP resolves this by discarding the previous rule and replacing it with a new one via the process in § 2.2.

For both underextension and overextension, when the list of rules is updated, the steps in § 2.3—combining generalizations, inducing natural classes, and ordering rules—are repeated. Since PLP can replace generalizations as needed as the vocabulary grows, it can learn incrementally, in batches, or once and for all over a fixed vocabulary.

3. Prior Models

3.1. Constraint-Based Models

Constraint-ranking models rank a provided set of constraints. Tesar & Smolensky (1998)'s Constraint Demotion algorithm was an early constraint-ranking model for OT. Others are built on stochastic variants of OT or Harmonic Grammar (HG) (Legendre *et al.*, 1990; Smolensky & Legendre, 2006), including the Gradual Learning Algorithm (Boersma, 1997; Boersma & Hayes, 2001) for Stochastic OT and a later model (Boersma & Pater, 2008) that provided a different update rule for HG (see Jarosz 2019 for an overview).

Constraint ranking models can capture the assumption of classical Optimality Theory that learning amounts to ranking a universal constraint set, or they can rank a learned constraint set. Hayes & Wilson (2008)'s Maximum Entropy model learns and ranks constraints, but it learns phonotactic constraints over surface forms, not alternations as PLP does.

Locality and identity biases are better reflected in the content of the constraint set than in the constraint ranking algorithm. Locality is determined in virtue of what segments are accessed in determining constraint violations.

Constraint ranking models usually initialize markedness constraints outranking faithfulness constraints (Smolensky, 1996; Tesar & Smolensky, 1998; Jusczyk *et al.*, 2002; Gnanadesikan, 2004). Consequently, any UR will initially undergo any changes necessary to avoid marked structures, even when lacking surface alternation to motivate discrepancies. Ranking faithfulness constraints above markedness constraints has been advocated for by Hale & Reiss (2008), but this approach has not been widely adopted. This in part due to arguments that such an initial ranking would render some grammars unlearnable (Smolensky, 1996), and in part due to the view that features of early child productions, in particular ‘emergence of the unmarked,’ reflect an early stage of the child’s grammar, rather than underdeveloped articulatory control.

3.2. Rule-Based, Neural Network, and Linear Discriminative Models

Johnson (1984) proposed an algorithm for learning ordered rules from words arranged in paradigms as a proof of concept about the learnability of ordered-rule systems. The algorithm did not incorporate a locality bias and was not extensively studied empirically or theoretically.

Albright & Hayes (2002, 2003) developed a model for learning English past tense morphology via probabilistic rules. The model can be applied to learn rules for any set of input-output word pairs, including phonological rules. It is called the *Minimum Generalization Learner* because when it seeks to combine rules constructed for multiple input-output pairs, it forms the merged rule that most tightly fits the pairs. A consequence of this generalization strategy is that the phonological context of the rule is as wide as possible around the target segment, only localizing around the target when less-local (and hence less-general) contexts cannot be sustained. This is the direct opposite of PLP and of experimental results that suggest human learners start with local patterns and only move to non-local patterns when local generalizations cannot be sustained (Finley, 2011; Baer-Henney & van de Vijver, 2012; McMullin & Hansson, 2019). We further discuss differences between PLP and MGL in § A.2.

Rasin *et al.* (2018) proposed a Minimum Description Length model for learning optional rules and opacity. The authors intended the model as a proof-of-concept and only evaluated it on two small, artificial datasets.

Peperkamp *et al.* (2006) proposed a statistical model for learning allophonic rules by finding segments with near-complementary distributions. The method is not applicable to learning rules involving non-complementary distributions. Calamaro & Jarosz (2015) extend the model to handle some cases of non-complementary distributions, if the alternation is conditioned in terms of the following segment (i.e., $a \rightarrow b / \_\_\ c$ where $|a| = |b| = |c| = 1$). These works attempt to model the very early stage of learning alternations (White *et al.*, 2008) prior to most morphological learning, whereas PLP models learning after abstract URs begin to be learned.

Beguš (2022) trained a generative, convolutional neural network on audio recordings of English-like nonce words, which followed local phonological processes and a non-local process (vowel harmony). The model was then used to generate speech. This model-generated speech followed the local processes more frequently than the non-local process, suggesting that it more easily learned local than non-local processes. This is possibly due to the use of convolution, which is a fundamentally local operation. As a model for generating artificial speech, it is not directly comparable in our setting of learning processes that map URs to SRs.

In a different direction, Baayen *et al.* (2018, 2019) proposed using Linear Discriminative Learning to map vector representations of form onto vector representations of meaning and vice versa. Since this model operates over vector representations of form and meaning, it is not directly comparable.

3.3. Formal-Language-Theoretic Models

Formal-language and automota-theoretic approaches analyze phonological generalizations in computational terms. Many resulting learning models attempt to induce a finite state transducer (FST) representation of the map between SRs and URs. These automata-theoretic models, together with precise assumptions about the data available for learning, allow for learnability results in the Gold (1967) paradigm of *identification in the limit*. Such results state that a learning algorithm will converge onto a correct FST representation of any function from a particular family, provided that the data presented to it meets certain requirements—called a *characteristic sample*. In phonology, the target class of functions is usually one that falls in the subregular hierarchy (Rogers *et al.*, 2013), which contains classes of functions more restrictive than the *regular* region of the Chomsky Hierarchy (Chomsky, 1956). These models are often chosen to demonstrate theoretical learnability results, and have seldom been applied to naturalistic data.

Gildea & Jurafsky (1996) developed a model, based on OSTIA (Oncina *et al.*, 1993), which learns subsequential FSTs. The class of subsequential functions is a sub-regular class of functions that may be expressive enough to capture any type of observed phonological map (Heinz, 2018), although some tonal patterns appear to be strong counter-examples (Jardine, 2016). The authors intended their model only as a proof-of-concept of the role of learning biases, and required unrealistic quantities of data to effectively learn. Indeed, they recognized the importance of faithfulness and locality as learning biases, which they attempted to embed into OSTIA. Their biases were, however, heuristics. In particular, a bias for locality was introduced by augmenting states with the features of their neighboring contexts. This in effect restricts the learner to local patterns, which is different from the current paper's proposal, in which locality is a consequence of the way that the algorithm proceeds over hypotheses.

As Chandlee (2014) observes, a more principled means of incorporating a locality bias into a finite state model is to directly target the class of strictly local functions. Chandlee *et al.* (2014) proposes such a model, called ISLFLA, and proves that it can learn any strictly local function in the limit in the sense of Gold (1967). However, the characteristic sample for the algorithm includes the set of input-output pairs for every language-theoretically possible string up to length k (a model-required parameter). As Chandlee (2014) discusses, this is problematic since natural language may in principle never provide all logically possible strings, due to phonotactic or morophological constraints. We implemented ISLFLA and attempted to run it on naturalistic data, and it does indeed fail to identify any FST on such data.[11]

Jardine *et al.* (2014) proposed a model, SOSFIA, for learning subsequential FSTs when the FST structure is known in advance; only the output for each arc in the FST needs to be learned. Strictly local functions are such a case, because the necessary and sufficient automata-theoretic conditions of strict locality include a complete FST structure (Chandlee, 2014). SOSFIA also admits learnability in the limit results, but has not been applied to naturalistic data.

[11] OSTIA will run on data not satisfying its characteristic sample; it is just not guaranteed to induce a correct FST in such cases. In contrast, ISLFLA is unable to proceed if the characteristic sample is not met: it exits at line 9 of the pseudocode in Chandlee *et al.* (2014, p. 499).

4. Evaluating the Model

This section evaluates PLP along a number of dimensions (36).

(36) **Q1**. Does PLP reflect human learners' preference for local generalizations?
Q2. How well does PLP learn local generalizations?
Q3. What are the learning effects of assuming UR-SR identity by default?

4.1. Model Comparisons

We compare to several alternative models.

4.1.1. Rule-Based, Neural Network, and Finite-State Models

MGL is the Minimal Generalization Learner from Albright & Hayes (2002, 2003). We used the Java implementation provided by the authors. MGL may produce multiple candidate SRs for a UR if more than one rule applies to the UR. In such cases, we used the rule with the maximum conditional probability scaled by scope (*confidence* in the terminology of Albright & Hayes 2002, sec. 3.2) to derive the predicted SR.

ED (encoder-decoder) is a neural network model. It is a successful neural network model for many natural language processing problems involving string-to-string functions, such as machine translation between languages (Sutskever *et al.* 2014), and morphological reinflection (Cotterell *et al.* 2016). It has also been used to revisit the use of neural networks in the 'past-tense debate' of English morphology (Kirov & Cotterell 2018), though its use as a computational model of morphology acquisition has been called into question (McCurdy *et al.* 2020; Belth *et al.* 2021). We follow Kirov & Cotterell (2018) and Belth *et al.* (2021) in its setup, using the same RNN implementation, trained for 100 epochs, with a batch size of 20, optimizing the log-likelihood of the training data. Both the encoder and the decoder are bidirectional LSTMs with 2 layers, 100 hidden units, and a vector size of 300.

