Deep Learning, Languages, Topology, Algebra

Joshua Ackerman & George Cybenko
Dartmouth

Work partially supported by DARPA award HR001119C0075 (SRI prime, Dartmouth subcontractor)

May 28, 2021 | LangSec 2021

Table of Contents

Problem statement:

To learn grammars from positive and negative examples.

Introduction

Foundational Questions

- Is grammar learning a practical task or is it “hard”?
- What does it even mean for a neural network to learn a grammar?

Introduction

Grammar Learning Hardness

- We have a finite set of examples from a language (both positive and negative)
- Can be exactly represented by a simple to construct Non-Determinisitc Finite Automaton (NDFA).
- But reducing the NDFA to a minimal DFA can be an exponentially hard task, IE. NP-Hard. (See Gold 1978, Pitt and Warmuth 1988, etc)
- So we have to be careful what we mean by “learning” a language from examples.

Introduction

Grammar Learning Definitions

- Many different possible definitions:
  - **Definition 1.** Recover the symbols and the specific rules to construct strings.
  - **Definition 2.** Estimate a probability density over the next token in the language, given previous tokens i.e., $p(x_i \mid x_1, \ldots, x_{i-1})$.
  - **Definition 3.** Learn to tell the difference between strings in the language and not in the language i.e., estimate the posterior distribution $p(y \mid x)$.
- For the purposes of this presentation we use definition 3.

Introduction

Grammar Learning Hardness Revisited

- **Theoretical Results with Unbounded Networks:** Neural networks with sigmoidal activations can approximate any function (Cybenko, 1989).

- **Theoretical Results with Bounded Networks:** Much more variety here, as the results are extremely architecture dependent. One brief highlights include

$$\text{Transformer, CNN} \subset \mathbf{REGULAR}.$$

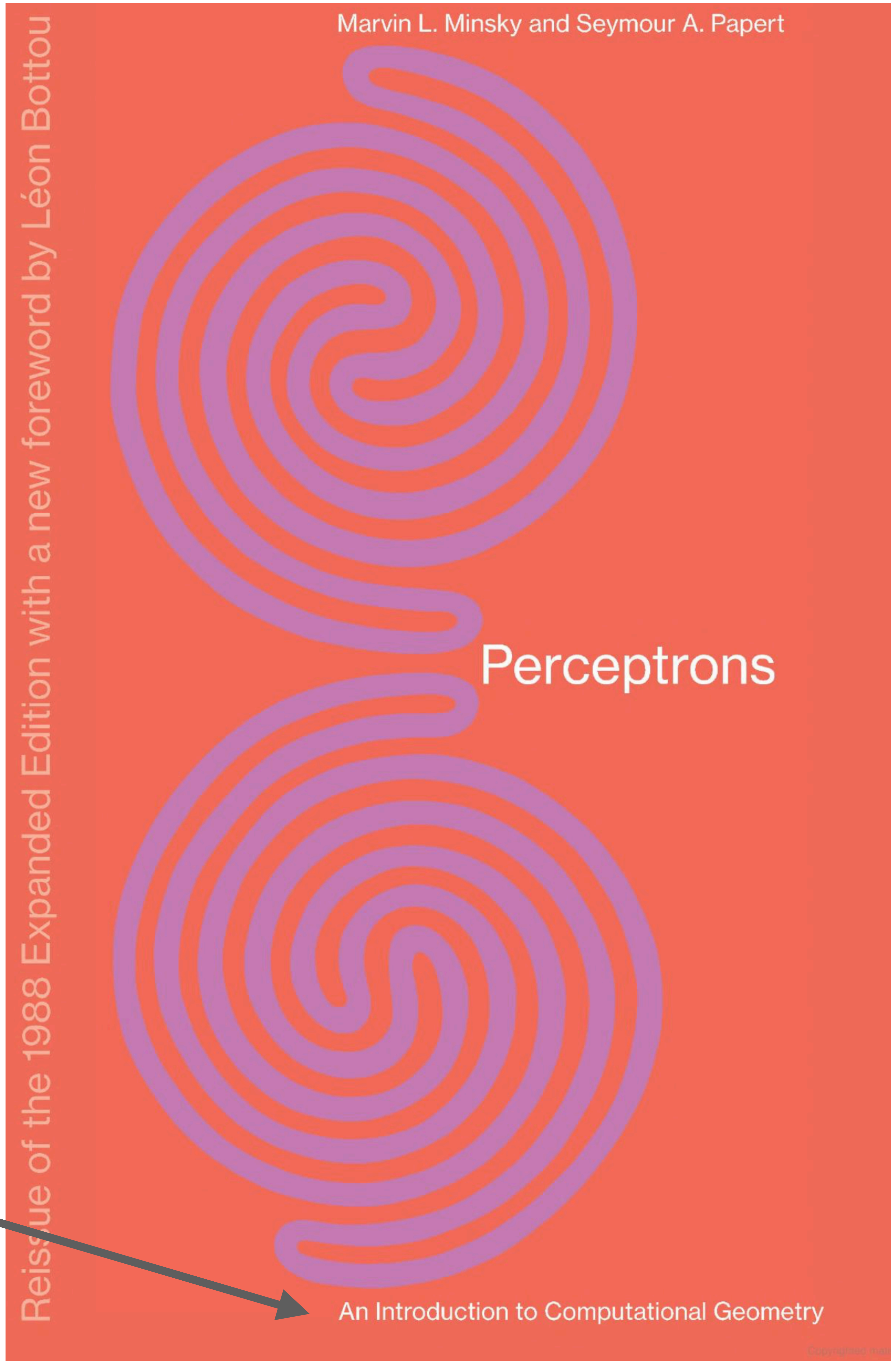
  See (Ackerman and Cybenko, 2020) for a survey of recent work.

