P-Map Biases Enable Joint Learning of Allomorphy and Phonology

LEVI DRISCOLL
MIT

1. Introduction. The task of learning unfaithful mappings is multifaceted and complex. A learner must be able to identify morphemes, recognize that two superficially distinct morphemes are actually derived from the same lexical item, and propose a grammar that oversees such unfaithful mappings. Suppose a child is learning a language in which the voicing of initial obstruents in suffixes is predictable as evidenced by the existence of forms like *patak-si* and *patag-zi* and total absence of words like **patak-zi* or **patag-si*. In such cases, we expect the child to conclude that *-si* and *-zi* are in fact the same lexical item and install a mechanism in their phonological grammar to account for the distribution of the voiced and voiceless variants. When the child encounters the forms *patag-da* and *patak-na*, should the learner treat *-na* as a nasal alternant of -da that follows voiceless obstruents?

Steriade (2008) observes that nasalization (*d → n) and lenition (*z → r) are typologically unattested repairs to prohibitions on final voiced obstruents. Such a finding suggests that nasal alternations in the above context would likewise be unattested, and the learning problem is resolved by storing *-da* and *-na* as distinct suffixes. These typological facts are codified in phonological theory by the P-Map, a bias in the ranking of faithfulness constraints based on the perceptual distance between input segments and candidates for the output of that segment. Because [t] is perceptually more similar to [d] than [n], the P-Map enforces a ranking that yields devoicing (d → t) over nasalization (d → n). There is some uncertainty about the strength of the P-Map bias—whether such repairs are categorically and universally prohibited by a strong bias or whether the P-Map imposes a soft bias that can be overcome in the face of sufficient evidence—but the core principle remains that the P-Map predisposes grammars to favor repairs that maximize perceptual similarity of unfaithful mappings.

The P-Map bears on the learning problem illustrated above; a learner endowed with the P-Map will never acquire *-da* ~ *-na* alternations in this context because constraint rankings that produce such alternations are explicitly made impossible by the strongest interpretation of the P-Map. A comparable learner without access to the P-Map in principle allows any alternations and may encounter difficulty reconciling (de)voicing and nasalization repairs to voicing disagreement in the same grammar.

The objective of this study is to build an unsupervised learner that can deduce a set of stems and affixes and a phonological grammar when presented with a corpus of surface forms to which some phonological processes have applied. The central question under investigation is whether introducing a learning bias informed by the P-Map enables the learner to identify related surface forms and converge on a grammar that replicates alternations found in training data more reliably and accurately than a learner that need not adhere to the conditions imposed by the P-Map. Concretely, this bias is implemented as a condition on the P-Map learner that the weight of certain faithfulness constraints must be a smaller value than the weight of other faithfulness constraints. This condition is absent for the unbiased learner, which may generate any weight for any constraint.

This model is novel in that it is tasked with segmenting a set of surface forms into proposed morphemes, determining whether these morphemes are related, and generating a constraint-based grammar with no prior information about meanings associated with forms or phonological processes that may have applied to the data. Other models are often primarily concerned with segmentation (Goldsmith 2000 et seq.), learn phonology given perfectly segmented input (Boersma & Pater 2016), or learn phonological rules rather than constraint weights or rankings (Albright & Hayes 2003, Calamaro & Jarosz 2015). The present model attempts to find a set of constraint weights that is consistent with the full set of alternations that it must recover from the data as it learns allomorphy.

Using pseudo-randomly generated languages, I demonstrate that a P-Map biased learner is more effective than its unbiased counterpart. Specifically, the P-Map learner identifies grammars that account for alternations more often, and is able to reproduce the paradigms from its input.

2. The Model. This model consists of two major components: a morphological parser and a phonological learner. The efficacy of the model is tested by a language generating algorithm that produces two sets of strings, concatenates them, and applies an ordered series of phonological operations to the concatenated form to provide the learner with its training data. The morphological parser identifies recurring strings in the training data, separates the concatenated forms, and constructs a lexicon. The phonological learner proposes sets of constraint weights to account for alternations and prunes redundant entries from the lexicon that can be derived from the proposed grammar.

Note that the model follows different procedures for language generation and phonological learning; namely the input to the model is constructed with ordered rules while the learner uses weighted constraints as in Harmonic Grammar (HG). While the decision to use HG in the learning stage of the model is significant and intentional, the use of ordered rules to modify the input to the model is inconsequential. The reasoning is no deeper than the fact that ordered rules are simpler to implement. This disparity is unproblematic as there are no opacity issues at play and the theoretical questions of interest are solely concerned with the learning procedure.

