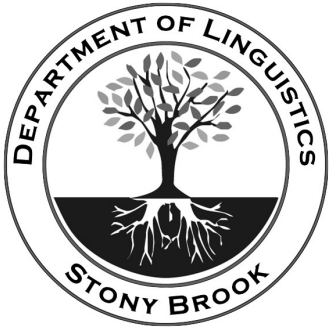
LEARNING PHONOLOGICAL TRANSDUCTIONS FROM SPARSE DATA OVER STRUCTURED REPRESENTATIONS

Magdalena Markowska



Stony Brook University

iACS
INSTITUTE FOR ADVANCED COMPUTATIONAL SCIENCE

Doctoral Defense, March 13

Phonological knowledge is part of what every speaker possesses.

Phonological knowledge is part of what every speaker possesses.

We store mental representations of words,
and we produce them in systematic ways.

Phonological knowledge is part of what every speaker possesses.

We store mental representations of words,
and we produce them in systematic ways.

Crucially, this system is not explicitly taught.
Yet learners still acquire it from limited input.

Phonological knowledge is part of what every speaker possesses.

We store mental representations of words,
and we produce them in systematic ways.

Crucially, this system is not explicitly taught.
Yet learners still acquire it from limited input.

The central question is: how?

Phonological knowledge is part of what every speaker possesses.

We store mental representations of words,
and we produce them in systematic ways.

Crucially, this system is not explicitly taught.
Yet learners still acquire it from limited input.

The central question is: how?

Language Acquisition

Learnability

Phonological knowledge is part of what every speaker possesses.

We store mental representations of words,
and we produce them in systematic ways.

Crucially, this system is not explicitly taught.
Yet learners still acquire it from limited input.

The central question is: how?

Language Acquisition

Learnability

LEARNABILITY IN PHONOLOGY: WHAT IS IT, AND WHY CARE?

- **Learnability** asks whether a learner can acquire the **productive sound patterns** of a language from the input they receive.
- It helps us distinguish genuine phonological knowledge from accidental regularities in the lexicon: speakers show learned knowledge when they generalize patterns to novel forms.
- It also gives us a way to evaluate phonological theories and models by asking whether their representations and mechanisms make acquisition from realistic data possible.

APPROACHES TO LEARNING PHONOLOGY

Rule-based learners:
Johnson (1984); Albright & Hayes (2002, 2003); Rasin et al. (2021); Belth (2023, 2024)

Optimality Theory:
Prince & Smolensky (1993); Tesar & Smolensky (1998, 2000); Prince & Tesar (2004); Boersma & Hayes (2001); Lamont (2021, 2022)

Neural networks:
Begus (2020); Prickett (2021)

Grammatical inference:
Gildea & Jurafsky (1996); de la Higuera (2010); Heinz (2010); Heinz et al.(2015); ; Chandlee et al. (2014); Jardine et al. (2014)

(and many more)

HOW DO THE MAIN APPROACHES COMPARE?

	Rule-based	**OT**	**Neural**	**GI**
Uses phonological representations	✓	✓	✗	✗
Performs well on learning experiments	✓	✓	✓	✗
Theorems about learnable class	✗	✗	✗	✓
Theorems about learning behavior	✗	✓	✗	✓

- This dissertation addresses the two ✗ in the GI column.
- A valuable direction for future research is to reduce these trade-offs by turning more of the ✗ into ✓ while making the different approaches directly comparable in terms of learning effectiveness.

WHY GRAMMATICAL INFERENCE MATTERS FOR PHONOLOGY?

- GI models phonological grammars as **explicit and interpretable computational devices**, so hypotheses are interpretable and testable
- The formal space itself provides **complexity restrictions**: we can ask what kinds of phonological processes are possible and where the typological boundaries lie
- Subregular classes cover a **broad phonological typology** while still supporting theoretical learning results