OSTIA (Oncina *et al.*, 1993) is a finite-state model for learning subsequential finite state transducers. We used the Python implementation from Aksënova (2020).

ID is a trivial baseline that simply copies every input segment to the output. This allows for interpreting the value of assuming UR-SR identity by default.

4.1.2. Learning as Constraint Ranking

We also compare to the view of learning as ranking a provided constraint set. Classic OT viewed constraints as part of UG; we represent this view with **Ucon**, for *universal constraint set*. An alternative view is that the constraint set is learned; we represent this view with **Oracle**, which effectively constitutes an upper-bound on how well a model that learns the constraint set to be ranked could do. Oracle is provided all and only the markedness constraints relevant to the grammar being learned. Ucon is provided the same constraints as Oracle, plus two extra markedness constraints that are violable in the adult languages and thus must be down-ranked.

It is important to emphasize that these models learn in a different setting than PLP and those in § 4.1.1. The latter receive as input only UR-SR training pairs, whereas Ucon and Oracle receive both training pairs and a constraint set. Consequently, Ucon and Oracle's accuracies at producing SRs are not directly comparable to the other models' accuracies. Our goal in comparing PLP to Ucon and Oracle is to highlight the ways in which PLP's account of phonological learning differs.

For Ucon and Oracle, we use the Gradual Learning Algorithm (GLA) (Boersma, 1997; Boersma & Hayes, 2001) to rank the constraints because it is robust to exceptions—an important property when learning from noisy, naturalistic data. We emphasize, however, that the comparison is not to the particular constraint-ranking algorithm—others could have been chosen. Because our experiments involve many random samples and tens of thousands of tokens, the implementation of GLA in Praat (Boersma *et al.* 1999) was not well-suited. Thus, we used our own Python implementation of GLA, with the same default parameters as in Praat (evaluation noise: 2.0, plasticity: 1.0). We initialize markedness constraints above faithfulness constraints.

4.2. Comparison to Humans' Preference for Locality

In an experimental study, Baer-Henney & van de Vijver (2012) found that allomorphic generalizations in an artificial language were more easily and successfully learned when the surface allomorph was determined by a segment two positions away than determined by a segment three positions away. The study involved three artificial languages in which plural nouns were formed by affixing either the vowel [-y] or [-u], which differ in backness: −back and +back, respectively. Each language involved a different phonological condition for determining which affix surfaced. Treating /-y/ as the underlying affix, the three generalizations are those in (37).

(37) a. [−back] → [+back] / [+vowel,+**back**][+cons] __
b. [−back] → [+back] / [+vowel,+**tense**][+cons] __
c. [−back] → [+back] / [+cons,+**son**][+vowel][+cons] __

All singular forms are CVC words; plurals add a vowel. The (37a) language is an example of vowel harmony, since the affix vowel assimilates in backness to the preceding vowel. The (37b) language is equally local, but lacks clear phonetic motivation, since the stem vowel's feature determining the affix' backness is [tense]. The (37c) language is both less local and phonetically unmotivated, since the backness of the vowel is determined by the initial consonant of the stem. Because all three languages have CVC stems and CVCV plurals, each pattern is strictly local, but (37a)-(37b) involve a sequence of three contiguous segments while (37c) involves four.

Since PLP starts locally around the affix when looking for an appropriate generalization, and only proceeds outward when the more local contexts become too inaccurate, we expect PLP to learn the (37a)-(37b) generalizations substantially more easily than the (37c) generalization, just as Baer-Henney & van de Vijver (2012) found for humans (**Q3**). For comparison, we use MGL, which generalizes in roughly the opposite way: it constructs the narrowest—and hence less local—generalization. We also compare to grammars resulting from ranking three different constraint sets. The markedness constraints for (37) are (38).

(38) a. *[+vowel,+back][+cons][−back,+vowel]
b. *[+vowel,+tense][+cons][−back,+vowel]

c. *[+cons,+son][+vowel][+cons][−back,+vowel]

The first constraint set encodes the assumption of a universal constraint set containing only grounded, universal constraints by including only (38a), because it is the only generalization viewed as phonetically motivated. Secondly, we consider a constraint set containing all three markedness constraints (38a)-(38c) regardless of which language is being learned. Thirdly, we consider a constraint set containing only the constraint relevant to the language being learned.

In addition to learning the local generalization more easily than the non local one, Baer-Henney & van de Vijver (2012) also found that the phonetically motivated generalization (37a) was learned slightly more easily than (37b). The authors argued that this is evidence for substantive bias in phonological learning. However, the question of substantive bias is largely orthogonal to the current paper, since our focus is on locality. Moreover, the performance gap between (37a) and (37b) was much smaller than the gap between them and (37c). For these reasons, we focus on the difference in models' performance on (37a)-(37b) vs. (37c) in this experiment.

4.2.1. Setup

Each of Baer-Henney & van de Vijver (2012)'s experiments involved presenting subjects with randomly selected singulars and plurals from the respective artificial languages. Each word was accompanied by a picture conveying the word's meaning; one item was present in the picture for singulars and multiple items for plurals. The singulars and plurals were presented independently, so the experimental setup did not separate phonological learning from learning the artificial languages' morphology and semantics. Due to this fact, the study participants likely only successfully acquired the underlying and surface representations for a subset of the exposure words, and what fraction of the exposure set they learned is entirely unknown. Since the models assume URs and SRs as training data, we factor out the fraction of the exposure set for which they have acquired URs and SRs by treating it as a free-variable that the models get to optimize over. We use the data released by Baer-Henney & van de Vijver (2012) and follow their setup to construct train (exposure) and test sets.[12] We ran each model over 100 randomized exposure sets to simulate 100 participants.

The MGL model from Albright & Hayes (2002, 2003) combines rules that target the same segment and carry out the same change to that target. For instance, if it has acquired the two word-specific rules in (39a)-(39b), it will attempt to combine them via Minimal Generalization—i.e., as conservatively as possible. The minimal generalization for (39a)-(39b) is (39c), which retains as much as possible of the original two rules. However, in the implementation of MGL from Albright & Hayes (2002, 2003), when two rules are combined, the longest substrings shared by the rules are retained—in this case /p/—but segment combination (e.g., {ʊ,o}) only proceeds *one position further*; everything else is replaced by a free variable X (see Albright & Hayes 2002, p. 60 for a complete description of this process). Thus, their implementation returns (39d), which is less conservative than the actual minimal generalization (39c).

[12] Baer-Henney & van de Vijver (2012) used both high and low frequency settings, where the high frequency setting included a higher fraction of plural forms in the exposure set. Since we already treat the amount of exposure data available for learning phonology as a free variable, we followed the high-frequency setting for our experiment.

(39) a. y → u / bʊp __ #
b. y → u / dop __ #
c. y → u / {b,d}{ʊ,o}p __ #
d. y → u / X{ʊ,o}p __ #

This issue does not arise in the original papers by Albright & Hayes (2002, 2003), because the object of study was the English past tense, in which the surface allomorph is determined by an immediately adjacent segment. Thus, any regularities beyond the adjacent segment, which their implementation would miss despite being more minimal generalizations, would be spurious regularities anyway. However, for the purposes of this experiment, the implementation is problematic. Consequently, we used our own implementation of MGL, which correctly generates the minimally general combination of rules.[13] We use the rule with the minimally general context in which /-y/ surfaces as [-u] to produce a surface form for each test instance.

4.2.2. Results

Fig. 2 shows the results. The *x*-axis of each plot is the free-variable discussed above, measuring the fraction of the exposure set that the learner successfully constructed a UR-SR pair for. The *y*-axis reports the average model performance (over the 100 simulations), for each language. The points marked with a color-coded 'X' provide the average *human* performance from Baer-Henney & van de Vijver (2012) (over the 20 participants). Since each model gets to optimize over the free variable, we select, for each respective model, the *x*-value where it best matches the human performance, averaged over all three languages. This point can be seen by where the 'X's are placed. We drew a color-coded vertical line from each human performance marker to the corresponding model performance to demonstrate the difference between each model's performance and the humans'.

PLP (Fig. 2a) is the best match to the human results, learning the more local generalizations (37a)-(37b) substantially more easily than the less local generalization (37c). This is because PLP requires sufficient evidence against local generalizations (via the Tolerance Principle) before it will abandon them for less local ones (§ 2.2.2). This is reminiscent of Gómez & Maye (2005, p. 199)'s characterization of human learners as attending to local contexts even 'past the point that such structure is useful' before eventually moving on to less local information. In contrast, MGL (Fig. 2b) learns all three generalizations equally well because it constructs the most conservative—and hence widest context—generalization that is sustainable. If a grammar is constructed by ranking a universal constraint set that includes only phonetically-motivated constraints (Fig. 2c), only the generalization from (37a) can be learned because that is the only phonetically-motivated generalization. On the other hand, if all relevant constraints are included together (Fig. 2d) or on their own (Fig. 2e), all three generalizations are learned roughly equally well. This is because learning reduces to constraint ranking—the constraints being provided—and thus fails to distinguish between more and less local constraints. For language 3, the number of exceptions against the more local generalization will eventually become too numerous under the Tolerance Principle,

[13] All other experiments involving MGL used Albright & Hayes (2002, 2003)'s implementation.

and PLP will construct a less-local rule, which will correctly characterize the language 3 alternation. Thus, PLP predicts that given sufficient time and data, learners will eventually be able to learn the language 3 alternation.