2.1. Language Generation. An adequate and direct test of the utility of P-Map biases in simultaneous learning of morphology and phonology requires the languages being learned to have certain properties. First, there must be genuine instances of allomorphy that are favored by P-Map biases: in this case, progressive voicing agreement. Second, there must be accidentally similar but distinct lexical items representing potential alternations that are dispreferred by the P-Map: nasalization and rhotacization repairs triggered by voicing markedness[1]. These morphemes serve as items that may trick a learner without knowledge of the P-Map to misidentify allomorphs. These design features guide the language generation algorithm.

Each time the model runs, a pseudo-randomly generated language consisting of 100 stems and 10 suffixes is produced. Stems may be between one and three syllables long; all monosyllabic stems have the shape CVC. In polysyllabic stems, syllables are either CV or CVC with a 65% bias in favor of open syllables. Suffixes can take the shape C, V, CV, or VC. Of the 10 suffixes, the first two are guaranteed to have near identical counterparts that probe issues related to the P-Map. These suffixes must be consonant-initial, and the initial consonant is chosen from a dictionary that maps a sonorant to an appropriate obstruent and vice versa (e.g. the dictionary keys 'm' and 'z' are associated with the values 'b' and 'r'). Thus if /-mu/ and /-z/ are generated as the first two suffixes, the language also automatically has /-bu/ and /-r/ (I will refer to such pairs of morphemes as 'decoys'). This guarantees that the learner will be forced to confront the possibility of typologically unattested repairs to voicing disagreement and final voiced obstruents. A learner informed by the P-Map knows that [-bu]~[-mu] and [-z]~[-r] are not plausible alternations whereas a learner without this bias considers nasalization and lenition reasonable options to remedy markedness issues related to voicing. The decoy alternations provide an opportunity to assess whether entertaining implausible allomorphy obscures an unbiased learner's ability to recover the input grammar relative to a P-Map learner, which rules out such alternations *a priori*.

With stems and suffixes generated, 500 unique combinations of stems and suffixes are concatenated and undergo final epenthesis, final devoicing, and progressive voicing assimilation as shown in (1). The epenthesis rule inserts a word-final [a] when the final two segments of a word are consonants (Ø → a / CC_#). Final devoicing makes voiced obstruents voiceless in word final position ([+voi, -son] → [-voi] / _#). Progressive voicing assimilation spreads the voicing

---

[1] This does not suggest that nasalization and rhotacization are universally unattested alternations or that such repairs always run afoul of the P-Map. Nasalization and rhotacization are found in e.g. Ponapean (Rehg 1981) and Hausa (Newman 1996) respectively. These cases differ from the present discussion because the nasal and rhotic repairs address different markedness concerns or occur in different contexts.

specification from a preceding obstruent to a following adjacent obstruent that disagrees in voicing ([αvoi, -son] → [-αvoi] / $C^{[-\alpha voi,\ -son]}$_).

(1)		/kazap/ + /d/	/rikev/ + /d/	/suma/ + /d/
	Concatenation	kazapd	rikevd	sumad
	Epenthesis	kazapda	rikevda	---
	Devoicing	---	---	sumat
	Agreement	kazapta	---	---
	Learner Input	[kazapta]	[rikevda]	[sumat]

These surface forms serve as input to the learner, with no access to information about underlying forms or morpheme boundaries. From this list of 500 items, the model must discover how to separate the morphologically complex forms, identify morphemes derived from the same lexical item, and propose a grammar to modulate alternations between morphemes that are determined to be related. The final grammar acquired by the model crucially relies on which morphemes the learner decides are related.

2.2. Segmentation and Morpheme Learning. The first step in the learning procedure is segmenting the morphologically complex input. The model begins segmentation by finding the longest common prefix (LCP) in each form. LCPs are stored as proposed stems, and the remainder of each form is categorized as a suffix[2]. While this procedure guarantees that each item in the corpus is fully parsed, it may erroneously classify accidentally identical substrings as morphemes. For instance, the LCP in the pair *poptimdafra* and *popra* is *pop-*; *poptimdafra* would then be segmented as *pop-timdafra*. To guard against such errors, every proposed stem is combined with every proposed suffix. If none of these combinations produce a form attested elsewhere in the corpus (i.e., *poptimdafra* cannot be used as evidence that *pop-* is a stem and -*timdafra* is a suffix), the proposed stem/suffix is eliminated from consideration as a morpheme and plays no further role in the model. The proposed morphemes that survive this pruning procedure are then stored as stems and suffixes along with their signatures (Goldsmith 2000).

