Learning Phonotactics from Linguistic Informants

Canaan Breiss*
University of Southern California
cbreiss@usc.edu

Alexis Ross*
MIT
alexisro@mit.edu

Amani Maina-Kilaas
MIT

Roger Levy
MIT

Jacob Andreas
MIT

Abstract

We propose an interactive approach to language learning that utilizes linguistic acceptability judgments from an informant (a competent language user) to learn a grammar. Given a grammar formalism and a framework for synthesizing data, our model iteratively selects or synthesizes a data-point according to one of a range of information-theoretic policies, asks the informant for a binary judgment, and updates its own parameters in preparation for the next query. We demonstrate the effectiveness of our model in the domain of phonotactics, the rules governing what kinds of sound-sequences are acceptable in a language, and carry out two experiments, one with typologically-natural linguistic data and another with a range of procedurally-generated languages. We find that the information-theoretic policies that our model uses to select items to query the informant achieve sample efficiency comparable to, and sometimes greater than, fully supervised approaches.

1 Introduction

In recent years, natural language processing has made remarkable progress toward models that can (explicitly or implicitly) predict and use representations of linguistic structure from phonetics to syntax (Mohamed et al., 2022; Hewitt and Manning, 2019). These models play a prominent role in contemporary computational linguistics research. But the data required to train them is of a vastly larger scale, and features less controlled coverage of important phenomena, than data gathered in the course of linguistic research, e.g. during language documentation with native speaker informants. How can we build computational models that learn more *like linguists*—from targeted inquiry rather than large-scale corpus data?

We describe a paradigm in which language-learning agents interactively select examples to learn from by querying an **informant**, with the goal of learning about a language as data-efficiently as possible, rather than relying on large-scale corpora to capture attested-but-rare phenomena. This approach has two important features. First, rather than relying on existing data to learn, our model performs **data synthesis** to explore the space of useful possible data-points. But second, our model can also **leverage corpus data** as part of its learning procedure by trading off between interactive elicitation and ordinary supervised learning, making it useful both *ab initio* and in scenarios where seed data is available to bootstrap a full grammar.

We evaluate the capabilities of our methods in two experiments on learning *phonotactic grammars*, in which the goal is to learn the constraints on sequences of permissible sounds in the words of a language. Applied to the problem of learning a vowel harmony system inspired by natural language typology, we show that our approach succeeds in recovering the generalizations governing the distribution of vowels. Using an ad-

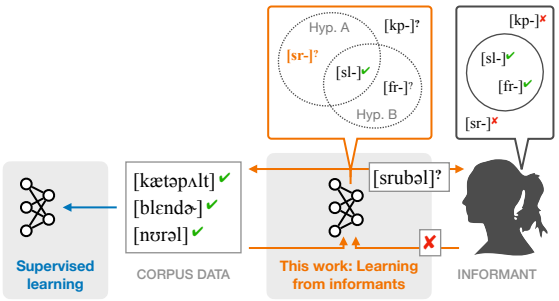

Figure 1: Overview of our approach. Instead of learning a model from a static set of well-formed word forms (left), we interactively elicit acceptability judgments from a knowledgeable language user (right), using ideas from active learning and optimal experiment design. On a family of phonotactic grammar learning problems, active example selection is sometimes more sample-efficient than supervised learning or elicitation of judgments about random word forms.

Both authors contributed equally to this work.

ditional set of procedurally-generated synthetic languages, our approach also succeeds in recovering relevant phonotactic generalizations, demonstrating that model performance is robust to whether the target pattern is typologically common or not. We find that our approach is more sample-efficient than ordinary supervised learning or random queries to the informant.

Our methods have the potential to be deployed as an aid to learners acquiring a second language or to linguists doing elicitation work with speakers of a language that has not previously been documented. Further, the development of more data-efficient computational models can help redress social inequalities which flow from the asymmetrical distribution of training data types available for present models (Bender et al., 2021).

2 Problem Formulation and Method

Preliminaries We aim to learn a language L comprising a set of strings x, each of which is a concatenation of symbols from some inventory Σ (so $L \subseteq \Sigma^+$). (In phonotactics, for example, Σ might be the set of phonemes, and L the set of word forms that speakers judge phonotactically acceptable.) A learned model of a language is a discriminative function that maps from elements $x \in \Sigma^+$ to values in $\{0, 1\}$ where 1 indicates that $x \in L$ and 0 indicates that $x \notin L$. In this paper, we will generalize this to **graded** models of language membership $f : \Sigma^+ \mapsto [0, 1]$, in which higher values assigned to strings $x \in \Sigma^+$ correspond to greater confidence that $x \in L$ (cf. Albright, 2009, for data and argumentation in favor of a gradient model of phonotactic acceptability in humans).

We may then characterize the language learning problem as one of acquiring a collection of pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $x_i \in \Sigma^+$, and $y_i \in \{0, 1\}$ correspond to **acceptability judgments** about whether $x_i \in L$. Given this data, a learner's job is to identify a language consistent with these pairs. Importantly, in this setting, learners may have access to both positive and negative evidence.

Approach In our problem characterization, the data acquisition process takes place over a series of time steps. At each time step t, the learner uses a **policy** π according to which a new string $x_t \in \mathcal{X}$ is selected; here $\mathcal{X}$ is some set of possible strings, with $L \subset \mathcal{X} \subset \Sigma^+$. The chosen string is then passed to an **informant** that provides the learner a value $y_t \in \{0, 1\}$ corresponding to whether x_t is in L. The new datum (x_t, y_t) is then appended to a running collection of (string, judgment) pairs $(\underline{x}, \underline{y})$, after which the learning process proceeds to the next time step. This procedure is summarized in Algorithm 1.