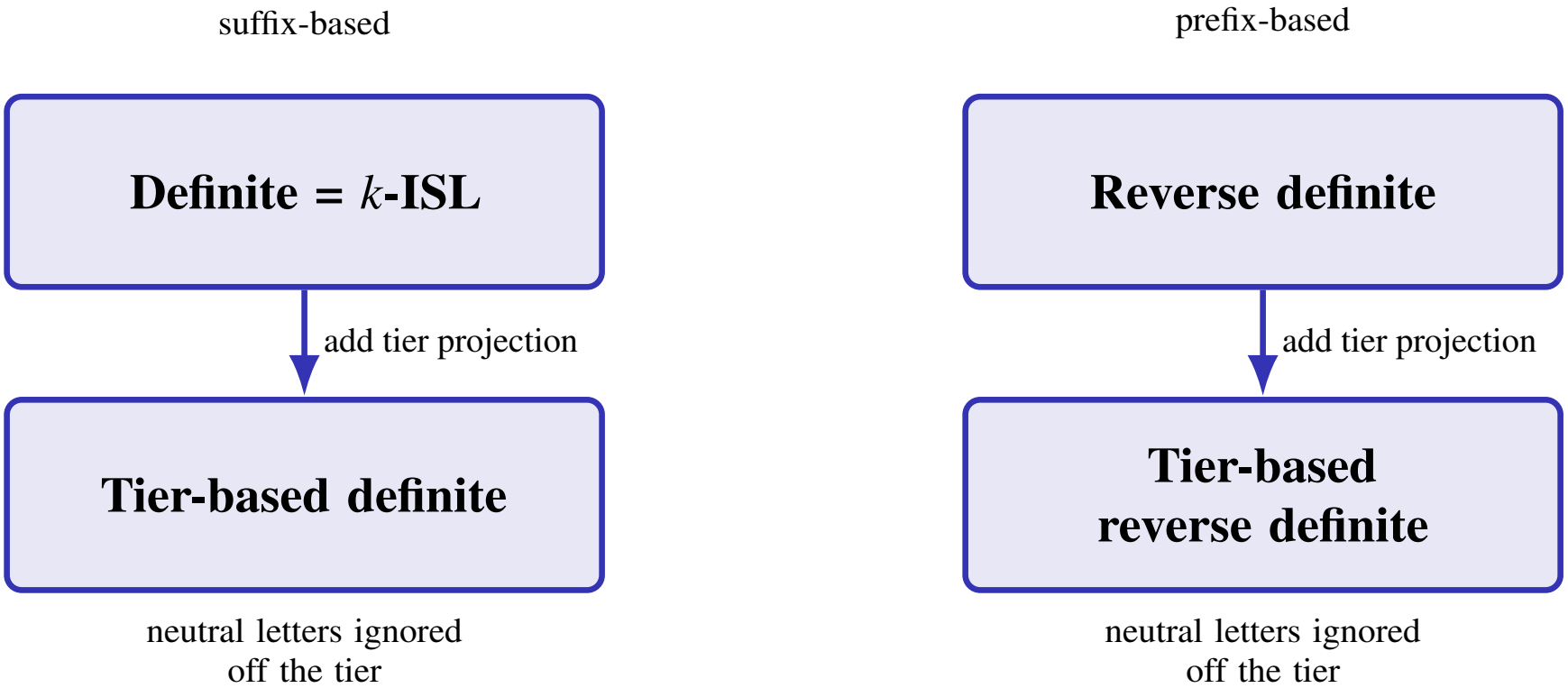
explicit
interpretable hypotheses

restricted
typologically meaningful complexity

broad
coverage of major phonological patterns

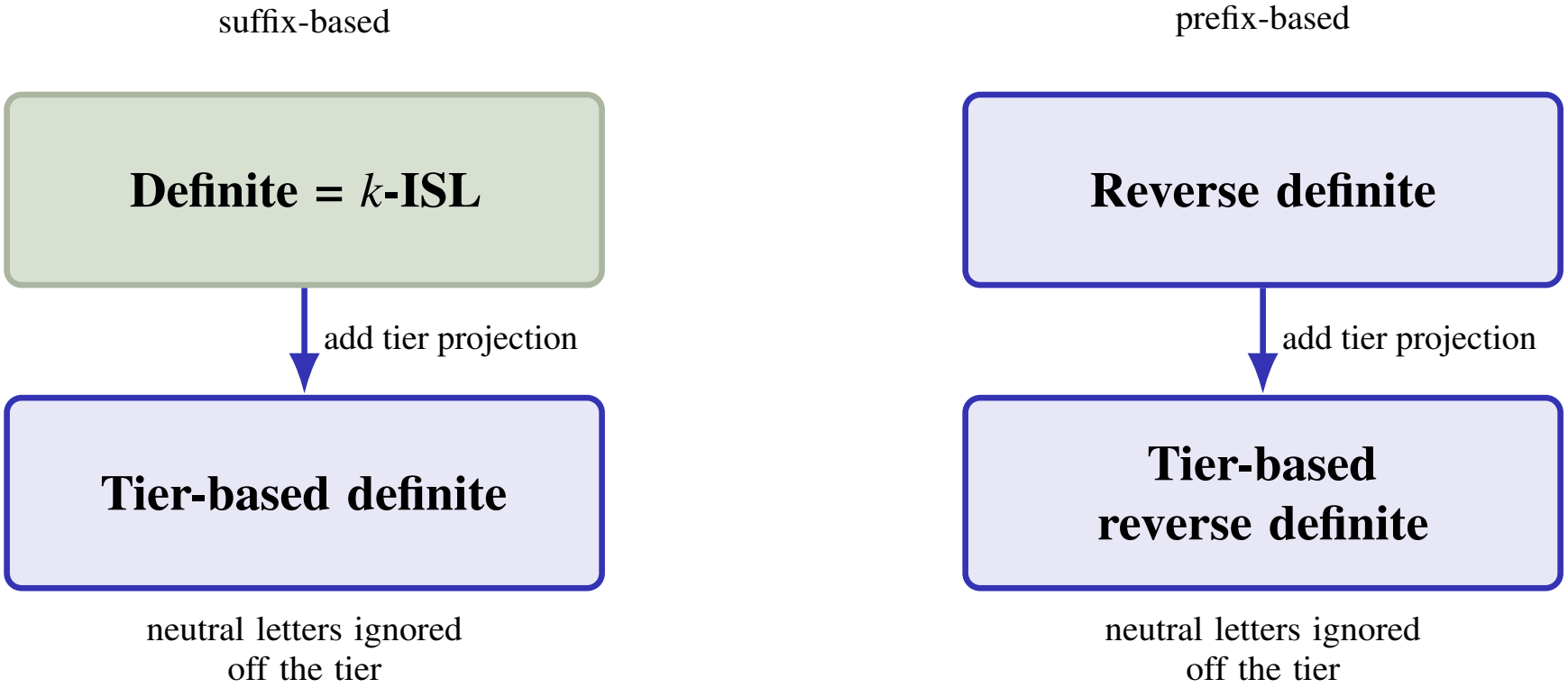
(de la Higuera (2010); Johnson (1972); Kaplan and Kay (1994); Heinz (2010); Chandlee and Heinz (2018); Heinz and Lai (2013); Chandlee et al. (2014); Jardine (2016))

COMPLEXITY CLASSES FOR PHONOLOGICAL PROCESSES



(Chandlee 2014; Chandlee and Heinz 2018; Lambert 2023; Lambert and Heinz 2023, 2024)

COMPLEXITY CLASSES FOR PHONOLOGICAL PROCESSES



Why focus on definite / k-ISL?

- establishes **locality** as the core baseline for comparison
- well-defined and interpretable structure
- majority of phonological processes fit naturally here

THE GAPS THIS DISSERTATION ADDRESSES

- Existing GI learners often require **large characteristic samples** and unusually **complete paradigms**. (e.g., Gildea and Jurafsky (1996).
- That expectation is poorly matched to **sparse acquisition data** and **low-resource documentation settings**, where evidence is limited and uneven. (e.g., Lingos and Yang (2016))

FIRST CONTRIBUTION

Directly incorporating **phonological representations (features)** into the grammatical inference learner SOSFIA (Jardine et al. 2014) **reduces** the characteristic sample needed for successful inference.

THE GAPS THIS DISSERTATION ADDRESSES

- Existing GI learners often require **large characteristic samples** and unusually **complete paradigms**. (e.g., Gildea and Jurafsky (1996).
- That expectation is poorly matched to **sparse acquisition data** and **low-resource documentation settings**, where evidence is limited and uneven. (e.g., Lingos and Yang (2016))

FIRST CONTRIBUTION

Directly incorporating **phonological representations (features)** into the grammatical inference learner SOSFIA (Jardine et al. 2014) **reduces** the characteristic sample needed for successful inference.

Worst case (Polish)

50% reduction
on synthetic data

THE GAPS THIS DISSERTATION ADDRESSES

- Existing GI learners often require **large characteristic samples** and unusually **complete paradigms**. (e.g., Gildea and Jurafsky (1996).
- That expectation is poorly matched to **sparse acquisition data** and **low-resource documentation settings**, where evidence is limited and uneven. (e.g., Lingos and Yang (2016))

FIRST CONTRIBUTION

Directly incorporating **phonological representations (features)** into the grammatical inference learner SOSFIA (Jardine et al. 2014) **reduces** the characteristic sample needed for successful inference.

Worst case (Polish)

50% reduction
on synthetic data

Best case (English)

96% reduction
on synthetic data

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

SECOND CONTRIBUTION

Introducing feature-based parameters, i.e., **processing direction** and **learning of** k, further reduces *some* sample requirements beyond factorization.

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

SECOND CONTRIBUTION

Introducing feature-based parameters, i.e., **processing direction** and **learning of** k, further reduces *some* sample requirements beyond factorization.

No reduction (Polish)

8,235 → 8,069

on synthetic data

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

SECOND CONTRIBUTION

Introducing feature-based parameters, i.e., **processing direction** and **learning of** k, further reduces *some* sample requirements beyond factorization.

No reduction (Polish)

8,235 → 8,069

on synthetic data

Biggest reduction (Yawel)

952 → 164

on synthetic data

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

THIRD CONTRIBUTION

Introducing String **Equation Adaptive Learner (SEAL)**, a new learner that substantially improves data efficiency over **SOSFIA**.

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

THIRD CONTRIBUTION

Introducing String **Equation Adaptive Learner (SEAL)**, a new learner that substantially improves data efficiency over **SOSFIA**.

Worst case (Chukchi)

SOSFIA vs. SEAL

2,352 : 672

on synthetic data

FURTHER CONTRIBUTIONS: SYNTHETIC DATA

THIRD CONTRIBUTION

Introducing String **Equation Adaptive Learner (SEAL)**, a new learner that substantially improves data efficiency over **SOSFIA**.