The results of segmentation in this model are ultimately imperfect. The learner generally acquires most if not all stems and affixes, but it additionally learns morphological pieces that

[2] Effectively the model knows from distributional information that a set of recurring strings occurs before others, a different set of recurring strings occurs after others, and stores them accordingly. Whatever theoretical terms we choose to discuss them makes no difference to the function of the learner.

were not part of the language generation process. This occurs most often when two suffixes begin with the same consonant and end with different vowels. In a language with suffixes -*se* and -*su*, the common [s] at the beginning of the suffixes will be counted as part of the LCP and (incorrectly) stored as part of the stem. The learner will therefore identify -*se*, -*su*, -*e*, and -*u* as well as the stem and the stem plus [s] from the suffix (eg. *pop*, *pops*). Considering that segmentation is not the primary objective of the model, but a necessary stepping stone in a larger process, the imperfect segmentation is sufficient.

This is further unproblematic in this model because imperfect segmentations of this type can be corrected at a later stage via allomorphy. When the initial segment of the suffix is counted as part of the LCP and -*se*/-*e* are compared as potential allomorphs, the phonological learner is likely to determine that this is a licit alternation and store -*se* and -*e* as two instances of the same lexical item. As an allomorph of -*se*, the erroneously identified suffix -*e* will be marked as a redundant entry and pruned from the lexicon because its distribution is phonologically predictable.

2.3. Learning Allomorphy and Phonology. After identifying morphemes, the model assesses whether any of these morphemes are related. Pairs of suffixes with a Levenshtein distance[3] of one are considered as possible allomorphs. To establish allomorphy, the learner calculates whether there is a set of HG constraint weights that can derive suffix /-α/ from suffix /-β/ if /-β/ is combined with a stem in /-α/'s signature. This is illustrated in (2).

(2)	Hypothesis:	**-ta** signature: [sakep, gon, …]	Desired winner:	**rob-ta** → robna
	[-ta] ≈ [-na]	-na signature: [**rob**, vinaf, …]	Candidates:	[robta, robna, ropta, robda, robt]

Constraint weights are sampled at random and EVAL checks whether that set of constraint weights produces the desired winner. If the intended candidate does not succeed, these weights are discarded and have no influence on future iterations on constraint weight generation. This process repeats until the learner succeeds at finding constraint weights that produce the alternation. If the learner succeeds, the suffixes are grouped together as allomorphs and new weights are no longer

[3] Levenshtein distance is a measure of similarity between two strings representing the number of changes needed to make one string identical to another. For example, -t/n and -t/-at both have a Levenshtein distance of one—a single substitution is needed to convert -t to -n, and adding a single character can convert -t to -at.

generated. If the learner fails to generate weights that cause the desired alternation within 50,000 attempts, it exits the loop and concludes that these forms are unrelated.[4]

Because the biased learner is informed by the P-Map, the model will always fail to select *robna* over *ropta* or *robda*, and will never consider these pairs allomorphs. Conversely, the unbiased learner may generate weightings that allow nasalization as a repair to voicing disagreement and the identification [-*ta*] and [-*na*] as allomorphs.

The model then attempts to account for the full set of alternations from the allomorphy learning stage with a single set of constraint weights. As before weights are sampled randomly up to 50,000 times, and the unbiased learner is free to consider any weights while the biased learner can only generate weights that are compliant with the P-Map. If the learner outputs constraint weights within these 50,000 attempts, these weights are used to evaluate whether the grammar was successfully learned. It should be noted that each time the model runs, the learner will produce at most one grammar. Once there is a hypothesis (a set of constraint weights) that accounts for all available data, there is no incentive to revise that hypothesis from the learner's perspective.

The upcoming subsections detail more explicitly the components of the phonological learning model, including how candidates to compete against desired winners are produced, how constraints are formally implemented, and the space of possible constraint weights.

2.4. GEN. This model autonomously generates candidates to evaluate against constraints. Suppose the learner encounters the forms [kobda], [kobi], [numut], and [numui] and identifies the morphemes [kob], [-t], [-da], [numu], and [-i]. It then asks if [-t] could plausibly alternate with [-da] and assesses this possibility by concatenating [-t] with a stem that occurs with [-da], [kob]. These morphemes become arguments of a candidate generating function which produces the competing outputs shown in (3), and marks [kobda] as the intended winner.

