Ke, Alan Hezao & Xu, Jingying & Ding, Lijun. 2025.
Modeling the Learning of Parametric Variation:
Implementing an input-driven hierarchical approach.
Glossa: a journal of general linguistics 10(1). pp. 1–42.
DOI: https://doi.org/10.16995/glossa.18053

Open Library
of Humanities

Modeling the Learning of Parametric Variation: Implementing an input-driven hierarchical approach

Alan Hezao Ke, Michigan State University, kehezao@msu.edu

Jingying Xu, Michigan State University, xujing21@msu.edu

Lijun Ding, University of California San Diego, l2ding@ucsd.edu

This paper introduces a computational model of a novel hierarchical approach to parameter setting. We argue that parameters, whether innate or derived from innate properties or another source, remain necessary within the Minimalist approach. We distinguish and compare two methods of parameter setting in the computational modeling of language acquisition: the direct parameter setting approach and the grammar selection approach. The merits and limitations of each approach are discussed. We observe that the strengths of one address the weaknesses of the other, and vice versa. Consequently, we suggest a hybrid method termed the Clustering Approach, according to which the parameters are hierarchically organized based on their occurrences in the input. The Clustering Approach retains the benefits of both grammar selection and direct parameter setting approaches while circumventing their shortcomings. We compare the Clustering Approach to a previous hybrid method, illustrating how it resolves a significant issue found in the latter. The Clustering Approach assumes that parameter setting is strictly data-driven due to its accurate update mechanism during learning. Our simulation results demonstrate that the Clustering Approach offers a unique framework for modeling the acquisition of parametric variation, especially in scenarios involving a larger set of parameters.

1 Introduction

Plato's Problem, which questions how children can acquire language given the impoverished input data, stands as a central issue in language acquisition and linguistic theorizing. Over the years, researchers have sought to explain the apparent gap between linguistic knowledge and external input from theoretical, empirical, or computational perspectives. Within the generative framework, the Principles and Parameters (P & P) approach offers a valuable attempt to address Plato's Problem. It proposes that children are biologically endowed with a finite set of principles, which is invariant across languages, and a finite set of parameters, which determines systematic language variation. The P & P approach assumes that the principles and parameters are part and parcel of Universal Grammar (UG), which by hypothesis should be sufficiently rich to guide children to locate their target grammar by setting the parameter values in the space of finite choices. This leads to a pressing question: In the actual language-learning process, how do children navigate through this hypothesis space to set the correct parameter values?

To answer this question, various models for parameter setting have been proposed within the P & P framework, among which we identify two main lines of research and call them the "grammar selection approach" and the "direct parameter setting approach." The former, exemplified by models like the Triggering Learning Algorithm (TLA, Gibson & Wexler 1994) model and the classical Variational Learner (VL, Yang 2002) model, regards language acquisition as the selection of the target grammar with the correct set of parameters out of all possible grammars defined by UG. In contrast, the direct parameter setting approach works with one single grammar and sets the values of parameters directly in the grammar. The direct parameter setting approach has been implemented in Fodor, Sakas, and colleagues' Structural Triggering Learner (STL, e.g., Fodor 1998a; Sakas & Fodor 2012; Sakas et al. 2017). The STL model assumes that parameters are learned directly and individually, that is, there is a direct mapping from knowledge of parameters to parts of the structure in the parsing of actual sentences.

Although the pros and cons of the various models have been extensively discussed in earlier studies (Fodor 1998a; Fodor & Sakas 2004; Sakas & Fodor 2012; Fodor 2017; Sakas et al. 2017), this paper is the first attempt to make a direct contrast of grammar selection and direct parameter setting approaches. Showing that the two approaches complement each other, we introduce a hybrid approach, termed the "Clustering Approach." This integrates both grammar selection and direct parameter setting, building upon a prior hybrid approach, Yang's (2002) Naive Parameter Learner (NPL). The hybrid approach not only addresses the challenges faced by grammar selection and direct parameter setting, but it also offers additional evidence supporting the continued viability of the parameter setting framework, alongside the overall thesis put forward by Sakas et al. (2017). This stands in contrast to the prevalent belief that parameters should be entirely eliminated from syntax under the Minimalist Program (Chomsky 2001; see Boeckx & Leivada 2014 for useful discussion).

Furthermore, we illustrate how the Clustering Approach resolves some critical issues found in the NPL. It achieves this by setting parameters in an order that is strictly informed by the input: We argue that only the parameters that are actually used in the input sentences will be updated. The order of acquisition is used to hierarchically cluster the possible grammars for more efficient learning. This results in the gradual learning of parameters, overcoming a critical hurdle in the computational modeling of parameter setting and enabling the model to handle grammars with a larger number of parameters.

We will focus on a conceptual introduction to the previous models, illustrating that the proposed model is theoretically feasible and intuitively sensible, while deliberately omitting many technical details. For example, we do not consider the batch-based system (Yang 2002; Pearl 2011) which deals with noisy data. Our simulations were run on synthesized corpora, which were constructed with the assumption that the parameters are set independently. We understand that, in reality, parameters interact in a more complex manner (Clark 1994: 480–483). However, the step we are taking in this paper is to demonstrate that the Clustering Approach is the only model, among those we considered, that effectively models the learning of a large number of parameters. We also do not discuss strategies to further improve the Clustering Approach when dealing with a longer list of parameters or with more realistic linguistic data where significant noise may disturb the learning.[1]

This paper is organized as follows. Section 2 summarizes the central assumptions of the Clustering Approach, where we provide a working definition of parameters and explain the clustering of possible grammars using parameters. Sections 3–5 review two competing approaches to the computational modeling of parameter setting: Section 3 presents the direct parameter setting approach; Section 4 presents the grammar selection approach; and Section 5 provides a direct comparison of the two. Section 6 presents the hybrid approach, which integrates both direct parameter setting and grammar selection. This section also contrasts our version of the hybrid approach with Yang's (2002) NPL and illustrates the improvements we have achieved through simulation results. Section 7 reviews Minimalist approaches to parameter setting, confirming that parametric knowledge is still necessary in the Minimalist Program to account for cross-language variation, although the parameters can be defined differently than in P & P. Section 8 concludes the paper.

2 Core assumptions of the clustering approach

Although the P & P approach offers an appealing solution to Plato's Problem of language acquisition, it significantly adds to the complexity of UG, the innate domain-specific knowledge

[1] We are currently undertaking a project that explores the application of the Clustering Approach to real language data, utilizing natural language processing methods such as automatic parsing and linguistic feature extraction.

of language, leaving the evolvability of the language faculty hard to explain (Berwick & Chomsky 2016; Chomsky 2017; Baker 2021). Therefore, the development of the Minimalist Program was motivated in the early 1990s to simplify UG. The systematic variations in languages were attributed to differences in the formal features of the functional heads in the lexicon, which can be learned through linguistic experience (Borer 1984; Chomsky 1995), referred to as the Borer-Chomsky Conjecture (BCC) by Baker (2008b). However, as we will argue in Section 7, current representative Minimalist approaches to parameters, especially Roberts (2019) and Crisma et al. (2020), do not completely remove parameters from UG. Instead, knowledge of parameter inventory must still be hard-wired into UG, although the presence of specific parameters in a particular language may be determined by relevant linguistic experience, namely, the primary linguistic data (PLD, see Lidz & Gagliardi 2015 for further discussion on the nature of the input).

However, this does not necessarily imply that parameters must be encoded in UG. As Chomsky (2005) and Boeckx (2011) have noted, the Minimalist Program does not leave much room for encoding variation in the narrow syntax. In this paper, we assume that parametric knowledge is necessary to account for the limited variation across languages. However, parameters can either reside in UG or be derived properties constrained by UG. To be more specific, we adopt the following working definition of parameters in a broad sense:

(1) *Definition of parameters in a broad sense*:
A parameter is systematic cross-linguistic variation that could itself be innate (minimized) or derived from other constraints, including logical options and cognitive constraints.

We will turn back to this definition in Section 7.2, explaining how it is connected to the concept of parameters in the Minimalist Program.

Given that we allow both innate parameters and parameters derived from (domain-specific and domain-general) innate and non-innate constraints and biases, a potential consequence is that a longer list of (innate and non-innate) parameters are needed for modeling systematic cross-linguistic variation. The challenge is, can we model the acquisition of syntax with a large number of parameters? Our computational model indicates that the Clustering Approach offers a valid approach to address this challenge.

In addition, the Clustering Approach is inspired by the idea that parameters can be hierarchically organized (Baker 2008a; Biberauer & Roberts 2015; Roberts 2019). Nevertheless, different from previous approaches, we do not presuppose that this hierarchy is predetermined by UG or a third factor. Instead, we assume that the grammar pool, encompassing all possible grammars generated by every possible combination of all parameters and their values, is clustered according to parameters that are learned in sequence. That is, a parameter's position in the hierarchy is determined by the chronological order in which its value is set. This position does not necessarily reflect cross-linguistic differences or typological distinctions.

To simplify the problem, we assume all the parameters are independent and therefore they are set independently.[2] That is, n parameters, each with two values, will generate 2^n possible grammars in the grammar pool. The grammar pool is hierarchically organized by clustering parameters. The clustering criterion includes the values of the parameters that have been set, and the hierarchical structure reflects the learning order of these parameters. Parameters acquired earlier serve as higher-level clustering parameters, while those acquired later serve as lower-level clustering parameters. **Figure 1** depicts this hierarchical organization. It represents an acquisition stage where two parameters have been set: P1 = 0 and P2 = 1. With these two parameters set, the grammar pool has shrunk to a quarter of its original size, given the assumption that the parameters are independent. That is, the learner at this point can disregard all possible grammars where P1 = 1 or P2 = 0.

After P1 and P2 have been set, the learner's subsequent goal is to identify which remaining parameter can serve as a clustering criterion among those still undergoing learning. This next clustering parameter is the parameter that has a most extreme probability away from 0.5, that is, a probability, Prob(P = 0), whose *absolute value* minus 0.5, i.e., |Prob(P = 0)–0.5|, is the highest (P3 in **Figure 1**). This parameter will be prioritized and moved to the top of the list as the focus of learning. The choice of this next clustering criterion is ultimately determined by the input. For instance, if a specific value of P3 in **Figure 1** consistently receives predominant and consistent support from the input, leading to a most extreme probability nearing 0 or 1, it will be taken as the clustering criterion that partitions the current grammar pool into two clusters: one cluster with the probability set at 0, and the other at 1.[3] Therefore, we will refer to this clustering parameter under learning as a sampling parameter, to distinguish it from other clustering parameters whose values have already been set. Once the probability of P3 passes a threshold, P3 is considered set. The parameters that have been set, i.e., P1 and P2, are set in the same way as described.

If a parameter is not used in a specific language, its probability will remain close to its initial value of 0.5, the default initial probability. This causes it to be pushed to the bottom of the hierarchy. While these parameters are present, they act as if they are irrelevant to the target grammar. In practice, they remain inactive, seeming as though they are not part of

[2] This assumption serves only as a working hypothesis to enable a computational simulation, as setting one parameter may indeed affect another. However, the interdependence of some parameters cannot resolve the setting of all parameter values, especially when the majority of parameters are independent of each other (cf. Biberauer et al. 2014; Roberts 2019). According to Guardiano et al. (2020), about 43% of their 5238 values associated with their 97 DP-parameters are dependent on other values. This still leaves most of the values needing to be set independently by learning from the input data.

[3] As will be detailed later, grammar selection from the current grammar pool is guided by the probabilities of P3 = 0 and P3 = 1. This is because grammar selection involves choosing a possible grammar from either the cluster of grammars with P3 = 0 or the cluster with P3 = 1. The probabilities of P3 (and other parameters detected in the input) will then be updated in the learning process.

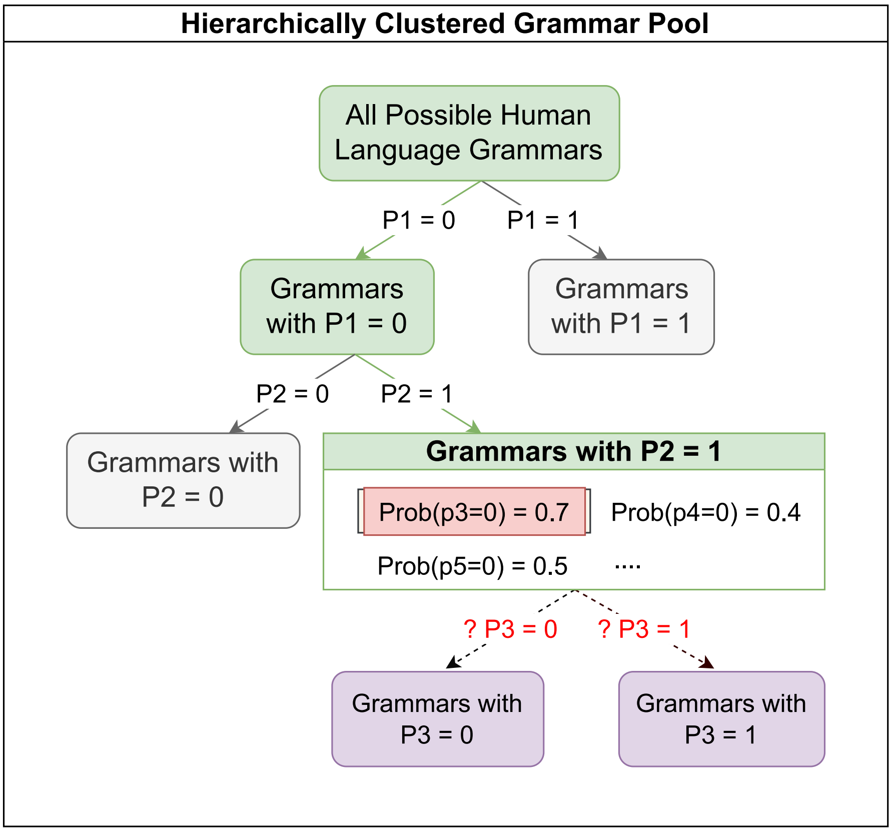


Figure 1: Grammar pool clustered by parameters that are hierarchically ordered.

the learner's mental representation of the target grammar, which is the desired outcome. In this sense, parameter setting in the Clustering Approach is strictly driven by the input and it provides an implementation of a novel emergentist approach to parameter setting in language acquisition.