- **Practical results & More Realistic Theory:** The goal of our work.

Neural Homology

[Guss et al., 2018] [Naitzat et al., 2020]

Note: Classic work from 1969



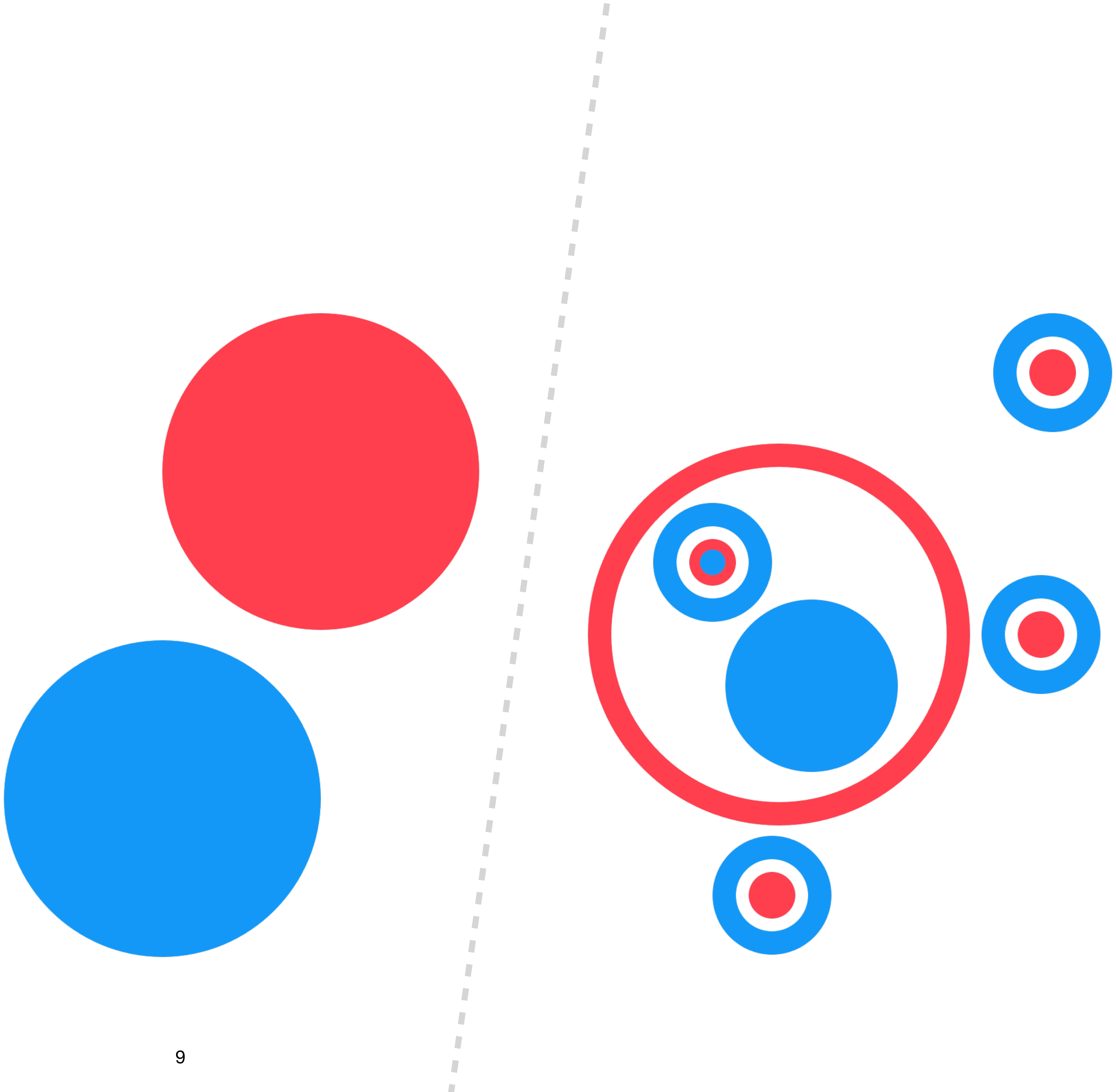
Neural Homology

Intuitive Example I

- Suppose red and blue shapes correspond to components of positive and negative labels in a binary classification problem.

- The left most dataset is certainly easier to linearly separate than the right.

- In generative modeling such a dataset is also easier to 'gaussianize'.



Neural Homology

Intuitive Example II

- A neural network with a hidden layer of width two is incapable of linearly separating this dataset.
- The animation shows the evolution of the neural network during training.