Worst case (Chukchi)

SOSFIA vs. SEAL

2,352 : 672

on synthetic data

Best case (Polish)

SOSFIA vs. SEAL

8,068 : 104

on synthetic data

FOURTH CONTRIBUTION: NATURALISTIC DATA

FOURTH CONTRIBUTION

Testing **SEAL** on sparse, incomplete, more naturalistic textbook-style datasets shows that it can generalize beyond the observed sample.

FOURTH CONTRIBUTION: NATURALISTIC DATA

FOURTH CONTRIBUTION

Testing **SEAL** on sparse, incomplete, more naturalistic textbook-style datasets shows that it can generalize beyond the observed sample.

Training consistency

5/5 case studies

FOURTH CONTRIBUTION: NATURALISTIC DATA

FOURTH CONTRIBUTION

Testing **SEAL** on sparse, incomplete, more naturalistic textbook-style datasets shows that it can generalize beyond the observed sample.

Training consistency

5/5 case studies

Generalization beyond sample

4/5 case studies

FIFTH CONTRIBUTION: NATRALISTIC DATA

FIFTH CONTRIBUTION

Adding another level of representation, **tiers**, extends the same underlying inference machinery to **long-distance processes**.

FIFTH CONTRIBUTION: NATRALISTIC DATA

FIFTH CONTRIBUTION

Adding another level of representation, **tiers**, extends the same underlying inference machinery to **long-distance processes**.

Tier content learner

CanTIERr-2 learns tiers for backness and roundness harmony in Turkish

FIFTH CONTRIBUTION: NATRALISTIC DATA

FIFTH CONTRIBUTION

Adding another level of representation, **tiers**, extends the same underlying inference machinery to **long-distance processes**.

Tier content learner

CanTIERr-2 learns tiers for backness and roundness harmony in Turkish

Same machinery

given tier content and $k = 2$, SEAL can be reused to learn the mapping

ROADMAP

1	2	3	4	5
FxF	k + direction	SEAL	naturalistic data	tiers

- **Chapter 3: FxF** — decomposes one segmental mapping into feature-wise subproblems over smaller alphabets.
- **Chapter 4: k + direction** — makes the locality bound k and processing direction adjustable for each feature.
- **Chapter 5: SEAL** — introduces the String Equation Adaptive Learner, which outperforms SOSFIA.
- **Chapter 6: naturalistic data** — shows that SEALS learns successfully from small, phonology-textbook-style datasets.
- **Chapter 7: tiers** — extends the same logic to long-distance segmental processes through tier learning.

PART I - FACTOR BY FEATURE (FXF)

FXF: THE CORE REPRESENTATIONAL MOVE

- Instead of learning one transducer over segments, learn one transducer τ per output feature ϕ.
- Each τ uses its own input projection Φ (features necessary to predict output for ϕ).
- Inference is carried out over *feature values* rather than fully specified segments.
- The final surface form is recovered by recombining the learned feature outputs.

GENERAL SCHEMA

segmental input

↓

project to relevant features Φ

↓

learn $\tau(\Phi, \phi)$ for each output feature ϕ

↓

recombine learned feature outputs

↓

surface representation

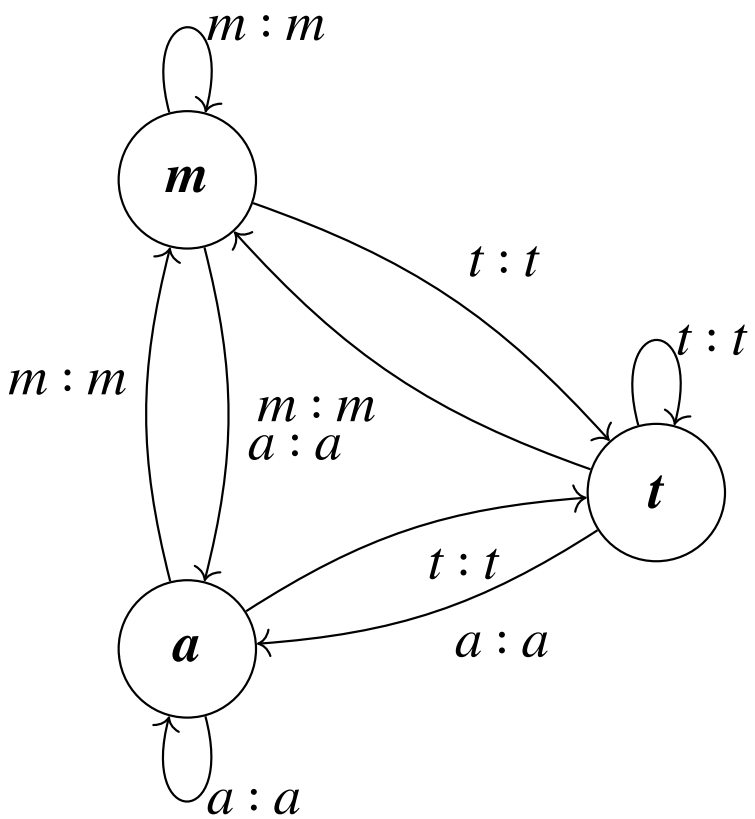
FXF LEARNING PIPELINE

1. Iterate through each $\phi \in F$.
2. Project the sample to $S_{(\Phi,\phi)}$.
3. Check whether that projected sample is functional.
4. If not, enrich Φ with another feature and project again.
5. Once the sample is functional, build the canonical k-ISL skeleton and learn the transition outputs.

Here we consider the first, which may not be the smallest, functional projection that results in successful inference.

EXAMPLE OF 2-ISL SEGMENTAL MACHINE

...		...
/kæm/	→	[kæ̃m]
/kkm/	→	[kkm]
/mæm/	→	[mæ̃m]
/mæt/	→	[mæt]
...		...



FEATURE CONSONANTAL

$$k = 2$$

$$f = [\textbf{consonantal}]$$

$$\begin{array}{ccc} \ldots & & \ldots \\ /+-+/ & \rightarrow & [+-+] \\ /+++/ & \rightarrow & [+++] \\ /+--/ & \rightarrow & [+--] \\ /---/ & \rightarrow & [---] \\ \ldots & & \ldots \end{array}$$

check_if_functional

FEATURE CONSONANTAL

$$k = 2$$

$$f = [\textbf{consonantal}]$$

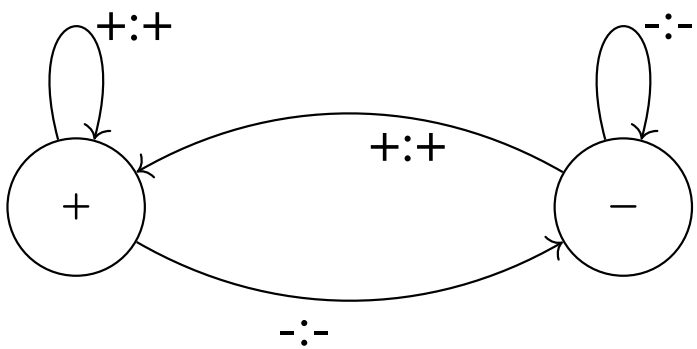
$$\begin{array}{ccc} \ldots & & \ldots \\ /{+}{-}{+}/ & \rightarrow & [\,{+}{-}{+}\,] \\ /{+}{+}{+}/ & \rightarrow & [\,{+}{+}{+}\,] \\ /{+}{-}{-}/ & \rightarrow & [\,{+}{-}{-}\,] \\ /{-}{-}{-}/ & \rightarrow & [\,{-}{-}{-}\,] \\ \ldots & & \ldots \end{array}$$