This strictly input-driven approach thus assumes that possible human grammars are hierarchically organized into parametric clusters, sharing significant properties with models that rely on parameter learning orders. The learning orders have been proposed to be due to constraints from UG that limit access to certain types of input (Lightfoot 1989) or learning order biases, stemming from either innate factors or input-related distributional information, such as those applied in models for acquiring phonological parameters (Dresher 1999; Pearl 2011). In the Clustering Approach, these parametric clusters are learned in order, specifically, from hierarchically higher parameters to lower ones, with the hierarchy being determined completely by the input. Since the learning order of the parameters is determined by the distribution of the parameters in the input, the hierarchical order of parameters can vary across different languages (see Section 6.2 for more details).

Our goal in this paper is to demonstrate that with the aforementioned assumptions, a computational model can be constructed, built upon prior efforts, to address important problems in previous parameter setting models. The model is strictly input-driven and is based on the hypothesis that possible grammars are organized hierarchically in clusters. These lead to two mechanisms in the Clustering Approach that are essential for its superior performance over the benchmark models: (i) It accurately updates parameters used in the input sentence after a successful parse, and (ii) it limits punishment to the sampling parameter after a parsing failure. This contrasts with the benchmark models, which reward and punish all parameters in the selected grammar. Rewarding and punishing all parameters in the selected grammar may sometimes accidentally promote a parameter value different from the value in the target grammar, potentially causing the learner to become stuck in an incorrect possible grammar. We demonstrate that such a model can learn a grammar with a large number of parameters, paving the way for the application of the model to realistic language acquisition settings.

In sections 3 to 5, we will review two camps of computational models for parameter setting and highlight the problems each camp encounters. We argue that the Clustering Approach as a hybrid approach can resolve these problems.

3 The direct parameter setting approach

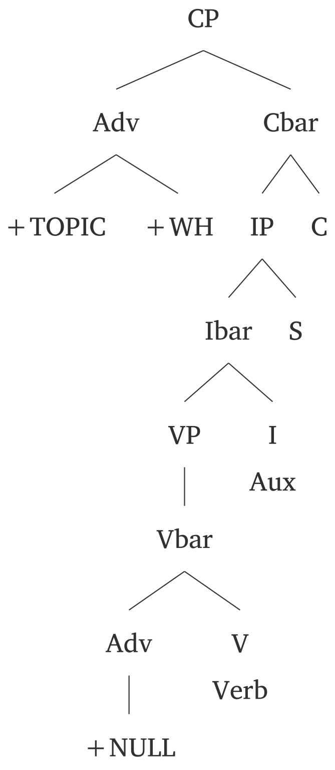
BCC and its followers highlight a crucial approach to parameter setting: Parameters are set directly as features on individual lexical heads, guided by the input. In this view, language acquisition is not formalized as a grammar selection process. The primary task in this direct parameter setting approach is determining whether a specific parameter (and its value) is present in a particular language. This idea of parameter setting has its roots in the P & P framework (e.g., Chomsky 1986; Chomsky & Lasnik 1993), which is in contrast to its alternative model based on grammar selection. Back in Chomsky (1965), language acquisition is framed as the selection of the target grammar out of the set of possible human grammars defined by UG. In fact, starting from Chomsky (1981), the concept of direct parameter setting was introduced to the theory, although grammar selection was still mentioned. The distinction between the direct parameter setting approach and the grammar selection approach turns out to be important for further advancement of child language acquisition modeling.

3.1 Computational implementations of the direct parameter setting approach

The direct parameter setting approach has been implemented in Fodor, Sakas, and colleagues's computational models, particularly in Fodor and Sakas's STL (e.g., Fodor 1998a; Fodor & Sakas 2004; Sakas & Fodor 2012; Fodor 2017; Sakas et al. 2017; also see Howitt et al. 2021 for a model that incorporates many of STL's features). The STL model characterizes parameters and their values as UG-specified treelets, where a treelet is a sub-structure of a larger sentential tree.

Essentially, the STL approach assumes a direct mapping from the knowledge of parameters to parts of the structure in the parsing of actual input sentences. For example, the following sentence and a possible syntactic structure that generates the sentence is taken from the CoLAG language domain (Sakas & Fodor 2012). The sentence consists of sequences of non-null lexical items (e.g., S(ubject), O(bject), Adv, Aux, Verb, etc.) and non-null features (e.g., DEC(larative), Q(uestion), WH, etc.).[4]

(2) *Sentence*: Adv[+TOPIC][+WH] Verb Aux S C



If the above syntactic structure is constructed during parsing, the parser can then infer the parameters that have been applied to generate the sentences, in the face of potential ambiguity in parsing:

(3) Relevant parameters:
Subject final, INFL final, C final, No null subject (probably), No null topic or obligatory wh-movement, etc.

This mapping provides a tool of setting parameters during online parsing: Whenever a treelet is used for structure building in parsing, the activation level of the parameter that is associated

[4] Therefore, Sakas & Fodor (2012) assume that children have acquired the syntactic categories of the relevant lexical items before parsing the structure.

with that treelet will be increased. The STL approach conceptualizes parameter setting as a search process: Language learners search a pool of all possible parametric treelets that is determined by UG. Once a treelet is used in online parsing, it joins the set of treelets employed for the target language, making it accessible for subsequent parsing. Consequently, when a new sentence is introduced, language learners first search their current set of treelets for ones that can analyze its structure. If no suitable treelets are found, the parser must then search the larger pool of possible parametric treelets to find new ones for the analysis.

Below is a description of the basic algorithm for all STL variants, where $G_{current}$ is the set of parametric treelets in use, and the Supergrammar is the parameter pool that includes all possible parametric treelets:

(4) *The basic algorithm of STL* (Sakas 2016: 712)
If $G_{current}$ can parse the current input sentence, *s*, retain the parametric treelets that make up $G_{current}$.
Otherwise, parse the sentence making use of any parametric treelets available in the Supergrammar, giving priority to those in $G_{current}$, and adopt those treelets that contribute to a successful parse.

Note that STL does not assume a grammar selection approach. It is not the grammars that are fit to and evaluated by the input; instead, it is the parameters, or more specifically the parametric treelets, that are used directly in the parsing of the input.

Another important property of STL is that the parser distinguishes between unambiguous and ambiguous triggers (cf. Roeper & Weissenborn 1990). Unambiguous triggers are input that is compatible with only a single (set of) parameter treelet(s), whereas ambiguous triggers are input that can be analyzed with different (sets of) parametric treelets. Therefore, the presence of an unambiguous trigger is a reliable cue for a specific parameter setting.

Different variants of the STL models deal with ambiguous input differently. Strong STL can learn from both unambiguous and ambiguous triggers. During online parsing, it performs a parallel search through both the active set of treelets and the entire pool of possible treelets to analyze ambiguous input.[5] Weak STL implements serial parsing and it learns only from unambiguous triggers, disregarding all ambiguous triggers. It is a deterministic model: once a parameter is set due to an unambiguous trigger, the value of the parameter will no longer be revised. Another STL variant, non-deterministic STL, employs serial parsing but makes a probabilistic choice for a specific structural analysis when encountering an ambiguous trigger, favoring the treelet option with a higher activation level.

[5] However, as noted by Fodor (1998b), this version of STL involves a computationally demanding parallel search during online parsing, which raises concerns about its psychological plausibility.

3.2 Problems of direct parameter setting approach

The direct parameter setting approach enjoys important advantages. It significantly reduces the size of the hypothesis space compared to the grammar selection approach. This method no longer requires a search space encompassing possible human grammars. Instead, the size of the hypothesis space is linearly proportional to the number of parameters. For instance, when 13 parameters are considered, each with two value options, there are 26 treelets in the parametric treelet pool to select from.

However, the direct parameter setting approach also faces some significant challenges. These challenges largely stem from the extensive search required during online parsing. Two main issues will be highlighted. First, to analyze a given input sentence, with the assumption that the parser has already learned the syntactic categories of words, this approach must search both the set of parametric treelets currently in use and potentially the entire pool of parametric treelets permitted by UG. Second, to determine whether a specific trigger is unambiguous, the parser must apply each of the treelets (including treelets in use and all possible treelets in the pool) in the analysis of every given input sentence. This ensures that no more than one set of treelets can successfully analyze a given sentence. If otherwise more than one set of treelets are applicable, the trigger will be instead identified as ambiguous. Furthermore, since there may be instances where individual treelets cannot fully analyze a sentence on their own, various combinations of treelets must be tried to achieve a successful analysis.

Therefore, all versions of the STL model—including non-deterministic STL and Howitt et al.'s (2021) approach—require an extensive search through all parametric treelets. This is mainly because they all maintain a distinction between ambiguous and unambiguous triggers. STL cannot sidestep this challenge since, as highlighted by Sakas (2016), the algorithm must uniquely weigh unambiguous triggers to ensure convergence. It is crucial to emphasize that such a search is seen as overly taxing for the parser, especially during online parsing, where cognitive resources are expected to be limited. Furthermore, this rigorous search process is undertaken even when the parser successfully parses a given input sentence. Identifying unambiguous triggers, which involves confirming that no alternative analysis could also successfully parse an input sentence, is even more challenging than merely finding a single successful interpretation. This intensive search process is obligatory for *every* input sentence, as long as not all parameters have been successfully set.

4 The grammar selection approach

In this section, we propose that the challenges faced by the direct parameter setting approach, particularly the requirement for extensive searching during online parsing, can be addressed by its alternative: the grammar selection approach. As mentioned earlier in Section 3, this alternative has been in consideration since Chomsky (1965) and was seriously explored in computational implementations before the emergence of the direct parameter setting approach.

The grammar selection approach assumes that the goal of language acquisition is to identify the target grammar from the pool of possible human grammars defined by UG. The grammar selection approach avoids the extensive search challenge during online parsing noted in the direct parameter approach, because the learner selects the grammar with a full set of parameters with their hypothetical values. Thus, any possible grammar selected from the grammar pool can be directly used as the grammar that guides the parser. There is no need to learn the parameter values during parsing since all the parameter values have been presupposed or hypothesized in possible grammars. In addition, some current grammar selection models do not distinguish ambiguous triggers from unambiguous triggers and make use of both for learning. In fact, there is no need to impose such a distinction. As we will explore further, computational models based on grammar selection can effectively set parameters without needing this differentiation, given sufficient input sentences.[6]

4.1 Computational implementations of the grammar selection approach

The grammar selection approach was implemented in various computational learning models. Due to space considerations, we will provide a succinct review of only two of its representative models: Gibson & Wexler's (1994) Triggering Learning Algorithm (TLA) and Yang's (2002) general Variational Learner (VL). The aim of this brief overview is to dip into the grammar selection approach to discern its major advantages and disadvantages.

4.1.1 The Triggering Learning Algorithm

Gibson & Wexler's (1994) TLA adopts the idea of grammar selection, and implements the learning algorithm in a rather conservative manner. TLA assumes that the learner entertains a particular grammar unless evidence suggests it is not the target grammar. For instance, if an input sentence cannot be analyzed by the adopted grammar, the learner is compelled to consider an alternative grammar; otherwise, the grammar should be retained. In other words, adjustments to the learner's knowledge state are driven by parsing failures. If an attempt of parsing with the current grammar G fails, the learner tries to parse the sentence again with a modified grammar G'. According to TLA, G' is obtained by resetting the value of only one parameter in G, chosen at random. This strategy of grammar selection is called the Single Value Constraint, adopted from Clark (1989): The TLA considers only grammars that differ from the current grammar by the value of one parameter, thereby acting conservatively. When considering an alternative grammar, another constraint, the Greediness Constraint, is applied: A proposed grammar is adopted only

[6] A key question in this approach is: How much input is sufficient? More critically, the input needed for convergence grows exponentially with each added parameter. We return to these challenges in Section 4.1.2.

if it can be used to parse the current sentence successfully. That is, if the grammar in use, G, cannot analyze a given sentence s, TLA shifts from G to an alternative grammar, G', if and only if G' renders a successful parse of s. This whole process repeats until the learner encounters no more unanalyzable input, culminating in the stabilization of the final grammar as the target grammar.

TLA postulates that learning is gradual and the shift from one grammar to its alternative is highly conservative. Both the Single Value Constraint and Greediness Constraint deter the learner from making drastic transitions to an alternative grammar when encountering input that the current grammar cannot analyze. Critically, since TLA evaluates only the current grammar and a possible alternative grammar at every point of learning, its computational demands remain relatively minimal. This mitigates a salient concern in earlier models about the psychological infeasibility of concurrently evaluating all possible grammars in the hypothesis space in parallel (e.g., in Clark and Roberts's Genetic Algorithm, as described in Clark 1992 and Clark & Roberts 1993).

However, this conservatism also introduces a critical issue for the TLA: The algorithm can potentially become “trapped” in a grammar that, while not the target grammar, bears a notable resemblance to it. In this case, if the current grammar does not differ from the target grammar in only one parameter, and if there is a competing grammar that is also sufficiently similar to the target grammar but deviates from the current grammar in only one single parameter, then the TLA may oscillate between the current and competing grammars upon encountering a parsing failure. This switching-back-and-forth process can lead the algorithm to become trapped in a local maximum. This problem has been confirmed by simulation results, in which TLA often struggles to converge on the target grammar (Kohl 1999; Sakas et al. 2017).