4.2.3. Takeaways

PLP reflects human learners' preference for local generalizations (**Q3**).

4.3. Learning German Devoicing

We now evaluate PLP on syllable-final obstruent devoicing in German (Wiese, 1996).

4.3.1. Setup

This experiment simulates child acquisition by using vocabulary and frequency estimations from child-directed speech in the Leo corpus (Behrens 2006). We retrieved the corpus from the CHILDES (MacWhinney 2000) database and intersected the extracted vocabulary with CELEX (Baayen *et al.* 1996) to get phonological and orthographic transcriptions for each word. We also computed the frequency of each word in the Leo corpus. The resulting dataset consists of 9,539 words. To construct URs and SRs, we followed Gildea & Jurafsky (1996), using the CELEX phonological representations as SRs and discrepancies between CELEX phonology and orthography to construct URs, since German orthography does not reflect devoicing. Specifically, we make the syllable-final obstruents voiced for the URs of all words where the corresponding orthography indicates a voiced obstruent. In this data, 8.2% of words involve devoicing, which means a substantial number of URs equal SRs w.r.t this process. However, this is an appropriate and realistic scenario, since the data was constructed from child-directed speech and is thus a reasonable approximation of the data that children have access to when learning this generalization.

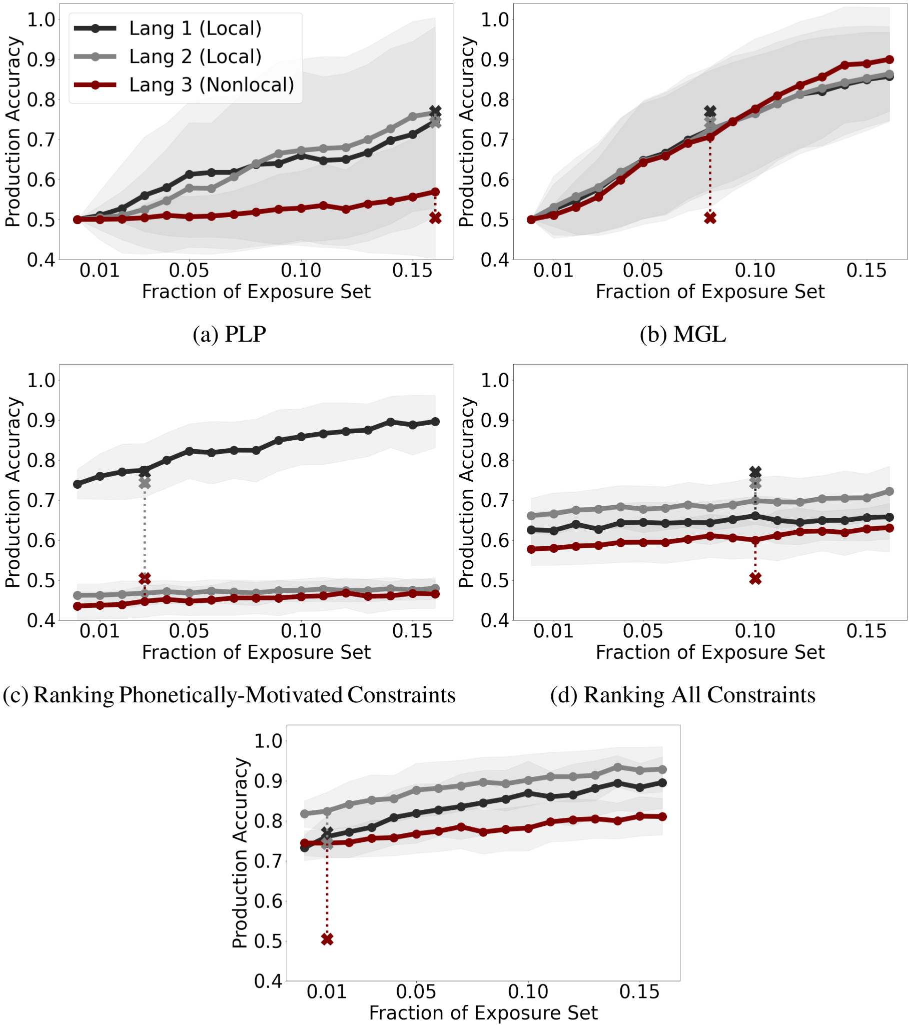
The experimental procedure samples one word at a time from the data, weighted by frequency. The word is presented to each model and added to its vocabulary. Sampling is with replacement, so the learners are expected to encounter the same word multiple times, at frequencies approximating what a child would encounter. When the vocabulary reaches a size of 100, 200, 300, and 400, each model is probed to produce an SR for each UR in the dataset that is *not* in the vocabulary (i.e. held-out test data). The fraction of these predictions that it gets correct is reported as the model's accuracy. The models MGL, ED, and OSTIA are designed as batch learners, so they are trained from scratch on the vocabulary prior to each evaluation period.[14] PLP, Ucon, and Oracle learn incrementally.

This simulation is carried out 10 times to simulate multiple learning trajectories. The results are averages and standard deviations over these 10 runs.

Oracle is provided with the constraint set (40).

(40) Con = { Max, Dep, Ident(voice), Ident(son), Ident(nas), *[+voi, −son]]$_{\sigma}$ }

[14] We provide MGL the frequency with which each vocabulary word has appeared, which it can make use of.



(a) PLP

(b) MGL

(c) Ranking Phonetically-Motivated Constraints

(d) Ranking All Constraints

(e) Ranking Language-Relevant Constraints

Figure 2: PLP best matches the locality results of Baer-Henney & van de Vijver (2012), where participants much more easily learned languages with local generalizations (languages 1-2) than a non-local generalization (language 3). In contrast, MGL fails to mirror these results, learning all generalizations equally well. Grammars constructed by ranking a provided constraint set also fail to match the results: if provided with phonetically-motivated constraints, only the first generalization can be learned, and if provided with all or language-relevant constraints, all generalizations are learned equally well.

The markedness constraint *[+voi, −son]$]_\sigma$, which marks syllable-final voiced obstruents, is the relevant markedness constraint for this process. Ucon is provided two additional constraints: *NC̥, which marks voiceless consonants following nasals, and *Complex. Both are frequently considered to be universal, violable constraints (Prince & Smolensky, 1993; Locke, 1983; Rosenthall, 1989; Pater, 1999). We included these to capture the assumption of a universal constraint set, which requires learning that *Complex and *NC̥ are violable in German; for instance /glaubənd/ → [$\underline{\text{gl}}$au.bə$\underline{\text{nt}}$.] ('believing') violates *Complex and *NC̥.

4.3.2. Results

The results are shown in Tab. 1. PLP learns an accurate grammar, which consists of the single generalizaton shown in (41), where '$]_\sigma$' denotes the end of a syllable.

(41) $[+\text{voi},-\text{son}] \rightarrow [-\text{voi}] \ / \ \_\_\ ]_\sigma$

While PLP achieves perfect accuracy by the time the vocabulary has grown to size 100, it does produce errors in the process of getting there. A primary example is underextensions. In our experiments, underlyingly voiced stops tended to enter the vocabulary earlier than voiced fricatives. Consequently, PLP sometimes fails to extend devoicing to fricatives until evidence of them devoicing enters the vocabulary. These underextensions are over *held-out test words*—i.e. words not in the learner's vocabulary. Thus, this is a prediction about an early state of the learner's phonological grammar, and not a prediction that children go through a stage of voicing final voiced fricatves. Indeed, as soon as an instance of fricative devoicing enters the vocabulary, we found that PLP extends the generalization to account for it.

Ranking a provided constraint set (Oracle and Ucon) can yield the same generalization as PLP: the sequence [+voi,−son]$]_\sigma$ is not allowed in German and violations of this restriction are repaired by devoicing. But the differences in how PLP learns this generalization are informative. Both Ucon and Oracle are provided the knowledge that the sequence [+voi,−son]$]_\sigma$ is marked. In contrast, PLP discovers the marked sequence in the process of learning.

In German, the onset [bl] is allowed (e.g., /blau/ → [blau]). PLP always produces the correct SR for /blau/ as a consequence of its identity default (Tab. 2). Whether a constraint-ranking model incorporates a preference for identity between inputs and outputs depends on what constraints it ranks. Because Oracle ranks only the constraints active in the language being learned, it—like PLP—does not produce unmotivated errors. If a universal constraint set is ranked (Ucon), then markedness constraints that are violable in the language being learned will lead to unmotivated errors. For instance, prior to downranking *Complex, Ucon sometimes produces [bəlau] for /blau/, with the complex onset /bl/ separated by a [ə], even though such onsets are allowed in German. However, deletion tends to be more common than epenthesis as a repair in child utterances, and it appears to occur due to articulatory limitations rather than by the child's hypothesized adult grammar.

