Sub-regular inductive biases in a phonological transformer

Micha Elsner & Donald S. Black
The Ohio State University, Independent Researcher

Transformers and related neural networks have shown significant promise as cognitive models. On linguistic tasks, they show an impressive capacity to learn syntactic (Millière, 2024) and morphological (Wu et al., 2021) phenomena, and their internal representations can be used to predict brain activity (Schrimpf et al., 2021). Neural learners have also been used to develop lexical representations (Sanabria et al., 2023) and phonological features (Beguš, 2020; Shain and Elsner, 2019) directly from acoustic data. This suggests a pathway toward joint modeling of phonetic and phonological acquisition and shows how quasi-symbolic representations may be synthesized from continuous (e.g., phonetic) data.

Neural networks are also capable learners of phonotactics (Muradoglu and Hulden, 2023; Mayer and Nelson, 2020; Mirea and Bicknell, 2019) and alternation phenomena like reduplication (Prickett et al., 2022). It therefore seems attractive to propose a model of language learning in which a single (perhaps transformer-like) mechanism acquires language across multiple levels of representation. Such a model has the advantage of parsimony and tallies with recent findings that the human brain's language network lacks specialized components for separate processing of syntax and phonology (Fedorenko et al., 2024; Regev et al., 2024).

Although neural learners do not yet completely mimic human acquisition (Kodner et al., 2023; Yedetore et al., 2023), insights that motivate the transformer's design could plausibly shed light on computational mechanisms of human language learning. Among these may be their use of distributed representations (Piantadosi et al., 2024) and attention as a form of latent structure building (Henderson, 2020).

The power of the transformer represents a potential problem for this model — if phonology and syntax are learned by the same mechanism, why does phonology appear formally so much simpler?[1] If a transformer or similar network is to serve as a model of phonological acquisition, it is necessary to show that such a network can be configured to exhibit an inductive bias toward the formal properties of phonological rather than syntactic phenomena.

A promising way to configure the inductive biases of a network is to pretrain it with synthetic data from a particular formal class (McCoy and Griffiths, 2025; Papadimitriou and Jurafsky, 2023). The Simulation-Induced Prior (SIP) framework of Lindemann et al. (2024), in which a pretraining step imposes a bias towards regular languages, is applicable to this task. We extend it in this work. Because a long tradition (Graf, 2022; Heinz, 2011) locates phonological phenomena within *sub*-regular classes of formal languages, we use the SIP mechanism to create models with subregular biases, which we hypothesize may be better suited to learning phonology.

We find that SIP training with different language classes is effective at creating transformer models with measurably different inductive biases. In general, however, the biases observed are not wholly due to SIP; even without this training, transformers exhibit a phonologically relevant preference for local rules over long-distance rules. Pretraining matters most for learning an unbounded process, where a model whose pretraining included an explicit notion of tiers succeeds faster than those with generic regular language

[1] Haley and Wilson (2021) shows empirically that LSTMs easily learn unnatural phonological patterns. In fact, Kallini et al. (2024) claim that LLMs can learn languages which are beyond the bounds of even syntactic structure, although these may be harder to learn than more natural ones; see Yang et al. (2025); Hunter (2025) for responses.

Proceedings of *AMP 2023/2024*

or local language pretraining. We also find a strong and linguistically implausible asymmetry between progressive and regressive conditioning.

Overall, these results are consistent with other work, which shows that the inherent biases of the transformer architecture are useful for language learning. Moreover, we show that further cognitively useful biases can be instilled in particular configurations of the transformer. These biases are not hard limits on the class of learnable languages but might still operate to shape observed typology. The progressive/regressive asymmetry, on the other hand, illustrates the limits of sequential symbolic processing as a metaphor for the complex time-varying processes underlying human speech production.

1 Evidence for locality biases in phonology

Almost all known phonological processes belong to the class of regular languages, and, in fact, to more restrictive classes (Graf, 2022; Heinz, 2010). 94% of the processes in P-base (Mielke, 2008) fall within the extremely restrictive class of k-Input Strictly Local (ISL) languages, which view only a k-length window of the input at a time (Chandlee et al., 2014:p138). The remaining processes, such as vowel and consonant harmony, are generally considered 'long-distance' phonology. But even these do not require the same formal machinery as syntax. Heinz (2010) shows that some harmony rules can be represented by precedence grammars that track subsequences of the string, but a more commonly studied sub-regular class is the Tier Strictly Local (TSL) languages, which generalize the ISL class by projecting some but not all input characters onto a tier and applying the local constraints there (Burness et al., 2021; McMullin and Hansson, 2014). TSL grammars capture a rich set of long-distance phenomena, including blocking (Chandlee et al., 2018).

A skeptic might claim that this typological evidence tells us nothing about phonological learning because the link between learning and typology is too weak. Perhaps all sorts of languages (including unattested ones) are equally learnable, and typological generalizations reflect only the sorts of patterns which historical change tends to create (Moreton and Pater, 2012; Morley, 2015; Harris, 2008). This may be a useful explanation for generalizations about sound classes or featural values for which participants in artificial grammar learning experiments seem relatively indifferent to whether they are learning a 'marked' grammar or a 'natural' one (Moreton and Pater, 2012). But it seems less satisfactory for claims about locality. While rules affecting unnatural classes are merely rare, the generalization that phonology is local appears to be much more robust. Secondly, evidence from artificial grammar learning tasks shows that human learners prefer to generalize processes so that they apply as locally as possible (Finley, 2017, 2011). Finally, although weighing up the potential impacts of a historical change is difficult (Morley, 2015), it does not seem impossible to us that language change could offer the learner at least ambiguous evidence for unattested long-distance phonological effects.[2]

If we conclude that a learning bias for local phonology is real, we must next ask what kind of learning mechanism is responsible. ISL languages are provably learnable by a deterministic algorithm (Chandlee et al., 2014)— perhaps this is what human learners are using. Apart from the question of how this mechanism might be implemented in neural tissue, the biggest problem with such a proposal is how to handle cases which require a larger class of languages. These might arise from interactions between individually k-ISL processes, which are not guaranteed to be ISL or retain a window size of k (Chandlee et al., 2018), or from inherently long-distance processes like harmony. Thus, a deterministic learner only capable of learning k-ISL languages seemingly requires a fallback plan.

One set of proposals to address these problems claims that such a fallback is not really necessary. In these theories, many supposedly 'long-distance' processes are in fact strictly local ones (Gafos, 2014; Ní Chiosáin and Padgett, 2001) — harmony does not leave intermediate segments untouched, but spreads the feature value onto them. For instance, backness harmony in Turkish leaves non-contrastive [+back] diacritics on the intervening segments (Ní Chiosáin and Padgett, 2001:p30). There is evidence for this kind of coarticulatory effect, in which the realization of intervening 'transparent' segments are affected by harmony triggers, but it is unclear that these effects are strong enough to learn harmony by tracking purely local statistics (Blaho and Szeredi, 2013). Finley (2021) finds that the phonetic realization of a transparent vowel makes relatively little

[2] For example, a CV language with vowel harmony could gain consonant clusters by vowel reduction. Speakers would need to decide whether harmony can skip any number of consonants (an attested system) or skips one consonant but is blocked by two (an unattested one) (McMullin and Hansson, 2014); without a learning bias, it is unclear why the second system cannot be the result.

difference to the learnability of an artificial harmony system.