4.1.2 The Variational Learner

Yang (2002) proposes another computational implementation of the grammar selection model, integrating a statistical learning algorithm, namely, the Linear Reward-Penalty Scheme (Bush & Mosteller 1951). Yang's (2002) VL assumes a hypothesis space which consists of possible human grammars as combinations of parameters and their values. The learner's task is to navigate through this hypothesis space to identify the target grammar. The navigation is guided by the Linear Reward-Penalty Scheme, essentially identical to a reinforcement learning algorithm. The algorithm works as follows: A grammar G_i is sampled from the hypothesis space (analogous to a grammar pool in this paper) according to the probability distribution of the hypothesis space. Given an input sentence s, if G_i can analyze s, then the probability of G_i is increased, and meanwhile the probabilities of all other possible grammars in the hypothesis space are decreased. Otherwise, if G_i cannot parse s, decrease the probability of G_i and meanwhile increase the probabilities of all other possible grammars in the hypothesis space. When another input sentence is presented, a grammar that could be the same as or different from G_i is sampled to parse this new sentence. Again, the probabilities of all grammars in the hypothesis space are updated based on whether this grammar can parse this sentence. This process continues until the probability of the target surpasses all competing grammars, signaling that the learner has learned the target grammar.

VL is a highly successful model for parameter setting, and it possesses several advantages. One is that it advances the line of research of integrating UG and statistical learning: UG provides a hypothesis space and statistical learning guides the learner toward the target grammar (see also Lidz & Gagliardi 2015; Pearl 2021). This learning algorithm has been demonstrated to consistently converge, given sufficient input (Straus 2008; Sakas et al. 2017). As a result, VL sidesteps the pressing issue faced by TLA, where TLA may not converge on the target grammar due to the learner becoming ensnared in a local maximum, regardless of the amount of input provided. In contrast, VL's grammar sampling allows the algorithm to predominantly utilize the most probable grammar, but it also promotes occasional exploration into grammars that appear less likely as the target grammar. This exploratory mechanism, to a certain extent, safeguards the algorithm from settling on a non-target grammar.

VL functions optimally with a relatively small grammar pool. For example, when considering only 3 parameters, and the grammar pool comprises $2^3 = 8$ possible grammars, navigation through the hypothesis space is relatively easy. However, as Sakas et al.'s (2017) simulation illustrates, when the number of parameters rises to just 13 (comprising only basic parameters), navigating the hypothesis space—specifically, their CoLAG domain, which contains significantly fewer grammars than $2^{13} = 8{,}192$ (it's 3,072 to be exact)—requires more than 360 thousand input sentences. Of course, the exact number of input sentences needed is also affected by the learning rate. What is important here is that, with each additional parameter, the number of possible grammars in the hypothesis space doubles. This suggests that if there is a significant increase in the number of parameters—a scenario that seems plausible when considering parameters as the explanatory factors for crosslinguistic systematic variation (Longobardi 2018; Baker 2021; Sheehan 2021)—VL would necessitate a vast number of input sentences. This point forms a primary criticism against VL (e.g., Fodor & Sakas 2004). Another potential limitation of VL pertains to its algorithm, which updates the probabilities of all possible grammars in the hypothesis space after parsing each input sentence. This concern is relevant to a grammar selection approach, as an intent of these extensive updates is to ensure that the cumulative probability of all possible grammars in the hypothesis space sums to 1. Such a hypothesis space provides a basis for grammar sampling according to the probability of the possible grammars.[7]

[7] Another potential rationale is that all possible grammars are in competition. Therefore, if the grammar currently being learned is not the target grammar, the target grammar must be among the others, and vice versa. This reasoning is irrelevant when considering that Yang (2002) allows for the coexistence of multiple competing grammars as a learning outcome.

In sum, VL requires extensive updates to probabilities after each parse. As a result, it demands a memory system to store and monitor the probabilities (and updates) of all potential grammars in the hypothesis space. This imposes a significant strain on computational resources. We propose that such challenges can be addressed using the direct parameter setting approach.

5 Contrasting direct parameter setting with grammar selection

We have discussed a previously raised concern regarding VL: It requires a substantial number of input sentences for the algorithm to converge on the target grammar, particularly when the number of parameters realistically reflects systematic cross-linguistic variations. Additionally, the model must keep track of the probabilities of all possible grammars in the hypothesis space and their updates. This could lead to considerable memory consumption as the number of parameters increases.

Crucially, both of these problems seem to stem from the concept of grammar selection. Again, according to the grammar selection approach, the learning algorithm is designed to identify the target grammar from a set of competing grammars. The number of competing grammars is determined by the number of parameters. The standard assumption is that all parameters are binary and thus each has two values. The number of possible grammars is equal to 2 raised to the power of *n*, where *n* is the number of parameters.[8] As the number of parameters increases, the total number of possible grammars increases exponentially. This dramatic growth also makes tracking their probabilities challenging due to constraints on memory resources.

In contrast, the direct parameter setting approach addresses these challenges inherent to the grammar selection approach. This is because, instead of dealing with possible grammars, it operates directly with parameters. There is no need to navigate through all possible grammars. The task becomes simpler, focusing on setting the values of a relatively small number of parameters, especially considering that the number of parameters is much smaller than the number of possible grammars. More importantly, within the direct parameter setting approach, introducing a new parameter does not drastically expand the hypothesis space. Thus, navigating this space is significantly more straightforward. Moreover, learners no longer need to track and update the probabilities of possible grammars. The probabilities of parameters (with their values) are instead tracked and updated. As a result, both challenges faced by VL are addressed by implementing the direct parameter setting approach.

[8] Although in reality the number is likely smaller due to various reasons, for example, not all languages generate a significantly distinguishable set of sentences. Please refer to the discussion of the CoLAG domain by Sakas & Fodor (2012).

However, as we have previously discussed, the direct parameter setting approach comes with its own set of challenges, notably the extensive search required during online parsing. Remember that models like STL operate effectively only when unambiguous triggers are differentiated from ambiguous triggers. The need to identify unambiguous triggers necessitates the exhaustive online search of all possible analyses for a given input sentence. An input sentence can be considered an unambiguous trigger only when all but one of these searches fail. Importantly, the grammar selection approach does not suffer from this issue. This is because it assumes that the parser simply samples a possible grammar with hypothetical parameter values from the grammar pool to analyze a given input sentence. Given that grammar sampling is both random and constrained (by the probability distribution), balancing both exploitation and exploration, the parser does not need prior knowledge of which parameters are pertinent before sampling. There is also no need to distinguish unambiguous triggers from ambiguous triggers, because models like VL have proven effective even when faced with ambiguous triggers. The differences between the grammar selection and the direct parameter approaches are summarized in **Table 1**.

Grammar selection approach	**Direct parameter setting approach**
Selects the grammar with a complete set of parameters with their hypothetical values.	Extensive search for treelets during online parsing.
No need to distinguish ambiguous triggers from unambiguous triggers.	Needs to distinguish ambiguous triggers from unambiguous triggers (All STL variants).
Requires a large amount of input sentences.	Requires fewer input sentences.
Tracks the probabilities of all possible grammars in the hypothesis space, which overburdens the computational resources.	Tracks only the probabilities of the parameters in use (Non-deterministic STL).

Table 1: Contrasting the grammar selection with the direct parameter setting approach.

From the above analysis, we seem to be caught in a dilemma. Both the grammar selection and the direct parameter setting approach face significant challenges, making neither of them an optimal solution within the parameter setting framework. However, it is important to note that they seem to counterbalance each other’s shortcomings. The issues of the grammar selection approach are resolved by the direct parameter setting approach, and vice versa. This observation gives hope for a more ideal model. By effectively merging these two approaches, we might harness the strengths of both while bypassing their limitations. We will delve into such hybrid approaches in the subsequent section.

6 The hybrid approaches

As we have pointed out, these two approaches, the direct parameter setting approach and the grammar selection approach, complement each other. We will now demonstrate how they can be integrated. In this section, we introduce an essential hybrid precursor of our Clustering Approach known as the Naive Parameter Learner (NPL) (Yang 2002: Section 2.4). Yang's NPL reserves grammar sampling for parsing from grammar selection but meanwhile adopts the update of the probabilities of parameters instead of possible grammars. Next, we will introduce our hybrid approach, the Clustering Approach, and highlight the challenges it addresses.

6.1 A precursor: Yang's (2002) Naive Parameter Learner

Yang's (2002) NPL positions his classical, theory-neutral VL algorithm under the P & P framework. NPL differs from VL in that it directly applies the reward and punishment algorithm to parameters. Like the direct parameter setting approach, NPL sets parameters directly and individually. Each time an input sentence is encountered, a grammar comprising a list of parameter-value pairs is sampled to parse it. This is how the NPL approach addresses the challenges associated with the grammar selection approach: It reduces the search space and does not need to update and track the probabilities of possible grammars, focusing only on the probabilities of individual parameters. Moreover, under NPL, adding a new parameter does not substantially expand the parameter pool/set. On the other hand, NPL, echoing the grammar selection approach, circumvents the immense online search dilemma inherent in the direct parameter setting approach. A grammar is sampled from the grammar pool for parsing purposes. There is no longer a need for the parser to search for parametric treelets to provide a plausible parse of an input sentence. In addition, quick learning is possible with general input without the need to distinguish unambiguous triggers from ambiguous triggers. Problems caused by ambiguous triggers do not significantly affect the NPL approach, as they can be properly handled by statistical learning, as demonstrated in Sakas et al. (2017) with simulations based on the CoLAG domain. This is possible because, as long as the amount of evidence consistent with a particular grammar is sufficient and outweighs the evidence (including ambiguous data) supporting a competing grammar, the target grammar will gain preference in grammar selection and will ultimately prevail. Similarly, noisy input is filtered out to some extent by NPL (like VL)'s frequency-sensitive learning algorithm.

However, an important concern with the NPL is that as the number of possible grammars increases (e.g., with more than 10 independent parameters), the NPL, as well as the VL, faces an extended period of exploration: Identifying the target grammar that always correctly parses the input sentences is challenging with a random search into a vast hypothesis space. For example, the probability of finding the right grammar out of a hypothesis space generated by 20 parameters is

$\frac{1}{2^{20}}$. This issue is inherent to all models that assume grammar selection for parsing but update all parameters in the selected grammar upon successful parsing. The issue is not merely the amount of input sentences the model requires to converge on the target grammar; more importantly, it implies that the learner does not set any parameters for a considerable amount of input sentences while exploring the hypothesis space. This prolonged exploration period is a consequence of the difficult random sampling of the target grammar from the grammar pool, irrespective of whether the learner has the competence to process language. When a non-target grammar is selected, it may either successfully parse the given input sentence or fail to do so. In the former situation, a list of parameters with incorrect values, differing from those in the target grammar, will be rewarded. In the latter instance, a list of parameters that may be part of the target grammar are penalized. The introduction of more parameters into the target grammar will simply result in greater confusion.

Such an issue is significant as we assume in the learning model that children are equipped with a parser capable of parsing the linguistic input, given a hypothesized grammar. This predicts that even if children already have the competence for parsing, they cannot learn parameters from the input simply because it is almost impossible to select the target grammar out of a big grammar pool. To compound this issue, a lack of parameter setting could persist even when the learner has successfully parsed a significant number of sentences. Such a prediction seems counter-intuitive and empirically unsupported, as most (but not all) well-explored parameters are acquired fairly early (Yang 2002: 46). Parameters that are frequently and consistently embodied in the input will be learned first, such as head-directionality parameters, comparing to parameters that are associated with more mixed or variational input, such as the null subject parameter and some agreement-related parameters (see Thornber & Ke 2024 for a computational analysis of the production data, see also Ayoun 2003: 104, and references therein as well as Tsimpli 2014; cf. Wexler 1998 and Rizzi 2008 for a relevant debate).

As will be discussed below, the Clustering Approach attempts to address this issue. It posits that only the parameters used in the input sentence will be updated upon a successful parse, adopting STL's concept of learning from parsing with parametric treelets. In addition, only a limited number of parameters that function as sampling parameters[9] will be penalized upon a parsing failure, due to the idea of clustering. As mentioned in Section 2, the sampling parameter, which exhibits the most extreme probability deviation from 0.5 (approaching either 0 or 1), is selected to partition the current grammar pool. This parameter is likely to be learned before others if it is frequently and consistently observed in the input.

[9] In this paper, we assume this number to be one for the sake of simplicity, though it could be larger. In fact, sampling based on multiple parameters and their probabilities may help select a grammar that is more likely the target, potentially accelerating the convergence process.

6.2 The Clustering Approach

In this subsection, we will first detail the main mechanisms of the Clustering Approach, followed by a more technical description of the mechanisms. We will then provide computational simulations that compare the VL, NPL, and the Clustering Approach, and discuss their implications. Note that our purpose is to demonstrate the logical plausibility and feasibility of the Clustering Approach in dealing with a hypothesized grammar with a large number of parameters.