check_if_functional True

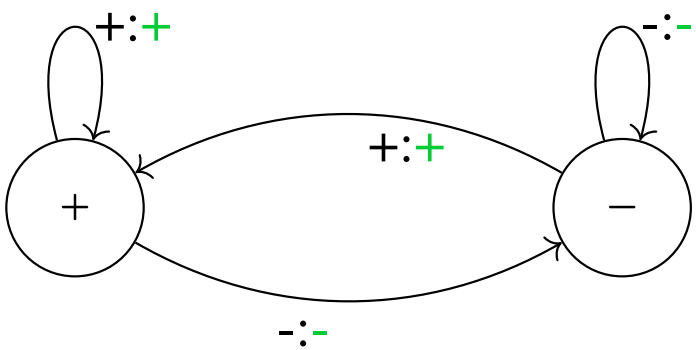
FEATURE CONSONANTAL

$$k = 2$$
$$f = [\textbf{consonantal}]$$

...		...
/+−+/	→	[+−+]
/+++/	→	[+++]
/+−−/	→	[+−−]
/−−−/	→	[−−−]
...		...



IDENT DFT$_{[consonantal]}$



FEATURE NASAL

$$k = 2$$
$$f = [\textbf{nasal}]$$

$$
\begin{array}{ccc}
\cdots & & \cdots \\
/--+/ & \rightarrow & [-++] \\
/--+/ & \rightarrow & [--+] \\
/+-+/ & \rightarrow & [+++] \\
/+-+/ & \rightarrow & [+-+] \\
\cdots & & \cdots
\end{array}
$$

FEATURE NASAL

$$k = 2$$
$$f = [\textbf{nasal}]$$

$$
\begin{array}{ccc}
\ldots & & \ldots \\
/--+/ & \rightarrow & [-++] \\
/--+/ & \rightarrow & [--+] \\
/+-+/ & \rightarrow & [+++] \\
/+-+/ & \rightarrow & [+-+] \\
\ldots & & \ldots
\end{array}
$$

check_if_functional

FEATURE NASAL

$$k = 2$$
$$f = [\textbf{nasal}]$$

...		...
/−−+/	→	[−++]
/−−+/	→	[−−+]
/+−+/	→	[+++]
/+−+/	→	[+−+]
...		...

check_if_functional False

FEATURES NASAL AND CONSONANTAL

$k = 2$

f, f' = [**nasal**, consonantal]

... ...

$/\begin{bmatrix}-\\+\end{bmatrix}\begin{bmatrix}-\\-\end{bmatrix}\begin{bmatrix}+\\+\end{bmatrix}/ \rightarrow \begin{bmatrix}-++\end{bmatrix}$

$/\begin{bmatrix}-\\+\end{bmatrix}\begin{bmatrix}-\\+\end{bmatrix}\begin{bmatrix}+\\+\end{bmatrix}/ \rightarrow \begin{bmatrix}--+\end{bmatrix}$

$/\begin{bmatrix}+\\+\end{bmatrix}\begin{bmatrix}-\\-\end{bmatrix}\begin{bmatrix}+\\+\end{bmatrix}/ \rightarrow \begin{bmatrix}+++\end{bmatrix}$

$/\begin{bmatrix}+\\+\end{bmatrix}\begin{bmatrix}-\\-\end{bmatrix}\begin{bmatrix}-\\+\end{bmatrix}/ \rightarrow \begin{bmatrix}--+\end{bmatrix}$

... ...

FEATURES NASAL AND CONSONANTAL

$$k = 2$$

$$f, f' = [\textbf{nasal}, \text{consonantal}]$$

$$\begin{array}{ccc} \ldots & & \ldots \\ / \begin{bmatrix} - \\ + \end{bmatrix} \begin{bmatrix} - \\ - \end{bmatrix} \begin{bmatrix} + \\ + \end{bmatrix} / & \rightarrow & [-++] \\ / \begin{bmatrix} - \\ + \end{bmatrix} \begin{bmatrix} - \\ + \end{bmatrix} \begin{bmatrix} + \\ + \end{bmatrix} / & \rightarrow & [--+] \\ / \begin{bmatrix} + \\ + \end{bmatrix} \begin{bmatrix} - \\ - \end{bmatrix} \begin{bmatrix} + \\ + \end{bmatrix} / & \rightarrow & [+++] \\ / \begin{bmatrix} + \\ + \end{bmatrix} \begin{bmatrix} - \\ - \end{bmatrix} \begin{bmatrix} - \\ + \end{bmatrix} / & \rightarrow & [--+] \\ \ldots & & \ldots \end{array}$$

check_if_functional

FEATURES NASAL AND CONSONANTAL

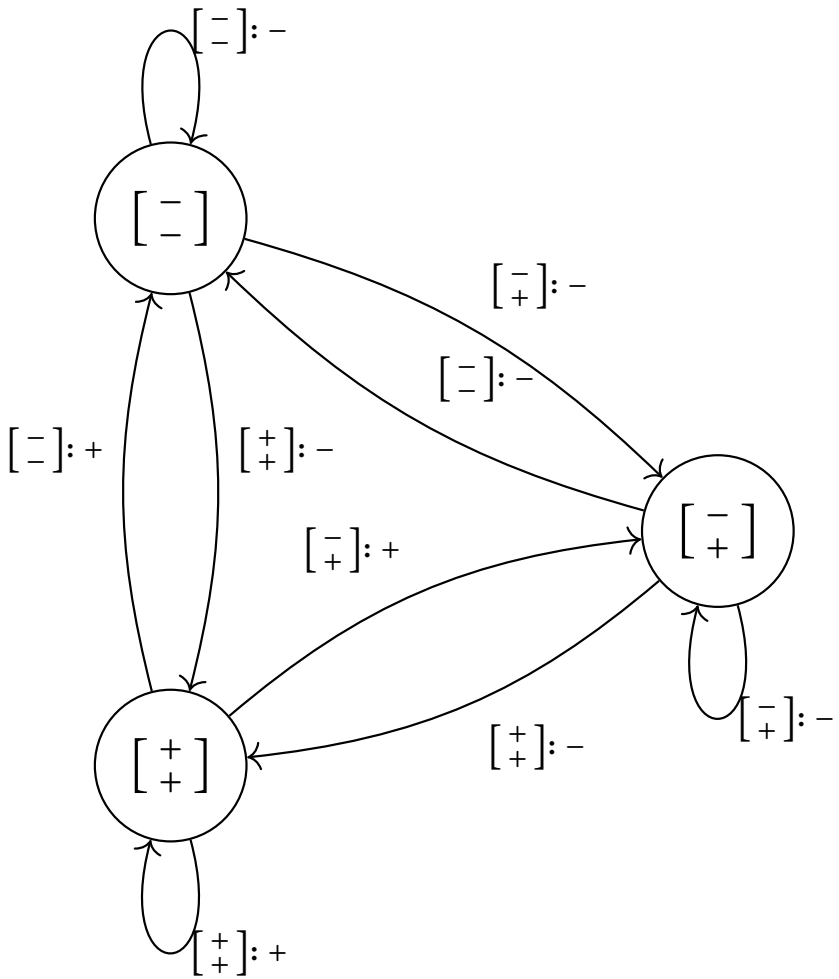
$$k = 2$$

$$f, f' = [\textbf{nasal}, \text{consonantal}]$$

...		...
/ [− / +] [− / −] [+ / +] /	→	[−++]
/ [− / +] [− / +] [+ / +] /	→	[−−+]
/ [+ / +] [− / −] [+ / +] /	→	[+++]
/ [+ / +] [− / −] [− / +] /	→	[+−−]
...		...

check_if_functional True

IDENT DFT$_{\left[\begin{smallmatrix} \textit{nasal} \\ \textit{cons} \end{smallmatrix}\right]}$



WHY FXF SHOULD HELP

- The size of the characteristic sample grows **linearly with the size of the target transducer**.
- **FxF** provides a way to reduce the alphabet size, which in turn shrinks the state space and reduces the size of the characteristic sample.
- When only some features change in a process, many outputs can be learned simply by **positing identity**.