	Input	Candidates	Winner
(3)	Input	Candidates	Winner
	kobt	kobt, kobda, kobd, kob, gobd, kopt, gobt	kobda

The function generates a faithful candidate [kobt] and the target output [kobda] as well as candidates in which the last segment is deleted from the faithful candidate [kob] and the intended

[4] This decision was a practical matter based on my inability to code more sophisticated functionality (e.g. error-driven learning) into the model. The design of this aspect of the model should not be interpreted as an endorsement or argument that human children learn phonological grammars by rearranging constraints at random.

winner [kobd]. Next are candidates with all voiced obstruents [gobd] and all voiceless obstruents [kopt]. The function then proceeds through each segment in the input and swaps the voicing specification of each obstruent one at a time, yielding the novel candidate [gobt] and duplicates [kopt] and [kobd].

Notably absent from the candidate set above are the nasal repairs [kobna] and [kobn]. The parameters for generating candidate sets were designed with the target grammar and the typology of repairs to voicing-related markedness in mind. Typology informs us that the expected repair to voicing issues is a voicing alternation, and the candidate sets are therefore focused on considering most (though admittedly not all) permutations of voicing specifications of obstruents. This is reflected in the candidate sets by prompting the learner to determine whether a nasal repair is superior to a voicing alternation only when the nasal repair is intended to win, rather than asking in all cases whether a voicing alternation is better than a nasal repair. While this does have some downstream effects (discussed in §3.2.2), their impact does not appear to be substantial.

2.5. CON. The model contains the following constraints and knowledge of the relevant natural classes: IDSON, IDNAS, IDCONT, AGRVOI, IDVOIROOT, IDVOIAFF[5], *D#, DEP, MAX, *VVV, IDPLACE, IDVOW, and *CC#. Constraints are implemented as functions that accept strings and a weight as arguments and return the number of times the string violates the constraint multiplied by the constraint weight. An example constraint written in pseudo-code is shown in (3).

(4)
```
def AgrVoi(output, weight):
    for adjacent segments x, y in output:
        if x and y in OBSTRUENTS:
            if (x in VOICED and y in VOICELESS) or (x in VOICELESS and y in VOICED):
                violations += 1
    cost = violations * weight
    return cost
```

[5] IDVOIROOT and IDVOIAFF in this model penalize voicing changes according to the index of a given segment; if the final two segments are both consonants, IDVOIROOT is violated when nonfinal segments change their voicing specification and IDVOIAFF only considers voicing changes of the final segment. In all other cases, IDVOIROOT is concerned with everything to the left of the final two segments, and IDVOIAFF penalizes the final two segments. Given the shape of morphemes in the languages generated here, these indices are essentially equivalent to genuine theoretical distinctions between root and affix. I pause here to note, however, that there is no mechanism in this model to endow certain strings with the theoretical status of root and others the status of affix to be later leveraged by EVAL.

These functions naturally differ slightly depending on whether they represent markedness or faithfulness constraints. Markedness constraints take a single string argument, the output, while faithfulness constraints require two string arguments, input and output.

2.6. Constraint weights and EVAL. In the case of the unbiased learner, constraint weights are randomly assigned any integer or non-integer value between 0 and 5, meaning constraint weights are chosen from an infinite space of possible values. The P-Map learner's weight assignment process is similar (any random value between 0 and 5 for *most* constraints), except that the weights of IDVOIROOT and IDVOIAFF are assigned last. Whichever constraint among IDSON, IDNAS, and IDCONT has the least weight decides the maximum value of IDVOIROOT and IDVOIAFF. For example, if the weights of IDSON, IDNAS, and IDCONT were 2, 3, and 4, IDVOIROOT and IDVOIAFF would be assigned any random value between 0 and 2. The weights of IDVOIROOT and IDVOIAFF relative to each other are not fixed. Once constraint weights and candidates are generated, weights are paired with their corresponding constraints, and candidate evaluation proceeds as normal in an HG framework.

3. Simulation and Results. 15 languages were generated according to the procedure in §2.1 and each one was given to the biased and unbiased version of the model for a total of 30 runs. All runs of the model identified at least 97 of the 100 stems that comprise each language. Overall, the biased learner converged on a grammar for 12 of the 15 languages whereas the unbiased learner only produced a grammar for 7 of the 15 languages. Where the models differ most drastically is in the paradigms the models produce when they encounter novel data. While the biased learner's paradigms repair voicing disagreements uniformly, the unbiased learner deploys various repairs to voicing disagreement depending on the identity of adjacent consonants. This suggests that the unbiased model is uniquely characterized by widespread ganging. These contrasts are explained in greater detail in the upcoming subsections.