As a hybrid approach, the Clustering Approach incorporates many mechanisms from the NPL. **Figure 2** presents a flowchart illustrating the Clustering Approach. We assume a grammar

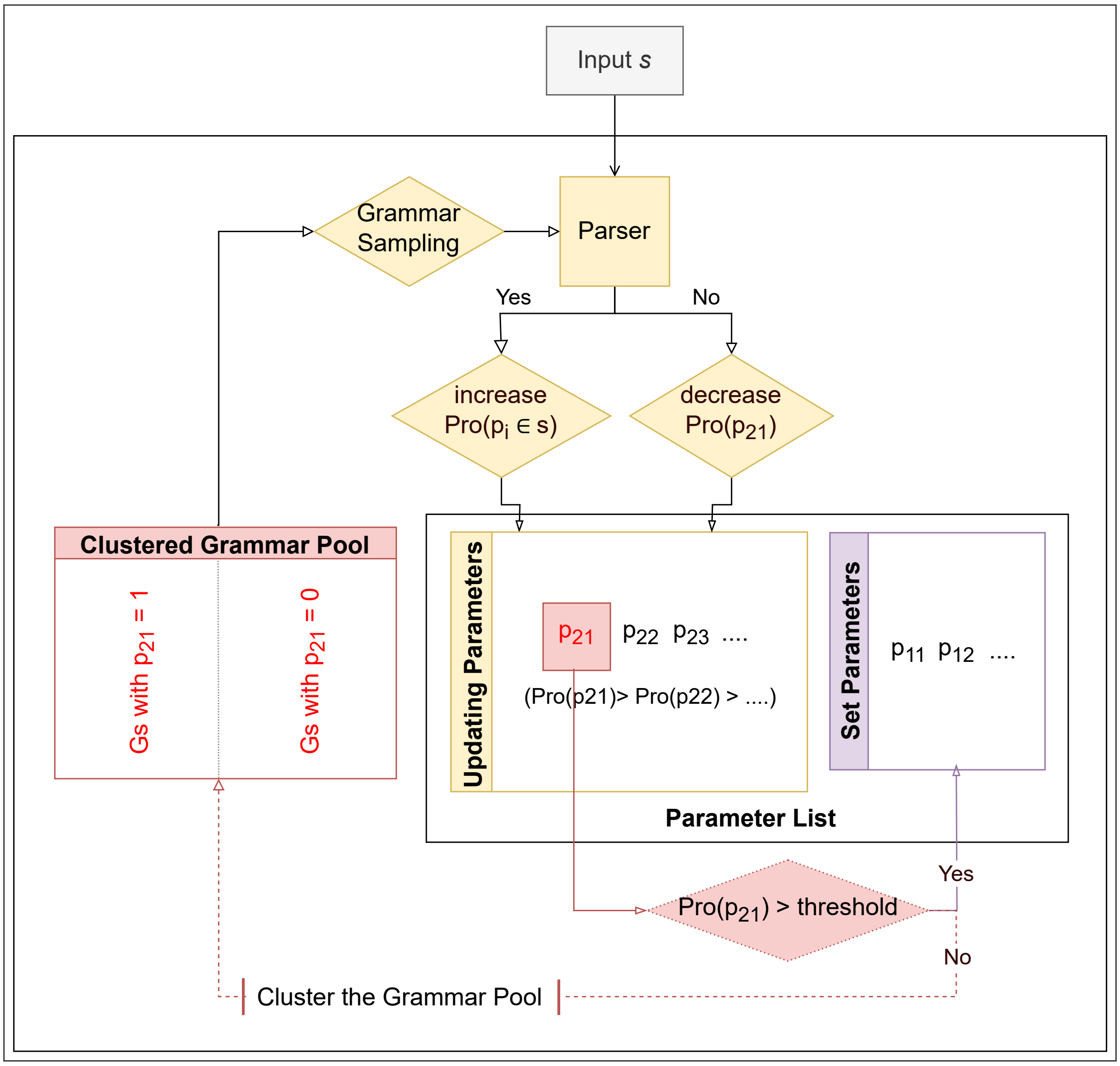


Figure 2: Parameter setting in the Clustering Approach.

pool that consists of all possible grammars determined by parameters from UG or derived from other sources. The size of this grammar pool corresponds to the number of all possible combinations of parameters and their values. Therefore, the initial size of the grammar pool is equal to what is assumed in both the VL and NPL models. Under the Clustering Approach, the grammar pool is hypothetical, designed solely to generate a possible grammar for the parser. It serves no other function beyond this. Consequently, the Clustering Approach does not keep track of the probabilities associated with possible grammars for grammar sampling. As such, it is "lightweight."

This grammar pool ensures that the parser adopts possible, yet restricted, ways to analyze the structure of the input sentences. The parser then interprets the input based on its selected grammar and learns from the parsing result, as suggested by Fodor (1998b) and Lightfoot (2020). The parameters manifest in the input as parts of tree structures (e.g., Sakas & Fodor 2012), allowing parameter values to be identified from the structures. Children's task is to obtain an interpretation of a given input sentence by constructing a syntactic tree for it. If a syntactic tree can be built, the corresponding parameter values encoded in the syntactic tree can be identified, which will be fed to a learning algorithm.[10] For example, if a learner's parser constructs a syntactic structure with a PP in which the P is to the left of its complement, as this is a plausible way to analyze the input sentence, then the parser can learn from the parsing result that this language is more likely head-initial in the structure of PP.

Before we proceed, it would be helpful to clarify the use of several terms within the Clustering Approach. All possible grammars, as combinations of parameters and their values, are accessible to the parser. These parameters may be innate or derived from interactions among innate domain-specific constraints, domain-general constraints such as logical options and cognitive biases, as well as other factors (see Section 7.2). Only the parameters that have been utilized in successful parsing will be collected into the set of *updating parameters*. We refer to the parameters that have been used to parse a specific sentence as the *parameters in use*. Only the parameter with the most extreme probability among the updating parameters will be employed for clustering the current grammar pool and guiding the sampling of a grammar from the grammar pool to parse the input. This parameter is termed the *sampling parameter*. Once a parameter is set, it will be classified into the *set parameters*, and its probability will no longer be updated. The set parameters reduce the size of the grammar pool as the possible grammars with different values than the set values are excluded from the future sampling process.

[10] This partially addresses the linking problem in language acquisition as children can relate the abstract knowledge of parameters to the parsing of specific input sentences. Of course, this approach assumes that children should have learned the syntactic categories of words before the learning of parameters.

Like the NPL, the learner samples a grammar from the grammar pool and uses that grammar to parse a given input sentence. This sampling process is guided by the probability of the sampling parameter and the set parameters. For example, as shown in **Figure 2**, if set parameters include two parameters P11 = 0 and P12 = 1, all the possible grammars of P11 = 1 or P12 = 0 will be excluded from the grammar pool. Say the sampled grammar has the following parameter-value pairs, and each pair is associated with a probability:

(5) *An example of a selected grammar*

a. *Set parameters*: P11 = 0, P12 = 1

b. *Updating parameters*: P21 = 1, P22 = 0, P23 = 1, P24 = 1

The updating parameters include P21. Suppose it has the most extreme probability value (closest to either 0 or 1). Therefore, P21 will be the sampling parameter at this point. The sampling parameter will be dynamically updated after each round of update of the parameters in use. A grammar will be sampled from the cluster of possible grammars with P21 = 0 and another cluster of possible grammars with P21 = 1. The probability of sampling a particular cluster is determined by the probability of P21 = 0 or P21 = 1. As mentioned, the Clustering Approach does not track the probabilities of all possible grammars. Similar to the STL but different from NPL, the Clustering Approach tracks only the probabilities of updating parameters.

If the parser successfully constructs a syntactic tree for an input sentence by applying the parametric treelets associated with the selected grammar, we consider the parsing successful. For instance, if an input sentence can be parsed with the selected grammar in (5), a syntactic structure can be built based on the parameters.

Not all parameters of the selected grammar are necessarily used in the structure of the particular input sentence. To give a more concrete example, if a sentence does not include a PP in the structure, P-stranding is not relevant for that sentence and will not be employed to parse that sentence. Therefore, P-stranding is not a parameter in use. The Clustering Approach will update only the parameters in use and will not affect the parameters that are not used in the parsing of a particular input sentence. We call this “accurate updating.”

In such an instance, the learner identifies which parameters and their values have been utilized in parsing. This identification of specific parameters in use is feasible due to STL’s assumption that the parametric values can be mapped to treelets in a structure and vice versa.[11] These parameters are added to the set of updating parameters, if they are not already included. The probabilities

[11] This mapping may involve analytical ambiguity, which is outside the current scope of consideration. We are examining the effect of ambiguity in one of our current projects.

of the updating parameters that are used in the parsed sentence are updated based on the Linear Reward-Penalty Scheme, which is also implemented in NPL.

It is important to note that the Clustering Approach accurately updates the probabilities only for the parameters used in a parsed sentence, a key aspect of its strictly input-driven nature. Via this process, a series of sampling parameters will be identified based on their consistent and frequent usage in the input sentences, leading to a learning order that is input-informed. This selective updating is a distinctive feature of the Clustering Approach, setting it apart from NPL, which updates all parameters in the selected grammar upon a successful parse. The reader can now relate this more targeted update process to the specification of the hierarchical order based on the learning order of the parameters. It is through this more targeted or accurate input-driven update process combined with the nature of the input, that is, some parameters are used more frequently and consistently than some others, that we have identified the learning order of parameters, and thus use the learning order to organize the grammar pool.

If parsing fails, there are, in principle, at least five options:

(6) a. The parser does not know which parameter(s) fails the parsing; thus, the input sentence is ignored, and there are no updates on any parameter probability (see, e.g., Yang 2012: 209).

b. The parser makes a random guess and punishes one of the parameters by decreasing its probability (similar to TLA's algorithm).

c. The parser discredits all parameters because of this failure (e.g., VL and NPL).

d. The parser knows which parameters or their values have caused the parsing failure, as in Fodor (1998a) and Lightfoot (2020).[12]

e. The parser discredits only the parameter that is used for sampling (the Clustering Approach).

We adopt the last option as our working hypothesis for the simulations in this article. This option is feasible because in the Clustering Approach, the current grammar pool is clustered by the sampling parameter and only the sampling parameter is used for sampling. If a grammar sampled according to this parameter fails to parse the sentence, then there is a high chance that the sampling parameter has the wrong value, although any other parameters may in fact be the real contributing factors. It is noteworthy that the last option is more conservative than

[12] A possible way to understand this option is that if a part of an input sentence cannot be successfully parsed, then the parser may be able to know which part of the sentence causes a problem. The parser can then guess whether changing certain parameter values would render the sentence parsable. We believe this probably requires too much from the learner/parser.

the fourth, as it avoids the added assumption that the parser can identify the specific parameter responsible for a parsing failure. Nonetheless, each of these options presents intriguing avenues for further study.

After a certain number of iterations, the probability of some specific parameter might reach the upper threshold (e.g., 0.9), causing the parameter to be set to 1, or it might fall under the lower threshold (e.g., 0.1), leading the parameter to be set to 0.[13] Once a parameter, P_i, has its value set to either 1 or 0, it will be utilized to cluster the grammar pool. Consequently, the grammar pool is grouped into two clusters: one where P_i is set to 1 and the other where it is set to 0. In future grammar sampling, only the cluster that has a P_i value consistent with the one set in the learning process will be considered. This effectively reduces the size of the grammar pool for grammar sampling by half, eliminating from consideration all grammars with the incorrect p_i value. As more parameters are set in their values, the grammar pool quickly shrinks. The target grammar emerges if the parameters are set according to their values in the target grammar.

More specifically, we assume each grammar is of a set of parameter-value pairs, $\{(i, p_i) \in \mathbb{Z} \times \{0, 1\}, i = 1, \ldots, n\}$ where n is the total number of parameters.[14] The number i represents i-th parameter and p_i represents its value. The model specifies (i) a corpus of m sentences $\{s_i\}_{i=1}^{m}$, where each sentence s_i is of a set of parameter-value pairs $\{(j, p_j) \in \mathbb{Z} \times \{0, 1\}, j \in I_i\}$ and I_i is a subset of $\{1, \ldots, n\}$, (ii) the learning rate $\gamma > 0$, and (iii) a threshold T for determining whether a parameter needs to be set to 0 or 1.

The parameter updating is recorded by the triplet $P = (P_{su}, P_u, P_s)$, which consists of three sets: The set P_{su} represents parameters being updated and used in grammar sampling; the set P_u represents parameters being only updated but not used in sampling; the set P_s collects parameters whose value has been set to 0 or 1. Thus, $P_{su} \cup P_u$ is the set of updating parameters, and P_s is the set of set parameters. Each element in $P_{su} \cup P_s \cup P_u$ should be of the parameter-value form $(i, p_i) \in \mathbb{Z} \times [0, 1]$. Here i is the i-th parameter, and p_i is the associated probability of i-th parameter being 1 estimated thus far. Their initial values are all empty sets.

The algorithmic description of the Clustering Approach can be found in **Algorithm 1** (the overall architecture), **Algorithm 2** (the grammar sampling algorithm), and **Algorithm 3** and **4** (sentence parsing and probability updating).

[13] There is a chance that the parameter may be set incorrectly if the input contains a high percentage of noisy or ambiguous sentences. However, we can decrease the likelihood of an incorrect setting by imposing a more extreme threshold, or employing a pseudo-batch approach similar to that in Yang (2002). The latter approach requires the selected grammar with the current parameter setting to successfully parse a certain number of input sentences consecutively before the model makes a commitment to a value even after the probability of the parameter has reached the threshold.

[14] The symbol $\mathbb{Z}$ represents the set of integers.