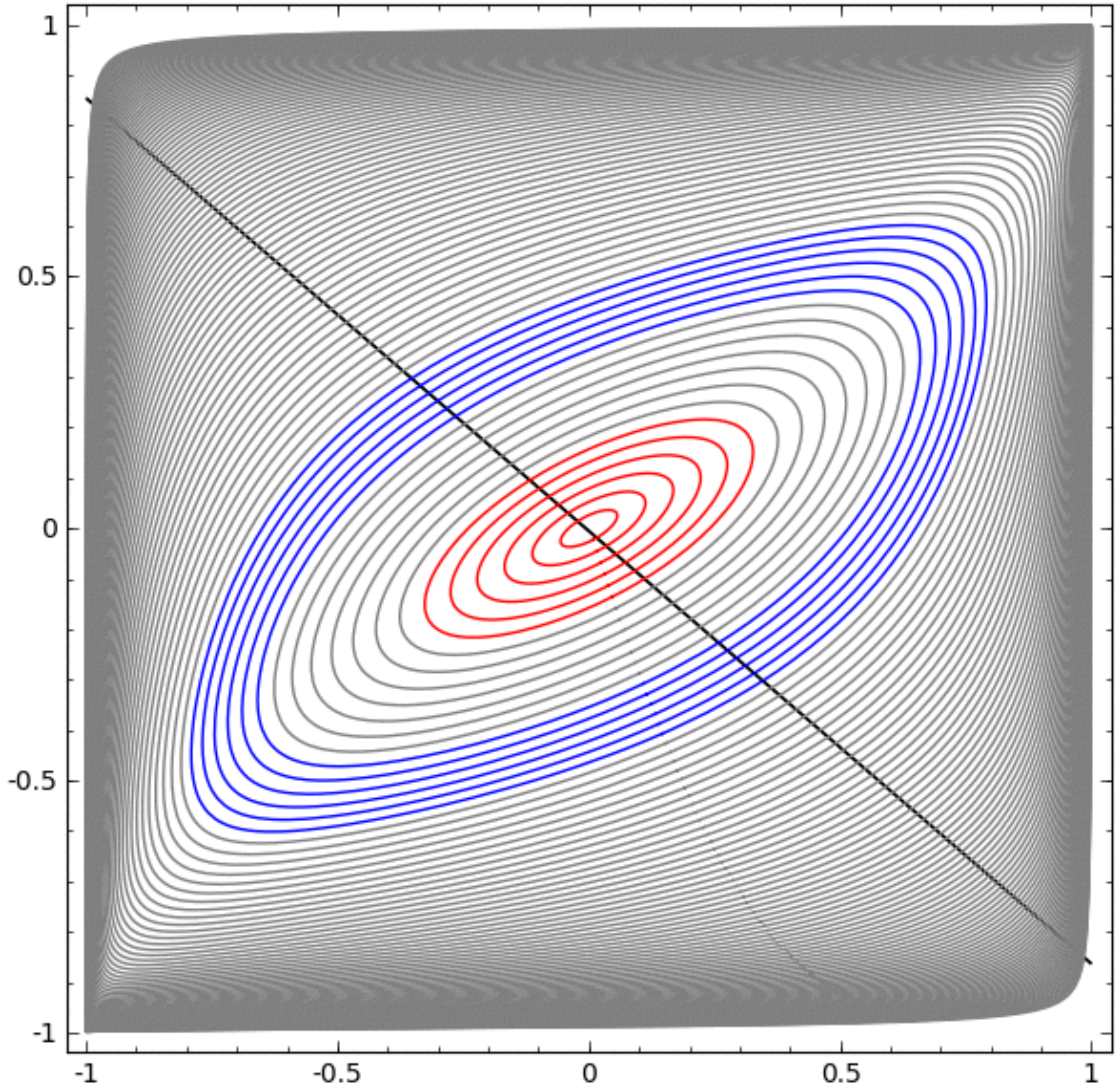


Image from https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Neural Homology

Intuitive Example II

- On the other hand, the task is trivial for a network with a hidden layer of width three.

- Again, the animation shows the evolution of the neural network during training.