Algorithm 1: Iterative Query Procedure

Input: policy π, total timesteps T
$(\underline{x}, \underline{y}) \leftarrow [\,]$; $t \leftarrow 0$;
while $t < T$ **do**
  $x_t \leftarrow \pi(x \mid \underline{x}, \underline{y})$;
  $y_t \leftarrow \text{informant}(x_t)$;
  append (x_t, y_t) to $(\underline{x}, \underline{y})$;
  $t \leftarrow t + 1$;
end

Conceptually, there are two ways in which a learner might gather information about a new language. One possibility is to gather examples well-formed strings already produced by users of the language (e.g. by listening to a conversation, or collecting a text corpus), similar to an "immersion" approach when learning a new language. In this case, the learner does not have control over the specific selected string x_t, but it is guaranteed that the selected string is part of the language: $x_t \in L$ and thus $y_t = 1$.

The other way of collecting information is to select some string x_t from $\mathcal{X}$, and directly elicit a judgment y_i from a knowledgeable informant. This approach is often pursued by linguists working with language informants in a documentation setting, where their query stems from a hypothesis about the structural principles of the language. Here, examples can be chosen to be maximally informative, and negative evidence gathered directly. In practice, learners might also use "hybrid policies" that compare which of multiple basic policies (passive observation, active inquiry) is expected to yield a new datum that optimally improves the learner's knowledge state. Each of these strategies is described in more detail below.

Model assumptions To characterize the learning policies, we make the following assumptions regarding the **model** trained from available data $(\underline{x}, \underline{y})$. We assume that the function $f : \Sigma^+ \to [0, 1]$ acquired from $(\underline{x}, \underline{y})$ can be interpreted as a conditional probability of the form $p(y \mid x, \underline{x}, \underline{y})$. We further assume that this conditional probability is determined by a set of parameters $\boldsymbol{\theta}$

for which a(n approximate) posterior distribution $P(\boldsymbol{\theta} \mid \underline{x}, \underline{y})$ is maintained, with $p(y \mid x, \underline{x}, \underline{y}) = \int_{\boldsymbol{\theta}} P(y|x, \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid \underline{x}, \underline{y})\, d\boldsymbol{\theta}$.

3 Query policies

In the framework described in Section 2, how should a learner choose which questions to ask the informant? Below, we describe a family of different policies for learning.

3.1 Basic policies

Train The first basic policy, $\pi_{\text{train}}(x \mid \underline{x}, \underline{y})$, corresponds to observing and recording an utterance by a speaker. For simplicity we model this as uniform sampling (without replacement) over L:

$$\pi_{\text{train}}(x|\underline{x}, \underline{y}) \sim U(\{x \in L - \underline{x}\}).$$

Uniform The second basic policy, $\pi_{\text{unif}}(x|\underline{x}, \underline{y})$, samples a string uniformly from $\mathcal{X}$ and presents it to the informant for an acceptability judgment:

$$\pi_{\text{unif}}(x|\underline{x}, \underline{y}) \sim U(\{x \in \mathcal{X}\}).$$

Label Entropy The $\pi_{\text{label-ent}}(x|\underline{x}, \underline{y})$ policy selects the string x^* with the maximum entropy $\mathcal{H}$ over labels y under the current model state:

$$x^* = \arg\max_{x \in \mathcal{X}} \mathcal{H}(y \mid x, \underline{x}, \underline{y}).$$

Expected Information Gain The $\pi_{\text{eig}}(x|\underline{x}, \underline{y})$ policy selects the candidate that, if observed, would yield the greatest expected reduction in entropy over the posterior distribution of the model parameters $\boldsymbol{\theta}$. This is often called the **information gain** (MacKay, 1992); we denote the change in entropy as $V_{\text{IG}}(x, y; \underline{x}, \underline{y})$:

$$\begin{aligned} &\boldsymbol{V_{\text{IG}}}(x, y; \underline{x}, \underline{y}) \\ &\quad = \mathcal{H}(\boldsymbol{\theta} \mid \underline{x}, \underline{y}) - \mathcal{H}(\boldsymbol{\theta} \mid x, y, \underline{x}, \underline{y}). \end{aligned} \tag{1}$$

The expected information gain policy selects the x^* that maximizes $\mathbb{E}_{y \in [0,1]}\, V_{\text{IG}}(x, y; \underline{x}, \underline{y})$, *i.e.,*:

$$\begin{aligned} x^* = \arg\max_{x \in \mathcal{X}} \; & V_{\text{IG}}(x, y = 1; \underline{x}, \underline{y}) \cdot p(y = 1 \mid x, \underline{x}, \underline{y}) \\ & + V_{\text{IG}}(x, y = 0; \underline{x}, \underline{y}) \cdot p(y = 0 \mid x, \underline{x}, \underline{y}), \\ \pi_{\text{eig}}(x|\underline{x}, \underline{y}) = \delta(x^*), & \end{aligned}$$

where $\delta(x)$ denotes the probability distribution that places all its mass on x.

3.2 Hybrid Policies

Hybrid policies dynamically choose at each time step among a set of basic policies Π based on some metric V. At each step, the hybrid policy estimates the expected value of V for each basic policy $\pi \in \Pi$, chooses the policy $\pi*$ that has the highest expected value, and then samples $x \in \Sigma^+$ according to π^*. Here, we study one such policy: $\Pi = [\pi_{\text{train}}, \pi_{\text{eig}}]$, with metric $V = V_{\text{IG}}$. We refer to the non-train policy as $\hat{\pi}$ and the metric used to select $\pi^* \in [\hat{\pi}, \pi_{\text{train}}]$ at each step as V.

We explore two general methods for estimating the expected value of V for each policy π^*: *history*-based and *model*-based. We also explore a *mixed* approach using a history-based method for π_{train} and a model-based method for $\hat{\pi}$.

History In the history-based approach, the model keeps a running average of empirical values of V for candidates previously selected by π_{train} and $\hat{\pi}$.