Segmental learner

inventory: $|\Sigma| = 44$

$\Downarrow$

2-ISL DFT: **46 states**

FxF learner

$\tau\left(\left[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}\right], \text{nasal}\right)$

$\Downarrow$

2-ISL DFT: **5 states**

SYNTHETIC BENCHMARK DESIGN FOR FXF

English	**Yawelmani**	**Chukchi**	**Polish**
vowel nasalization	vowel shortening	ə-epenthesis	opacity
one feature	one feature	all features	three features
2-ISL	3-ISL	3-ISL	3-ISL

- These four case studies span single feature change, insertion affecting every feature, and opaque process interaction.
- They provide a controlled setting for asking how much **representation alone** can buy us.

EXPERIMENTAL SETUP

- For each case study, we generate idealized characteristic samples from the "gold" transducer.
- The alphabets were hand-picked to represent the relevant targets and conditioning environments of each process.
- Each generated sample was capped at **20,000** input-output pairs.
- We then used a minimal sample search procedure, **AMβA**, to estimate sample requirements:
  - AMβA draws **10 random pairs** at a time,
  - tests whether the learned hypothesis is equivalent to the original machine.
- Successful featural samples are then synthesized and compared against successful segmenatl samples.

FXF RESULTS: SEGMENTAL LEARNING VS. FEATURE-WISE LEARNING

Process	Generated sample	Segmental	FxF	Reduction
English	18,278	6,157	188	97%
Yawelmani	17,206	16,583	952	94%
Chukchi	17,206	17,181	2,467	85%
Polish	17,206	15,169	8,235	46%

- **Beneficial across the board**: even the smallest reduction (46%) is substantial.
- **Largest gains** arise when the function targets a single feature change.

SUMMARY OF THE FIRST RESULT

- English and Yawelmani show the cleanest gains because their targets and environments for change require minimal feature specification.
- Chukchi remains harder because epenthesis changes every feature at once, so no feature can assume identity.
- Polish remains difficult because it combines multiple processes, one of which targets segments as opposed to a natural class.
- Feature-based learning is therefore highly effective, but not equally so for all process types.

WHAT IS THE LESSON HERE?

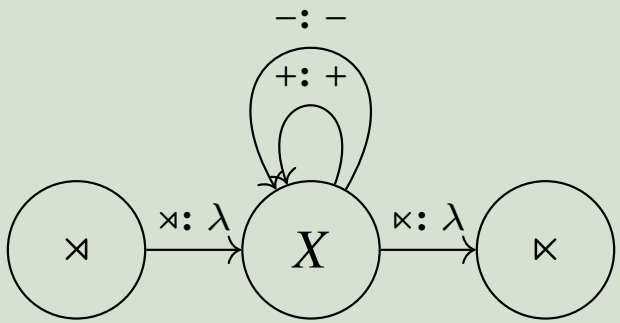
Feature-based representation not only **improves learnability**, but it also opens the door to further reductions in sample complexity: we now can, separately for each feature, adjust **direction of processing** and learn the memory window k.

PART II - LEARNING k AND CHOOSING DIRECTION

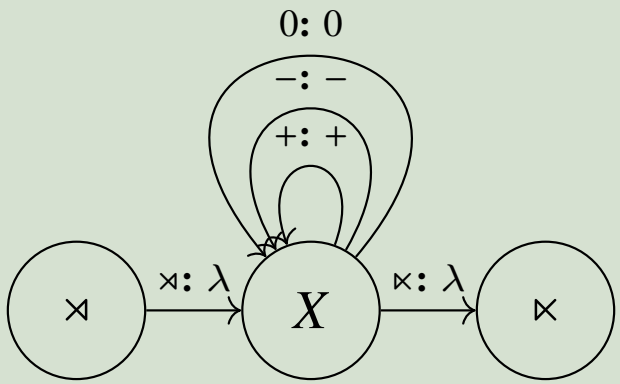
WHY LEARN k PER FEATURE?

- Many output features never change at all across the mapping.
- For those **faithful features**, the successful hypothesis may have the smallest possible machine: **one-state** τ (excluding the boundary markers).
- Learning k per feature prevents the whole system from paying for memory it does not need.

FAITHFUL **BINARY** ϕ



FAITHFUL **TERNARY** ϕ



WHY PROCESSING DIRECTION MATTERS

- The underlying structure of the transducer remains the same under **left-to-right** (L2R) and **right-to-left** (R2L) processing.
- What changes is which transitions contribute to **non-identity** behavior.
- If identity is the default for the learner, the best direction is the one that leaves fewer genuinely non-trivial transitions to learn.

INTUITION FOR L2R PREFERENCE

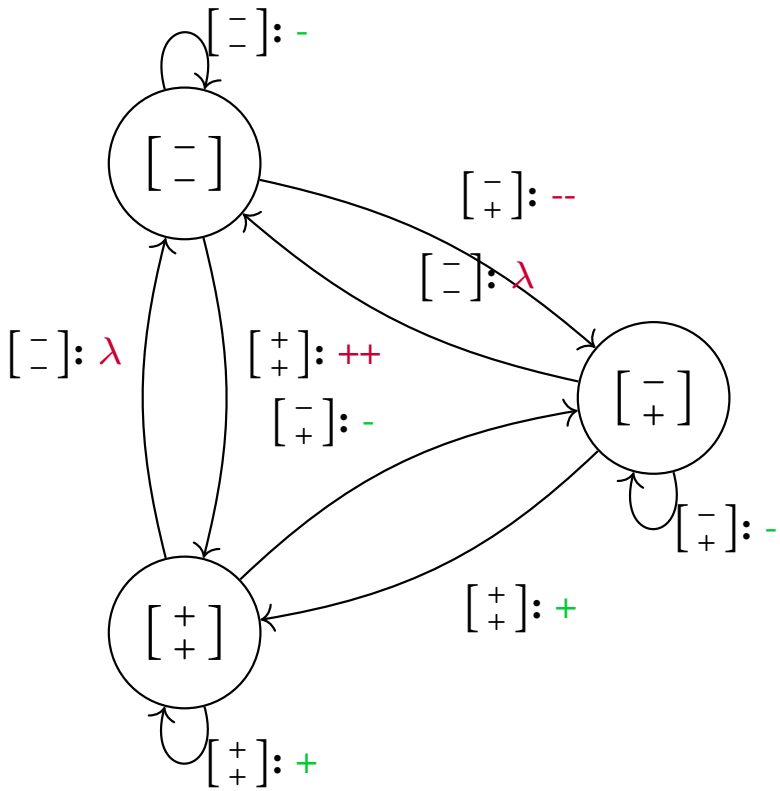
Best for **left-context** processes.