3.1. Comparison of Models. One way to compare the models' performance is to examine how successful the model was in identifying allomorphy. Due to the way that suffixes were generated, not every target language had exactly the same number of suffixes that exhibited allomorphy, but all had between 2 and 7. The acquisition of allomorphy is evaluated as successful if all variants derived from an underlying suffix are grouped together with no extraneous forms. As (1) shows, the underlying suffix /-d/ is expected to appear as [-t], [-ta], and [-da]. Anything other than the grouping [-t], [-ta], [-da] constitutes a failure in the acquisition of a set of allomorphy; this applies equally to the biased and unbiased learners. If the learner incorporates decoy allomorphy into a set (e.g., [-t], [-ta], [-da], [-na]), this also counts as a failure to learn a set of allomorphy.

Pairwise comparison of competing learners' performance on a single language shows that the unbiased learner correctly identified more sets of related forms than the biased learner twice; in all other cases, the biased learner performed as well as or better than the unbiased learner. These cells are shaded in (5), which articulates how many sets of allomorphs were found by each version of the model for each language (expressed as the number of sets identified over the number attested in the training data).

Another way to compare these models is whether the learner converged on a grammar for a given language, and whether that grammar faithfully replicates the sequence of phonological processes in the training data. This is shown in (5) with a ✓ if the learned grammar replicates the training pattern, Y if the learner produced a grammar that fails to reproduce the training pattern, and N if no grammar is produced at all.

(5)

Language #		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Allomorphs	**Biased**	2/6	5/6	3/4	5/6	4/6	1/2	3/6	6/7	5/5	3/6	3/5	3/4	2/4	5/7	3/4
	Unbiased	1/6	2/6	3/4	1/6	4/6	1/2	4/6	3/7	3/5	1/6	1/5	3/4	0/4	3/7	4/4
Grammars	**Biased**	✓	✓	✓	N	✓	✓	N	Y	✓	N	Y	✓	Y	✓	✓
	Unbiased	N	N	Y	N	N	Y	N	Y	Y	N	Y	Y	N	N	Y

The next matter to consider in evaluating the success of each version of the model is whether the set of constraint weights that is consistent with the allomorphy acquired by the learner replicates the phonological operations in (1) which produced the learners' input. Out of the 12 languages for which the P-Map learner produced a grammar, 9 of those grammars mimicked the paradigm of alternations in the training data. Conversely, none of the grammars learned by the unbiased version of the model reproduced the patterns from its input. Furthermore, no two grammars generated by the unbiased learner yield the same paradigm when confronted with novel data. The erroneous paradigms acquired by the learners are displayed in (6).

(6)

	apa-z	apak-z	apag-z	apa-d	apak-d	apag-d
Expected Forms	apas	apaksa	apagza	apat	apakta	apagda
Biased Learner # 8	*<u>apaz</u>	apaksa	apagza	*<u>apad</u>	apakta	apagda
Biased Learner # 11	apas	*<u>apak</u>	apagza	apat	*<u>apak</u>	apagda
Biased Learner # 13	*<u>apaza</u>	apaksa	apagza	*<u>apada</u>	apakta	apagda
Unbiased Learner #3	*<u>apar</u>	*<u>apakra</u>	apagza	*<u>apar</u>	*<u>apakra</u>	apagda
Unbiased Learner #6	*<u>apaza</u>	*<u>apakra</u>	apagza	*<u>apada</u>	*<u>apakra</u>	apagda
Unbiased Learner #8	*<u>apar</u>	*<u>apak</u>	apagza	*<u>apar</u>	*<u>apak</u>	apagda
Unbiased Learner #9	*<u>apaza</u>	*<u>apakra</u>	apagza	*<u>apada</u>	*<u>apak</u>	apagda
Unbiased Learner #11	*<u>apar</u>	*<u>apak</u>	apagza	*<u>apan</u>	*<u>apak</u>	apagda
Unbiased Learner #12	*<u>apaza</u>	*<u>apakra</u>	apagza	*<u>apada</u>	apakta	apagda
Unbiased Learner #15	*<u>apaza</u>	apaksa	apagza	*<u>apada</u>	apakta	apagda

The patterns in (6) illustrate the effect of the P-Map quite clearly. As expected, nasalization and lenition are never deployed as repairs in the biased learner. However when these repairs are not specifically forbidden, they are quite robustly attested; UL-15 is the only unbiased paradigm that does not include alternations made impossible by the P-Map. (6) reveals an additional unintended asymmetry between the biased and unbiased learners. Ganging (as exhibited by UL-9, for instance) almost exclusively affects the paradigms of the unbiased learner, even though nothing about the P-Map bias explicitly precludes this possibility (I expand on this topic in §4).