For instance, for history-based hybrid policy $\pi_{\text{eig-history}}(x|\underline{x}, \underline{y})$, $V = V_{\text{IG}}$ (see Table 1b). Suppose at a particular step, the basic policy π^* selected by $\pi_{\text{eig-history}}$ chose query x, which received label y from the informant. Then the history-based $\pi_{\text{eig-history}}$ would store the empirical information gain between $p(\boldsymbol{\theta} \mid \underline{x}, \underline{y}), p(\boldsymbol{\theta} \mid x, y, \underline{x}, \underline{y})$ for the chosen π^*; in future steps, it would then select the $\pi^* \in [\pi_{\text{train}}, \hat{\pi}]$ with the highest empirical mean of V, in this case the empirical mean information gain, over candidates queried by each basic policy.

More formally, let $\boldsymbol{S}^{\text{EMP}}(\pi; \underline{x}, \underline{y})$ refer to the mean of observed values V for candidates x_i selected by π before step t, where $\pi \in [\pi_{\text{train}}, \hat{\pi}]$:

$$\boldsymbol{S}^{\text{EMP}}(\pi; \underline{x}, \underline{y}) = \frac{\sum_{i \in I_\pi} V(x_i, y_i; \underline{x}_{<i})}{|I_\pi|},$$
$$\text{where } I_\pi = \{i \mid x_i \text{ was selected by } \pi, i < t\}.$$

$V(x_i, y_i; \underline{x}_{<i})$ denotes V's score for the i'th datapoint x_i selected by π under a model that as observed data $\underline{x}_{<i}, \underline{y}_{<i}$.

Then at step t, the history-based hybrid policies sample π^* according to:

$$\pi^* = \arg\max_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S^{\text{EMP}}(\pi; \underline{x}, \underline{y}).$$

For $t < 2$, we automatically select π_{train} and $\hat{\pi}$ in a random order, each once, to ensure we have empirical means for both policies.

Model The model-based approach is prospective and involves using the current posterior distribution over model parameters to compute an expected value for the target metric under the policy. We define two ways of computing these expectations.

$S^{\text{EXP}(y)}$ computes an expectation over possible **labels** y for the candidate x^* that will be chosen by policy π. We use $S^{\text{EXP}(y)}$ to score **non-train** basic policies $\hat{\pi}$ because they select x^* deterministically given $\mathcal{X}$, *i.e.,* selecting the inputs that maximize the objectives described in §3.1. More formally:

$$\boldsymbol{S^{\text{EXP}(y)}}(\hat{\pi}; \underline{x}, \underline{y}) = \mathop{\mathbb{E}}_{y \in [0,1]} V(x^*, y; \underline{x}, \underline{y}), x^* \sim \hat{\pi}.$$

$S^{\text{EXP}(x)}$ computes an expectation over possible **inputs** $x \in L$ and assumes a fixed label $(y = 1)$. We score the **train** basic policy π_{train} with $S^{\text{EXP}(x)}$ because the randomness for π_{train} is over forms in the lexicon that could be sampled by π_{train}, and labels are always 1. More formally:

$$\boldsymbol{S^{\text{EXP}(x)}}(\pi_{\text{train}}; \underline{x}, \underline{y}) = \mathop{\mathbb{E}}_{x \in L} V(x, y = 1; \underline{x}, \underline{y}).$$

In practice, however, we approximate this expectation with samples from $\mathcal{X}$, since we do not assume that the model has access to the lexicon used by the informant. In particular, we model the probability that a form x is in the lexicon as $p(y = 1 \mid x; \underline{x}, \underline{y})$.

Using the policy-specific expectations defined above, the model-based approach selects the policy π^* according to:

$$\pi^* = \mathop{\arg\max}_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S(\pi; \underline{x}, \underline{y}).$$

Mixed Finally, the mixed policies combine the retrospective evaluation of the history-based method and the prospective evaluation of the model-based method. In particular, we use the **model**-based approach for non-train $\hat{\pi}$ (*i.e.,* scoring with $S^{\text{EXP}(y)}$) and the **history**-based approach for **train** policy π_{train} (*i.e.,* scoring with S^{EMP}):

$$\begin{aligned} S(\hat{\pi}; \underline{x}, \underline{y}) &= S^{\text{EXP}(y)}(\hat{\pi}; \underline{x}, \underline{y}), \\ S(\pi_{\text{train}}; \underline{x}, \underline{y}) &= S^{\text{EMP}}(\pi_{\text{train}}; \underline{x}, \underline{y}), \\ \pi^* &= \mathop{\arg\max}_{\pi \in [\hat{\pi}, \pi_{\text{train}}]} S(\pi; \underline{x}, \underline{y}). \end{aligned}$$

For $t = 0$, we always select π_{train} to ensure we have an empirical mean for π_{train}. Table 1 provides an overview of the query policies described in the preceding sections.

4 A Grammatical Model for Phonotactics

We implement and test our approach for a simple categorical model of phonotactics. The grammar consists of two components. First, a finite set of phonological feature functions $\{\phi_i\} : \Sigma^+ \mapsto \{0, 1\}$; if $\phi_i(x) = 1$ we say that feature i is **active** for string x. This set is taken to be universal and available to the learner before any data are observed. Second, a set of binary values $\boldsymbol{\theta} = \{\theta_i\}$, one for each feature function; if $\theta_i = 1$ then feature i is **penalized**. In our simple categorical model, a string is grammatical if and only if no feature active for it is penalized. $\boldsymbol{\theta}$ thus determines the language: $L = \{x : \sum_i \theta_i(x)\phi_i(x) = 0\}$. We assume a factorizable prior distribution over which features are active: $p(\boldsymbol{\theta}) = \prod_{\theta_j \in \boldsymbol{\theta}} p(\theta_j)$. To enable mathematical tractability, we also incorporate a noise term α which causes the learner to perceive a judgment from the informant as noisy (reversed) with probability $1 - \alpha$.