Both Ucon and Oracle sometimes produce /kɪnd/ as *[kɪndə] and *[kɪn], rather than [kɪnt], because they must figure out the relative ranking of faithfulness constraints in order to capture which repair German uses to avoid *[+voi,−son]$]_\sigma$ violations. In

Table 1: Model accuracies (with standard deviations) on held-out test data at different training vocabulary sizes. PLP readily learns an accurate generalization for German final-obstruent devoicing.

Model	Vocabulary Size			
	100	200	300	400
PLP	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**	**1.000 ± 0.00**
MGL	0.918 ± 0.00	0.918 ± 0.00	0.918 ± 0.00	0.919 ± 0.00
ED	0.008 ± 0.00	0.178 ± 0.03	0.389 ± 0.04	0.543 ± 0.04
OSTIA	0.023 ± 0.02	0.022 ± 0.01	0.031 ± 0.01	0.040 ± 0.00
Ucon	0.960 ± 0.03	0.988 ± 0.00	0.992 ± 0.00	0.995 ± 0.00
Oracle	0.982 ± 0.01	0.997 ± 0.00	0.998 ± 0.00	0.999 ± 0.00
ID	0.918 ± 0.00	0.918 ± 0.00	0.918 ± 0.00	0.918 ± 0.00

Table 2: Analysis of the types of errors each of the models that learn an accurate grammar make in the process. Because it only adds generalizations to the grammar when necessitated by surface-alternation, PLP produces no unmotivated errors.

Error Type	Example	PLP	Ucon	Oracle
Unmotivated	/blɑu/ → *[bə̲lɑu]	No	Yes	No
Wrong-Repair	/kɪnd/ → *[kɪnd̲ə̲]	No	Yes	Yes
Under/Over Extension	/kɪnd/ → *[kɪnd̲]	Yes	Yes	Yes

contrast, PLP infers the repair—devoicing—directly from what discrepancy it observes in the data.

None of the other models perform competitively: PLP outperforms them all by a statistically significant amount ($p < 0.01$), as measured by a paired t-test against the null hypothesis that each model's performance over the 10 simulations has the same average accuracy as PLP's. MGL, which generalizes as conservatively as possible, struggles to generalize beyond the training data. This is seen in its slow rate of improvement. ED is a powerful model in natural language processing when substantial amounts of data are available, but it struggles to learn on the small vocabularies at the scale children learn from. OSTIA struggles even more, consistent with the negative results of Gildea & Jurafsky (1996), who presented it with much larger vocabularies.

4.3.3. Takeaways

PLP is readily able to learn German syllable-final devoicing (**Q2**) and never introduces unmotivated generalizations (**Q3**).

4.3.4. Opacity

The associate editor observed that devoicing in Polish interacts opaquely with o-raising, in which /ɔ/ surfaces as [u] before final, underlyingly voiced, oral consonants (Kenstowicz, 1994; Sanders, 2003). As a proof-of-concept, we ran PLP on the data in

Sanders (2003, chap 2; ex. 2-5). PLP learns rules (42) and correctly ordered them in Counterbleeding order with raising r_1 before devoicing r_2.[15]

(42) $G = r_2 \circ r_1$, where

$$r_1 = \text{ɔ} \rightarrow \text{u} \,/\, \_\_\, [\text{+voi}]\#$$
$$r_2 = [\text{+voi}, -\text{son}] \rightarrow [-\text{voi}] \,/\, \_\_\, \#$$

Rule r_2 accounts for devoicing both in isolation (43a) and in words exhibiting raising (43c). Rule r_1 accounts for raising both in isolation (43b) and when its underlying context is opaquely obscured by devoicing (43c).

(43) a. /klub/ → [klup] ‘club’ SG
b. /bɔl/ → [bul] ‘ache’ NOM.SG
c. /bɔb/ → [bup] ‘bean’ NOM.SG

The correct ordering was achieved because, in the reverse ordering, devoicing bleeds raising, resulting in errors like *[bɔp] for /bɔb/ that are not present when in Counterbleeding order. This demonstrates that PLP is capable of handling at least this case of opacity. We leave a systematic study of opacity for future work (see § 2.3.3).

4.4. Learning a Multi-Process Grammar

This experiment evaluates PLP at learning multiple generalization simultaneously. The processes modeled are the alternating plural and Prs 3rd Sg affix /-z/ (44a), the alternating past tense affix /-d/ (44b), and vowel nasalization (44c).

(44) a. /dɑg-z/ → [dɑgz]
/wɔk-z/ → [wɔks]
/hɔrs-z/ → [hɔrsəz]
b. /smɛl-d/ → [smɛld]
/wɔk-d/ → [wɔkt]
/foʊld-d/ → [foʊldəd]
c. /ðɛm/ → [ðɛ̃m]
/sʌmθɪŋ/ → [sʌ̃mθɪ̃ŋ]
/dæns/ → [dæ̃ns]

4.4.1. Setup

This experiment, like the first, simulates child language acquisition. The child-directed speech is aggregated across English corpora in CHILDES (MacWhinney, 2000), including the frequency of each word. Only words with ‘%mor’ tags were retained, because the morphological information was needed to construct URs. Transcriptions from the CMU pronunciation dictionary (Weide, 2014) served as SRs, with nasalization added to vowels preceding nasal consonants. URs had all vowels recorded without

[15] The examples from Sanders (2003) were too sparse to distinguish between [+voi]# and [+cons,+voi,−nas]# as the context for raising; a more realistic lexicon should drive PLP to the more nuanced context.

Table 3: Model accuracies (with standard deviations) on held-out test data at different training vocabulary sizes. PLP readily learns an accurate grammar for the English processes in (44).

Model	Vocabulary Size			
	1000	2000	3000	4000
PLP	**0.984** ± **0.01**	**0.992** ± **0.00**	**0.995** ± **0.00**	**0.997** ± **0.00**
Ucon	0.969 ± 0.00	0.982 ± 0.00	0.987 ± 0.00	0.990 ± 0.00
Oracle	0.980 ± 0.00	0.989 ± 0.00	0.991 ± 0.00	0.992 ± 0.00
ID	0.510 ± 0.00	0.510 ± 0.00	0.510 ± 0.00	0.510 ± 0.00

nasalization. The surface affixes for all past tense verbs, plural nouns, and Prs 3rd Sg verbs were set to /d/, /z/, and /z/, respectively in the URs. The resulting dataset contains 20,421 UR-SR pairs.

The experimental procedure is the same as for German, sampling words weighted by frequency and reporting accuracies at predicting SRs from URs over held-out test words when each learner's vocabulary reaches certain sizes: 1K, 2K, 3K, and 4K words.

We omit results from MGL, ED, and OSTIA because they continued to be non-competitive. Oracle once again ranks only the relevant constraints (45) and Ucon receives, in addition to (45), *Complex and *NC̥.

(45) Con = {
Max, Dep, Ident(voice), Ident(son), Ident(nas),
Agree(Voice), *SS, *[+vowel, −nas][+cons, +nas],
*[−cont,−dist,−son][−cont,−dist,−son]
}

All faithfulness constraints other than Dep were split into two—one for stems and one for affixes—so that, for instance, *[wɔgz] could be ruled out for input /wɔk-z/ in (44).

4.4.2. Results

The models' accuracies on held-out test words, shown in Tab. 3, reveal that PLP learns an accurate grammar by the time its vocabulary grows to about 2000 words. PLP's output is shown as an ordered list of rules in (46).

(46) $G = r_5 \circ r_4 \circ r_3 \circ r_2 \circ r_1$, where

$$r_1 = [\text{+syl}] \longrightarrow [\text{+nas}] \,/\, \_\_ \, [\text{+nas}]$$
$$r_2 = \emptyset \longrightarrow \text{ə} \,/\, [\text{+sib}] \, \_\_ \, [\text{+sib}]$$
$$r_3 = [\text{+sib, +voi}] \longrightarrow [-\text{voi}] \,/\, [-\text{voi}] \, \_\_$$
$$r_4 = \emptyset \longrightarrow \text{ə} \,/\, [\text{+cor}, -\text{cont}, -\text{nas}] \, \_\_ \, [\text{+cor}, -\text{cont}, -\text{nas}]$$
$$r_5 = [\text{+cor}, -\text{cont}, -\text{nas}, \text{+voi}] \longrightarrow [-\text{voi}] \,/\, [-\text{voi}] \, \_\_$$

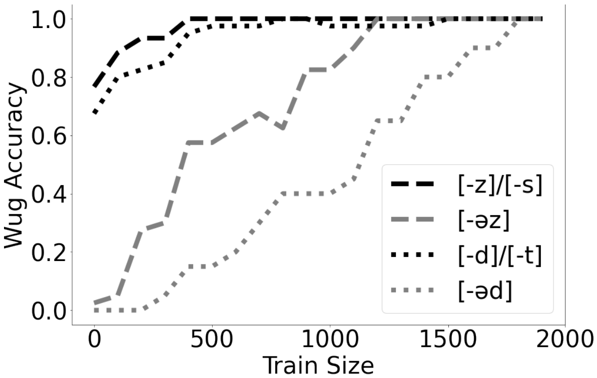


Figure 3: PLP's accuracy on the plural and past tense nonce words from Berko (1958) as training progressed. The black dashed line denotes plurals that should take [-z] or [-s] and the gray dashed lines those that should take [-əz]. The dotted lines represent the analogues for past-tense. The fact that [z]/[s] accuracy converges before [-əz] and [d]/[t] before [-əd] matches Berko (1958)'s finding that children learn [-z]/[-s] and [-d]/[-t] before [-əz] and [-əd].