Another proposal is to infer the appropriate subregular class of a particular grammar (e.g., ISL or TSL) from the data. Belth (2024) proposes a preference hierarchy, in which 2-ISL grammars are simpler than 2-TSL grammars, and searches for a satisfactory grammar in that order. If his learner cannot find an ISL grammar with few enough exceptions to satisfy the Tolerance Principle (Yang, 2016), it tries again using a tiered representation. This kind of hierarchy accounts for the learnability of non-2-ISL patterns in phonology, but extending it to cover more cases can be difficult. The strict ordering requires a decision about (e.g.) whether a 2-TSL grammar is simpler than a 3-ISL grammar— and this decision must be made *a priori*, not on the basis of how well each one accounts for the data.

An alternate cognitive modeling paradigm is to express preferences for one class of grammars over another as defeasible prior biases. Learners would then balance a grammar's preferability versus its goodness of fit. Biases of this type are often expressed as Bayesian priors (Goldwater and Johnson, 2004), but the inductive biases of neural networks can also be described in terms of defeasible biases (Griffiths et al., 2024, 2012).

Unlike learners with stages that search within a specific langauge class, a learner with a defeasible bias towards locality is not limited to searching within a specific class of subregular grammars. Such a unified approach is cognitively preferable to one involving sharp discontinuities. For example, the statistics it tracks need not be discarded each time a smaller subregular class is rejected and the search expands to a larger one. But searching a larger hypothesis space presents challenges; Hayes and Wilson (2008) find that unconstrained search through a space of finite-space rules cannot discover a vowel harmony rule. The effectiveness of the transformer at various learning tasks suggests that it may offer an efficient solution to this kind of search problem. We discuss this hypothesis in more detail in the following section.

2 The inductive biases of transformers

In purely formal terms, transformer networks themselves search a limited space of grammars. Hahn (2020) proves that even some finite-state languages are beyond the capacity of fixed-dimensional self-attention networks (although Strobl et al. (2025) demonstrates that they are capable of representing all star-free regular transductions, a class which includes the ISL and TSL classes discussed here (Heinz et al., 2011)). In practice, however, they are capable of learning even context-free languages if either the length or embedding depth of the inputs is bounded (Bhattamishra et al., 2020; Yao et al., 2021).

Within the set of learnable languages, networks have biases which push them strongly toward simpler functions. Valle Pérez et al. (2019) show that feed-forward networks have a bias toward expressing functions of low Kolmogorov complexity (operationalized in their experiments by Lempel-Ziv compressibility); most random initializations of the network express such functions. Bhattamishra et al. (2023) find that randomly initialized transformers tend to express functions with low boolean sensitivity, (the number of pairs of binary input strings which differ by a single bit but have different labels). Moreover, on k-sparse functions whose output is affected by only k bits of the input, transformers learn faster and are more robust to noise than LSTMs. k-sparsity is not a locality property, since it does not specify where in the input the k critical bits are located, but it does at least represent a preference for small factors which may predispose the transformer toward phonologically useful representations.

The speed and sample efficiency with which transformers learn can be further improved by suitable pretraining. The use of synthetic pretraining data has been motivated on applied grounds in order to teach models to copy and/or denoise their input Bergmanis et al. (2017). More recently, synthetic pretraining has been used to instill inductive biases into transformers in order to test cognitive modeling hypotheses (see Griffiths et al. (2024) for an overview of this research program).

Papadimitriou and Jurafsky (2023) and McCoy and Griffiths (2025) both show that networks trained on structured artificial languages learn the syntax of human languages more rapidly when fine-tuned on small datasets. McCoy et al. (2020) studies the phonological problem of syllabification. Their artificial languages are generated by randomly ranking four Optimality Theoretic constraints. They show that the resulting model has a preference for languages with a consistent constraint ranking and learns such languages faster than an unbiased model. Lindemann et al. (2024) learn a bias towards regular languages, which they apply to a grapheme-to-phoneme task but without detailed linguistic analysis.

Although all these projects involve synthetic pretraining, there are methodological differences in the

training procedure. Papadimitriou and Jurafsky (2023) simply generate the synthetic data and train on it. McCoy and Griffiths (2025) and McCoy et al. (2020) use MAML (Finn et al., 2017) as a meta-learning strategy. In MAML training, a sub-model is used to learn each new synthetic language, and the base model is then tweaked to provide a better initializer which would have enabled that language to be learned faster.

Lindemann et al. (2024) propose a different method, which they call SIP (Simulation Induced Prior). During SIP training, the system is given a mechanistic description of the finite-state transducer (FST) that generated a particular input-output pair. A training input therefore has the format:

$$\underbrace{h_1, h_2, \ldots, h_k}_{\text{FST encoding}}, \underbrace{x_1, x_2, \ldots, x_n}_{\text{Input to FST}}$$

The h_i are embeddings of the transitions of the generating FST F, and the x_i are the characters of its input. The prediction target is the string $y_i \ldots y_n = F(x)$. Lindemann et al. (2024) show that SIP training can induce a robust bias toward regular languages, while being computationally more efficient than MAML. Moreover, the SIP transformer appears to mechanistically simulate an FST, since a trained probe can extract the state sequence from its internal representations.

After pretraining, the SIP transformer can be fine-tuned on new processes for which no FST transition table is available. To do so, the FST embedding h is replaced with a series of tunable embeddings h', and then both the transformer and the h' embeddings are fine-tuned— a procedure similar to prefix tuning. The learning rate for h' is higher than for the transformer parameters, encouraging the system to make larger changes to the prefix. Loosely speaking, fine-tuning h' searches for the embeddings of a finite-state transducer that might have generated the fine-tuning inputs. But there is an important difference: there is no requirement that h' actually represent such an FST! This means that SIP can learn processes for which the transition table could not fit within the allocated k-element prefix, or non-regular transductions for which no FST exists. This is exactly the sort of defeasible bias we hypothesize for phonology.

3 Models

We investigate the learnability of synthetic and natural language datasets by various transformer models trained with SIP. We begin with **ByT5** (Xue et al., 2022), a multilingual transformer encoder-decoder trained on a byte-level tokenization of a large multilingual dataset. Lindemann et al. (2024) perform SIP fine-tuning on ByT5 to produce a model we call **SIP-FST**. SIP-FST is trained on transition tables and input-output pairs from 50,000 random finite-state transducers with up to 4 states; the sampling process is biased to favor identity transductions for many characters and has some other tunable parameters described in the paper.