INTUITION FOR R2L PREFERENCE

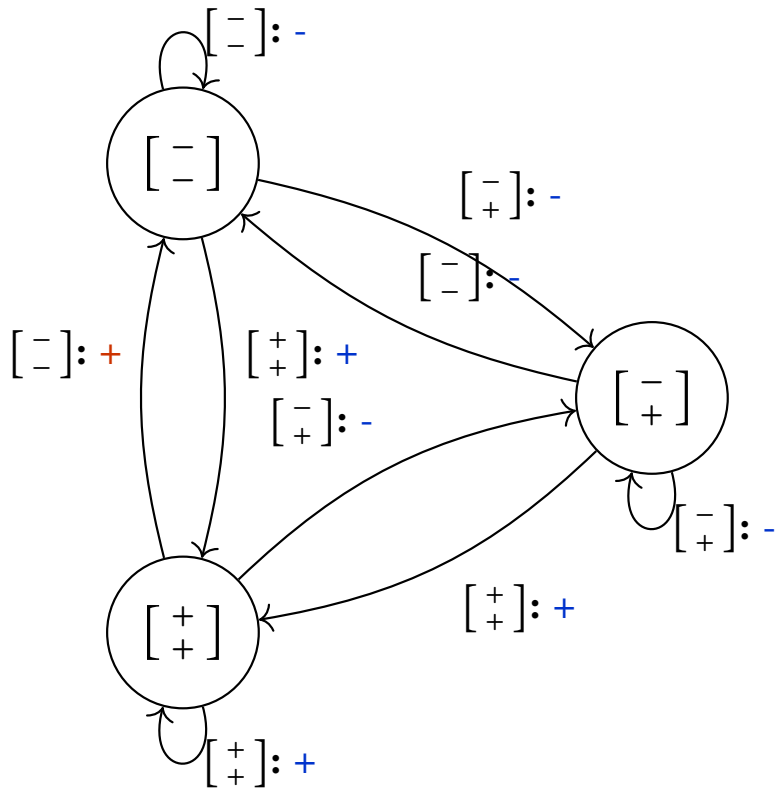
Best for **right-context** processes.

EXAMPLE: $\tau_{([\substack{\text{NAS}\\\text{CONS}}],\text{NAS})}$ IN L2R AND R2L SETTINGS

left-to-right



right-to-left



DID THE PARAMETERS HELP?

	2ISL	3ISL		
Setting	**Eng**	**Yawel**	**Chuk**	**Pol**
Segmental Baseline	6,157	16,583	17,181	16,583
Features	189	952	2,331	8,235
Features + direction	137	851	2,352	8,114
Features + direction + k-learning	**44**	**164**	2,412	8,068

WHY THE GAINS ARE UNEVEN ACROSS PROCESSES

- English and Yawelmani benefit most:
  - the relevant conditioning information is encountered earlier in R2L
  - and k is reduced to 1 for all but one ϕ.
- Chukchi does not show any gain:
  - epenthesis affects every feature, so reducing k is not possible
  - surprisingly no visible advantage of changing direction
- Polish shows some benefits:
  - reducing k for individual *unfaithful* feature, e.g. from 3 to 2 for **[voice]**
  - changing directions shows no effect for features targeting ɔ-raising

WHY THE GAINS ARE UNEVEN ACROSS PROCESSES

- English and Yawelmani benefit most:
  - the relevant conditioning information is encountered earlier in R2L
  - and k is reduced to 1 for all but one ϕ.
- Chukchi does not show any gain:
  - epenthesis affects every feature, so reducing k is not possible
  - surprisingly no visible advantage of changing direction
- Polish shows some benefits:
  - reducing k for individual *unfaithful* feature, e.g. from 3 to 2 for **[voice]**
  - changing directions shows no effect for features targeting ɔ-raising

WHAT IS THE LESSON HERE?

Feature-specific directionality and k again help most in simple cases. Accident, or signal? If the bias is real, perhaps it is **the learner** itself that needs to be reconsidered?

Part III - String Equation Adaptive Learner (SEAL)

WHY SEAL IS NEEDED?

- SOSFIA assigns outputs using onwardness, using the **longest common prefix** (`lcp`) .
- In sparse data, SOSFIA will push the lcp too early.
- This obstructs the gains promised by FxF, directionality, and adaptive k.

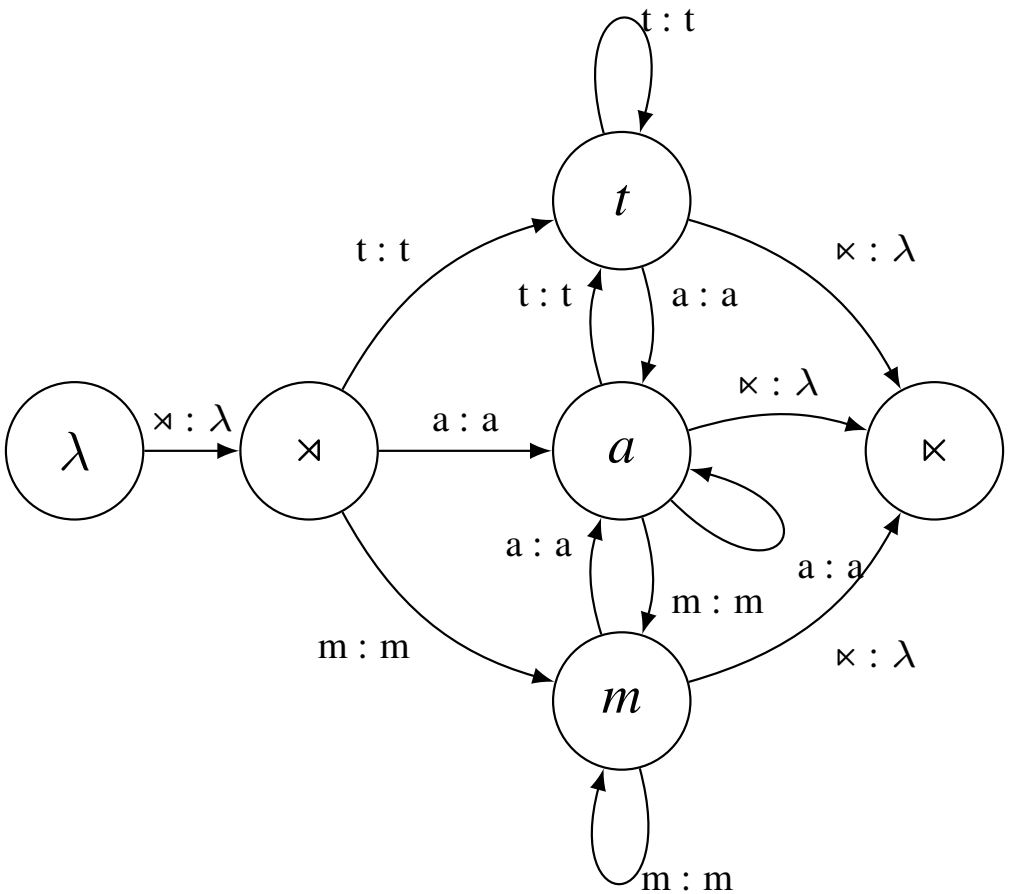
FOR EXAMPLE

If string $w = abc$ is the only one that starts with 'a' and $abc \mapsto abc$, then SOSFIA will assign output abc on the transitions from $\rtimes$ to state a.