The rules were ordered as described in § 2.3.3, with r_2 before r_3 and r_4 before r_5 (i.e. Bleeding order) being the inferred ordering dependencies. Thus, as described in § 2.3.3, PLP learned that epenthesis bleeds devoicing. The rules r_2-r_5 do not encode a word-final context because doing so would require expanding PLP's search window, which is not necessary because the rules without word-final context pass the Tolerance Principle. The extension of [+cor,−cont,−nas] is {t, d} and of [+cor,−cont,−nas,+voi] is {d}.

The reason no model achieves 100% accuracy is due to a handful of words that do not follow the generalizations in (44). For instance, compounds like [bɛdtaĩm] allow the sequence [dt], but the models predict there should be an epenthetic vowel to split the sequence. Such exceptions are easily accounted for if we assume the learner recognizes the word as a compound. Since exceptions are inevitable in naturalistic data, we chose to not remove these exceptions.

In Berko (1958)'s seminal study, Berko found that children aged 4-7yrs could accurately inflect nonce words that take the [-z], [-s], [-d], or [-t] suffixes, but that they performed much worse at inflecting nonce words taking the [-əz] or [-əd] suffixes. Adults could inflect nonce words with [-əz] or [-əd], suggesting that voicing assimilation process may be learned earlier than the epenethesis process. We show PLP's accuracy on Berko (1958)'s different categories of nonce words in Fig. 3 as the vocabulary grows (x-axis). PLP's accuracy on nonce words taking [-z] or [-s] (black dashed line) converges earlier than its accuracy on nonce words taking [-əz] (gray dashed line); similarly the accuracy for nonce words taking [-d] or [-t] (black dotted line) converges earlier than for nonce words taking [-əd] (gray dotted line). Thus, the order of acquisition matches Berko (1958)'s finding.

4.4.3. Takeaways

The results in this more challenging setting, where multiple processes are simultaneously active, support the takeaways from the prior experiment. PLP successfully learns all the generalizations (**Q2**) and does not introduce unmotivated generalizations (**Q3**).

4.5. Learning Tswana's Post-Nasal Devoicing

Although a majority of phonological patterns may be phonetically grounded, some processes nevertheless appear to lack or even oppose phonetic motivation (Anderson, 1981; Buckley, 2000; Johnsen, 2012; Beguš, 2019). Moreover, these must still be learnable, because children continue to successfully acquire them (Johnsen, 2012, p. 506). An example of such a pattern is post-nasal devoicing in Tswana shown in (2), which Coetzee & Pretorius (2010) confirmed to be productive despite operating against the phonetic motivation for post-nasal *voicing* (Hayes & Stivers, 2000; Beguš, 2019). Beguš (2019, p. 699) found post-nasal devoicing to be reported as a sound change in thirteen languages and dialects, from eight language families.

Models of phonological learning should account for the fact that non-phonetically-grounded, yet productive patterns are successfully learned by humans. A consequence of PLP's identity default is that generalizations are added to the grammar whenever they are motivated by surface alternation. Since surface alternation in Tswana motivates a generalization for post-nasal devoicing, its learnability should be accounted for with PLP. This experiment attempts to confirm this (**Q3**).

4.5.1. Setup

For this experiment we used the 10 UR-SR pairs from Coetzee & Pretorius (2010, p. 406) as training data. Five pairs involve devoicing resulting from the 1st Sg Obj clitic /m/ attaching to a stem that starts with a voiced obstruent. The other five pairs involve the 1st Pl Obj clitic /re/ attaching to the same stems, which serve as negative examples since the clitic does not introduce a nasal. This data is not necessarily representative of the data that a child would have during acquisition, and thus serves as a proof-of-concept learnability experiment.

The test data consists of the same 20 /b/-initial nonce words presented to the participants in Coetzee & Pretorius (2010, p. 407)—10 stems each combined with /m/ and /re/.

4.5.2. Results

The results in Tab. 4 demonstrate that PLP can learn Tswana's post-nasal devoicing without requiring the existence of a universal, phonetically unmotivated constraint.[16] Constraint-ranking models can also learn the generalization, but depend on an account of how the constraint *NC̬, which is not usually considered to be a universally marked sequence (Locke, 1983; Rosenthall, 1989; Pater, 1999; Beguš, 2016, 2019), is added to the constraint set.

[16] PLP learns *[mb] rather than *NC̬ because the training data only included [mb] instances; if more representative training data were available, PLP would induce natural classes, as in the previous experiments.

Table 4: PLP learns precisely the set of processes active in its experience. This provides a straight-forward account of how productive phonological processes can be learned even if they operate against apparent phonetic motivation, like devoicing in Tswana following nasals (Coetzee & Pretorius, 2010). With PLP, the unmotivated constraint *NC̬ need not be assumed universal.

Model	Generalization	Test Accuracy
PLP	b → [−voi] / m __	1.0
Ranking without *NC̬	{*NC̥, Ident(Voice)}	0.5
Ranking with *NC̬	*NC̬ » {*NC̥, Ident(Voice)}	1.0

4.5.3. Takeaways

Because PLP assumes UR-SR identity by default, it constructs precisely the generalizations necessary to account for the discrepancies active in its experience, providing a straight-forward account of how productive generalizations can be learned even if they are opposed to apparent phonetic motivation, as humans evidently do (Seidl & Buckley 2005, Johnsen 2012, p. 506; Beguš 2018, ch. 6) (**Q3**).

5. Discussion

5.1. The Nature of Locality

One reviewer asked what sort of tendency we view locality to be. We view the cognitive tendency for humans to prefer constructing local generalizations to be a *gemoetric, computational* consequence. That is, if words are viewed, at least to a first-approximation, as linear objects, this linear geometry introduces the notion of locality as *small linear distance.* In our view, the reason that a human is more likely to construct a generalization that conditions x_i on x_{i-1} than on x_{i-2} in a sequence $..., x_{i-2}, x_{i-1}, x_i$ (see § 1.1) is that a search outward from x_i encounters x_{i-1} before it encounters x_{i-2}. PLP is an attempt to state this in explicit computational terms. An immediate consequence of this hypothesis is that if x_{i-1} is sufficient to account for whatever the uncertainty in x_i is (e.g., what its surface form is), then x_{i-2} will never be considered, even if there is some statistical dependency between the two. We believe this prediction is consistent with the experimental results from sequence learning, which we discuss in § 1.1, where participants would track adjacent dependencies even when non-adjacent dependencies were more statistically informative (Gómez & Maye, 2005) and would construct local generalizations over less local ones when the exposure data underdetermined the two (i.e. the poverty-of-stimulus paradigms of Finley 2011, McMullin & Hansson 2019). We note that words may not be *exactly* linear—segment articulations have gestural overlap, syllables are often viewed as hierarchical structures, and representations like autosegmental tiers may be present. However, we think treating words as linear sequences is a good first approximation. Work on tier-locality also recognizes that string-locality is a special case of tier locality in which all segments are present on the tier (e.g. Hayes & Wilson 2008; Heinz *et al.* 2011; McMullin 2016).

An alternative view could be that locality is distributional: a learner may track the dependency between x_i and both x_{i-1} and x_{i-2}, and may find that x_{i-1} is more statistically robust as a generalization, preferring it for that reason. However, this view is inconsistent with the findings that when statistical robustness is controlled (Finley, 2011; McMullin & Hansson, 2019) and even when it *favors* the less-local dependency (Gómez & Maye, 2005) humans systematically generalize locally. The distributional approach could be combined with a stipulated bias (prior) favoring local dependencies, but this would simply describe the phenomena, not explain it.

5.2. Future Directions

Research on phonological representations recognizes that strict locality arises not only over string representations, but also over representations like tiers and metrical grids (Goldsmith, 1976; Heinz *et al.*, 2011; Hayes & Wilson, 2008; McMullin, 2016). PLP could be applied over these representations to find, e.g. tier-strictly local generalizations. In our view, an account of how a learner may construct increasingly abstract representations should first be given, and PLP may provide a framework for such an account. In computational terms, generalizations over tier-contiguous sequences require more computation than generalizations over string-contiguous sequences, because they require projection onto the tier in addition to recognizing the sequence (Heinz, 2018). Thus, if we extend the concept of *computational parsimony*, a learner would first seek out a generalization over the minimally-abstract string representation. However, if a satisfactorily accurate generalization cannot be found, then this motivates the additional computation of projection. We see this type of account to be a promising line of inquiry.

PLP currently learns categorical processes. Variation is an important aspect of phonology (Coetzee & Pater, 2011), and future work should investigate extending PLP to handle variation.

The implications of PLP to language change and typology could also be investigated.