This model is based on a decades-long research tradition in theoretical and experimental phonology into what determines the range and frequency of possible word forms in a language. A consensus view of the topic is that speakers have fine-grained judgments about the acceptability of nonwords (for example, most speakers judge *blick* to be more acceptable than *bnick*; Chomsky and Halle, 1968), and that this knowledge can be decomposed into the independent, additive effects of multiple prohibitions on specific sequences of sounds (in phonological theory, termed MARKEDNESS constraints). Further, speakers form these generalizations at the level of the phonological feature, since they exhibit structured judgments that distinguish between different unattested forms: speakers systematically rate *bnick* as less English-like than *bzick*, despite no attested words having initial *bn-* or *bz-*. We reflect this knowledge in our generative model: to determine the distribution of licit strings in a language, we first sample some parameters which govern subsequences of features which are penalized in the language.

In our model we take $\{\phi_i\}$ to be a collection of phonological **feature trigrams**: an ordered triple of three phonological features with values that pick out some class of trigrams of segments in the language (see §5.1 for more details and examples). Since our phonotactics are variants on vowel harmony, these featural trigrams are henceforth assumed to be relativized to the vowel tier, regulat-

Basic Policy	Quantity Maximized
π_{train}	–
π_{unif}	–
$\pi_{\text{label-ent}}$	Label entropy
π_{eig}	Expected info gain

(a) Basic policies (§3.1).

Hybrid Policy	Basic Choices Π	Basic Policy Selection: Method	Metric V	π_{train} score	Non-train score
$\pi_{\text{eig-history}}$	π_{train}, π_{eig}	History	Info gain (V_{IG}, Eq 1)	S^{EMP}	S^{EMP}
$\pi_{\text{eig-model}}$		Model		$S^{\text{EXP}(x)}$	$S^{\text{EXP}(y)}$
$\pi_{\text{eig-mixed}}$		Mixed		S^{EMP}	$S^{\text{EXP}(y)}$

(b) Hybrid policies (§3.2).

Table 1: Summary of query policies (§3). S^{EMP} refers to empirical mean. $S^{\text{EXP}(y)}$ and $S^{\text{EXP}(x)}$ refer to the expectation metrics for the non-train $\hat{\pi}$ and train π_{train} strategies, respectively. Basic policies select inputs to query the informant. Hybrid policies choose between a set of basic policies Π by scoring them with a metric V and one of the scoring functions.

ing vowel qualities in three adjacent syllables. In order to capture generalizations that may hold differently in edge-adjacent vs. word-medial position, we pad the representation of each word treated by the model with a boundary symbol "#" — omitted generally in this paper, for simplicity — which bears the [+ word boundary] feature that the trigram constraints can refer to (following the practice of Hayes and Wilson, 2008, inspired by Chomsky and Halle, 1968).

4.1 Implementation details

Our general approach and specific model create several computational tractability issues that we address here. First, all policies aside from π_{train} and π_{unif} in principle require search for an optimal string x within $\mathcal{X}$. In practice, we consider $\mathcal{X} = \Sigma^{+}\{2, 5\}$, *i.e.,* $\mathcal{X}$ is the set of strings with 2-5 syllables. This resulting set is still very large, so we approximate the search over $\mathcal{X}$ by uniformly sampling a set of k candidates and choosing the best according to V. We sample candidates by uniformly sampling a length, then uniformly sampling each syllable from the inventory of possible onset-vowel combinations in the language (with replacement). We then de-duplicate candidates and filter $\underline{x}$, excluding previously observed sequences and those that were accidental duplicates with items in the test set.

Second, although the model parameters $\boldsymbol{\theta}$ are independent in the prior, conditioning on data renders them conditionally dependent and computing with the true posterior is in general intractable. To deal with this, we use mean-field variational Bayes to approximate the posterior as $p(\boldsymbol{\theta} \mid \underline{x}, \underline{y}) \approx \prod_{\theta_j \in \boldsymbol{\theta}} q(\theta_j = 1 \mid \underline{x}, \underline{y})$. We use this approximation to both estimate the model's posterior (used by $\pi_{\text{label-ent}}$ and π_{eig}) and to make predictions about individual new examples. See Appendix D for details.

5 Experiments

We now describe our experiments for evaluating the different query policies. We evaluate on two types of languages. We call the first the ATR Vowel Harmony language (§5.1), which has grammar that regulates the distribution of types of vowels, inspired by those found in many languages of the world. The purpose of evaluating on this language is to evaluate how well our new approach, and specifically the various non-baseline query policies, work on naturalistic data. We also evaluate on a set of procedurally-generated languages (§5.2) that are matched on statistics to ATR Vowel Harmony, *i.e.,* they have the same number of feature trigrams that are penalized, but differ in *which*. This second set of evaluations aims to determine how robust our model is to typologically-unusual languages, so we can be confident that any success in learning ATR Vowel Harmony is attributable to our procedure, rather than a coincidence of the typologically-natural vowel harmony pattern.

These experiments lead to three sets of analyses: in the first (§5.4), we both select hyperparameters and evaluate on procedurally-generated languages through k-fold cross validation. These results can be interpreted as an in-distribution analysis of the query policies. In the second set of results (§5.5), we evaluate the policies out-of-distribution by selecting hyperparameters on the procedurally-generated languages and evaluating on the ATR Vowel Harmony language. In the last analysis (§5.6), we evaluate the upper bound of policy performance by selecting hyperparameters and evaluating on the same language, ATR Vowel Harmony.