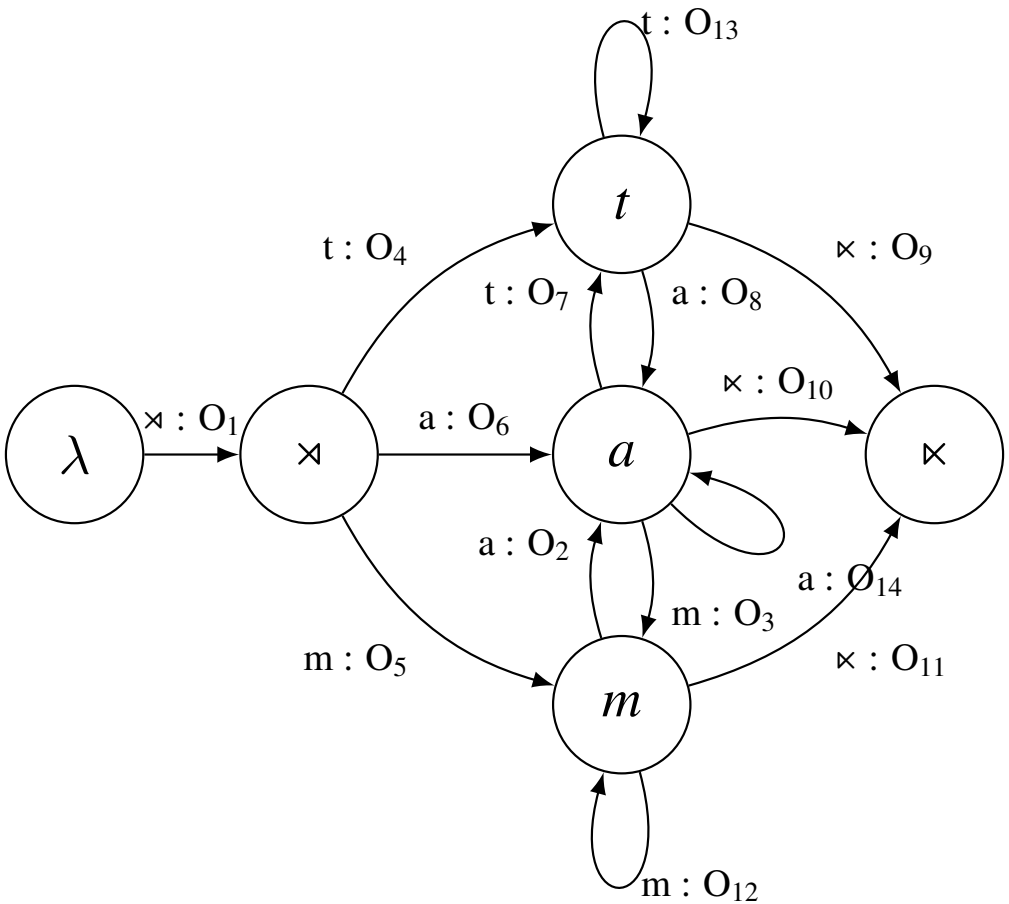
SEAL IN FOUR STEPS

1. Fix the canonical k-ISL skeleton.
2. Treat each transition output as an unknown O_i.
3. Encode each training input as a path of O_i variables.
4. Solve for the O_i variables that make all equations simultaneously true.

STEP 1: BUILD IDENTITY k-ISL τ



STEP 2: ASSIGN VARIABLES



STEP 3: GENERATE ENCODED INPUTS

Training pairs				**Encoded equations**		
$\langle mam \rangle$	$\mapsto$	mãm		$O_1O_5O_2O_3O_{11}$	=	mãm
$\langle mat \rangle$	$\mapsto$	mat		$O_1O_5O_2O_7O_9$	=	mat
$\langle amm \rangle$	$\mapsto$	ãmm	$\Longrightarrow$	$O_1O_6O_3O_{12}O_{11}$	=	ãmm
$\langle tata \rangle$	$\mapsto$	tata		$O_1O_4O_8O_7O_8O_{10}$	=	tata
$\langle tamat \rangle$	$\mapsto$	tãmat		$O_1O_4O_8O_3O_2O_7O_9$	=	tãmat
$\langle atama \rangle$	$\mapsto$	atãma		$O_1O_6O_7O_8O_3O_2O_{10}$	=	atãma

STEP 4:

$O_1O_5O_2O_3O_{11}$	=	mãm
$O_1O_5O_2O_7O_9$	=	mat
$O_1O_6O_3O_{12}O_{11}$	=	ãmm
$O_1O_4O_8O_7O_8O_{10}$	=	tata
$O_1O_4O_8O_3O_2O_7O_9$	=	tãmat
$O_1O_6O_7O_8O_3O_2O_{10}$	=	atãma

- Now the learning task is to distribute the output substrings over the encoded input elements.
- We propose two modes of such distributions:
  - Push-Right Distribution (PRD-SEAL)
  - Length-Constrained Distribution (LCD-SEAL)

HOW SEAL DISTRIBUTES OUTPUT MATERIAL

PRD-SEAL

- Local **push-right** distribution
- Starts from 1:1 alignment
- Reassigns output only when conflict forces it
- Best for feature-changing processes

LCD-SEAL

- ***k*-length constrained** distribution
- Solves under a k-length bound
- Can return material at the final transition
- Needed for epenthesis and deletion

SHARED GOAL

Both modes seek one globally consistent assignment of output strings to transition variables.

SEAL EXPERIMENTAL SETUP

- SEAL is evaluated on the exact same functions as before.
- To isolate the learner difference, the feature sets Φ and required k values are reused from earlier chapters.
- The criterion is once again the exact recovery of the gold transducer from incrementally sampled data.

GOAL

To test whether FxF(SEAL) can recover the same four functions from substantially smaller samples than FxF(SOSFIA).

SEAL VS. SOSFIA ON SYNTHETIC BENCHMARKS

Language	Direction	SOSFIA	SEAL
English	L2R	46	34
	R2L	44	**15**
Yawelmani	L2R	258	312
	R2L	164	**20**
Chukchi	L2R	2,338	2,466
	R2L	2,412	**672**
Polish	L2R	8,114	8,206
	R2L	8,068	**104**

- k was learned for each feature separately in each case

RESULTS INTERPRETATION

- In L2R settings, SEAL is often comparable to SOSFIA.
- In R2L settings, especially for right-context processes, SEAL finally realizes the benefits of directionality and faithful defaults.
- The most notable improvement is Polish: from over 8,000 examples to about 104 in the best setting.
- Chukchi also improves substantially: SEAL achieves a 72% reduction over SOSFIA, but the absolute sample size remains relatively high.

WHAT IS THE LESSON HERE?

SEAL substantially **lowers the sample requirements** relative to SOSFIA, bringing them into a range where evaluation on more realistic datasets becomes feasible. Now off to textbook datasets!