We create a **SIP-ISL** model by changing the training distribution. Rather than sampling the state transitions directly (which would not necessarily yield an ISL transducer), we sample up to 4 string replacement patterns drawn from the same distribution of character alphabets as SIP. Each pattern contains up to two input characters and up to two replacements, $ab \rightarrow cd$. Our sampling procedure ensures that 25% of these patterns have $a = c$ (the rule is progressive, with a fixed context on the left and a changing target on the right). 25% of the patterns have $b = d$ (regressive rules). 5% of the rules have either $a = \epsilon$ or $c = \epsilon$ (the rule epenthesizes a segment). Otherwise, characters within patterns are sampled uniformly at random from the alphabet.

We apply the algorithm described in Chandlee et al. (2014)[\$5.1] to induce a 2-ISL transducer from the sampled patterns; if the language is not 2-ISL, we skip it. We continue until 50,000 random languages have been sampled. We represent 2-ISL transducers by minimizing them to yield what Chandlee et al. (2014) calls the *canonical* representation.

We also create a **SIP-TSL** model. In this model, we sample a character alphabet for the transducer in the same way as SIP, but then pick 1-3 random characters from that alphabet to form a tier T. We sample replacement patterns and create an ISL transducer operating only on that tier. We then add self-loops to each state of the transducer which read and copy each other character in the alphabet. When sampling input/output pairs for TSL languages, we ensure that the inputs match the regular expression $\Sigma^* T \Sigma^* T \Sigma^*$, that is, at least two characters in T appear somewhere within the input.

We embed transition tables in the same way as in baseline SIP except that 2-ISL/TSL machines must be

allowed to output two characters from a single transition.[3] Thus, we embed transitions as tuples s, i, o_1, o_2, d: current state s, input character i, outputs o_1, o_2 and destination state d, using a null character when o_2 is absent.

At test time, SIP models are applied to new transductions via the process described in Lindemann et al. (2024:§4.2). Briefly, the "FST embedding" part of the input is initialized at random, and then the entire network, including this random prefix, is fine-tuned on the task examples. The learning rate for the prefix is set at a higher rate, to encourage prefix-tuning-like behavior.[4]

4 Experiments

We conduct experiments on synthetic and natural transductions with each of the four models. In general, the models' tendency is to copy their inputs; this is especially true for SIP models, since the transducers they are trained on are biased toward copying. This means that they may achieve a deceptively high accuracy on tasks where the rule to be applied is very specific, and that accuracy cannot be compared across tasks with different rates of rule application. To avoid this issue, we report our main results in terms of *informedness*, defined as the average of the rate of correct rule application and correct non-application (Powers, 2011). A classifier that always copies and never applies the rule will always score 0 (0 true application rate + 1 true non-application rate - 1). Over triples of input x, target y, prediction $\hat{y}$:

$$I = \frac{\#[y = \hat{y} \neq x]}{\#[y \neq x]} + \frac{\#[y = \hat{y} = x]}{\#[y = x]} - 1 \tag{1}$$

4.1 *Synthetic transductions* We create synthetic datasets by beginning with a set of strings, then applying deterministic artificial rewrite rules to them. The rules are phonetically unmotivated, so that they are not affected by any knowledge of natural language learned during ByT5's training process. However, we use natural language words as the input strings so that we have a naturalistic distribution of segments.

As our vocabulary, we sample Spanish words from the set of inflected forms in Unimorph (McCarthy et al., 2020). Not every word in the vocabulary offers an opportunity to learn something about a particular rule; for example, a word without a d character offers no evidence for a rule $d \rightarrow D$. For each group of synthetic languages, we define a set of critical environments which must be represented when sampling instances. These critical environments cannot be defined as simply the domain of application of the rule and its converse— in this case, it might be difficult to tell whether the model over- or under-generalizes the rule. For example, consider the pair of target rules $d \rightarrow D / a \_ a$ (intervocalic) and $d \rightarrow D / a \_$ (progressive). , If the dataset does not provide evidence for the case adX, these two rules cannot be distinguished, so either might be considered a correct learning outcome in theories that allow different degrees of generalization beyond the input. To avoid this sort of ambiguity, we ensure that all triggering environments are equally represented in training and test sets.

We test the transductions given in Table 1. We compute the critical environments by searching each word for the regular expressions *ea*, *e.a*, *e..a*, *e.*a*, *a* and . in order, and assigning it to the first match; we balance across these groups in our sampled datasets. For each experiment, we sample random training sets with $2 \ldots 40$ instances of each critical environment at increments of 2 and random test sets with 8 instances of each. We run 16 random samples at each dataset size. Results are shown in Figure 1; see Appendix B for numerical scores.

We find a clear trend in the ease of learning of the first three processes, which differ in their degree of locality (and therefore in the smallest k for which they are k-ISL). The most local processes are learned from fewest examples; this is true for every model, even T5. T5 is generally slightly slower to learn than SIP models. In particular, SIP-ISL also shows increasing deficits when attempting to learn less local languages, but these deficits are gradual— the bias toward ISL languages is defeasible rather than absolute. The unbounded process A4 shows a different pattern of results, with SIP-TSL learning it faster than A3 while

[3] This is needed for the rules like the regressive one in Figure 5. When the machine reads an input d, it cannot output yet— it has to wait to see whether the next character is a or not. An epsilon transition would also suffice, but ISL languages are otherwise epsilon-free, a property we aim to preserve in our representations.

[4] We use the parameters from the original SIP paper, using a fixed-length 50 element prefix, tuning the prefix at rate 1.0 and the rest of the parameters at rate 3e-4.

Ref	SPE	Formal
A1	a → A / e _	2-ISL
A2	a → A / e [ ] _	3-ISL
A3	a → A / e [ ] [ ] _	4-ISL
A4	a → A / e [ ]* _	2-TSL

Table 1: The 4 rules used in Experiment 1, written in SPE notation.

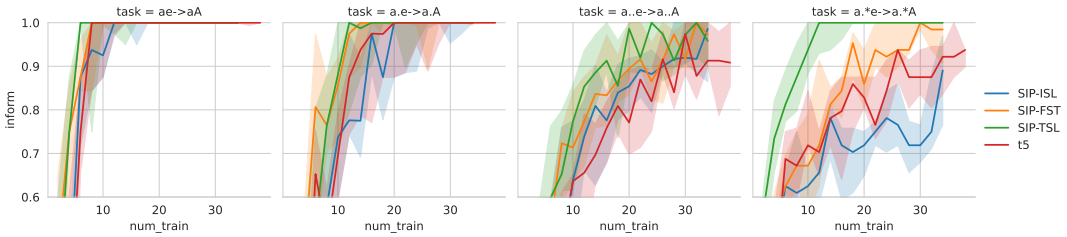

Figure 1: Informedness by number of training examples per critical condition across four languages from Table 1. Shaded region shows 50% confidence interval of the data.

the other models do not.