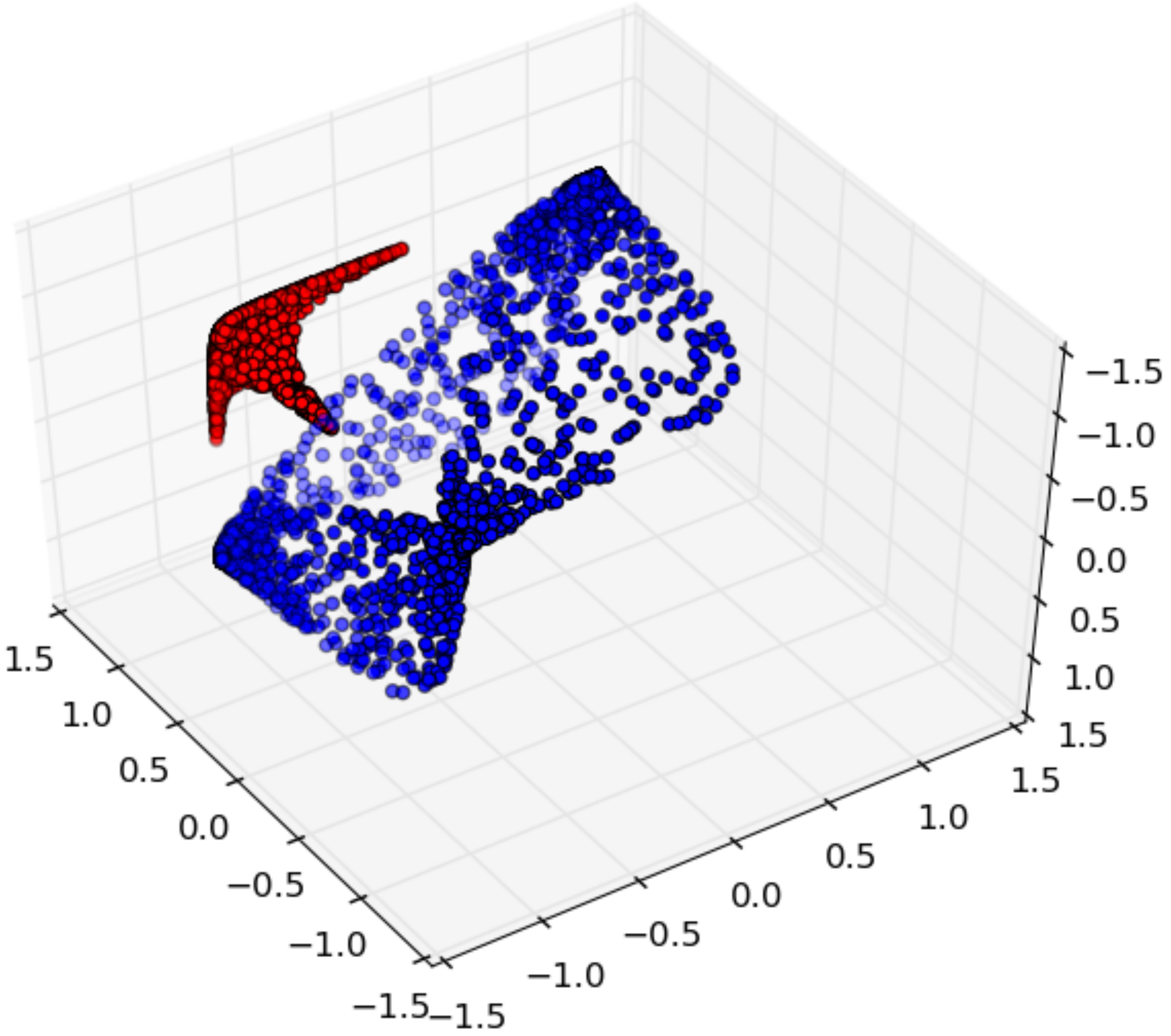


Image from https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