PART IV - LEARNING FROM NATURALISTIC DATA

FIVE TEXTBOOK-STYLE CASE STUDIES

Language	**Process(es)**	$\lvert sample \rvert$
Koasati	nasal assimilation	48
Kerewe	/h/-hardening	28
Lumasaaba	four interacting processes	14
Samoan	front-vowel and final-consonant deletion	116
Serbo-Croatian	*a*-epenthesis and final *l*-vocalization	132

- Datasets adapted from Ellis et al. (2022), based on textbook and problem-set data.
- The original sources provide surface forms and glosses; underlying forms are reconstructed here.
- These datasets are sparse by design, with larger alphabets and less complete paradigms than the synthetic benchmarks.

EXPERIMENTAL SETUP

- As before, we explore processing direction, k, and feature combinations.
- Earlier, search expanded by increasing k and then considering larger feature sets.
- Here, we switch to a different strategy: candidate transducers are sorted by **machine size**, and the smallest machines are tested first.
- This gives the learner a stronger **simplicity bias** in the sparse-data setting.

WHY THIS CHANGE?

Previous work showed that **smaller transducers can be inferred from less data**, so we now search by machine size first.

EVALUATION

- In the synthetic setting, evaluation could compare the learned machine directly to the gold transducer.
- Here that is less reliable: sparse lexicons leave *many paths unattested*.
- We therefore separate two questions:

1. Is the learned transducer **consistent with the training pairs**?
2. Does it **generalize beyond the sample**, on unseen forms and unseen paths?

OVERALL RESULT

- SEAL converges on **all five** textbook-style datasets.
- In **four out of five** cases, the learned transducer generalizes beyond the observed sample.
- The main exception is **Lumasaaba**: only 14 training pairs, four interacting processes, and limited room for featural generalization.

UNSEEN PATHS AND THE QUESTION OF GENERALIZATION

WHAT IS THE LESSON HERE?

FxF(SEAL) is strong enough to move beyond synthetic benchmarks, but its behavior on unseen paths reveals an important limitation.

- In sparse lexicons, many transducer paths remain unattested.
- On those paths, the learner defaults to **identity** — but this default is assigned **transition by transition**, not path by path.
- As a result, when output has been withheld, identity can restore it in the wrong place, or fail to restore it at all.

BROADER IMPLICATIONS

1. How should identity be distributed over **unseen paths**?
2. Should a learner aim to define behavior over **all** paths at all? (**total** vs. **partial** functions problem)

FROM LOCAL TO LONG-DISTANCE PROCESSES

WHERE WE ARE SO FAR

So far, we have shown that successful inference is possible for **local phonological processes**, that is, processes that fall within the k**-ISL** class.

THE REMAINING CHALLENGE

Phonological processes may also exhibit **long-distance dependencies**. An adequate learner should be able to handle such patterns as well, rather than being limited to strictly local generalizations.

FINAL SECTION

In the final section, we introduce an algorithm designed to address this limitation.

Part V - Learning tiers for long-distance dependencies

TIERS AS A REPRESENTATION FOR LONG-DISTANCE DEPENDENCIES

Some phonological patterns are not local. A common solution is to project appropriate elements on a **tier** to make the dependency local.

ON THE FULL STRING

- Trigger and target may be separated by arbitrarily many interveners.
- No fixed surface window k is enough.

ON A TIER

- Project only the relevant segments.
- Then the same dependency becomes local again.

Case study: Turkish round and back harmony

LEARNING TIERS IN PRIOR GRAMMATICAL INFERENCE

Burness & McMullin (2019)

- learn **canonical tiers** for **2-OTSL** functions
- work in an **output-oriented** setting
- **delete** symbols that behave inconsistently

CORE INTUITION

If a symbol is truly irrelevant to the dependency, removing it should not change the relevant local behavior.

SHORTCOMING

The original learner assumes richer, more characteristic-sample-like data than textbook datasets actually provide.

WHAT CHANGES IN THIS DISSERTATION?

Prior GI	**This dissertation**
tier over outputs	tier over **inputs**
assumes rich evidence	relaxed for **sparse naturalistic data**

Long-distance harmony can be brought back to the same input-local transducer architecture using algebraic insights from Lambert and Heinz (2024).

CANTIERR-2: A CONSERVATIVE 2-TIER LEARNER

1. Consider only **attested continuations**.
2. Delete a symbol from the tier only with **double-witness** evidence.
3. Prevent `lcp` from running ahead by enforcing $|lcp(p)| = |p|$.

INFERENCE BIAS

CanTIERr-2 prefers the **safest tier consistent with the sample**, even if that tier is not maximally small.

TURKISH HARMONY CASE STUDY

WHAT IS BEING LEARNED

Tiers over features for backness and roundness harmony from a naturalistic dataset of **265** input–output pairs (Ellis et al. (2022)).

SAMPLE DATA

Stem	Genitive	Plural
ip	ip-in	ip-ler
ev	ev-in	ev-ler
kɯz	kɯz-ɯn	kɯz-lar
sap	sap-ɯn	sap-lar

SETUP

- Learn **tier content** for harmony.
- Exclude */j/*-epenthesis cases.
- Supply a feature inventory Φ.
- After tier inference, fix $k = 2$ and use **SEAL** to infer outputs.

learn tier → build the transducer → infer mapping with SEAL

RESULTS: LEARNING TIERS FOR TURKISH HARMONY

- For **[back]** and **[front]**, the learner successfully removes the **consonantal bundle** from the tier.
- The resulting tiers are **conservative**: they may retain */j/* and underspecified-vowel bundles when the data do not provide enough **double-witness evidence** to eliminate them.
- For **[round]**, adding **[high]** yields a finer partition of the vowel space, which is necessary because roundness harmony is restricted to high vowels.
- Even when the learned tier is not minimal, it still supports the correct **tier-sensitive transducer construction**, after which **SEAL** can infer the mapping.

WHAT IS THE LESSON HERE?

Tier learning can be carried out **separately** and then incorporated into the inference system. This extends the framework beyond strictly local processes, allowing the same general architecture to capture both **local** and **long-distance** phonological dependencies.

CONCLUSION

This dissertation shows that grammatical inference can be successfully adapted to linguistically structured, sparse phonological data.

- **FxF** decomposes the problem **feature by feature**, yielding **smaller transducers** and requiring **less data** for correct generalization.
- **Per-feature parameterization** — including directionality, and adaptive k — further improves **sample efficiency** by reducing both the quantitative and qualitative burden on the learner.
- **SEAL** replaces greedy local output assignment with **global consistency constraints**, allowing the system to recover correct generalization in cases where **SOSFIA** fails on incomplete data.
- **Tier learning** extends the same general framework beyond strictly local patterns, making it possible to handle both **local** and **long-distance** phonological dependencies.

NEXT STEPS

1. **Unseen paths and controlled generalization:** improve behavior on unattested paths by revising how identity is distributed, and by asking more explicitly when phonological mappings should be treated as **total** versus **partial** functions.
2. **Richer representations:** extend the framework to more complex representational domains, such as morphological domains and tones.
3. **More realistic learning settings:** evaluate the system on larger and more naturalistic datasets, including **child-directed speech**, and move toward learning not only the mapping but also the **lexicon** itself (Hua and Jardine (2021); Chandlee and Jardine (forthcoming)).
4. **Broader empirical comparison:** compare this framework directly with alternative learners, including **neural sequence-to-sequence models** and **rule-based / algorithmic learners** such as Belth (2024).

Thank you!