We also present results comparing progressive and regressive local processes. These are part of a larger set of experiments fully described in Appendix A which also test bidirectional processes and processes which insert segments; in general, these experiments did not reveal interesting differences between models other than the one described here. Comparing a regressive rule (B1) to a progressive rule (B2), however, we see a substantial difference; B1 is more difficult for every model. This occurs despite both languages being 2-ISL, both being part of the training distribution for SIP-ISL/TSL, and our sampling procedure selecting replacement patterns in either direction at the same rate. We return to this issue in the discussion.

4.2 *Natural languages* We run our system on natural language datasets from Belth (2023, 2024). The datasets representing local processes are German (final devoicing), Polish (final devoicing and a related vowel alternation) and English (plural suffix voice assimilation, 3rd person singular *-d* voice assimilation, and vowel nasalization). The datasets representing non-local processes are Finnish front/back vowel harmony and Turkish (CHILDES) front/back and secondary roundness vowel harmony. Following Belth, we sample random training datasets based on token frequency. To generate a dataset with k types, we sample tokens

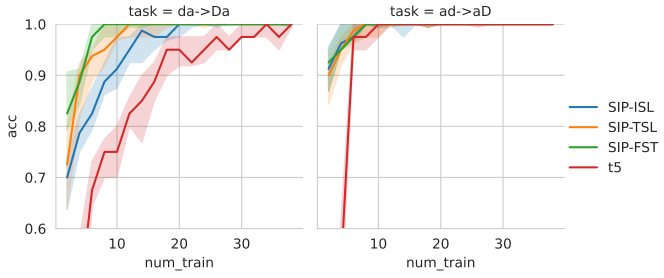

Figure 2: Informedness by number of training examples per critical condition across a regressive local change ($d \rightarrow D$ _ a) and a progressive one ($d \rightarrow D$ a _). Shaded region shows 50% confidence interval of the data.

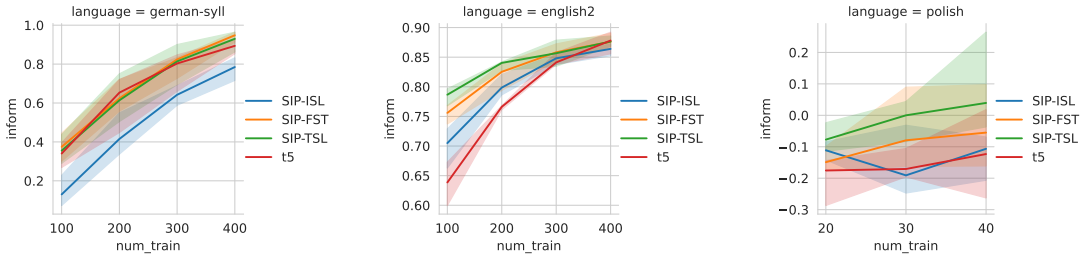

Figure 3: Informedness by total training examples for three natural language datasets from Belth (2023). Shaded region shows 50% confidence interval of the data.

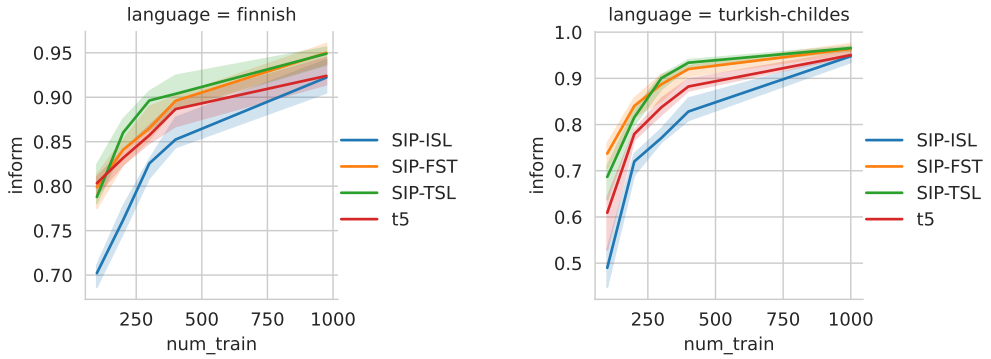

Figure 4: Informedness by total training examples for two natural language datasets from Belth (2024). Shaded region shows 50% confidence interval of the data.

from the unigram distribution until we have sampled k distinct types, then hold out the remaining types for testing. Each word type appears only once in the training or test set presented to the network.[5]

On German, SIP-ISL underperforms the other models, but T5 is surprisingly effective in comparison to the synthetic datasets and does not show the same trend of early poor performance. On English, the trends are more similar to those on synthetic data, with T5 performing worse early on and TSL performing better. On English, 400 examples are enough for differences between models to disappear, but for German, ISL continues to lag behind. Most errors are underapplications of the change (failure to devoice), rather than overextensions (unmotivated voicing); stem copying errors also occur. Results on Polish are poor; with the 40 examples available, no model reaches a clearly positive informativity. Underapplication of the process is very severe, with the best true positive rate (SIP-TSL) no higher than 30%. The Polish dataset thus appears too small for transformer models to learn the appropriate generalizations, while Belth's local learner is capable of doing so.

For Finnish, we observe the same trends as with a synthetic harmony process: SIP-ISL performs poorly, while SIP-TSL has a clear advantage early on. For Turkish, SIP-ISL also clearly performs poorly, but SIP-FST is competitive with SIP-TSL.

[5] For German, we follow Belth in sampling 100 to 400-type training vocabularies in increments of 100 for training. For Finnish and Turkish, we use the same sizes, plus the larger sizes (975 and 1k) used by Belth. His English experiment uses a larger vocabulary for which we did not observe meaningful differences between models; we use the German sizes instead. For Polish, only 80 types are available; we use 20-40 types for training.

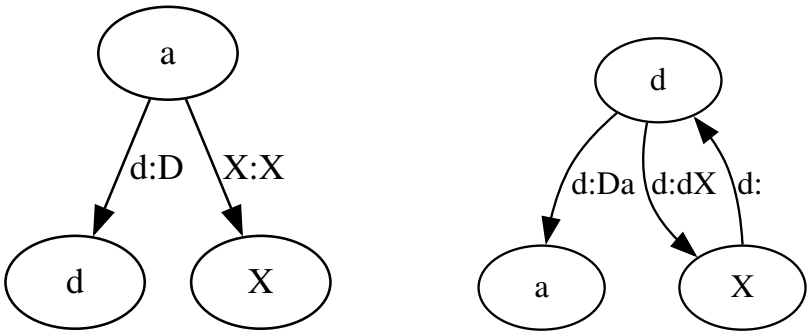

Figure 5: ISL transducer fragments for (left) progressive $ad \rightarrow aD$, (right) regressive $da \rightarrow Da$.