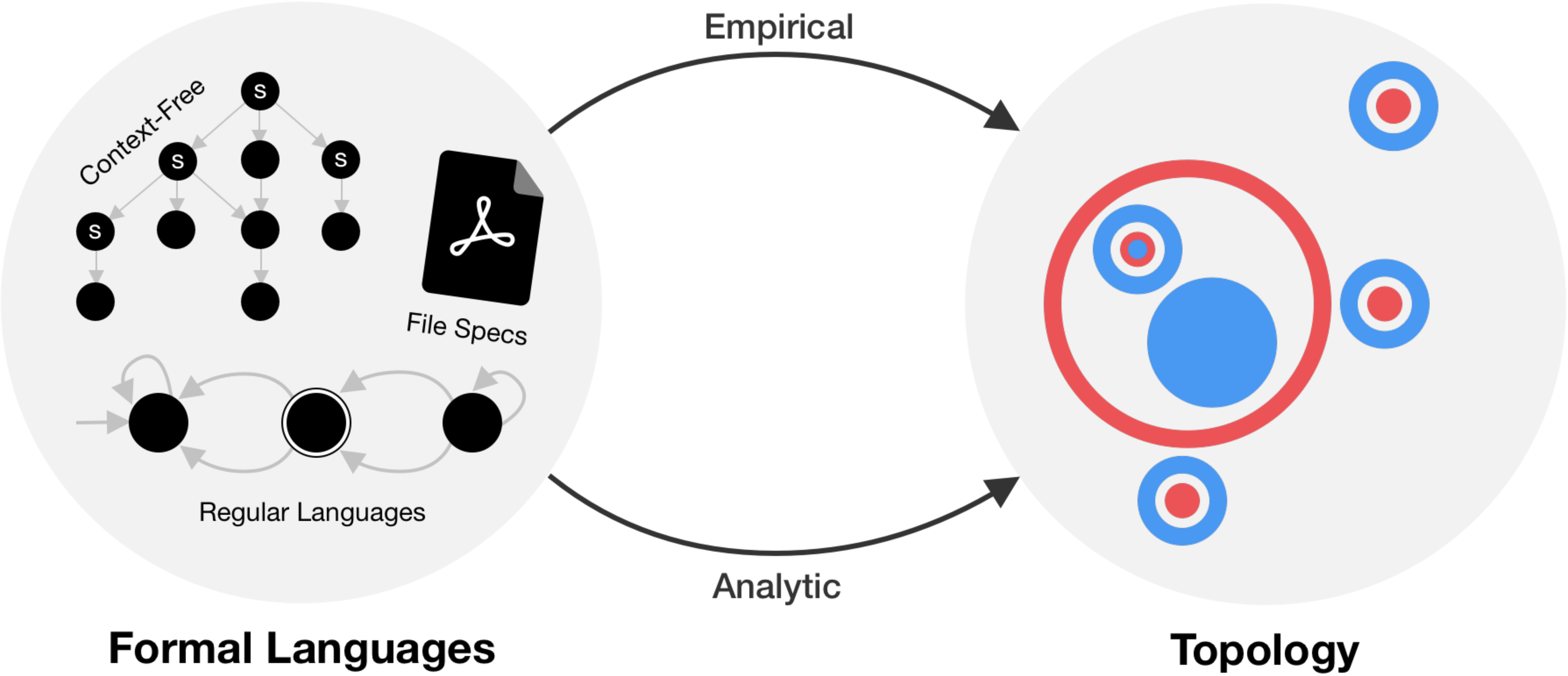
Neural Homology

Basic Definitions (Guss et al., 2018)