Algorithm 1 The Clustering Approach.

```
Input: corpus of sentences $\{s_i\}_{i=1}^{m}$, learning rate $\gamma > 0$, and a threshold $T \in [0,1]$
Output: A set of parameters with their associated estimated probabilities $P$
  Initialize $P_{us} \leftarrow \{\}$, $P_u \leftarrow \{\}$, and $P_f \leftarrow \{\}$
  for $t \leq m$ do
                                   ▷ Sample a grammar and use it to analyze the sentence
     Sample a grammar $G_t$ according to Algorithm 2
     Analyze $s_i = \{(j,p_j)|j \in I_i\}$ with $G_t$ and obtain result $A \in \{0,1\}$ according to Algorithm 3
                                   ▷ Update probabilities according to the success or failure of the analysis
     if $A = 0$ then
        Update the probability in $P_{us}$ according to Algorithm 4 using $\gamma$ and $G_t$
     else
        Update the probability in $P_{us}$ and $P_u$ according to Algorithm 4
        for any parameter $j \in I_i$ that has not appeared as a parameter in $P_{us} \cup P_u \cup P_f$,
        Add $(j, |p_j - 0.4|)$ to $P_u$, where $(j,p_j) \in s_i$.
     end if
                                   ▷ Clustering by fixing parameter values
     if $A = 1$ and a parameter-value pair in $(s,p) \in P_{us}$ satisfies $\max\{p, 1-p\} > T$ then
        Set $p = 1$ if $p > T$ and $p = 0$ otherwise.
        Add $(s,p)$ to $P_f$ and delete $(s,p)$ from $P_{us}$
     end if
                                   ▷ Changing $P_{us}$ and $P_u$
     Let $P_t = P_{us} \cup P_u$ and $(s',p')$ be the parameter-value in $P_t$ with largest $\max\{p', 1-p'\}$.
     Set $P_{us} = \{(s',p')\}$ and $P_u$ be the set difference of $P_t$ and $P_{us}$.
  end for
```

Algorithm 2 Grammar sampling.

```
Input: Two sets of parameters and their associated estimated probabilities $P_{us}$ (the updated and
  used for sampling set), $P_f$ (the set with fixed parameter values), and the number of parameters $n$
Output: A sampled Grammar $G$
  Initialize $G = \{\}$
  for each $1 \leq i \leq n$ do
     If $i$ is a parameter in $P_{us}$ or $P_f$, sample a Bernoulli random variable with success probability,
  the probability of being 1, $p_i$. Let the realized value be $l_i$. Add $(i, l_i)$ to $G$
     Otherwise, sample a Bernoulli random variable with success probability, the probability of
  being 1, 1/2. Let the realized value be $l_i$. Add $(i, l_i)$ to $G$
  end for
```

Algorithm 3 Grammar analyzing a sentence.

```
Input: A Grammar $G$ and a sentence $s$
Output: A binary value $A \in \{0, 1\}$                ▷ 1 for success of analysis
  Initialize $A = 1$
  for each parameter $i$ in $s$ do
      If the associated value $p_i$ in $s$ is different from the $p_i$ in $G$ for parameter $i$, set $A = 0$
  end for
```

Algorithm 4 Updating probabilities in a set.

```
Input: A set of parameter-probability pairs, $P$, a learning rate $\gamma$, a given language $G$, and a binary
  value $A$                                            ▷ $A = 1$ for success of analysis
  for each parameter key $i$ in $P$ do
      Compute $p_1 = \min(p_i \times (1 + \gamma), 1)$
      Compute $p_2 = (1 - \gamma) \times p_i$
      if the parameter $i$ in $G$ has value 1 then
          Compute $p_i = p_1 \times A + (1 - A) \times p_2$
      else
          Compute $p_i = p_1 \times (1 - A) + A \times p_2$
      end if
  end for
```

6.3 Simulations with synthesized data

A number of simulations are conducted to explore the properties of the Clustering Approach in the learning of artificial languages, in comparison to two closely related benchmark models, namely, the VL and NPL. Recall that the primary goal is to verify whether the Clustering Approach can address a crucial issue in VL and NPL: namely, that with a larger number of parameters, the learner does not begin learning the target grammar or setting its parameters until after an extended period of exploration (even in the case where the learner has successfully parsed a significant number of input sentences). On the contrary, what we would like to see is that, even with a large hypothesis space, at least some parameters are learned earlier and overall the parameters are gradually learned. For example, the parameters that are frequently and consistently observed in the input should be learned earlier than the parameters that occur less frequently.

We construct a few corpora of synthesized sentences as part of the input to **Algorithm 1**. The corpora are of m sentences, where $m = c{*}n/2$, with the integers c representing a constant number and n the number of parameters. The sentences in the corpora are partitioned to groups with

$4 + 2 + 2 + 2 \ldots + 2$, if more than 4 parameters are used. This is to create sentences with a varying number of parameters. For example, if 8 ($= 4 + 2 + 2$) parameters are used for a language, the distribution of which is given by their relative frequencies, the corpus will comprise three groups of sentences: (i) a random sampling of short sentences with 1 to 4 parameters from the overall distribution of parameters; (ii) a random sampling of longer sentences with 5 to 6 parameters; and (iii) a random draw of sentences with 7 to 8 parameters. If there are 4 parameters or less, all sentences will be drawn from the distribution of these 4 parameters. In the simulations with 12 parameters or less, we set c to be 5,000. That is, 12 parameters will have $c * n/2 = 5000 \times 12/2 = 30000$ sentences. c is increased to 8000 for 16 or more parameters as more parameters generally require more input for successful learning.

As mentioned, the initialization consists of three sets: the set P_{su} represents parameters being updated and used in sampling possible grammars; the set P_u represents parameters in use, whose probabilities are updated when parsing is successful, but excluding the sampling parameter; the set P_s represents parameters whose values have been set to 0 or 1. In the simulations, we set their initial values to be empty sets.

We construct eight artificial corpora of sentences which includes 2, 4, 8, 12, 16, 20, 24, and 28 parameters. The distribution of the parameters in sentences approximates a Zipf distribution (Zipf 1949), with some parameters occurring more frequently than others.[15] The sentences are created by randomly combining the parameters in the range of parameters.[16]

Figure 3 contrasts the Clustering Approach (represented by the solid green line) with the VL model (dashed blue line) and the NPL model (dash-dotted orange line). The latter two act as benchmarks in the computational simulations. **Figure 3** results from 400 iterations, with a learning rate set to 0.05 for all three approaches. The target grammar probability results are averages over the results obtained from all iterations. We set the high-bound and low-bound threshold to 0.9 and 0.1.[17] For the Clustering Approach, this implies that if a parameter with a certain value has a probability higher than 0.9 or lower than 0.1, it is considered settled, and it will be used to cluster the grammar pool. Consequently, only the cluster of possible grammars with the settled parameter value will be used for grammar sampling for future parsing of the input. This reduces the size of the grammar pool by half, and thus increases the chance of identifying the target grammar from the grammar pool in the future.

[15] The list below represents relative frequencies of the parameter in the language corpus with 16 parameters: $p_1 = 1, p_2 = 1/2, p_3 = 1/2, p_4 = 1/3, p_5 = 1/3, p_6 = 1/3, p_7 = 1/6, p_8 = 1/6, p_9 = 1/8, p_{10} = 1/8, p_{11} = 1/10, p_{12} = 1/10, p_{13} = 1/10, p_{14} = 1/13, p_{15} = 1/13$, and $p_{16} = 1/13$. The actual frequency of p_2 in a corpus with 2 parameters, with relative frequencies of $p_1 = 1$ and $p_2 = 1/2$, for example, can be calculated by $c \cdot \frac{1/2}{1+1/2}$.

[16] See the shared scripts in the Supplementary Files for detailed specifications of the corpora.

[17] The threshold could be closer to 0.5, in which case the learning may still converge with more noise but may cause parameters to be misset; or the threshold could be further from 0.5, which will reduce the likelihood of parameter missetting but increase the learning time or reduce tolerance to noise.

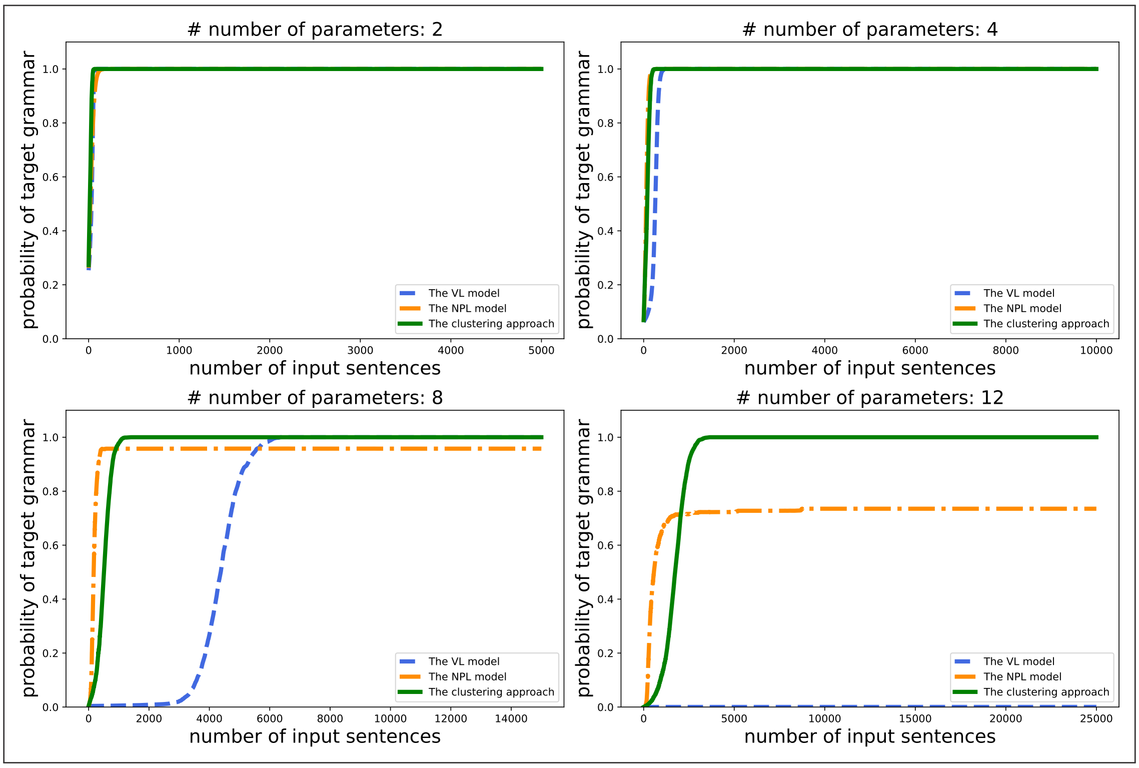


Figure 3: Comparing the Clustering Approach and Yang's (2002) VL and NPL modeled on synthesized data with 2, 4, 8, and 12 parameters.

These results suggest that the Clustering Approach method is indeed distinctive in terms of its beginning to learn from the input early and steadily converging on the target grammar. In addition, the number of input sentences the model requires for convergence does not increase too much with the increase in parameter number. The Clustering Approach can learn the parameters with a more reasonable number of input sentences when the number of parameters is large, for example, 12 or more. This is due to two crucial properties of the Clustering Approach: It learns the parameters that are used in the input (accurate updating) and the parameters are clustered as they are learned in an order informed by the input.

By contrast, although the VL and NPL learners perform as well as the Clustering Approach when the parameter number equals 2 and 4, the number of sentences the learners require for them to even begin learning the target grammar increases exponentially as the number of parameters increases. Imagine that when the parameter number is increased to more than 50 (see, for example, Longobardi 2018 and Crisma et al. 2020), VL will require too many input sentences for the algorithm to identify the target grammar and actually start learning. NPL will face a similar challenge. This is because when the parsing of an input sentence is successful with a selected grammar, the algorithm rewards all parameters in that grammar, including those not used in the input. This ultimately adds noise to the probabilities of the parameters and thus hinders learning. Again, the primary challenge for VL and NPL is not necessarily the quantity of sentences required to successfully learn the target grammar. By contrast, an important challenge for VL lies in the vast hypothesis space; with many possible grammars, learners may struggle to identify the target grammar. In addition, a more significant challenge for both VL and NPL is that the updates of the parameters are not accurately targeted at those that need updating; instead, the algorithm updates all parameters indiscriminately. This approach can lead to prolonged exploration periods during which parameters remain essentially unchanged, despite the learner having successfully processed a considerable number of input sentences. Note that obviously the current simulations have made the learning much easier due to the fact that the corpora do not include noise and ambiguous sentences, as the short actual learning periods (after the probability of the target grammar starts to increase) show: This further highlights the significance of the problematic "flat" learning period. For example, VL does not learn the target grammar at all from input derived from a corpus generated by 12 parameters, whereas NPL stops setting any additional parameters of the target grammar after learning most of the parameters (between 8–10) under the 12 parameter condition.

To further compare the performance of the Clustering Approach and the NPL at a synthesized language with more parameters, we simulated their learning of languages with 16, 20, 24, and 28 parameters. **Figure 4** is based on 100 iterations with otherwise the same parameters as the previous simulations. The results confirm that a hypothesis space generated by 16 to 28 parameters is challenging for the NPL model, but much less so for the Clustering Approach.

Another important aspect of the Clustering Approach is its emergent property. This is related to a potential concern that for some parameters, their triggers might be so rare in the input that they might never reach either the upper or lower thresholds. The Clustering Approach allows parameters that are most frequently and consistently found in the input to be set first, and the parameters which rarely occur or associate with significant variation will be set later (cf. Legate & Yang 2007). Could some of the parameters which are extremely rare be left unset in the end? The model's answer is yes. This suggests that children are uncertain about some parameters even though they have learned many other parameters. This predicts that in experimental studies, we should be able to observe some uncertainty with regard to those parameters (see e.g., Ke & Gao 2020). Which parameters are they? One of the central goals of our ongoing project is to identify the learning trajectories of different parameters, which has the potential to further test the empirical predictions of the Clustering Approach in experimental studies.