5.1 ATR Vowel Harmony

We created a model language whose words are governed by a small set of known phonological principles. Loosely inspired by harmony systems

common among Niger-Congo and Nilo-Saharan languages spoken throughout Africa, the vowels in this language can be divided into two classes, defined with respect to the phonological feature Advanced Tongue Root (ATR); for typological data, see Casali (2003, 2008, 2016); Rose (2018), among others. In this language, vowels that are [+ATR] are {i, e}, and have pronunciations that are more peripheral in the vowel space; those that are [-ATR] are {ɪ, ɛ}, and are more phonetically centralized. For the sake of simplicity, we restrict the simulated language to only have front vowels. A fifth vowel in the system, [a], is not specified for ATR. This language has consonants {p, t, k, q}, which are distributed freely with respect to one another and to vowels with the exception that syllables must begin with exactly consonant and must contain exactly one vowel, a typologically common restriction. Since consonants are not regulated by the grammar we are working with, the three binary features (leaving out [word boundary]) create a set of 512 possible feature trigrams which characterize the space of all possible strings in the language. The syllable counts of words follows a Poisson distribution with $\lambda = 2$.

The single rule active in this language governs the distribution of vowels specified for the feature [ATR]: vowels in adjacent syllables had to have the same [ATR] specification. This means that vowel sequences in a word can be [i…e] or [ɛ…ɪ], but not [e …ɛ] or [e …ɪ]. Since [a] is not specified for ATR, it creates boundaries that allow different ATR values to exist on either side of it: for example, while the vowel sequence [e …ɛ] is not permitted, the sequence [e …a …ɛ] is allowed, because the ATR-distinct vowels are separated by the unspecified [a]. This yielded sample licit words like [katipe], [tɛpɪ], and [qekatɪ], and illicit ones [kɛkiqa], [tɪtaqikɛ], and [qiqɪka].

Feature trigrams were composed of triples of the features and specifications shown in Appendix Table 3, any one of which picks out a certain set of vowel trigrams in adjacent syllables.

Data We sampled 157 unique words as the lexicon L, and a set of 1,010 random words, roughly balanced for length, as a test set. The model was provided with the set of features in Appendix Table 3, and restrictions on syllable structure for use in the proposal distribution.

Informant The informant was configured to reject any word that contained vowels in adjacent syllables that differed in ATR specification (like [pekitɛ] or [qetatɪkipe]), and accept all others.

5.2 Procedurally-Generated Languages

We also experimented with languages that share the same feature space, and thus the same set of 512 feature trigrams, as ATR Vowel Harmony (§5.1) but were *procedurally generated* by sampling 16 of the 512 total feature trigrams to be "on" (*i.e.,* penalized) and set all others to be off, creating languages with different restrictions on licit vowel sequences in adjacent syllables.

Data For each "language" (*i.e.,* set of sampled feature trigrams to be penalized), we carried out a procedure to sample the lexicon L, as well as evaluation datasets. For each set of 16 θ values representing penalized phonological feature trigrams, we created random strings as in Experiment 1, filtering them to ensure that the train and test set are of equal size, and the test set is balanced for length of word and composed of half acceptable and half unacceptable forms.

5.3 Experimental Set-Up

Hyperparameters The model has several free parameters: a noise parameter α that represents the probability that an observed label is correct (versus noisy), and θ_{prior}, the prior probability of a feature being *on* (penalized), *i.e.,* $p_{\text{prior}}(\theta_j = 1)$. There are also hyperparameters governing the optimization of the model: we denote by s the number of optimization steps in the variational update.[1] When $s = \infty$, we optimize until the magnitude of the change in $\boldsymbol{\theta}$ is less than or equal to an error threshold $\epsilon = 2e^{-7}$. We also experiment with $s = 1$, in which we perform a single update.

We ran a grid-search over the parameter space of $\log(\log(\alpha)) \in \{0.1, 0.25, 0.5, 1, 2, 4, 8\}$, $\theta_{\text{prior}} \in \{0.001, 0.025, 0.05, 0.1, 0.2, 0.35\}$, and $s \in \{1, \infty\}$. We ran 10 random seeds (9 for the procedurally generated languages)[2] and all query policies in Table 1 for each hyperparameter setting. Each experiment was run for 150 steps.

For non-train policies, we generated $k = 100$ candidates from $\mathcal{X}$.

[1] These optimization parameters govern both the model's learning and the evaluation of candidate queries for prospective strategies, *i.e.,* π_{eig}, and the hybrid strategies.

[2] For the generated languages, seed also governed the "language," *i.e.,* phonological feature trigrams sampled as "on."

Evaluation At each step, we compute the AUC (area under the ROC curve) on the test set. We then compute the mean AUC value across steps, which we refer to as the **mean-AUC**; a higher mean-AUC indicates more efficient learning. We report the median of the mean-AUC values over seeds.

5.4 In-Distribution Results

Assessing the in-distribution results, shown in the left column of Figure 2, we see that interactive elicitation is on par with, if not higher than, baseline strategies (top left plot). The difference between the *train* and *uniform* baselines was not significant according to a two-sided paired t-test, and the only strategy that performed significantly better than *train* after correcting for multiple comparisons was *Info. gain / train (model)*. This difference is more visually striking in the plot of average AUC over time (middle left plot), where *Info. gain / train (model)* both ascends faster, and asymptotes earlier, than *train*, although with greater variance across runs. In the bottom left plot of Figure 2, we see that the numerically-best-performing *Info. gain / train (model)* strategy moves rather smoothly from an initial *train* preference to an *Info. gain* preference as learning progresses. That is, information in known-good words is initially helpful, but quickly becomes less useful as the model learns more of the language and can generate more targeted queries.

5.5 Out-Of-Distribution Results

The out-of-distribution analysis on the ATR Vowel Harmony language found greater variance of median mean-AUC between strategies, and also greater variance within strategies across seeds (top center plot). We note that this performance is lower than what is found in the upper-bound analysis, since the hyperparameters (listed in Appendix Table 2) were chosen based on the pooled results of the procedurally-generated languages. As in the in-distribution analysis, we found no statistical difference between the two baselines, nor between the *Info. gain* strategy and *uniform*, although *Info. gain* performed numerically better. In terms of average AUC over time (middle center plot), we find again that the top two non-baseline strategies rise faster and peak earlier than *uniform*, but exhibit greater variance.