- **Definition.** (*Homeomorphism*, *Informal*). *We say that two topological spaces X and Y are homeomorphic ($X \cong Y$) if there is a continuous function $f : X \mapsto Y$ that has an inverse f^{-1} that is also continuous.*
- **Definition.** *(Homology, Informal). Let β_n be the nth Betti number, that counts the number of 'holes' of dimension n in X. Then, if X is a topological space, $H_n(X) = Z^{\beta_n}$ is called the nth homology group of X.*
- *We write $H(X) = (H_n(X))_{i \geq 0}$ to describe the homology of a set X.*
- **Definition.** *(Topological Complexity)* The topological complexity is the sum

$$\omega(X) = \sum_i \beta_i(X).$$

A Two-Pronged Approach



Analytic

Analytic

Algebraic Group Refresher

- A *group*, G, is a set of elements with a unique “identity”, 1, and an operation, $*$, that satisfies:
  - **Closure:** If $a, b \in G$ then $a * b \in G$
  - **Associativity**: $a * b * c = (a * b) * c = a * (b * c)$
  - **Identity Element:** For every $a \in G, 1 * a = a * 1 = a$
  - **Inverse Elements:** If $a \in G$, there exists an element $a^{-1} \in G$ so that $a * a^{-1} = a^{-1} * a = 1$

Analytic

Free Groups

- **Definitions**:
  - S is a set of "generators" or "symbols"
  - Define $\hat{S} := S \cup S^{-1}$ where $S^{-1} = \{a^{-1} | a \in S\}$
  - Define $G = \hat{S}* \cup \{1\}$, Kleene Closure (i.e., the set of all finite strings based on $\hat{S}$) with 1.
  - Associative and $a * a = a^2$ etc. and $a * a^{-1} = 1$.
- **Examples**:
  - Integers, $\mathbf{Z}$, is the free group based on one generator with addition operator and identity, 0.
  - Free group with two generators, a, b is the set of strings of the form:

$$a^{n_1}b^{n_2}a^{n_3}\cdots b^{n_k}, n_j \in \mathbf{Z}\,.$$

Analytic

Finitely Presented Groups

- **Definition:** $G = \langle S \,|\, R \rangle$, S are generators and R is a finite number of relations equal to the identity.
- **Examples:**
  - $G = \langle a \,|\, a^n \rangle$ is the cyclic group of order n.
  - $G = \langle a, b \,|\, aba^{-1}b^{-1} \rangle$ is the free Abelian group with two generators (i.e., $\mathbf{Z}^2$, the 2-D integer lattice).
  - $G = \langle a, b \,|\, a^n, b^m, aba^{-1}b^{-1} \rangle$ is the product of cyclic groups of order n and m.

Analytic

The Group Word Problem (Dehn 1911)

- **Group Word Problem:** Given a finitely presented G and a word, $w \in \hat{S}*$, is $w = 1$?
- **Group Word Language:** What kind of language is $L = \{w \,|\, w = 1\}$?
- **General Results:**
  - L is regular iff G is finite (Animisov 1971, uses pumping lemma);
  - L is context-free iff G is *virtually free* (Muller & Schupp, complex).
  - *Virtually free* means it has a proper subgroup of finite index (finite number of cosets) i.e., free apart from some finite number of structural relationships.

Analytic

Group Presentation vs CFG Production Rules

- **Group Presentation:** Relations simplify/shorten strings
- **CFG Productions:** Productions build up strings.

Analytic

Homotopy I

Continuously deform curves and surfaces…

The first homology group is the "Abeleanized" first homotopy group.