3.2. Error Analysis. By far the simplest of the incorrectly generated patterns is BL-8. Here, the only process that fails to apply is final devoicing. The only evidence the learner has that such a process occurs is that there are no final voiced obstruents anywhere in the corpus. As (7) illustrates, the grammar that generates the training data produces identical outputs for single-obstruent suffixes whether they are voiced or voiceless.

(7)

	/kazap/ + /d/	/rikev/ + /d/	/suma/ + /d/
Concatenation	kazapd	rikevd	sumad
Epenthesis	kazapda	rikevda	---
Devoicing	---	---	sumat
Agreement	kazapta	---	---
Learner Input	[kazapta]	[rikevda]	[sumat]

	/kazap/ + /t/	/rikev/ + /t/	/suma/ + /t/
Concatenation	kazapt	rikevt	sumat
Epenthesis	kazapta	rikevta	---
Devoicing	---	---	---
Agreement	---	rikevda	---
Learner Input	[kazapta]	[rikevda]	[sumat]

Since these forms never surface with a final voiced obstruent and the underlying voicing specification has no impact on the output, the learner has no reason to store these suffixes as underlyingly voiced. There are therefore no alternations to suggest the relative weights of IDVOI constraints and *D#. In the absence of a default M»F ranking bias, the learner fails to replicate the pattern from the training data accidentally because the weight of IDVOIAFF is greater than the weight of *D# by random chance. While including such a bias would likely have resolved this particular error, it is not clear that it would have any bearing on the alternations of interest: the choice between voicing alternations, nasalization, and rhotacization.

Understanding what went wrong with attempts to learn the other languages is a more involved endeavor. It is instructive at this juncture to separate errors into two categories: languages in which decoys were learned as allomorphs and languages in which they were not.

3.2.1. Errors Without Decoy Allomorphy. It is a foregone conclusion that the biased models never acquired the decoy allomorphs, so we will begin with these.

In BL-11, word-final clusters are remedied by epenthesis when the consonants agree in voicing and by deletion when they disagree. Ganging prohibits the learner from reproducing the alternation pattern from the training data. Although the relevant constraints are appropriately ordered to produce the expected paradigm (*CC#, MAX > DEP and AGRVOI, IDVOIRT > IDVOIAFF), the combined violation cost of DEP and IDVOIAFF is greater than a violation of MAX. Thus epenthesis occurs only if there are no additional markedness violations to repair (i.e., voicing disagreement).

BL-13 and UL-15 have the same paradigm with epenthesis instead of final devoicing. In this case the weight of *D# is greater than that of IDVOIAFF, but the weight of DEP is less than the weight of IDVOIAFF.

Alongside the biased learners, a number of unbiased learners produced grammars without acquiring decoy allomorphs. These include UL-3, 6, and 12 (as well as UL-15 above).

The errors in UL-3, though unattested in natural languages, are relatively straightforward to derive. Final voiced obstruents and voicing disagreements in UL-3 are repaired by replacing the offending segment with [r], regardless of its underlying continuancy. The combined weights of IDSON and IDCONT are less than the weight of IDVOIAFF and *D# individually. This results in [r]-substitution when voiced obstruents are in final position (*D# > IDSON + IDCONT) or adjacent obstruents disagree in voicing (IDVOIAFF > IDSON + IDCONT). This is particularly unexpected since the allomorphy UL-3 acquired did not require IDSON to have a lower weight than IDVOI constraints.

UL-6's paradigm differs slightly from UL-3. Here, IDVOIAFF and IDSON have greater weights than *D#, which prevents the expected final devoicing or lenition as in UL-3. Because DEP has the lowest weight of the faithfulness constraints, final voiced obstruents are repaired with epenthesis instead. Among IDVOIAFF and IDSON, IDSON assigns a less heavy penalty which yields the mappings apakz/apakd → apakra.

In UL-12, a single final voiced obstruent is repaired by epenthesis. If the final consonant in a cluster is a fricative, the fricative undergoes lenition (apakz → apakra) because IDVOIAFF > IDSON. If the final consonant is a stop, it simply agrees in voicing with the preceding consonant (apakd → apakta). Although IDSON has a lower weight than IDVOIAFF, the additional cost of violating IDCONT for [r]-substitution or the cost of violating IDNAS for nasalization makes devoicing the preferred option.

3.2.2. Errors With Decoy Allomorphy. Among the unbiased learners, three of them acquired the decoy allomorphs and produced a grammar: UL-8, 9, and 11.