5 Discussion

Our most basic finding is that SIP training fulfills its intended function, producing models with different biases based on their training distribution. SIP-ISL is noticeably slow to learn a long-distance language (A4) while SIP-TSL is fast. On the other hand, SIP training does not *restrict* models to learning languages within their training distribution; the learned biases are defeasible. SIP-ISL is slightly slower than SIP-FST at learning 3-ISL languages that skip a character (A2) or have bidirectional conditioning (see Appendix A), but not much slower. Moreover, much of the locality bias which determines relative learning rates among these languages is already present in T5. Natural language pretraining is apparently already capable of instilling a strong bias towards learning local transductions; SIP training only mildly strengthens this existing prior.

Just as important as where models succeed is where they fail. The ISL-biased model learns long-distance phenomena much more slowly than the other transformer models. This suggests that transformer networks can indeed operate in a regime where their locality preferences are strong enough to typologically differentiate phonology from syntax— the same learning architecture can learn both kinds of system, but can be differently configured to produce different results.

One area in which this pretrained bias is not sufficient to explain the typological facts is the case of unbounded versus count-based long-distance languages (A4 vs A3). While humans prefer to interpret these cases as harmony (Finley, 2012), T5, SIP-FST and SIP-ISL learn the count-based language faster. SIP-TSL has the more cognitively plausible preference for an unbounded process rather than a count-based one. This suggests that, while locality bias is a robust phenomenon arising from generic statistical learning mechanisms and language exposure, tier-based assimilation and dissimilation are more sensitive. Not all plausible learners will inevitably have biases that lead to tier-based harmony systems. Hayes and Wilson (2008) is still correct that models must be explicitly nudged towards the possibility of tier-based analyses (if not the tiers themselves).

Another area where our results diverge from the cognitive facts is in the progressive-regressive asymmetry observed in Figure 2. This difference is comparatively large— at 10 critical examples, SIP-ISL has succeeded perfectly for the progressive language but has a median informedness of 83 for the regressive one, despite both types being well-represented in the training distribution. Moreover, the typological asymmetry is (mildly) in the other direction for segmental phonology; P-base has twice as many regressive assimilation patterns as progressive ones (Brohan and Mielke, 2018).

We view the asymmetry as an artifact of left-to-right processing. The default behavior of phonological transducers is to copy unless some specific criterion is met. A regressive transduction involves more non-copying behavior than a progressive transduction, as shown by the state diagrams for the two processes (Fig. 5). The progressive transducer has only a single non-copying arc (from a to d) while the regressive transducer has a non-copying arc for every transition to and from d. While t5 does not literally implement this transduction as a state machine, it still appears to be affected by this difference in complexity, suggesting that its internal representation of string transductions involves some analogue of state tracking.

This asymmetry does not occur in human languages. This is likely because local phonological processing is not strictly left-to-right. Anticipatory planning is necessary for speech motor control and phonetic representations contain a lot of local context, including following segments (Levelt, 1999), so there is probably more built-in temporal blending at the process level than is induced by building a byte-level LM. This reduces the cost in complexity caused by having to delay the output of the d until a is detected; in reality, the two representations are closely entwined from the start.

Our results on natural languages are generally less impressive than those reported by Belth (2023, 2024), whose learner achieves near-perfect accuracy on small datasets. But we argue that this is not a reason to write off the transformer (or the broader class of connectionist models) for phonological learning. First, our results are generally much better than the ones Belth reports for connectionist models, which struggle even to copy the word stems. For German, for instance, Belth reports an encoder-decoder with an accuracy of 54% on 400-word datasets; since the German devoicing process applies to only 8% of the words, the vast proportion of these errors are simple copying mistakes. On datasets of the same size, all four of the models reported here are 98% accurate or better. The regularization effect of either natural language or SIP training is substantial; while our models still make copying errors, these are few, and even humans make speech errors as part of normal production.

Secondly, this project addresses one area of phonological inductive bias (formal language class) but Belth's learner includes another important bias we do not attempt to model: natural classes. Belth's learners include featural representations for each segment. While natural class biases are violable (as discussed in §2), they allow the learner to share statistical strength across related contexts, rather than learning a family of per-segment processes individually. Such classes can be acquired from rich enough datasets (Shain and Elsner, 2019), but not from the small character-level sets used here.

6 Conclusion

Our results are broadly in line with previous work that views the inductive biases of transformers as inherently well-suited for certain kinds of language data (Bhattamishra et al., 2023), as well as with work exploring how particular biases can be instilled into the transformer through specialized training (McCoy et al., 2020; Griffiths et al., 2024). In our case, we find a pre-existing bias for locality and demonstrate a mechanism for creating a human-like preference for harmony over counting. These biases are not absolute, but they do not need to be to explain observed typology. Learning biases matter when language change creates an ambiguous situation in which learners might acquire one of several grammars (Morley, 2015). In such a situation, the learner does not need to be *incapable* of acquiring a disfavored grammar in order to eliminate it; they merely have to prefer a different one.

On the other hand, evidence of continued discrepancies between transformer biases and human language typology shows the limits of some of our machine metaphors. First, we should not deceive ourselves into thinking that the Chomsky hierarchy provides the only or even the most meaningful way of measuring how hard a language is to learn (Yang and Piantadosi, 2022)— other notions of complexity, like sensitivity, may offer better ways of describing the relative complexity of different grammars. In particular, metaphors based on strictly sequential processing may not correctly describe the kinds of locality that emerge from the interplay of time-varying phonetics with motor planning. We believe that investigating the phonological biases of models which process acoustic phonetic input rather than symbol strings is a promising direction. Such models might offer a richer account of how locality operates at different timescales, leading to a strong tendency for harmony but no bias towards progressive rules for adjacent segments, and begin to address the relationship between learning biases due to natural classes/features and the computational biases presented in this work.

Acknowledgments

This work was supported by the Ohio Supercomputer Center (1987). We thank Matthias Lindemann for his help with the SIP software, and Becca Morley, Yue Yin, three anonymous reviewers and the attendees of AMP 2025 for their comments. This work is dedicated to the memory of our advisor, Eugene Charniak, a great collaborator who loved applying old insights to new formalisms.

References

Beguš, G. (2020). Generative adversarial phonology: Modeling unsupervised phonetic and phonological learning with neural networks. *Frontiers in Artificial Intelligence*, 3:44.

Belth, C. (2024). A learning-based account of phonological tiers. *Linguistic Inquiry*.

Belth, C. A. (2023). A learning-based account of local phonological processes. *Phonology*, 40(1-2):1–33.

Bergmanis, T., Kann, K., Schütze, H., and Goldwater, S. (2017). Training data augmentation for low-resource morphological inflection. In Hulden, M., editor, *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver. Association for Computational Linguistics.

Bhattamishra, S., Ahuja, K., and Goyal, N. (2020). On the practical ability of recurrent neural networks to recognize hierarchical languages. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1481–1494, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bhattamishra, S., Patel, A., Kanade, V., and Blunsom, P. (2023). Simplicity bias in transformers and their ability to learn sparse Boolean functions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Blaho, S. and Szeredi, D. (2013). Hungarian neutral vowels. *Nordlyd*, 40(1):20–40.