Acknowledgments. This work has been greatly improved through the thoughtful and constructive comments of three anonymous reviewers, and the associate editor. I thank Andries Coetzee, Charles Yang, Jane Chandlee, and Jeffrey Heinz for many discussions throughout the development of this project. This work was supported by an NSF GRF. All mistakes are my own.

References

Aksënova, Alëna. (2020). *Sigmapie*. https://github.com/alenaks/SigmaPie.

Albright, Adam. (2009). Feature-based generalisation as a source of gradient acceptability. *Phonology*, **26**(1), 9–41.

Albright, Adam, & Hayes, Bruce. (2002). Modeling english past tense intuitions with minimal generalization. *Pages 58–69 of: Proceedings of the ACL workshop on morphological and phonological learning*.

Albright, Adam, & Hayes, Bruce. (2003). Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition*, **90**(2), 119–161.

Anderson, Stephen R. (1981). Why phonology isn't "natural". *LI*, **12**(4), 493–539.

Aslin, Richard N, Saffran, Jenny R, & Newport, Elissa L. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological science*, **9**(4), 321–324.

Baayen, R. Harald, Piepenbrock, Richard, & Gulikers, Léon. (1996). *CELEX2*. Linguistic Data Consortium.

Baayen, R Harald, Chuang, Yu-Ying, & Blevins, James P. (2018). Inflectional morphology with linear mappings. *The mental lexicon*, **13**(2), 230–268.

Baayen, R Harald, Chuang, Yu-Ying, Shafaei-Bajestan, Elnaz, & Blevins, James P. (2019). The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de) composition but in linear discriminative learning. *Complexity*, **2019**.

Baer-Henney, Dinah, & van de Vijver, Ruben. (2012). On the role of substance, locality, and amount of exposure in the acquisition of morphophonemic alternations. *Laboratory phonology*, **3**(2), 221–249.

Baković, Eric. (2011). Opacity and ordering. Goldsmith, John, Riggle, Jason, & Yu, Alan C.L. (eds), *The handbook of phonological theory*, 2nd edn. Malden, MA: Wiley-Blackwell.

Beguš, Gašper. (2016). Post-nasal devoicing and a probabilistic model of phonological typology. *Ms., Harvard University*.

Beguš, Gašper. (2018). *Unnatural phonology: A synchrony-diachrony interface approach*. Ph.D. thesis, Harvard University.

Beguš, Gašper. (2019). Post-nasal devoicing and the blurring process. *JL*, **55**(4), 689–753.

Beguš, Gašper. (2022). Local and non-local dependency learning and emergence of rule-like representations in speech data by deep convolutional generative adversarial networks. *Computer speech & language*, **71**, 101244.

Behrens, Heike. (2006). The input–output relationship in first language acquisition. *Language and cognitive processes*, **21**(1-3), 2–24.

Belth, Caleb, Payne, Sarah, Beser, Deniz, Kodner, Jordan, & Yang, Charles. (2021). The greedy and recursive search for morphological productivity. *Proceedings of CogSci 2021*.

Berent, Iris. (2013). The phonological mind. *Trends in cognitive sciences*, **17**(7), 319–327.

Berent, Iris, & Lennertz, Tracy. (2007). What we know about what we have never heard before: Beyond phonetics. *Cognition*, **104**(3), 638–643.

Berko, Jean. (1958). The child's learning of English morphology. *Word*, **14**(2–3), 150–177.

Berkson, Kelly, Davis, Stuart, & Strickler, Alyssa. (2017). What does incipient /ay/-raising look like?: A response to josef fruehwald. *Lg*, **93**(3), e181–e191.

Boersma, Paul. (1997). How we learn variation, optionality, and probability. *Pages 43–58 of: Proceedings of the institute of phonetic sciences of the university of amsterdam*, vol. 21.

Boersma, Paul, & Hayes, Bruce. (2001). Empirical tests of the gradual learning algorithm. *LI*, **32**(1), 45–86.

Boersma, Paul, & Pater, Joe. (2008). *Convergence properties of a gradual learning algorithm for harmonic grammar*. Tech. rept.

Boersma, Paul, *et al.* . (1999). Optimality-theoretic learning in the praat program. *Pages 17–35 of: IFA proceedings*, vol. 23.

Buckler, Helen, & Fikkert, Paula. (2016). Dutch and german 3-year-olds representations of voicing alternations. *Language and speech*, **59**(2), 236–265.

Buckley, Eugene. (2000). On the naturalness of unnatural rules. *Pages 1–14 of: Proceedings from the second workshop on american indigenous languages. UCSB working papers in linguistics*, vol. 9.

Calamaro, Shira, & Jarosz, Gaja. (2015). Learning general phonological rules from distributional information: A computational model. *Cognitive science*, **39**(3), 647–666.

Chandlee, Jane. (2014). *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Chandlee, Jane, Eyraud, Rémi, & Heinz, Jeffrey. (2014). Learning strictly local subsequential functions. *Transactions of the association for computational linguistics*, **2**, 491–504.

Chandlee, Jane, Heinz, Jeffrey, & Jardine, Adam. (2018). Input strictly local opaque maps. *Phonology*, **35**(2), 171–205.

Chomsky, Noam. (1956). Three models for the description of language. *IRE transactions on information theory*, **2**(3), 113–124.

Chomsky, Noam. (2005). Three factors in language design. *LI*, **36**(1), 1–22.

Chomsky, Noam, & Halle, Morris. (1965). Some controversial questions in phonological theory. *JL*, **1**(2), 97–138.

Chomsky, Noam, & Halle, Morris. (1968). *The sound pattern of english*. Cambridge, MA: Harper & Row.

Coetzee, Andries W. (2009). Learning lexical indexation. *Phonology*, **26**(1), 109–145.

Coetzee, Andries W., & Pater, Joe. (2011). The place of variation in phonological theory. Goldsmith, John, Riggle, Jason, & Yu, Alan C.L. (eds), *The handbook of phonological theory*, 2nd edn. Malden, MA: Wiley-Blackwell.

Coetzee, Andries W., & Pretorius, Rigardt. (2010). Phonetically grounded phonology and sound change: The case of tswana labial plosives. *JPh*, **38**(3), 404–421.

Cormen, Thomas H, Leiserson, Charles E, Rivest, Ronald L, & Stein, Clifford. (2009). *Introduction to algorithms*. MIT press.

Cotterell, Ryan, Kirov, Christo, Sylak-Glassman, John, Yarowsky, David, Eisner, Jason, & Hulden, Mans. (2016). The sigmorphon 2016 shared taskmorphological reinflection. *Pages 10–22 of: Proceedings of the 14th sigmorphon workshop on computational research in phonetics, phonology, and morphology*.

Emond, Emeryse, & Shi, Rushen. (2021). Infants' rule generalization is governed by the Tolerance Principle. *Pages 191–204 of:* Dionne, Danielle, & Vidal Covas, Lee-Ann (eds), *Proceedings of the 45nd annual Boston University Conference on Language Development*.

Ernestus, Mirjam Theresia Constantia, & Baayen, R Harald. (2003). Predicting the unpredictable: Interpreting neutralized segments in dutch. *Lg*, **79**(1), 5–38.

Fikkert, Paula. (1994). *On the acquisition of prosodic structure*. Ph.D. thesis, Leiden University.

Finley, Sara. (2011). The privileged status of locality in consonant harmony. *Journal of memory and language*, **65**(1), 74–83.

Finley, Sara, & Badecker, William. (2009). Artificial language learning and feature-based generalization. *Journal of memory and language*, **61**(3), 423–437.

Fiser, József, & Aslin, Richard N. (2002). Statistical learning of higher-order temporal structure from visual shape sequences. *Journal of experimental psychology: Learning, memory, and cognition*, **28**(3), 458.

Freitas, M João. (2003). The acquisition of onset clusters in european portuguese. **15**(1), 27–46.

Gafos, Adamantios I. (2014). *The articulatory basis of locality in phonology*. Routledge.

Gildea, Daniel, & Jurafsky, Dan. (1996). Learning bias and phonological-rule induction. *Computational linguistics*, **22**(4), 497–530.

Gnanadesikan, Amalia. (2004). Markedness and faithfulness constraints in child phonology. *Constraints in phonological acquisition*, 73–108.

Gold, E Mark. (1967). Language identification in the limit. *Information and control*, **10**(5), 447–474.

Goldsmith, John. (1976). *Autosegmental phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Gómez, Rebecca, & Maye, Jessica. (2005). The developmental trajectory of nonadjacent dependency learning. *Infancy*, **7**(2), 183–206.

Gómez, Rebecca L. (2002). Variability and detection of invariant structure. *Psychological science*, **13**(5), 431–436.

Grijzenhout, Janet, & Joppen, Sandra. (1998). First steps in the acquisition of german phonology: A case study. Seminar für Allgemeine Sprachwissenschaft, Heinrich-Heine-Univ.

Grijzenhout, Janet, & Joppen-Hellwig, Sandra. (2002). The lack of onsets in german child phonology. *The process of language acquisition*, 319–339.

Hale, Mark, & Reiss, Charles. (2008). *The phonological enterprise*. Oxford University Press.