UL-8 learned the intended decoy allomorphy pair -z/-r as well as an unintended, equally implausible alternation -g/-n (instead of -t/-n). This model is similar to BL-11, except that a single final voiced obstruent is replaced with [-r] because the cost of violating IDSON is lesser than that of IDVOIAFF. Given this state of affairs, we might then expect voicing disagreements in final clusters to be repaired via epenthesis and lenition apakz → apakra instead of deletion apakz → apak. In UL-8 the weight of MAX is less than the combined weights of IDSON and DEP, which makes deletion the preferred repair.

A nasal alternation -b/-m was acquired in UL-9. After a single final voiced obstruent, UL-9 epenthesizes a vowel which is straightforward to characterize (*D#, IDVOIAFF > DEP). When a final cluster ends with a voiced fricative, the fricative lenites and a final epenthetic vowel is inserted (apakz → apakra). If a final cluster ends with a voiced stop, the stop deletes (apakd → apak). The mapping apakz → apakra violates only IDSON and DEP which is a lower cost than violating IDVOIAFF and DEP (apakz → apaksa) or MAX alone (apakz → apak). Mapping apakd to apakra additionally entails a violation of IDCONT, at which point deletion is the least costly repair (apakd → apak).

UL-11 similarly learned nasalization allomorphy -t/-n. This alternation is reflected in the paradigm; UL-11 nasalizes final voiced stops (apad → apan) and lenites final voiced fricatives (apaz → apar) because IDSON and IDNAS combined are less costly than any other repair. This was demanded of the grammar because the decoys -d/-n were identified as related forms. The weights of MAX is less that of DEP plus IDSON, so final clusters are met with deletion (apakd/z → apak) instead of epenthesis and nasalization/lenition, regardless of the underlying continuancy of the suffix.

We can see a common tendency among the models that acquired decoy allomorphs to exhibit asymmetrical paradigms. Specifically, decoy allomorphy appears to create a predisposition to employ different repairs depending on the continuancy of the suffix. A further effect of acquiring decoy allomorphy has been invisible throughout this section of the paper so far, which could obscure how detrimental decoy allomorphy is to acquiring phonological grammars. Generally, the models that acquired decoy allomorphs failed to learn any grammar. UL-1, 2, 4, 7, 10, 13, and 14 acquired at least one pair of decoys and were unable to reconcile the competing demands of voicing alternations and nasalization/lenition. In the majority of cases, the decoy allomorphs impaired the biased learners' ability to learn phonological grammars.

An unanticipated quirk of these paradigms is that alternations learned from decoy allomorphy do not necessarily correspond to the repairs applied to novel data. For instance, UL-9 acquired a nasal decoy allomorphy pair, but voicing disagreement is resolved by rhotacization, not nasalization. Although this is unusual, there are two reasons that I do not consider the disconnect between allomorphy and repair strategies to be the fault of the unbiased learner.

First, nasalizing an obstruent entails violating at least two constraints—IDSON and IDNAS—while rhotacization of obstruents minimally violates only a single constraint, given a continuant input. Nasalization faces an inherent disadvantage against rhotacization; as (8) shows, rhotacization is a valid repair to underlying stops and fricatives both, but nasalization is only available to underlying stops.

(8)

/d/	IDSON	IDCONT	IDNAS
[n]	*		*
[r]	*	*	

/z/	IDSON	IDCONT	IDNAS
[n]	*	*!	*
☞ [r]	*		

This problem derives from general facts about feature theory. From this fact alone, we should expect to see rhotacization in (6) more frequently than nasalization.

The other reason that I hold the unbiased learner blameless for the unusual relationship between decoy allomorphy and paradigms is that the candidate generating algorithm did not allow the learner to meaningfully distinguish between nasalization and rhotacization. Because the algorithm does not generate nasalized or rhotacized candidates, the learner compared nasalization to (de)voicing and rhotacization to (de)voicing separately. Without direct competition between nasalization and rhotacization, the model did not learn that these were distinct processes from each other, per se. It seems that in practice, acquiring decoy allomorphy informed the unbiased learner of a distinction between IDVOI violations and IDSON violations, where the latter may be implemented as nasalization or rhotacization, depending on the arbitrarily assigned relative weights of IDCONT and IDNAS, as well as the continuancy of the underlying obstruent.

4. Discussion. As previously mentioned, the paradigms of unbiased learners in (6) show widespread gang effects. The biased and unbiased learners' performance on Language 9, whose paradigm are repeated in (9), provide a useful case study of ganging in this model.