Brohan, A. and Mielke, J. (2018). Frequent segmental alternations in P-base 3. *Phonological Typology*, pages 196–228.

Burness, P. A., McMullin, K. J., and Chandlee, J. (2021). Long-distance phonological processes as tier-based strictly local functions. *Glossa: a Journal of General Linguistics*, 6(1).

Chandlee, J., Eyraud, R., and Heinz, J. (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–504.

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.

Fedorenko, E., Ivanova, A. A., and Regev, T. I. (2024). The language network as a natural kind within the broader landscape of the human brain. *Nature Reviews Neuroscience*, 25(5):289–312.

Finley, S. (2011). The privileged status of locality in consonant harmony. *Journal of Memory and Language*, 65(1):74–83.

Finley, S. (2012). Testing the limits of long-distance learning: Learning beyond a three-segment window. *Cognitive Science*, 36(4):740–756.

Finley, S. (2017). Locality and harmony: Perspectives from artificial grammar learning. *Language and Linguistics Compass*, 11(1):e12233.

Finley, S. (2021). Coarticulation and learnability of transparent vowels in vowel harmony. *Proceedings of the Linguistic Society of America*, 6(1):92–106.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

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

Goldwater, S. and Johnson, M. (2004). Priors in Bayesian learning of phonological rules. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 35–42. Association for Computational Linguistics.

Graf, T. (2022). Subregular linguistics: Bridging theoretical linguistics and formal grammar. *Theoretical Linguistics*, 48(3-4):145–184.

Griffiths, T., Austerweil, J., and Berthiaume, V. (2012). Comparing the inductive biases of simple neural networks and Bayesian models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34.

Griffiths, T. L., Zhu, J.-Q., Grant, E., and Thomas McCoy, R. (2024). Bayes in the age of intelligent machines. *Current Directions in Psychological Science*, 33(5):283–291.

Hahn, M. (2020). Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Haley, C. and Wilson, C. (2021). Deep neural networks easily learn unnatural infixation and reduplication patterns. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 427–433.

Harris, A. C. (2008). On the explanation of typologically unusual structures. In Good, J., editor, *Linguistic universals and language change*, pages 54–76. Oxford University Press.

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

Heinz, J. (2010). Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Heinz, J. (2011). Computational phonology–Part I: foundations. *Language and Linguistics Compass*, 5(4):140–152.

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, pages 58–64.

Henderson, J. (2020). The unstoppable rise of computational linguistics in deep learning. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6294–6306, Online. Association for Computational Linguistics.

Hunter, T. (2025). Kallini et al. (2024) do not compare impossible languages with constituency-based ones. *Computational Linguistics*, 51(2):641–650.

Kallini, J., Papadimitriou, I., Futrell, R., Mahowald, K., and Potts, C. (2024). Mission: Impossible language models. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Kodner, J., Khalifa, S., Payne, S. R., and Liu, Z. (2023). Re-evaluating the evaluation of neural morphological inflection models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45.

Levelt, W. J. (1999). Models of word production. *Trends in cognitive sciences*, 3(6):223–232.

Lindemann, M., Koller, A., and Titov, I. (2024). SIP: Injecting a structural inductive bias into a Seq2Seq model by simulation. In *Proceedings of ACL*.

Mayer, C. and Nelson, M. (2020). Phonotactic learning with neural language models. *Society for Computation in Linguistics*, 3(1).

McCarthy, A. D., Kirov, C., Grella, M., Nidhi, A., Xia, P., Gorman, K., Vylomova, E., Mielke, S. J., Nicolai, G., Silfverberg, M., Arkhangelskiy, T., Krizhanovsky, N., Krizhanovsky, A., Klyachko, E., Sorokin, A., Mansfield, J., Ernštreits, V., Pinter, Y., Jacobs, C. L., Cotterell, R., Hulden, M., and Yarowsky, D. (2020). UniMorph 3.0: Universal Morphology. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

McCoy, R. T., Grant, E., Smolensky, P., Griffiths, T. L., and Linzen, T. (2020). Universal linguistic inductive biases via meta-learning. In *42nd Annual Meeting of the Cognitive Science Society: Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020*.

McCoy, R. T. and Griffiths, T. L. (2025). Modeling rapid language learning by distilling Bayesian priors into artificial neural networks. *Nature Communications*, 16.

McMullin, K. and Hansson, G. Ó. (2014). Long-distance phonotactics as tier-based strictly 2-local languages. In *Proceedings of the annual meetings on phonology*.

Mielke, J. (2008). *The emergence of distinctive features*. Oxford University Press.

Millière, R. (2024). Language models as models of language. *The Oxford Handbook of the Philosophy of Linguistics*.

Mirea, N. and Bicknell, K. (2019). Using LSTMs to assess the obligatoriness of phonological distinctive features for phonotactic learning. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1595–1605, Florence, Italy. Association for Computational Linguistics.

Moreton, E. and Pater, J. (2012). Structure and substance in artificial-phonology learning, Part II: Substance. *Language and Linguistics Compass*, 6(11):702–718.

Morley, R. L. (2015). Can phonological universals be emergent?: Modeling the space of sound change, lexical distribution, and hypothesis selection. *Language*, 91(2):e40–e70.

Muradoglu, S. and Hulden, M. (2023). Do transformer models do phonology like a linguist? In *Findings of ACL*, pages 8529–8537.

Ní Chiosáin, M. and Padgett, J. (2001). Markedness, segment realisation, and locality in spreading. *Segmental phonology in Optimality Theory*, pages 118–156.

Ohio Supercomputer Center (1987). Ohio supercomputer center.

Papadimitriou, I. and Jurafsky, D. (2023). Injecting structural hints: Using language models to study inductive biases in language learning. In *Findings of EMNLP*, pages 8402–8413.

Piantadosi, S. T., Muller, D. C., Rule, J. S., Kaushik, K., Gorenstein, M., Leib, E. R., and Sanford, E. (2024). Why concepts are (probably) vectors. *Trends in Cognitive Sciences*, 28(9):844–856.

Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Prickett, B., Traylor, A., and Pater, J. (2022). Learning reduplication with a neural network that lacks explicit variables. *Journal of Language Modelling*, 10(1):1–38.

Regev, T. I., Kim, H. S., Chen, X., Affourtit, J., Schipper, A. E., Bergen, L., Mahowald, K., and Fedorenko, E. (2024). High-level language brain regions process sublexical regularities. *Cerebral Cortex*, 34(3):bhae077.

Sanabria, R., Tang, H., and Goldwater, S. (2023). Analyzing acoustic word embeddings from pre-trained self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Schrimpf, M., Blank, I. A., Tuckute, G., Kauf, C., Hosseini, E. A., Kanwisher, N., Tenenbaum, J. B., and Fedorenko, E. (2021). The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118.