We also find, with additional experiments (results not shown here), that when the input involves a large number of complex structures generated by many parameters, NPL can fail to learn the target grammar due to problems previously termed as accomplices and hitchhikers (Yang 2002; Nazarov & Jarosz 2021). Specifically, given the difficulty in selecting the target

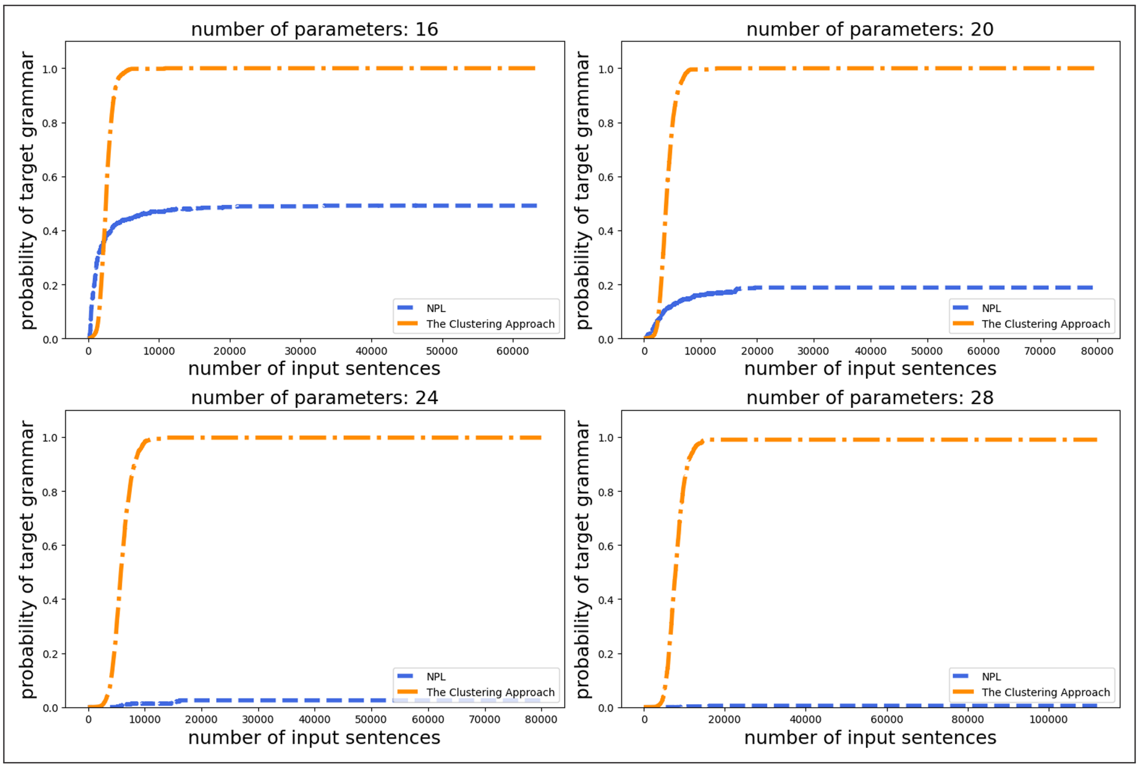


Figure 4: Comparing the Clustering Approach and Yang's (2002) NPL modeled on synthesized data with 16, 20, 24, and 28 parameters.

grammar from a vast hypothesis space at random, most parsing attempts fail. This results in penalties for parameters that are part of the target grammar (referred to as "accomplices"). On the other hand, when parsing is incidentally successful with a non-target grammar—due to cases where the input sentences may not include all parameters distinguishing the selected grammar from the target grammar (or due to ambiguous input compatible with multiple grammars)—the learner rewards all parameters of the selected grammar, including those not part of the target grammar, coined as "hitchhikers."[18] In other words, punishing or rewarding all parameters in the selected grammar when parsing fails or succeeds causes learning problems. Nazarov & Jarosz (2021) demonstrated that a significant amount of ambiguous structures in the input leads to similar issues. The Clustering Approach addresses these problems because the parser accurately updates parameters that are used in the input sentences, and when parsing fails, the parser penalizes only the sampling parameter rather than all parameters in the selected grammar. In

[18] Yang (2002) acknowledges the existence of hitchhikers but expresses the hope that this issue can be resolved in the long run. This may be true. However, our simulations suggest that the model may get stuck at a wrong parameter value for too long. The wrong parameter value may be further promoted without being corrected because (i) grammar selection is guided by the probabilities of these parameters, and (ii) the parameter may not be frequent in the input sentences.

our additional simulations, we observed that the learner can converge on the target without problems. However, in principle, there is still a possibility that accidental continuous penalization of a sampling parameter could lead to the missetting of this parameter, although the likelihood is low. To prevent a parameter from being set incorrectly, we require that the decision to set the parameter be made only after a reward process, as the reward process accurately reflects the parameters in the target grammar (see **Algorithm 1**). Our next step is to evaluate whether ambiguous input will cause problems for this learning algorithm.

Finally, the Clustering Approach, although always superior to the NPL model in our simulations, may not be able to learn the target grammar 100% of the time in the following circumstances: (i) Some parameters are too infrequent (e.g., 0.010 to 0.015 of the input) when the parameter number is increased to 20 or more; (ii) the corpus includes a fair amount of long sentences that consist of more than 20 parameters; (iii) the number of parameters is increased but the number of sentences in the corpus stays the same. Since these restrictions are reasonable constraints on a computational model of language acquisition, they do not raise immediate concerns to us. Whether they will cause serious problems for the Clustering Approach awaits future research.

7 Parameters in the Minimalist Program

In this section, our goal is to relate some of the assumptions of the Clustering Approach to Minimalist approaches to parameter setting. We demonstrate that representative Minimalist approaches we reviewed still assume the necessity of parametric knowledge, although this knowledge is not necessarily represented as the classical parameters. The term "parametric knowledge" implies that the knowledge can be translated into parameters or used to derive parameters.

7.1 The necessity of parametric knowledge

To validate our assumption that parametric knowledge is necessary even in the Minimalist Program (MP) approaches, we will firstly provide an (incomplete) review of some representative MP approaches to cross-language variation, regardless of where the knowledge is stored.

MP conceptualizes the (core) language system as an optimal solution to satisfy the requirements imposed by the external systems interfacing with the language faculty. Optimality is defined such that the system contains only what is necessary (i.e., being optimal). Consequently, parameters are eliminated from UG, primarily the narrow syntax, leaving UG to comprise only basic syntactic operations such as Merge, including internal and external Merge, yielding recursion (Hauser et al. 2002; but see Pinker & Jackendoff 2005), and more controversially, also Agree (Chomsky et al. 2019) and Labeling (see e.g., Biberauer et al. 2014; Ke 2024). Chomsky (2005) identifies the three factors of optimal language design as follows:

i. Universal Grammar (UG): the initial state of language development, determined by human genetic endowment.

ii. Experience: the source of language variation.

iii. Third factor: principles that are not exclusive to the language faculty but applicable in other cognitive domains.

Under the MP framework, language acquisition is a result of the interaction between UG, experience, and domain-general strategies (third factor) rather than from setting the values of an inventory of parameters specified by UG through learning.

However, if parameters are indeed eliminated from UG, the question arises: How do children acquire the systematic variation in different languages? Borer (1984) and Chomsky (1995) attribute language's systematic variation to differences in the formal features of the functional heads in the lexicon, which can be learned through linguistic experience. Borer (1984) emphasizes the benefit of associating functional features (FFs) with heads, noting that "associating parameter values with lexical entries reduces them to the one part of a language which clearly must be learned: the lexicon (Borer 1984: 29)." This hypothesis is termed the Borer-Chomsky Conjecture (BCC) by Baker (2008b).[19] Nevertheless, this approach still leaves the question open: How is the inventory of the FFs determined such that they can capture constrained systematic variation across languages? We thus agree with Roberts (2019: 99–101) that a predetermined inventory of features is necessary to limit cross linguistic variation.

By integrating Chomsky's three factors of language design, Roberts (2019) argues that parameter settings are not necessarily directly predetermined by UG; instead, they are emergent properties of the interaction between UG, experience, and the third factor. Roberts (2019) considers two domain-general optimization strategies as relevant third factor:

(7) a. Feature Economy (FE): Postulate as few formal features as possible.
b. Input Generalization (IG): Maximize available features.

In Roberts's model, FE and IG interact with UG and experience, giving rise to the NO > ALL > SOME learning algorithm (Biberauer & Roberts 2015; Biberauer & Roberts 2017; Roberts 2019), which can be illustrated with a comparison of the ϕ-feature parameter setting procedures by Mohawk-, English-, and Japanese-learning children. These languages represent a continuum concerning the presence of ϕ-features (p. 285).

[19] See Boeckx (2011) and citations therein for discussion of other approaches that may exclude parameters from the narrow syntax.

(8) a. Mohawk has a "rich" inflection system and has fully-specified ϕ-features (Person, Number and Gender) on all functional heads.
b. English is inflectionally "poor"; it only has Person and Number features on some functional heads (i.e., D).
c. Japanese, by contrast, lacks overt ϕ-feature agreement marking entirely, thus is considered as having no ϕ-features in the system.

Given this setup, children learning Mohawk, English, and Japanese are expected to set different values for the ϕ-features:

(9) a. Starting with the ϕ-parameter unspecified, Japanese children detect no ϕ-features in their language, and thus keep the initial setting, satisfying both FE and IG (vacuously).
b. Both Mohawk and English learners observe the presence of ϕ-features (either Person, Number or Gender) in their linguistic input and thus spread the presence of all ϕ-features to (all) the functional heads in the system. In this way, linguistic experience causes IG to override FE.
c. The SOME Stage: English learners may further detect that Gender is not present in their linguistic input, thus restrict ϕ-features in their language to only allow some features: Person and Number.

The ϕ-feature parameter setting procedure can be illustrated as in **Figure 5**.

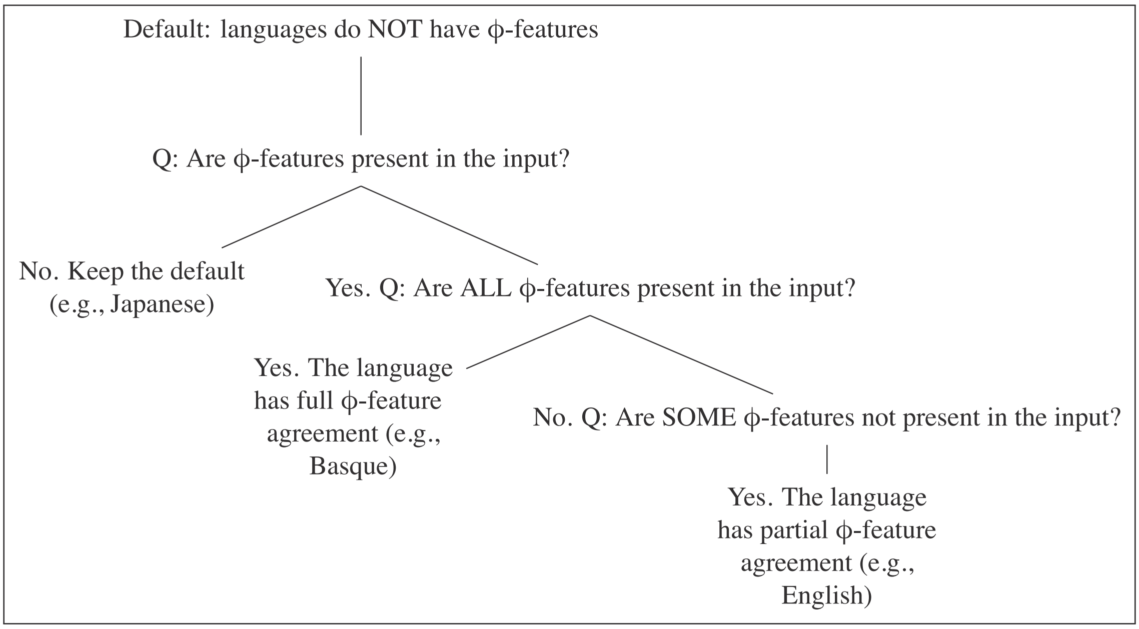

Default: languages do NOT have ϕ-features
Q: Are ϕ-features present in the input?
No. Keep the default (e.g., Japanese)
Yes. Q: Are ALL ϕ-features present in the input?
Yes. The language has full ϕ-feature agreement (e.g., Basque)
No. Q: Are SOME ϕ-features not present in the input?
Yes. The language has partial ϕ-feature agreement (e.g., English)

Figure 5: Roberts' (2019) application of the NO > ALL > SOME learning path.

The NO > ALL > SOME learning path results in a hierarchical taxonomy of parameters. This hierarchy encompasses four levels: macroparameters, mesoparameters, microparameters, and nanoparameters (Biberauer & Roberts 2012).

However, the NO > ALL > SOME learning algorithm highlights that, in Roberts's model, parametric knowledge is not completely removed from UG. The knowledge may not be expressed directly as a list of parameters. Instead, they are represented as features that define the limitation of cross-linguistic variation. For example, in the process of setting the ϕ-features, while it can be assumed that Japanese learners do nothing in the NO stage, and thus do not need knowledge of the sub-ϕ-features (Biberauer 2019: 60). In the ALL > SOME stage, in order to answer the question "Are some ϕ-features not present in the input?", Mohawk- and English-acquiring children must verify each of Person, Number, and Gender individually. This implies that children must have innate knowledge of the full inventory of ϕ-features. After checking the presence or absence of each sub-parameter, Mohawk children would find all ϕ-features present in their language and would stay in the ALL stage, while English children would find that their language only has partial ϕ-features and would move to the SOME stage. Given that a learner cannot predict which language they will be exposed to, this parameter-setting procedure implies that all learners must initially have the inventory of Person, Number, and Gender features in UG, even if their language eventually lacks certain sub-ϕ-features (as in English) or entirely lacks ϕ-features (as in Japanese).