5.6 Upper Bound Results

Greedily selecting for the best test performance in a hyperparameter search conducted on ATR Vowel Harmony yields superior performance compared to the out-of-distribution analysis hyperparameters, as seen in the top right plot in Figure 2. Appendix Table 2 lists the hyperparameter values used. However, we found no significant difference between the stronger baseline (*uniform*) and any other strategy after correcting for multiple comparisons.

6 Related Work

The goal of **active learning** is to improve learning efficiency by allowing models to choose which data to query an oracle about (Zhang et al., 2022). *Uncertainty sampling* (Lewis and Gale, 1994) methods select queries for which model uncertainty is highest. Most closely related are uncertainty sampling methods for probabilistic models, including least-confidence (Culotta and McCallum, 2005), margin sampling (Scheffer et al., 2001), and entropy-based methods.

Disagreement-based strategies query instances that maximize disagreement among a group of models (Seung et al., 1992). The distribution over a single model's parameters can also be treated as this "group" of distinct models, as has been done for neural models (Gal et al., 2017). Such methods are closely related to the feature entropy querying policy that we explore.

Another class of *forward-looking methods* incorporates information about how models would change if a given data-point were observed. Previous work includes methods that sample instances based on expected loss reduction (Roy and McCallum, 2001), expected information gain (MacKay, 1992), and expected gradient length (Settles et al., 2007). These methods are closely related to the policies based on information-gain that we explore.

Our hybrid policies are also related to previous work on dynamic selection between multiple active learning policies, such as DUAL (Donmez et al., 2007), which dynamically switches between density and uncertainty-based strategies.

7 Conclusion

We have described a method for parameterizing a formal model of language via efficient, iterative querying of a black box agent. We demonstrated that on an in-distribution set of toy languages, our query policies consistently outperform baselines numerically, including a statistically-significant improvement for the most effective policy. The model struggles more on out-of-distribution languages,