Shain, C. and Elsner, M. (2019). Measuring the perceptual availability of phonological features during language acquisition using unsupervised binary stochastic autoencoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 69–85.

Strobl, L., Angluin, D., Chiang, D., Rawski, J., and Sabharwal, A. (2025). Transformers as transducers. *Transactions of the Association for Computational Linguistics*, 13:200–219.

Valle Pérez, G., Louis, A. A., and Camargo, C. Q. (2019). Deep learning generalizes because the parameter-function map is biased towards simple functions. In *7th International Conference on Learning Representations, ICLR 2019*.

Wu, S., Cotterell, R., and Hulden, M. (2021). Applying the transformer to character-level transduction. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2022). Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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

Yang, X., Aoyama, T., Yao, Y., and Wilcox, E. (2025). Anything goes? A crosslinguistic study of (im)possible language learning in LMs.

Yang, Y. and Piantadosi, S. T. (2022). One model for the learning of language. *Proceedings of the National Academy of Sciences*, 119(5):e2021865119.

Yao, S., Peng, B., Papadimitriou, C., and Narasimhan, K. (2021). Self-attention networks can process bounded hierarchical languages. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3770–3785, Online. Association for Computational Linguistics.

Yedetore, A., Linzen, T., Frank, R., and McCoy, R. T. (2023). How poor is the stimulus? Evaluating hierarchical generalization in neural networks trained on child-directed speech. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9393, Toronto, Canada. Association for Computational Linguistics.

Ref	Rule	Formal	In train?
B1	d → D / _ a	2-ISL regressive	yes
B2	d → D / a _	2-ISL progressive	yes
B3	d → DD /_ a	2-ISL regressive+epenth.	no
B4	d → DD / a _	2-ISL progressive+epenth.	yes
B5	d → D / a _ a	3-ISL bidir.	no
B6	d → DD / a _ a	3-ISL bidir.+epenth.	no

Table 2: The 6 local transitions considered, written in SPE notation.

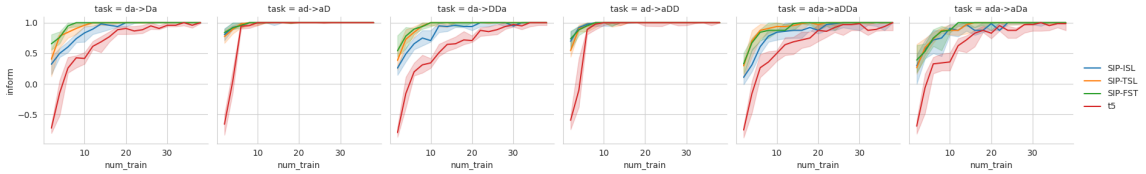

Figure 6

A Additional local transductions

We run a group of experiments to test the model's capacity to learn local transductions within and outside the training distribution. These languages are given in Table **??**. The critical environments are defined as ada, adX, Xda and XdX. Results are shown in Figure **??**. Languages B1 and B3 are compared in Figure 2 in the main text. Numerical scores appear in the next section.

All models reach informativity 1.0 for the largest datasets (which contain 40 instances of each critical environment = 160 total). However, for smaller datasets, T5 (the red line) learns slower than other models.

Several of the languages lie outside the training distribution for the ISL/TSL models. Languages with an a _ a conditioning environment (A5, A6) are 3-ISL but not 2-ISL. A3 (which inserts DD in a regressive context) is also outside the distribution, in this case not for formal reasons but because the embedding format allows only two output characters from a transition, but the required output sequence in this case would be DDa. However, none of these languages appear to impose meaningful learning delays beyond the delay caused by a regressive environment.

B Numerical scores

This section gives the numerical scores for the experiments in the main text and the previous appendix. For synthetic experiments, we show scores only for datasets with 2, 10, 20 and 30 critical examples, as the full range of sizes is too large. We report the actual median and the 95% confidence interval of the median obtained by bootstrapping with `scipy.stats.bootstrap` and the `percentile` method.

A1	2	10	20	30
SIP-FST	0.25 (0.17 − 0.44)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.12 (0.06 − 0.25)	0.94 (0.88 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-TSL	0.55 (0.41 − 0.72)	0.98 (0.88 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.80 (-0.83 − -0.66)	1.00 (0.94 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
A2	2	10	20	30
SIP-FST	0.15 (0.00 − 0.38)	0.88 (0.75 − 0.92)	1.00 (0.98 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.00 (0.00 − 0.12)	0.66 (0.50 − 0.75)	1.00 (0.88 − 1.00)	1.00 (1.00 − 1.00)
SIP-TSL	0.18 (0.03 − 0.30)	0.88 (0.75 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.75 (-0.81 − -0.65)	0.73 (0.62 − 0.80)	0.93 (0.88 − 1.00)	1.00 (1.00 − 1.00)
A3	2	10	20	30
SIP-FST	0.08 (-0.01 − 0.25)	0.66 (0.29 − 0.75)	0.81 (0.77 − 0.97)	0.86 (0.78 − 1.00)
SIP-ISL	0.12 (0.00 − 0.33)	0.39 (0.25 − 0.50)	0.83 (0.75 − 0.86)	0.83 (0.73 − 1.00)
SIP-TSL	0.09 (-0.03 − 0.23)	0.71 (0.47 − 0.88)	0.84 (0.73 − 1.00)	1.00 (0.95 − 1.00)
t5	-0.76 (-0.82 − -0.70)	0.31 (0.22 − 0.47)	0.67 (0.50 − 0.76)	0.83 (0.74 − 0.86)
A4	2	10	20	30
SIP-FST	0.36 (0.23 − 0.48)	0.73 (0.66 − 0.77)	0.93 (0.87 − 1.00)	1.00 (0.94 − 1.00)
SIP-ISL	0.34 (0.25 − 0.44)	0.65 (0.52 − 0.69)	0.74 (0.69 − 0.81)	0.74 (0.69 − 0.77)
SIP-TSL	0.51 (0.42 − 0.56)	0.91 (0.81 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.71 (-0.77 − -0.64)	0.81 (0.77 − 0.85)	0.85 (0.80 − 0.90)	0.94 (0.88 − 0.96)

Table 3: Informedness scores for synthetic processes (Table 1) as a function of number of examples per critical environment. (That is, 2 examples per environment = 12 total.) Median and bootstrapped .95 confidence interval from 16 random samples.