To summarize, a paradox in the NO > ALL > SOME learning algorithm is that setting the ϕ-feature macroparameter requires children to identify specific ϕ-features, i.e., Person, Number, and Gender. This presupposes children's prior knowledge of these features and their association with particular linguistic data in the input. Consequently, the knowledge of specific ϕ-features must be in UG. Therefore, as acknowledged by Roberts (2019) himself, this formalization of parameters under the MP does not entirely remove parametric knowledge from UG. The inventory of features (in the case of the Minimalist approach), like parameters (in the case of the P & P approach), still needs to be stored in UG, although the number of features assumed can be smaller than the number of parameters, as parameters can be derived from features, potentially in interaction with other factors such as the NO > ALL > SOME learning algorithm (see also Longobardi 2018; Biberauer 2019).[20,21]

[20] As a reviewer points out, given that the parametric hierarchies take a very specific "route" from macro to micro/nano parameters, the "ordering" of the questions matters in Roberts' (2019) analysis. This specific ordering requires some specific language-related knowledge. For example, asking a general question about ϕ-features before asking specific questions about Number, Gender, and Person features requires the learner to know that ϕ-features include these three specific features.

[21] It is also relevant to note that Roberts's parameter hierarchy does not intend to reduce the size of possible languages or possible variation. The question this hierarchy addresses is not "can we reduce the number of feature categories with this design?" What it is designed to solve is a different question: Can the learner learn the parameters in a more efficient way?

Crisma et al.'s (2020) model also works under the MP framework and does not assume a list of parameters as part of the initial state S_0 of the mental grammar. This model makes two core assumptions about parameter setting: First, only parameters with a positive value are set; second, parameters can be set exclusively on the basis of a core subset of positive evidence. In other words, parameter setting is viewed as a process of adding parameters with the [+] value to the mental grammar when the relevant manifestation is present in the primary linguistic data (PLD). To show the sufficiency of the positive evidence in the PLD for setting the parameters, Crisma et al. presented a judgment experiment based on 94 real parameters from 69 real languages. It is shown that at least one value of each parameter in the collection can be unambiguously associated with positive evidence available in the PLD. Specifically, each parameter can be associated with a set of YES/NO questions about the occurrence of a set of observable patterns in the following form:

(10) "Does a (set of) structure(s)/interpretation(s) so-and-so occur in language L?"

A "Yes" answer to the question (i.e., positive evidence of the 'structure(s)/interpretations so-and-so' in the PLD) indicates the addition of the [+] value of the parameter to the mental grammar, while a "No" answer to the question (i.e., lack of positive evidence of the 'structure(s)/interpretations so-and-so') indicates no addition of the parameter to the mental grammar. It is further assumed that only the core primary manifestations, which are called the *Restricted List*, are used to set the parameters. Learners can set the parameter by identifying just one manifestation in the Restricted List per parameter, and the nonprimary manifestations will follow from that setting. For example, the Person parameter can have various manifestations corresponding to a "Yes" answer to the following questions:

(11) a. Is there agreement in person between an argument and a verb?
b. Are there overt expletive pronouns in subject function?
c. Are there overt expletive pronouns in object function?
d. Pick a 3rd person pronoun that can occur with no coreferential item in the sentence. Can it also occur as a variable bound by a quantified antecedent like *no-one/everyone*?
e. Are argument proper names introduced by a functional morpheme that is not required in other (non-argument) occurrences?

According to Crisma et al. (2020), only (11a)–(11c) form the *Restricted List*. Therefore, children who encounter one of the three manifestations (corresponding to a "Yes" answer to the questions) in *Restricted List*, such as Mohawk-learning children and English-learning children, will add [+Person] to their mental grammar, while children who never encounter any of the three manifestations (corresponding to a "No" answer to the questions), such as Japanese-learning children, will not add [+Person] to their grammar, and thus set [-Person] as the default option.

However, although Crisma et al. (2020) claim that their model does not commit to the assumption that the list of parameters is part of UG, knowledge of the *manifestations* and *Restricted List* as well as the associations related to the positive value of each parameter entails parametric knowledge. Therefore, like Roberts' NO > ALL > SOME learning algorithm, parametric knowledge is still not completely removed from the system. The only place that can keep this language-specific knowledge is UG.

Similarly, we are sympathetic to the idea that a good number of parameters is needed to connect to the primary linguistic data and account for cross-linguistic variation (cf. Longobardi 2005; Guardiano et al. 2020), although these parameters may be derivable from a smaller set of primitives, such as a smaller set of features (Longobardi 2005; Crisma et al. 2020). That is, on the one hand, UG cannot avoid parametric knowledge completely; on the other hand, it should not be overloaded with too much language-specific knowledge. Following various proposals in the Minimalist framework, in the next subsection we provide a definition of parameters in a broad sense to mitigate this tension. We will argue that a large number of parameters will not necessarily burden UG but may cause a learnability problem, which this paper aims to address.

7.2 Parameters in a broad sense

The parameters that are hard-wired into UG, as in the P & P approach, are considered parameters in a narrow sense in this study. While we agree that UG should be minimized in its language-specific knowledge, we also believe that parametric knowledge is necessary to account for systematic cross-language variation. Additionally, we do not believe that all parameters must be innately set in UG. In this paper, we adopt the following definition of "parameters" in a broader sense:

(12) *Definition of parameters in a broad sense*:
A parameter is systematic cross-linguistic variation that could itself be innate (minimized) or derived from other constraints, including logical options and cognitive constraints.

An example of a parameter in a broad sense is that UG might specify binary Merge without linear order, allowing the order between a head and its complement to be either head-final or head-initial during externalization. Because UG allows only binary Merge, only head-final or head-initial structures are possible in cases where a head merges with its complement, leading to cross-linguistic parametric variation. That is, the head-initial or head-final parameter does not need to be encoded in UG, as it can be derived from Merge of a head and a complement plus logical options. Therefore, the Clustering Approach is not parasitic on the P & P framework and can be applied to model parameter learning in other framework, e.g., the MP. The current approach is compatible with the leading idea proposed in Longobardi (2005) and Biberauer (2019), among many others, in the sense that, by applying third factors such as logical options or cognitive biases, a minimal system can be assumed in the UG to derive a good number of parameters, which could be evaluated directly against linguistic data from various languages (Longobardi 2018).

The Clustering Approach proposed in this paper is also compatible with parameters that arise from the underspecification of rule orderings in syntactic derivation (Obata et al. 2015; Blümel et al. 2021; Sugimoto & Pires 2023), which we consider to be a special instance of logical options applied to the output of Merge in the course of derivation. In addition, the Clustering Approach fits well with proposals that encode parameters or patterned crosslinguistic variation at the externalization/sensorimotor interface/PF interface (Berwick & Chomsky 2011; Sigurðsson 2020). Such variation might arise from underspecification in the narrow syntax (Richards 2008; Boeckx 2011). However, we would like to note that not all forms of language variation qualify as "parameters" (see Yang 2011), even if we do not assume parameters are directly specified by UG. For instance, microparameters that are merely associated with individual lexical items in particular languages, including nanoparameters in Biberauer & Roberts (2017) and Roberts (2019) and superficial microparameters that are associated with particular lexical items defined in Culicover (1999), should in principle not be treated as parameters. Only when several microparameters can be analyzed as connected to a single parameter may they be relevant (see Ayoun 2003 for relevant discussion). We thus restrict ourselves to systematic cross-language syntactic variations that are not language-specific.

A comparison between our assumption of the initial state of the learner to Crisma et al.'s (2020) strong emergentist approach will be helpful here. Crisma et al.'s (2020) approach needs to postulate three sets of knowledge: (i) a UG that includes all possible parameters for human languages, with their values underspecified, (ii) the possible manifestations of each parameter, and empirical questions targeting the manifestations, which are further specified as *p*-expressions, the core manifestations for parameter setting, and then Restricted List, the primary *p*-expressions that compose the "core primary evidence for specific parameters," and (iii) a mental grammar that includes only parameters whose values are set due to positive evidence primarily from the Restricted List. By contrast, the Clustering Approach assumes that all parameters, regardless of their origin or derivation method, are part of the grammar pool that the learner's language parser uses to analyze input sentences. The grammar pool is not necessarily UG. Instead, it is a set of possible human grammars determined by UG and other logical and cognitive constraints.

8 Conclusions

In this paper, we have briefly reviewed some representative approaches to parameter setting for child language acquisition and made a contrast between what we have termed the grammar selection approach and the direct parameter setting approach. We argue that parameters are not eliminated from UG under several representative formalizations of parameter setting in the MP, contrary to the general assumption in the field. However, we highlighted a valuable insight that the Minimalist approach has underscored: Parameters are assumed to be learned directly and individually. This conception of parameter setting has roots in the P & P approach.

Regarding the computational implementations of the direct parameter setting approach, we have illuminated both its strengths and weaknesses. A significant merit of this approach is that it does not assume a vast hypothesis space. Instead, it tracks a single grammar, comprising a set of parameters that are set directly based on relevant structural information extracted from the primary linguistic data. This method sidesteps the intricate challenge of navigating through an immense hypothesis space in search of the target grammar. Yet, its primary drawback is the necessity for identifying relevant parametric treelets as well as ambiguous and unambiguous triggers during online parsing.

By contrast, a critical alternative to the direct parameter setting approach, namely, the grammar selection approach, assumes that parameter setting is instead a grammar selection process. Since the grammar selection approach assumes that the parser adopts the grammar that is sampled from the hypothesis space and applies it in the analysis of input sentences, it no longer needs to initiate immense searches to find appropriate parameters/parametric treelets for the analysis of the given input sentences.

However, the grammar selection approach encounters a serious challenge: It requires a significant number of input sentences to successfully navigate a vast hypothesis space. Finally, it presupposes close tracking of a potentially large set of possible grammars, comparing them for evaluation. These assumptions raise critical empirical questions to this approach: Do children receive enough input to support such navigation? And do they possess the computational resources to monitor all potential grammars in the hypothesis space? Interestingly, it is observed that the direct parameter setting approach addresses these issues: Navigating through all possible grammars in the hypothesis space becomes unnecessary when parameters are set directly.

Reflecting on the advantages and disadvantages of the two competing approaches to parameter setting introduces a dilemma. However, a significant observation of this paper is that these two approaches seem to complement each other in their major aspects. Consequently, we propose a hybrid method, the Clustering Approach, that integrates the grammar selection and direct parameter setting approaches. On one side, the Clustering Approach envisions a pool of possible grammars without necessitating the tracking and evaluation of each; on the flip side, its algorithm sets parameter probabilities directly and individually, eliminating the need for extensive online searches, thanks to the grammar sampling from the pool during parsing. The Clustering Approach accurately updates the parameters used in the input sentences, and thus allows some parameters that are frequently and consistently observed in the input be set first and thus serve as clustering criteria. The clustering of parameters tracks the learning order of parameters and helps the algorithm to gradually reduce the size of the grammar pool. Accurate updating and clustering of the grammar pool in the Clustering Approach distinguish it from its alternative, the NPL. Based on the simulation results for the Clustering Approach and NPL, we contend that the NPL instead predicts the learner would gain minimal knowledge about the target grammar during the initial phase of exploration unless the target grammar is accidentally identified. Hence, the Clustering Approach offers a more realistic, input-driven model for the learning of parametric variation, paving the way for modeling child language acquisition with real linguistic data.

Supplementary files

Supplementary file: Appendix. Python scripts for the generation of the corpora for simulations in this article.

https://osf.io/qwef3/?view_only=a03ca2b8709c4dfeaa8aa1574effc457

Acknowledgements

The authors would like to thank Charles Yang, William Sakas, and Acrisio Pires for their insightful comments on various aspects of this project. We also thank the audience at the 48th Boston University Conference on Language Development (BUCLD) for their helpful questions and comments. We greatly appreciate the helpful comments provided by the anonymous reviewers and the handling editor, Lyn Tieu. Any remaining errors are our own. The simulations in this paper were conducted using computational resources provided by MSU's High-Performance Computing Center.

Competing interests

The authors have no competing interests to declare.

References

Ayoun, Dalila. 2003. *Parameter setting in language acquisition*. New York, NY: Continuum.

Baker, Mark. 2021. On Chomsky's legacy in the study of linguistic diversity. In *A companion to Chomsky*, chap. 10, 158–171. Wiley-Blackwell. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119598732.ch10.

Baker, Mark C. 2008a. *The atoms of language: The mind's hidden rules of grammar*. Basic books.

Baker, Mark C. 2008b. The macroparameter in a microparametric world. In Biberauer, Theresa (ed.), *The limits of syntactic variation*, 351–373. Amsterdam: Benjamins. DOI: https://doi.org/10.1075/la.132.16bak

Berwick, Robert C. & Chomsky, Noam. 2011. The biolinguistic program: The current state of its development. In Di Sciullo, Anna Maria & Boeckx, Cedric (eds.), *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty*, 19–41. Cambridge, MA: Oxford University Press.

Berwick, Robert C. & Chomsky, Noam. 2016. *Why only us: Language and evolution*. Cambridge, MA: The MIT Press. DOI: https://doi.org/10.7551/mitpress/9780262034241.001.0001

Biberauer, Theresa. 2019. Factors 2 and 3: Towards a principled approach. *Catalan Journal of Linguistics: Special issue* 45–88. DOI: https://doi.org/10.5565/rev/catjl.219

Biberauer, Theresa & Holmberg, Anders & Roberts, Ian & Sheehan, Michelle. 2014. Complexity in comparative syntax: The view from modern parametric theory. In Newmeyer, Frederick & Preston, Laurel (eds.), *Measuring grammatical complexity*, 103–127. Oxford: Oxford University Press. DOI: https://doi.org/10.1093/acprof:oso/9780199685301.003.0006

Biberauer, Theresa & Roberts, Ian. 2012. Towards a parameter hierarchy for auxiliaries: Diachronic considerations. In Chancharu, James N. & Hu, Xuhui Freddy & Mitrovic, Moreno (eds.), *Cambridge Occasional Papers in Linguistics*, vol. 6, 267–294.