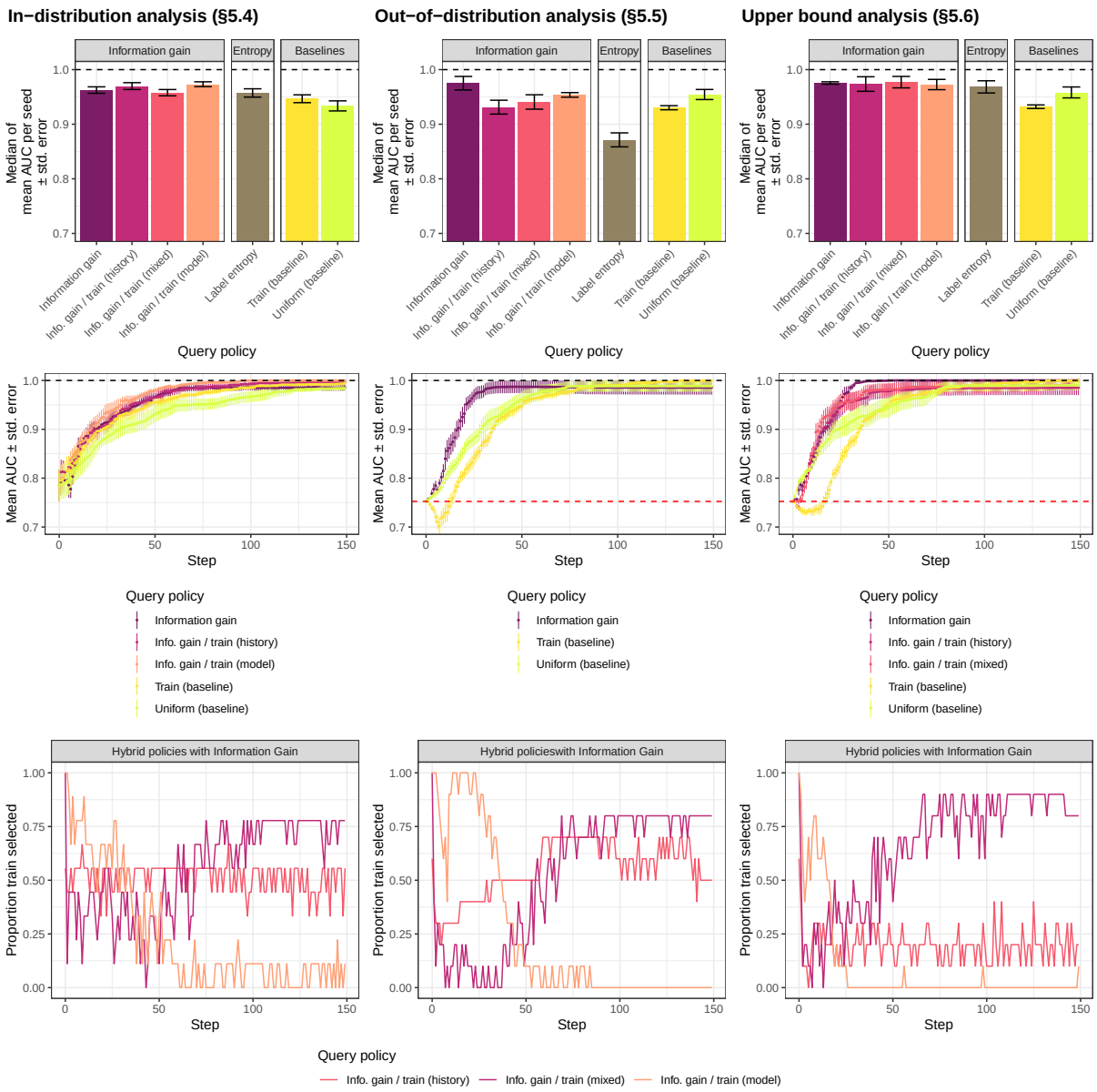

Figure 2: We report three analyses of the toy ATR Vowel Harmony language and our procedurally-generated languages: in-distribution (left column, see §5.4), out-of-distribution (center column, see §5.5), and an upper-bound assessment (right column, see §5.6). For each, we report the median and standard error of the mean-AUC over steps aggregated across runs (top row; numerical values and hyperparameters reported in Appendix Table 2), average AUC at each step aggregated across runs (middle row), and at each step the proportion of runs where the basic *train* strategy was selected by the hybrid strategies (bottom row). **Results:** In terms of median mean-AUC (top row), our query strategies are numerically on par with, if not beating, the stronger of the two baseline conditions; statistically, only the difference between *Info. gain / train (model)* and *uniform* was significant in the in-distribution analysis (top left). Average AUC over time (middle row) shows a similar pattern across all three analyses, with the non-baseline strategies rising faster and asymptoting sooner than baseline strategies, but usually with greater variance. Finally, though all hybrid strategies prefer non-train some portion of the time, the *Info. gain / train (model)* exhibits an interpretable shift from early preference for *train* data to later preference for its own synthesized queries in all three analyses.

though in all cases the query policies are numerically comparable to the best baseline. We note that a contributing factor to the difficulty of the query policies consistently achieving a *significantly* higher performance than baselines is the small number of seeds, which exhibit nontrivial variance, particularly in hybrid policies. Future work may address this with more robust experiments.

References

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

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Roderic F Casali. 2003. [atr] value asymmetries and underlying vowel inventory structure in niger-congo and nilo-saharan.

Roderic F Casali. 2008. Atr harmony in african languages. *Language and linguistics compass*, 2(3):496–549.

Roderic F Casali. 2016. Some inventory-related asymetries in the patterning of tongue root harmony systems. *Studies in African Linguistics*, pages 96–99.

Noam Chomsky and Morris Halle. 1968. *The sound pattern of English*. Harper & Row New York.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, page 746–751. AAAI Press.

Pinar Donmez, Jaime G. Carbonell, and Paul N. Bennett. 2007. Dual strategy active learning. In *Proceedings of the 18th European Conference on Machine Learning*, ECML '07, page 116–127, Berlin, Heidelberg. Springer-Verlag.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1183–1192. JMLR.org.

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

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR '94*, pages 3–12, London. Springer London.

David J. C. MacKay. 1992. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*.

Sharon Rose. 2018. Atr vowel harmony: new patterns and diagnostics. In *Proceedings of the Annual Meetings on Phonology*, volume 5.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through monte carlo estimation of error reduction. In *International Conference on Machine Learning*.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *Advances in Intelligent Data Analysis*, pages 309–318, Berlin, Heidelberg. Springer Berlin Heidelberg.

Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 287–294, New York, NY, USA. Association for Computing Machinery.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Phonological features for Toy Languages

As described in §5.1, the ATR Vowel Harmony language is based on the categorization of vowels as [+ATR], [-ATR], or unspecified. The features [high] and [low] also serve to distinguish vowels in the language, but are not governed by a phonotactic. In contrast, any of the 512 logically possible trigrams of specified phonological features may be penalized for the procedurally-generated languages. Table 3 displays the phonological features for each of the vowels in the languages.

B Hyperparameters for out-of-distribution and upper-bound analyses

In §5.3, we described the hyperparameters of our grammatical model and the process by which val-

Out-of-distribution analysis						Upper-bound analysis					
Policy	log(log(α))	prior	s	Median mean-AUC	Std. err.	Policy	log(log(α))	prior	s	Median mean-AUC	Std. err.
Info. gain / train (model)	0.5	0.1	∞	0.973	0.004	Info. gain / train (mixed)	0.25	0.1	∞	0.977	0.010
Info. gain / train (history)	1	0.1	∞	0.970	0.006	Information gain	0.1	0.025	∞	0.975	0.002
Info. gain / train (mixed)	2	0.2	∞	0.969	0.005	Info. gain / train (history)	0.1	0.05	∞	0.974	0.013
Information gain	0.25	0.025	∞	0.966	0.004	Info. gain / train (model)	1	0.001	1	0.973	0.009
Label entropy	0.1	0.1	∞	0.964	0.009	Label entropy	0.5	0.05	1	0.968	0.011
Train (baseline)	1	0.1	∞	0.947	0.007	Uniform (baseline)	0.5	0.025	1	0.958	0.010
Uniform (baseline)	1	0.1	1	0.940	0.008	Train (baseline)	8	0.35	1	0.932	0.003

Table 2: Hyperparameters for the out-of-distribution analysis (§5.5) and upper-bound analysis (§5.6).

ues were selected for the out-of-distribution analysis. These selected hyperparameter values are presented in Table 2.

C Query Policy Implementation

We now revisit the query strategies introduced in §3 and describe how they are implemented for the model described in §4. In particular, under the described generative model, $p(y = 1 \mid x, \underline{x}, \underline{y}) = \prod_{j \in \phi(x)} q(\theta_j = 0 \mid x, \underline{x}, \underline{y})$, as described above.

Let $q_y = \prod_{j \in \phi(x)} q(\theta_j = 0 \mid x, \underline{x}, \underline{y})$, *i.e.,* q_y is the probability of label $y = 1$ for input x under the variational posterior; this is equivalent to the probability of all features in $\phi(x)$ being "off". Let $q_{\theta_j} = q(\theta_j = 1 \mid \underline{x}, \underline{y})$ indicate the probability of parameter θ_j being "on" (*i.e.,* penalized) under the current variational $q(\boldsymbol{\theta})$. For this model, the quantities used by the query policies in §3 are computed as follows:

Label Entropy Policy $\pi_{\text{label-ent}}$ selects x^* according to:

$$x^* = \arg\max_{x \in \mathcal{X}} H(q_y), \text{where}$$
$$H(p(y \mid x, \underline{x}, \underline{y})) = -\, q_y \log q_y - (1 - q_y) \log(1 - q_y).$$

	[high]	[low]	[ATR]
i	+	−	+
ɪ	+	−	−
e	−	−	+
ɛ	−	−	−
a	−	+	0

Table 3: Phonological features for vowels used in the toy languages. The feature [word boundary] is omitted for simplicity, as it has the value '−' for all segments.