B1	2	10	20	30
SIP-FST	0.66 (0.56 − 0.79)	1.00 (0.96 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.32 (0.17 − 0.43)	0.83 (0.69 − 0.94)	1.00 (0.94 − 1.00)	1.00 (1.00 − 1.00)
SIP-TSL	0.41 (0.12 − 0.58)	0.96 (0.94 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.72 (-0.81 − -0.65)	0.42 (0.33 − 0.52)	0.91 (0.82 − 0.94)	0.96 (0.94 − 1.00)
B2	2	10	20	30
SIP-FST	0.84 (0.77 − 0.89)	1.00 (0.97 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.81 (0.72 − 0.89)	1.00 (0.98 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-TSL	0.77 (0.68 − 0.87)	1.00 (0.96 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.66 (-0.83 − -0.50)	1.00 (0.96 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
B3	2	10	20	30
SIP-FST	0.54 (0.45 − 0.79)	1.00 (0.94 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.26 (0.15 − 0.29)	0.71 (0.69 − 0.88)	0.94 (0.88 − 1.00)	0.95 (0.94 − 1.00)
SIP-TSL	0.39 (0.25 − 0.57)	1.00 (0.96 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.79 (-0.85 − -0.71)	0.34 (0.27 − 0.42)	0.71 (0.69 − 0.77)	0.97 (0.94 − 1.00)
B4	2	10	20	30
SIP-FST	0.74 (0.58 − 0.85)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.70 (0.59 − 0.85)	1.00 (0.95 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
SIP-TSL	0.54 (0.44 − 0.71)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.59 (-0.73 − -0.52)	1.00 (0.96 − 1.00)	1.00 (0.95 − 1.00)	1.00 (0.96 − 1.00)
B5	2	10	20	30
SIP-FST	0.39 (0.19 − 0.62)	0.88 (0.84 − 0.92)	1.00 (0.88 − 1.00)	1.00 (0.94 − 1.00)
SIP-ISL	0.30 (0.00 − 0.47)	0.91 (0.81 − 0.98)	0.98 (0.88 − 1.00)	1.00 (0.94 − 1.00)
SIP-TSL	0.27 (0.11 − 0.62)	0.88 (0.81 − 0.97)	1.00 (1.00 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.69 (-0.81 − -0.59)	0.36 (0.22 − 0.50)	0.83 (0.75 − 0.88)	0.97 (0.88 − 1.00)
B6	2	10	20	30
SIP-FST	0.33 (0.19 − 0.50)	0.88 (0.81 − 1.00)	1.00 (0.88 − 1.00)	1.00 (1.00 − 1.00)
SIP-ISL	0.11 (0.00 − 0.16)	0.84 (0.73 − 0.88)	0.88 (0.81 − 0.98)	1.00 (0.88 − 1.00)
SIP-TSL	0.30 (0.12 − 0.42)	0.94 (0.88 − 1.00)	0.97 (0.88 − 1.00)	1.00 (1.00 − 1.00)
t5	-0.75 (-0.88 − -0.69)	0.50 (0.38 − 0.53)	0.88 (0.72 − 0.97)	0.98 (0.88 − 1.00)

Table 4: Informedness scores for synthetic processes (Table **??**) as a function of number of examples per critical environment. (That is, 2 examples per environment = 12 total.) Median and bootstrapped .95 confidence interval from 16 random samples.

B.1 *Synthetic data*

german-syll	100	200	300	400
SIP-FST	0.37 (0.27 − 0.54)	0.62 (0.55 − 0.75)	0.82 (0.70 − 0.84)	0.95 (0.91 − 0.97)
SIP-ISL	0.13 (0.07 − 0.25)	0.42 (0.32 − 0.57)	0.64 (0.59 − 0.72)	0.78 (0.70 − 0.84)
SIP-TSL	0.35 (0.28 − 0.44)	0.61 (0.48 − 0.77)	0.81 (0.67 − 0.92)	0.93 (0.86 − 0.97)
t5	0.34 (0.26 − 0.44)	0.65 (0.44 − 0.77)	0.80 (0.65 − 0.87)	0.89 (0.84 − 0.95)
english2	100	200	300	400
SIP-FST	0.76 (0.74 − 0.77)	0.83 (0.79 − 0.84)	0.86 (0.84 − 0.88)	0.88 (0.86 − 0.90)
SIP-ISL	0.70 (0.66 − 0.73)	0.80 (0.78 − 0.82)	0.85 (0.84 − 0.86)	0.86 (0.85 − 0.87)
SIP-TSL	0.79 (0.76 − 0.80)	0.84 (0.82 − 0.84)	0.86 (0.83 − 0.88)	0.88 (0.87 − 0.89)
t5	0.64 (0.59 − 0.68)	0.77 (0.76 − 0.79)	0.84 (0.84 − 0.86)	0.88 (0.85 − 0.89)
polish	20	30	40	
SIP-FST	-0.15 (-0.18 − -0.09)	-0.08 (-0.17 − 0.10)	-0.05 (-0.19 − 0.13)	
SIP-ISL	-0.11 (-0.16 − -0.08)	-0.19 (-0.25 − 0.00)	-0.11 (-0.23 − -0.03)	
SIP-TSL	-0.08 (-0.16 − -0.02)	0.00 (-0.10 − 0.05)	0.04 (-0.13 − 0.29)	
t5	-0.18 (-0.32 − -0.14)	-0.17 (-0.26 − -0.09)	-0.12 (-0.28 − 0.03)	

finnish	100	200	300	400	975
SIP-FST	0.80 (0.77 − 0.82)	0.84 (0.82 − 0.86)	0.87 (0.85 − 0.89)	0.90 (0.89 − 0.90)	0.95 (0.93 − 0.96)
SIP-ISL	0.70 (0.68 − 0.71)	0.76 (0.74 − 0.78)	0.83 (0.80 − 0.84)	0.85 (0.84 − 0.88)	0.92 (0.90 − 0.94)
SIP-TSL	0.79 (0.78 − 0.83)	0.86 (0.84 − 0.88)	0.90 (0.86 − 0.91)	0.90 (0.88 − 0.93)	0.95 (0.94 − 0.96)
t5	0.80 (0.78 − 0.81)	0.83 (0.82 − 0.84)	0.86 (0.85 − 0.87)	0.89 (0.87 − 0.90)	0.92 (0.91 − 0.95)
turkish-childes	100	200	300	400	1000
SIP-FST	0.74 (0.70 − 0.77)	0.84 (0.81 − 0.86)	0.89 (0.84 − 0.91)	0.92 (0.90 − 0.93)	0.96 (0.95 − 0.98)
SIP-ISL	0.49 (0.44 − 0.55)	0.72 (0.67 − 0.74)	0.77 (0.76 − 0.80)	0.83 (0.80 − 0.86)	0.95 (0.93 − 0.95)
SIP-TSL	0.69 (0.63 − 0.73)	0.82 (0.79 − 0.84)	0.90 (0.87 − 0.91)	0.93 (0.92 − 0.94)	0.97 (0.96 − 0.97)
t5	0.61 (0.51 − 0.65)	0.78 (0.77 − 0.79)	0.84 (0.82 − 0.86)	0.88 (0.87 − 0.90)	0.95 (0.94 − 0.96)

Table 5: Informedness scores for natural languages as a function of training set size: Median and bootstrapped .95 confidence interval from 10 random samples.

B.2 *Natural languages*