(9)							
		apa-z	apak-z	apag-z	apa-d	apak-d	apag-d
	BL-9	apas	apaksa	apagza	apat	apakta	apagda
	UL-9	***apaza**	***apakra**	apagza	***apada**	***apak**	apagda

UL-9 is among the unbiased learners that identified decoy morpheme pairs as allomorphs. The proposed allomorphy of UL-9 includes voicing alternations [so]~[zo], [su]~[zu], [tu]~[du], as well as ternary alternations consisting of both voicing and nasalization [pa]~[ba]~[ma]. The other decoy pair, -ru/-su was not acquired as an alternation. This means the learner had conflicting evidence with regard to the relative weights of IDVOIAFF and IDSON. To produce the nasal alternation (pa/ba → ma), the combined weights of IDSON and IDNAS must be lower than IDVOIAFF while avoiding lenition (su → ru) requires that IDVOIAFF's weight is lower than that of IDSON. The only way to arrive at a set of weights that is consistent with these facts is if the individual alternations are the result of ganging, where some additional factor overpowers the difference between the weights of IDSON and IDVOIAFF. We therefore find that these gang effects persist when the learner is confronted with new data: the paradigm we see in (9).

In contrast, BL-9 shows no gang effects because there is no need to reconcile conflicting evidence about the weight of IDVOIAFF with respect to other faithfulness constraints. All of the intended allomorphy was learned (-p/-pa/-ba, -m/-ma, -so/-zo, -su/-zu, -tu/-du) without acquiring nasalization or lenition alternations from the decoy morphemes -b/-m and -ru/-su. The allomorphy learned by BL-9 can be derived simply by assigning a low weight to IDVOIAFF, so no ganging is necessary to find a set of weights that is consistent with the proposed alternations. As a result, the paradigm precisely mirrors the phonological processes from the training data, and the grammar of BL-9 replicates this pattern.

It is clear that the biased model's performance on this learning task was superior. The P-Map learner rarely failed to converge on a grammar and nearly every grammar it learned replicated the alternations from the training data. The unbiased learner failed to learn a grammar for over half of the languages and never managed to reproduce the pattern from the training data. Crucially, the only difference between these models is whether they were forced to adhere to the P-Map. These results suggest that a bias informed by the P-Map substantially aids the process of learning allomorphy and constraint-based phonological grammars.

5. Conclusion. Although this model provides a novel perspective on the difficulties of learning unfaithful mappings, it is not without its deficiencies. At present, there is no quantitative evaluation metric in the model to assess the quality of segmentation. Allowing the model to make numerous attempts at segmenting input and selecting the optimal parse in an MDL framework (Rissanen 1989, Goldsmith 2001) could reduce the margin of error in biased and unbiased learners alike.

In future iterations of this model, the candidate generation algorithm detailed in §2.4 should be revised to include nasalized and rhotacized candidates, as this could have a significant impact on how and whether the unbiased model learns phonological grammars. GEN in principle produces an infinite set of candidates which would consistently reinforce the ranking of certain constraints over others, those relevant here being IDVOI, IDNAS, IDCONT, and IDSON. The present method of generating candidates effectively assumes voicing alternations are the best repair and informs the learner how best to deploy these alternations unless it is directly presented with alternatives such as nasalization.

An updated model would also ideally extract information from its unsuccessful attempts to generate grammars, as in error-driven learning, rather than simply discarding the weights that led to failure. Converting the current system from rejection sampling to a hillclimbing algorithm would be more efficient and guard against the possibility of incorrectly concluding that constraint weights that account for the observed data cannot be found. Updating incorrect constraint weights in accordance with evidence from feedback would be a more faithful simulation of how we believe human children learn grammars, and this would be an objective improvement to the current version of the model.

Human children also presumably make use of a wider range of information in assessing allomorphy than the present model. Although the model collects distributional information about suffixes, such data do not inform the categorization of a suffix as a distinct lexical item or an allomorph. Rather than leaving these determinations entirely to the phonology, the model could check whether pairs of suffixes occur with the same stem and mark those with no overlap in their signatures as allomorphs.

A model equipped with these upgrades would likely be able to investigate a larger variety of learning problems. While parsing ambiguity was deliberately foregrounded as a problem in this model, it remains an inescapable challenge for computational models and human language acquisition. One of the strengths offered by this model is how allomorphy addresses difficulties with segmentation. In the most common cases of missegmentation, the learner proposes that a missegmented suffix has an allomorphic relationship with the actual lexical item it is derived from.