Halle, Morris. (1978). Knowledge unlearned and untaught: What speakers know about the sounds of their language. *Pages 294–303 of:* Halle, Morris, Bresnan, Joan, & Miller, George A. (eds), *Linguistic theory and psychological reality*. Cambridge, MA: MIT Press.

Hayes, Bruce, & Stivers, Tanya. (2000). Postnasal voicing. *Ms., UCLA*.

Hayes, Bruce, & Wilson, Colin. (2008). A maximum entropy model of phonotactics and phonotactic learning. *LI*, **39**(3), 379–440.

Heinz, Jeffrey. (2018). The computational nature of phonological generalizations. *Phonological typology, phonetics and phonology*, 126–195.

Heinz, Jeffrey, Rawal, Chetan, & Tanner, Herbert G. (2011). Tier-based strictly local constraints for phonology. *Pages 58–64 of: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*.

Howard, Irwin. (1972). *A directional theory of rule application in phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

Hyman, Larry M. (2018). Why underlying representations? *JL*, **54**(3), 591–610.

Jakobson, Roman, & Halle, Morris. (1956). *Fundamentals of language*. The Hague: Mouton.

Jardine, Adam. (2016). Computationally, tone is different. *Phonology*, **33**(2), 247–283.

Jardine, Adam, Chandlee, Jane, Eyraud, Rémi, & Heinz, Jeffrey. (2014). Very efficient learning of structured classes of subsequential functions from positive data. *Pages 94–108 of: The 12th international conference on grammatical inference*, vol. 34. Kyoto, Japan: PMLR.

Jarosz, Gaja. (2019). Computational modeling of phonological learning. *Annual review of linguistics*, **5**(1), 67–90.

Johnsen, Sverre Stausland. (2012). A diachronic account of phonological unnaturalness. *Phonology*, **29**(3), 505–531.

Johnson, Mark. (1984). A discovery procedure for certain phonological rules. *Page 344347 of: Proceedings of the 10th international conference on computational linguistics and 22nd annual meeting on association for computational linguistics*. ACL.

Joos, Martin. (1942). A phonological dilemma in Canadian English. *Lg*, **18**(2), 141–144.

Jusczyk, Peter W, Smolensky, Paul, & Allocco, Theresa. (2002). How english-learning infants respond to markedness and faithfulness constraints. *Language acquisition*, **10**(1), 31–73.

Kenstowicz, Michael. (1994). *Phonology in generative grammar*. Malden, MA: Blackwell.

Kenstowicz, Michael, & Kisseberth, Charles. (1979). *Generative phonology: Description and theory*. San Diego: Academic Press.

Kerkhoff, Annemarie Odilia. (2007). *Acquisition of morpho-phonology: The Dutch voicing alternation*. Ph.D. thesis, Landelijke Onderzoekschool Taalwetenschap.

Kiparsky, Paul. (1968). *How abstract is phonology?* Indiana University Linguistics Club.

Kirov, C., & Cotterell, Ryan. (2018). Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the association for computational linguistics*, **6**, 651–665.

Koulaguina, Elena, & Shi, Rushen. (2019). Rule generalization from inconsistent input in early infancy. *Language acquisition*, **26**(4), 416–435.

Legendre, Géraldine, Miyata, Yoshiro, & Smolensky, Paul. (1990). *Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations*. Tech. rept. 90-5. Institute of Cognitive Science, University of Colorado, Boulder.

Locke, John L. (1983). *Phonological acquisition and change*. Academic Press.

MacWhinney, Brian. (2000). *The childes project: Tools for analyzing talk. transcription format and programs*. Vol. 1. Psychology Press.

Mayer, Connor. (2020). An algorithm for learning phonological classes from distributional similarity. *Phonology*, **37**(1), 91–131.

McCarthy, John J. (2007). *Derivations and levels of representation*. Cambridge Handbooks in Language and Linguistics. Cambridge University Press. Pages 99–118.

McCurdy, Kate, Goldwater, Sharon, & Lopez, Adam. (2020). Inflecting when there's no majority: Limitations of encoder-decoder neural networks as cognitive models for German plurals. *Pages 1745–1756 of: Proceedings of the 58th annual meeting of the association for computational linguistics*. Association for Computational Linguistics.

McMullin, Kevin. (2016). *Tier-based locality in long-distance phonotactics: learnability and typology*. Ph.D. thesis, University of British Columbia.

McMullin, Kevin, & Hansson, Gunnar Ólafur. (2019). Inductive learning of locality relations in segmental phonology. *Laboratory phonology*, **10**(1).

McNaughton, Robert, & Papert, Seymour A. (1971). *Counter-free automata (mit research monograph no. 65)*. The MIT Press.

Mortensen, David R., Littell, Patrick, Bharadwaj, Akash, Goyal, Kartik, Dyer, Chris, & Levin, Lori. (2016). PanPhon: A resource for mapping IPA segments to articulatory feature vectors. *Pages 3475–3484 of: Proceedings of the 26th international conference on computational linguistics: Technical papers*. Osaka, Japan: The COLING 2016 Organizing Committee.

Newport, Elissa L., & Aslin, Richard N. (2004). Learning at a distance: I. statistical learning of non-adjacent dependencies. *Cognitive psychology*, **48**(2), 127–162.

Oncina, José, García, Pedro, & Vidal, Enrique. (1993). Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE trans. pattern anal. mach. intell.*, **15**(5), 448–458.

Pater, Joe. (1999). Austronesian nasal substitution and other nc effects. *The prosody-morphology interface*, 310–343.

Peperkamp, Sharon, Le Calvez, Rozenn, Nadal, Jean-Pierre, & Dupoux, Emmanuel. (2006). The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition*, **101**(3), B31–B42.

Prince, Alan, & Smolensky, Paul. (1993). *Optimality Theory: Constraint interaction in generative grammar*. Tech. rept. Rutgers University.

Rasin, Ezer, Berger, Iddo, Lan, Nur, & Katzir, Roni. (2018). Learning phonological optionality and opacity from distributional evidence. *Pages 269–282 of: NELS*, vol. 48.

Richter, Caitlin. (2018). Learning allophones: What input is necessary. *Proceedings of the 42nd annual boston university conference on language development.* Cascadilla Press.

Richter, Caitlin. (2021). *Alternation-sensitive phoneme learning: Implications for children's development and language change*. Ph.D. thesis, University of Pennsylvania.

Ringe, Don, & Eska, Joseph F. (2013). *Historical linguistics: Toward a twenty-first century reintegration*. Cambridge University Press.

Rogers, James, Heinz, Jeffrey, Fero, Margaret, Hurst, Jeremy, Lambert, Dakotah, & Wibel, Sean. (2013). Cognitive and sub-regular complexity. *Pages 90–108 of:* Morrill, Glyn, & Nederhof, Mark-Jan (eds), *Formal grammar*. Berlin, Heidelberg: Springer.

Rosenthall, Samuel. (1989). *The phonology of nasal-obstruent sequences*. M.A. Thesis, McGill University.

Saffran, Jenny R, Aslin, Richard N, & Newport, Elissa L. (1996). Statistical learning by 8-month-old infants. *Science*, **274**(5294), 1926–1928.

Saffran, Jenny R., Newport, Elissa L., Aslin, Richard N., Tunick, Rachel A., & Barrueco, Sandra. (1997). Incidental language learning: Listening (and learning) out of the corner of your ear. *Psychological science*, **8**(2), 101–105.

Sanders, Robert Nathaniel. (2003). *Opacity and sound change in the polish lexicon*. Ph.D. thesis, University of California, Santa Cruz.

Santelmann, Lynn M., & Jusczyk, Peter W. (1998). Sensitivity to discontinuous dependencies in language learners: Evidence for limitations in processing space. *Cognition*, **69**(2), 105–134.

Schuler, Kathryn, Yang, Charles, & Newport, Elissa. (2016). Testing the Tolerance Principle: Children form productive rules when it is more computationally efficient to do so. *The 38th cognitive society annual meeting*.

Seidl, Amanda, & Buckley, Eugene. (2005). On the learning of arbitrary phonological rules. *Language learning and development*, **1**(3–4), 289–316.

Smith, Nelson V. (1973). *The acquisition of phonology: A case study*. Cambridge: Cambridge University Press.

Smolensky, Paul. (1996). *The initial state and "richness of the base" in Optimality Theory*. Tech. rept. Johns Hopkins University.

Smolensky, Paul, & Legendre, Géraldine. (2006). *The harmonic mind: From neural computation to optimality-theoretic grammar (cognitive architecture)*. Vol. 1. MIT press.

Sutskever, Ilya, Vinyals, Oriol, & Le, Quoc V. (2014). Sequence to sequence learning with neural networks. *Pages 3104–3112 of: Advances in neural information processing systems*.

Tesar, Bruce, & Smolensky, Paul. (1998). Learnability in optimality theory. *LI*, **29**(2), 229–268.

Van de Vijver, Ruben, & Baer-Henney, Dinah. (2014). Developing biases. *Frontiers in psychology*, **5**.