Biberauer, Theresa & Roberts, Ian. 2015. Rethinking formal hierarchies: A proposed unification. *Cambridge Occasional Papers in Linguistics* 7(1). 1–31.

Biberauer, Theresa & Roberts, Ian G. 2017. Parameter setting. In Ledgeway, Adam & Roberts, Ian G. (eds.), *The Cambridge handbook of historical syntax*, 134–162. Cambridge: Cambridge University Press. DOI: https://doi.org/10.1017/9781107279070.008

Blümel, Andreas & Goto, Nobu & Sugimoto, Yushi. 2021. When the grammar doesn't mind which merge it chooses. In *The 39th meeting of the west coast conference on formal linguistics*. https://arizona.figshare.com/articles/presentation/Oral_Presentation_for_When_the_grammar_doesn_t_mind_which_Merge_it_chooses_/14481798.

Boeckx, Cedric. 2011. Approaching parameters from below. In Di Sciullo, Anna Maria & Boeckx, Cedric (eds.), *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty*, 205–221. Cambridge, MA: Oxford University Press.

Boeckx, Cedric & Leivada, Evelina. 2014. On the particulars of universal grammar: Implications for acquisition. *Language Sciences* 46. 189–198. DOI: https://doi.org/10.1016/j.langsci.2014.03.004

Borer, Hagit. 1984. *Parametric syntax: Case studies in Semitic and Romance languages*. Dordrecht: Foris. DOI: https://doi.org/10.1515/9783110808506

Bush, Robert R. & Mosteller, Frederick. 1951. A mathematical model for simple learning. *Psychological Review* 58(5). 313. DOI: https://doi.org/10.1037/h0054388

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: The MIT Press. DOI: https://doi.org/10.21236/AD0616323

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht, Netherlands: Foris.

Chomsky, Noam. 1986. *Knowledge of language: Its nature, origin, and use*. New York: Praeger Publishers.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, MA: The MIT Press.

Chomsky, Noam. 2001. Derivation by phase. In Kenstowicz, Michael (ed.), *Ken Hale: A life in language*, 1–52. Cambridge, MA: MIT Press. DOI: https://doi.org/10.7551/mitpress/4056.003.0004

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36(1). 1–22. DOI: https://doi.org/10.1162/0024389052993655

Chomsky, Noam. 2017. Language architecture and its import for evolution. *Neuroscience & Biobehavioral Reviews* 81. 295–300. DOI: https://doi.org/10.1016/j.neubiorev.2017.01.053

Chomsky, Noam & Gallego, Ángel J. & Ott, Dennis. 2019. Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics* 229–261.

Chomsky, Noam & Lasnik, Howard. 1993. The theory of principles and parameters. In Jacobs, Joachim & von Stechow, Arnim & Sternefeld, Wolfgang & Vennemann, Theo (eds.), *Syntax: An international handbook of contemporary research*, vol. 1, 506–569. Berlin, Germany: de Gruyter. DOI: https://doi.org/10.1515/9783110095869.1.9.506

Clark, Robin. 1989. On the relationship between the input data and parameter setting. In *The proceedings of the North East Linguistics Society*, vol. 19, 48–62. University of Massachusetts, Amherst: GLSA.

Clark, Robin. 1992. The selection of syntactic knowledge. *Language Acquisition* 2(2). 83–149. DOI: https://doi.org/10.1207/s15327817la0202_1

Clark, Robin. 1994. Finitude, boundedness, and complexity: Learnability and the study of first language acquisition. In *Syntactic theory and first language acquisition: Cross-linguistic perspectives*, 473–489. Psychology Press. DOI: https://doi.org/10.4324/9781315789200-22

Clark, Robin & Roberts, Ian. 1993. A computational model of language learnability and language change. *Linguistic Inquiry* 24(2). 299–345.

Crisma, Paola & Guardiano, Cristina & Longobardi, Giuseppe. 2020. Syntactic diversity and language learnability. *Studi e saggi linguistici* 58(2). 99–130.

Culicover, Peter W. 1999. *Syntactic nuts: Hard cases, syntactic theory, and language acquisition*. New York, NY: Oxford University Press. DOI: https://doi.org/10.1093/oso/9780198700241.001.0001

Dresher, Bezalel Elan. 1999. Charting the learning path: Cues to parameter setting. *Linguistic Inquiry* 30(1). 27–67. DOI: https://doi.org/10.1162/002438999553959

Fodor, Janet Dean. 1998a. Learning to parse? *Journal of Psycholinguistic Research* 27(2). 285–319. DOI: https://doi.org/10.1023/A:1023258301588

Fodor, Janet Dean. 1998b. Parsing to learn. *Journal of Psycholinguistic Research* 27(3). 339–374. DOI: https://doi.org/10.1023/A:1023255705029

Fodor, Janet Dean. 2017. Ambiguity, parsing, and the evaluation measure. *Language Acquisition* 24(2). 85–99. DOI: https://doi.org/10.1080/10489223.2016.1270948

Fodor, Janet Dean & Sakas, William Gregory. 2004. Evaluating models of parameter setting. In Brugos, Alejna & Micciulla, Linnea & Smith, Christine E. (eds.), *The proceedings of the 28th Annual Boston University Conference on Language Development*, vol. 1, 1–27. Somerville, MA: Cascadilla Press.

Gibson, Edward & Wexler, Kenneth. 1994. Triggers. *Linguistic Inquiry* 25(3). 407–454.

Guardiano, Cristina & Longobardi, Giuseppe & Cordoni, Guido & Crisma, Paola. 2020. Formal syntax as a phylogenetic method. In Janda, Richard D. & Joseph, Brian D. & Vance, Barbara S. (eds.), *The handbook of historical linguistics*, vol. 2, 145–182. Wiley Blackwell. DOI: https://doi.org/10.1002/9781118732168.ch7

Hauser, Marc D. & Chomsky, Noam & Fitch, W. Tecumseh. 2002. The faculty of language: what is it, who has it, and how did it evolve? *Science* 298(5598). 1569–1579. DOI: https://doi.org/10.1126/science.298.5598.1569

Howitt, Katherine & Dey, Soumik & Sakas, William Gregory. 2021. Gradual syntactic triggering: The gradient parameter hypothesis. *Language Acquisition* 28(1). 65–96. DOI: https://doi.org/10.1080/10489223.2020.1803329

Ke, Alan Hezao. 2024. Can agree and labeling be reduced to minimal search? *Linguistic Inquiry* 55(4). 849–870. DOI: https://doi.org/10.1162/ling_a_00481

Ke, Alan Hezao & Gao, Liqun. 2020. Domain restriction in child mandarin: Implications for quantifier spreading. *Linguistics* 58(6). 1839–1875. DOI: https://doi.org/10.1515/ling-2020-0246

Kohl, Karen Thompson. 1999. *An analysis of finite parameter learning in linguistic spaces*: Massachusetts Institute of Technology master's thesis.

Legate, Julie Anne & Yang, Charles. 2007. Morphosyntactic learning and the development of tense. *Language Acquisition* 14(3). 315–344. DOI: https://doi.org/10.1080/10489220701471081

Lidz, Jeffrey & Gagliardi, Annie. 2015. How nature meets nurture: Universal grammar and statistical learning. *Annual Review of Linguistics* 1(1). 333–353. DOI: https://doi.org/http://www.annualreviews.org/doi/abs/10.1146/annurev-linguist-030514-125236

Lightfoot, David. 1989. The child's trigger experience: Degree-0 learnability. *Behavioral and Brain Sciences* 12(2). 321–334. DOI: https://doi.org/10.1017/S0140525X00048883

Lightfoot, David W. 2020. *Born to parse: How children select their languages*. Cambridge, MA: MIT Press. DOI: https://doi.org/10.7551/mitpress/12799.001.0001

Longobardi, Giuseppe. 2005. A minimalist program for parametric linguistics. In Broekhuis, Hans & Corver, Norbert & Huybregts, Riny & Kleinhenz, Ursula & Koster, Jan (eds.), *Organizing grammar: Linguistic studies for Henk van Riemsdijk*, 407–414. Berlin, Germany: Mouton de Gruyter. DOI: https://doi.org/10.1515/9783110892994.407

Longobardi, Giuseppe. 2018. Principles, parameters, and schemata: A radically underspecified UG. *Linguistic Analysis* 517–558.

Nazarov, Aleksei & Jarosz, Gaja. 2021. The credit problem in parametric stress: A probabilistic approach. *Glossa* 6(1). 1–26. DOI: https://doi.org/10.16995/glossa.5884

Obata, Miki & Epstein, Samuel & Baptista, Marlyse. 2015. Can crosslinguistically variant grammars be formally identical? Third factor underspecification and the possible elimination of parameters of UG. *Lingua* 156. 1–16. DOI: https://doi.org/10.1016/j.lingua.2014.12.003

Pearl, Lisa S. 2011. When unbiased probabilistic learning is not enough: Acquiring a parametric system of metrical phonology. *Language Acquisition* 18(2). 87–120. DOI: https://doi.org/10.1080/10489223.2011.554261

Pearl, Lisa S. 2021. How statistical learning can play well with universal grammar. In Allott, Nicholas & Lohndal, Terje & Rey, Georges (eds.), *A companion to Chomsky*, 267–286. Wiley-Blackwell. DOI: https://doi.org/10.1002/9781119598732.ch17

Pinker, Steven & Jackendoff, Ray. 2005. The faculty of language: What's special about it? *Cognition* 95(2). 201–236. DOI: https://doi.org/10.1016/j.cognition.2004.08.004

Richards, Marc. 2008. Two kinds of variation in a minimalist system. In Heck, Fabian & Müller, Gereon & Trommer, Jochen (eds.), *Varieties of competition*, 133–162. Universität Leipzig: Linguistische Arbeits Berichte 87.

Rizzi, Luigi. 2008. On the grammatical basis of language development. In Cinque, Guglielmo & Kayne, Richard (eds.), *The Oxford handbook of comparative syntax*, 70–109. New York: Oxford University Press.

Roberts, Ian. 2019. *Parameter hierarchies and universal grammar*. Oxford University Press. DOI: https://doi.org/10.1093/oso/9780198804635.001.0001

Roeper, Thomas & Weissenborn, Jürgen. 1990. How to make parameters work: Comments on Valian. In *Language processing and language acquisition*, 147–162. Springer. DOI: https://doi.org/10.1007/978-94-011-3808-6_6

Sakas, William. 2016. Computational approaches to parameter setting in generative linguistics. In Lidz, Jeffrey & Snyder, William & Pater, Joe (eds.), *The Oxford handbook of developmental linguistics*, 696–724. Oxford, UK: Oxford University Press. DOI: https://doi.org/10.1093/oxfordhb/9780199601264.013.29

Sakas, William Gregory & Fodor, Janet Dean. 2012. Disambiguating syntactic triggers. *Language Acquisition* 19(2). 83–143. DOI: https://doi.org/10.1080/10489223.2012.660553

Sakas, William Gregory & Yang, Charles & Berwick, Robert C. 2017. Parameter setting is feasible. *Linguistic Analysis* 41. 391–408.

Sheehan, Michelle. 2021. Parameters and linguistic variation. In *A companion to Chomsky*, chap. 11, 158–171. Wiley-Blackwell. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119598732.ch11.

Sigurðsson, Halldór Ármann. 2020. Universality and variation in language: The fundamental issues. *Evolutionary Linguistic Theory* 2(1). 5–29. DOI: https://doi.org/10.1075/elt.00013.sir

Straus, Kenneth Jerold. 2008. *Validation of a probabilistic model of language acquisition in children*: Northeastern University Boston doctor's dissertation.

Sugimoto, Yushi & Pires, Acrisio. 2023. A parameter-free underspecification approach to complementizer agreement. *Revista Linguistica* 18(1). 62–81. DOI: https://doi.org/10.31513/linguistica.2022.v18n1a56350

Thornber, Molly & Ke, Alan Hezao. 2024. Modeling the learning of syntactic parameters from parsed data. In AlThagafi, Hayat & Ray, Jupitara (eds.), *The proceedings of the 48th annual Boston University Conference on Language Development (BUCLD 48)*, Cascadilla Press. https://www.lingref.com/bucld/48/BUCLD48-47.pdf.

Tsimpli, Ianthi Maria. 2014. Early, late or very late?: Timing acquisition and bilingualism. *Linguistic Approaches to Bilingualism* 4(3). 283–313. DOI: https://doi.org/10.1075/lab.4.3.01tsi

Wexler, Ken. 1998. Very early parameter setting and the unique checking constraint: A new explanation of the optional infinitive stage. *Lingua* 106(1–4). 23–79. DOI: https://doi.org/10.1016/S0024-3841(98)00029-1

Yang, Charles. 2011. Three factors in language variation. In Di Sciullo, Anna Maria & Boeckx, Cedric (eds.), *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty*, 205–221. Cambridge, MA: Oxford University Press.

Yang, Charles. 2012. Computational models of syntactic acquisition. *Wiley Interdisciplinary Reviews: Cognitive Science* 3(2). 205–213. DOI: https://doi.org/10.1002/wcs.1154

Yang, Charles D. 2002. *Knowledge and learning in natural language*. New York, NY: Oxford University Press.

Zipf, George Kingsley. 1949. *Human behavior and the principle of least effort*. Cambridge, MA: Addison-Wesley Press.