Expected Information Gain Policy π_{eig} selects x^* according to:

$$x^* = \arg\max_{x \in \mathcal{X}} V_{\text{IG}}(x, y = 1; \underline{x}, \underline{y}) \cdot q_y + V_{\text{IG}}(x, y = 0; \underline{x}, \underline{y}) \cdot (1 - q_y),$$

where V_{IG} is given by

$$V_{\text{IG}}(x, y; \underline{x}, \underline{y}) = \sum_{j \in |\theta|} \Big(H(q(\theta_j \mid \underline{x}, \underline{y})) - H(q(\theta_j \mid x, y, \underline{x}, \underline{y})) \Big),$$

and H is given by

$$H(q(\theta_j)) = -q_{\theta_j} \log q_{\theta_j} - (1 - q_{\theta_j}) \log(1 - q_{\theta_j}).$$

D Derivation of the Update Rule

We want to compute the posterior $p(\theta|y, x, \alpha)$, which is intractable. Thus, we approximate it with a variational posterior, composed of binomial distributions for each θ_i. We further assume that the individual dimensions of the posterior (the individual components of θ) have values that are not correlated. This allows us to perform coordinate ascent on each dimension of the posterior separately; thus we express the following derivation in terms of $q(\theta_i)$, where *i* is the index in the feature *n*-gram vector.

The variational posterior is optimized to minimize the KL divergence between the true posterior $p(\boldsymbol{\theta}|X, Y, \alpha)$ and $q(\boldsymbol{\theta})$; we do this by maximizing the ELBO.

The coordinate ascent update rule for each dimension of the posterior, that is, for each latent variable, is:

$$q(\theta_i) \propto \exp\left[\mathbb{E}_{q \neg i} \log p(\theta_i, \theta_{\neg i}, y, x)\right].$$

Given the generative process, we can rewrite:

$$p(\theta_i, \theta_{\neg i}, y, x) = p(\theta_i) \cdot p(\theta_{\neg i}) \cdot p(y|x, \theta_i, \theta_{\neg i}).$$

$\mathbb{E}_{q\neg i} \log p(\theta_{\neg i})$ is assumed to be constant across values of θ_i (expressing the lack of dependence between parameters), so we can rewrite the update rule as:

$$q(\theta_i) \propto \exp\big[\mathbb{E}_{q\neg i}[\log p(\theta_i) + \log p(y|x, \theta_i, \theta_{\neg i})]\big].$$

Further, since $\log p(\theta_i)$ is constant across values of $q\neg i$, we can rewrite it once more:

$$q(\theta_i) \propto \exp\big[\log p(\theta_i) + \mathbb{E}_{q\neg i} \log p(y|x, \theta_i, \theta_{\neg i})\big].$$

Since our approximating distribution is binomial, we describe in turn the treatment of each of the two possible values of θ. First, we derive the update rule for when the label y is acceptable ($y = 1$).

We know that there are two subsets of $q\neg i$ cases where this can happen. In α proportion of them, y is a correct label, which can only happen when $\theta_j = 0$ for all $j \neq i \in \phi(x)$. This occurs with probability $p_{\text{all\_off}} = \prod_{j\neq i\in\phi(x)} q(\theta_j = 0)$. There is also, then, the $1-\alpha$ proportion of cases in which y is an incorrect label, and the true judgement is unacceptable. Under this assumption, at least 1 feature is on, which occurs with probability $1 - p_{\text{all\_off}}$.

We can rewrite the expectation term to get approximate probabilities for both the $\theta_i = 0$ and $\theta_i = 1$ cases when $y = 1$:

$$q(\theta_i = 0) \propto \exp\big[\log p(\theta_i = 0) + \big(p_{\text{all\_off}} \cdot \log\alpha + (1 - p_{\text{all\_off}}) \cdot \log(1-\alpha)\big)\big].$$

If $\theta_i = 1$, we know that $\log p(y|x, \theta_i, \theta_{\neg i}) = \log(1-\alpha)$ for all $q\neg i$, since we know that y must be a noisy label. Thus:

$$q(\theta_i = 1) \propto \exp\big[\log p(\theta_i = 1) + \log(1-\alpha)\big].$$

We can normalize these quantities to get a proper probability distribution, i.e. we can set $q(\theta_i = 1)$ to the following quantity:

$$q(\theta_i = 1) := \frac{q(\theta_i = 1)}{q(\theta_i = 1) + q(\theta_i = 0)}.$$

Using the expression $q(\theta_i)$ as shorthand for $q(\theta_i = 1)$, this results in the following update rule:

$$q(\theta_i = 1) = \sigma\Big(\log p(\theta_i) - \log(1 - p(\theta_i)) - p_{\text{all\_off}} \cdot \log\frac{\alpha}{1-\alpha}\Big).$$

In practice, we update over batches of inputs/outputs rather than single datapoints, *i.e.,*

$$\mathbf{m_{i,j}} = \sum_{j'\neq j\in\phi(x_i)} \log(1 - p(\theta'_j)) + \log\log\frac{\alpha}{1-\alpha},$$

$$q(\theta_j) = \sigma(\log p(\theta_j) - \log(1 - p(\theta_j)) - \sum_{i<t} y_i \cdot \exp(\mathbf{m_{i,j}})).$$

We update each $q(\theta_j)$ either for a fixed number of steps s, or until convergence, *i.e.,* when:

$$\left|\sum_{j\in|\boldsymbol{\theta}|} q_j^{\delta+1} - q_j^{\delta}\right| < \epsilon,$$

where ϵ is an error threshold.