Weide, Robert. (2014). *The carnegie mellon pronuncing dictionary v0.7b*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

White, James, Kager, René, Linzen, Tal, Markopoulos, Giorgos, Martin, Alexander, Nevins, Andrew, Peperkamp, Sharon, Polgárdi, Krisztina, Topintzi, Nina, & van De Vijver, Ruben. (2018). Preference for locality is affected by the prefix/suffix asymmetry: Evidence from artificial language learning. *Pages 207–220 of: NELS*.

White, Katherine S, Peperkamp, Sharon, Kirk, Cecilia, & Morgan, James L. (2008). Rapid acquisition of phonological alternations by infants. *Cognition*, **107**(1), 238–265.

Wiese, Richard. (1996). *The phonology of German*. Oxford: Clarendon.

Yang, Charles. (2016). *The price of linguistic productivity: How children learn to break the rules of language*. Cambridge, MA: MIT press.

Yang, Charles, Crain, Stephen, Berwick, Robert C., Chomsky, Noam, & Bolhuis, Johan J. (2017). The growth of language: Universal grammar, experience, and principles of computation. *Neuroscience and biobehavioral reviews*, **81**(Part B), 103–119.

A. Appendix

A.1. PLP and Strict Locality

We discuss how PLP's generalizations can be characterized in the formal-language-theoretic terms of *strict locality*. We show that the sequences PLP learns are strictly-local definitions, and thus have the interpretation of banning substrings (Heinz, 2018, p. 28) (§ A.1.1). We then discuss how PLP's generalizations describe input-strictly local maps (§ A.1.2).

A.1.1. Strict-Locality of Sequences

Stirctly local stringsets (McNaughton & Papert, 1971) are stringsets whose members 'are distinguished from non-members purely on the basis of their k-factors' (Rogers *et al.*, 2013, p. 98). A k-factor of a string is a length-k substring, and (p. 96) the set of k-factors over an alphabet Σ is $F_k(\Sigma^*) = \{w \in \Sigma^* : |w| \leq k\}$. A *Strictly k-Local Definition* $\mathcal{G}$ is a subset of the k-factors over Σ, i.e., $\mathcal{G} \subseteq F_k(\Sigma^*)$.[17] A definition is a strictly-local definition if it is strictly k-local for some k. We wish to demonstrate that the sequences PLP learns, as defined in (6) and repeated in (47), are strictly-local definitions.

(47) $\mathcal{S} \triangleq \bigcup_{k=1}^{\infty}\{s_1 s_2 ... s_k : s_i \subset \Sigma\}$

Since $\bar{s} \in \mathcal{S}$ is a sequence of *sets* of segments $s_i \subset \Sigma$, we define the *extension*, $E_{\bar{s}}$, of $\bar{s}$ as the set of sequences of *segments* that match $\bar{s}$, as in (48), where $k = |\bar{s}|$.

(48) $E_{\bar{s}} \triangleq \{a_1 a_2 ... a_k : a_i \in s_i \; \forall i \in 1...k\}$

For example, the sequence of two adjacent sibilants (49a) has the extension (49b).

(49) a. $\bar{s}$ = [+sib][+sib]

b. $E_{\bar{s}}$ = {ss, sz, zs, ʃz, ʒs, ...}

Theorem 1 *The instances $E_{\bar{s}}$ of any $\bar{s} \in \mathcal{S}$ form a Strictly Local definition over the alphabet Σ.*

Proof. For any $a_1 a_2 ... a_k \in E_{\bar{s}}$, each a_i is an element of s_i (i.e., $a_i \in s_i$) by (48) and thus an element of Σ (i.e., $a_i \in s_i \subset \Sigma$) by (47). Thus, every $a_1 a_2 ... a_k \in E_{\bar{s}}$ is a length-k string from Σ^*. It follows that $E_{\bar{s}} \subseteq F_k(\Sigma^*)$ and that, for $k = |\bar{s}|$, $E_{\bar{s}}$ is a Strictly Local Definition. □

A.1.2. Strict-Locality of Generalizations

Chandlee (2014, p. 40) provides formal-language-theoretic and automata-theoretic definitions of *Input Strictly Local* string-to-string functions, which, for input and output alphabets Σ and Γ, have the following interpretation:

Definition 1 (kISL function - Informal) *A function (map) $f : \Sigma^* \to \Gamma^*$ is Input Strictly Local (ISL) iff $\exists k \in \mathbb{N}$ such that each output symbol $o \in \Gamma$ is determined by a length-k window around its corresponding input symbol.*[18]

Each of PLP's generalizations is interpretable as a rule of the form (50) with a target context (cad) of finite length $|cad|$,[19] and under simultaneous application (cf. § 2.3.4).

(50) $a \to b \;/\; c \,\_\_\, d$

Chandlee (2014, p. 41) provides an algorithm for constructing, from any such rule (i.e., with finite target context and under simultaneous application), a Finite State Transducer with the necessary and sufficient automata-theoretic properties of an ISL map. Consequently, if Chandlee's algorithm is a valid *constructive proof*, it follows that each generalization that PLP constructs describes an ISL map. When these are combined into a grammar, it is unknown whether the resulting grammar is also ISL because it is unknown whether ISL maps are closed under composition (Chandlee, 2014, p. 149).

[17] Rogers *et al.* (2013) add word-initial '⋊' and word-final '⋉' markers to Σ. We assume the learner's segment inventory already contains symbols for syllable and word boundaries.

[18] Length k includes the corresponding input symbol.

[19] Under the realistic assumption that input strings are of finite length.

A.2. Differences between PLP and MGL

PLP differs in several ways from the MGL model of Albright & Hayes (2002, 2003). Note that PLP is designed to learn phonology, while MGL was designed for producing English past-tense inflections from verb stems, though it can be extended to other settings.

A.2.1. Generalization Strategy

PLP and MGL use different generalization strategies. PLP generalizes as *locally* as possible and MGL generalizes as *conservatively* as possible. As discussed in § 1.1 and tested in § 4.2, we believe that PLP's generalization strategy is better-supported by studies of human learning.

A.2.2. Number of Rules

Another difference between PLP and MGL is the number of rules they generate. For German syllable-final devoicing at a vocabulary size of 400 (§ 4.3), PLP learns a single rule (51).

(51) [+voi,−son] → [−voi] / __]$_\sigma$

In contrast, MGL learns 102 rules for where devoicing should take place and 4138 for where it should not. An example of the former is (52a) and the latter (52b) (both are presented with the extensions of the natural classes for clarity). Rules like (52b) are learned because not every word involves devoicing, and thus MGL needs such rules in order to produce those words (§ A.2.3).

(52) a. g → k / {a,e,i,o,u,y,ø,æ,ɔ,ɘ,ɛ,ɪ,ʊ} __]$_\sigma$ #
⋮
b. ∅ → ∅ / {f,k,p,t,x}]$_\sigma$ #
⋮

A.2.3. Production

MGL may produce multiple candidate outputs for an input, because every rule that applies to the input generates a candidate output. The quality of a candidate output is 'the confidence of the best rule that derives it' (Albright & Hayes, 2002, sec. 3.2). We used the candidate with the highest confidence as MGL's prediction. This differs from PLP's production (§ 2.3.4), which applies all rules (here just one) in order. This difference is not significant when learning a single phonological process, but it is not straight-forward to use MGL to learn multiple processes simultaneously. For instance, in § 4.4, for input /ɪnsɛkt-z/, MGL may have rule(s) for vowel nasalization that produce the candidate *[ɪ̃nsɛktz] and rule(s) for pluralization that produce the candidate *[ɪnsɛkts]. However, MGL does not provide a mechanism to apply *both* rules to produce the correct output [ɪ̃nsɛkts].

A.2.4. Natural Classes

MGL's natural class induction differs from PLP's in two ways. First, MGL does not form natural classes for every part of a rule. For example, the two rules in (53a) will combine to form a third (53b)—and similarly for (53c) and (53d)—but the rules (53b) and (53d) will not combine to form (53e) because only contexts (not targets) are merged.

(53) a. ə → ə̃ / __ n
ə → ə̃ / __ m
b. ə → ə̃ / __ {n, m}
c. ʌ → ʌ̃ / __ n
ʌ → ʌ̃ / __ m

d. ʌ → ʌ̃ / __ {n, m}

e. {ə, ʌ} → [+nas] / __ {n, m}   (formed by PLP but not MGL)

Moreover, when rules are combined, the new rule and the original rules are all retained. In contrast, PLP will construct (53e), and only it will be present in the grammar (§ 2.3.1).

Second, when MGL creates natural classes for a set of segments, it retains *all* features shared by those segments, whereas PLP only retains those needed to keep the rule satisfactorily accurate. Thus, for (54a), MGL will construct (54b) while PLP will construct (54c).

(54) a. ə → ə̃ / __ {n, m}

b. ə → ə̃ / __ [+ant, +cons, +lab, +nas, +son, +voi, −back, −cg, −cont, −cor, −delrel, −hi, −lat, −lo, −long, −round, −sg, −syl, −velaric]

c. ə → ə̃ / __ [+nas]

Consequently, PLP will correctly extend ə-nasalization when preceding ‘ŋ,’ but MGL will need to wait for such an instance in the training data before constructing the full generalization.