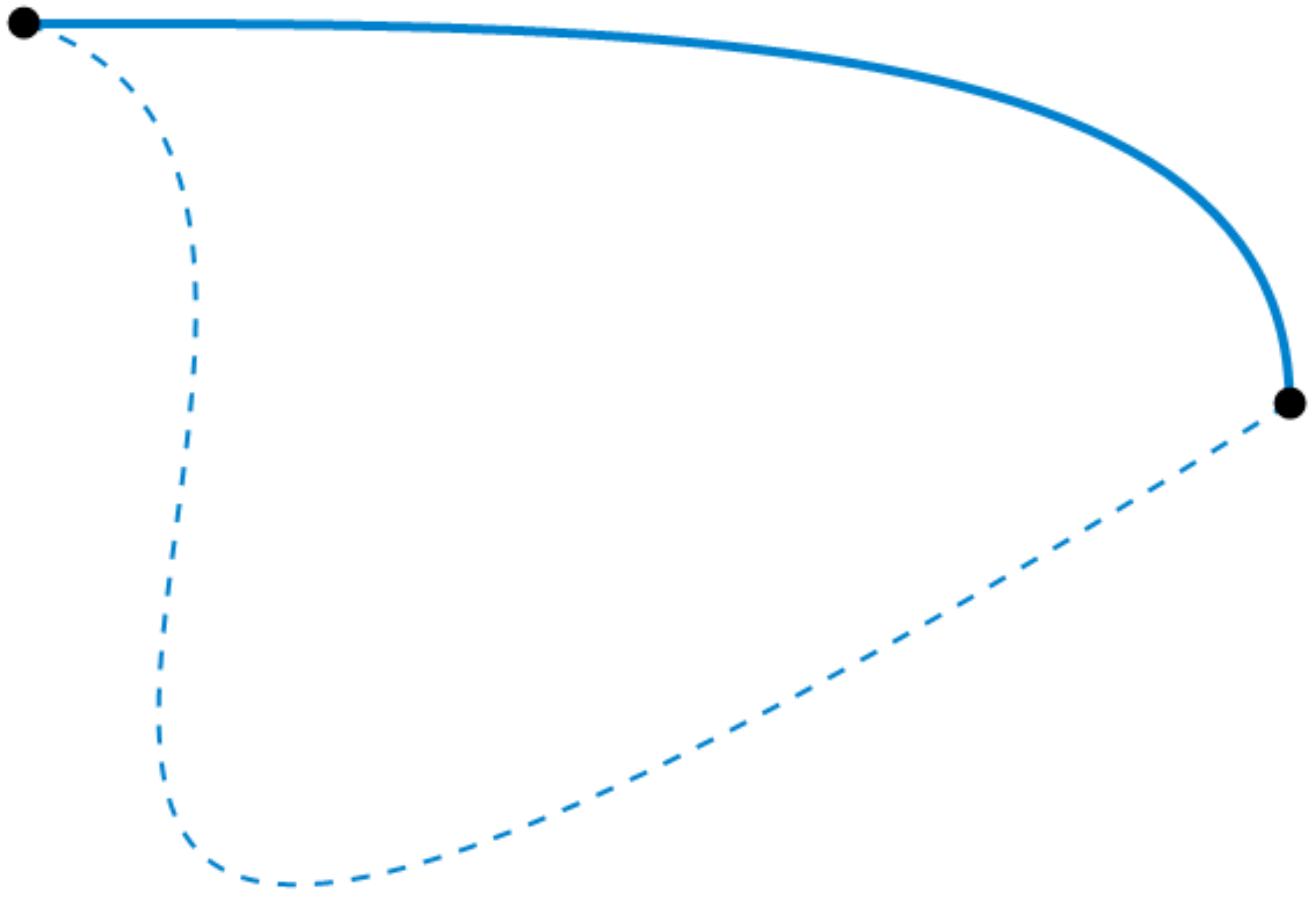


Image from https://en.wikipedia.org/wiki/Homotopy

Analytic

Homotopy II

Continuously deform curves and surfaces...
So a coffee cup and a donut are “Homotopically Equivalent”

Image from https://en.wikipedia.org/wiki/File:Mug_and_Torus_morph.gif

Analytic

Homotopy III

Consider an annulus in 2-D. These two curves are NOT “homotopically equivalent” because the green curve goes around the hole but the blue one does not.

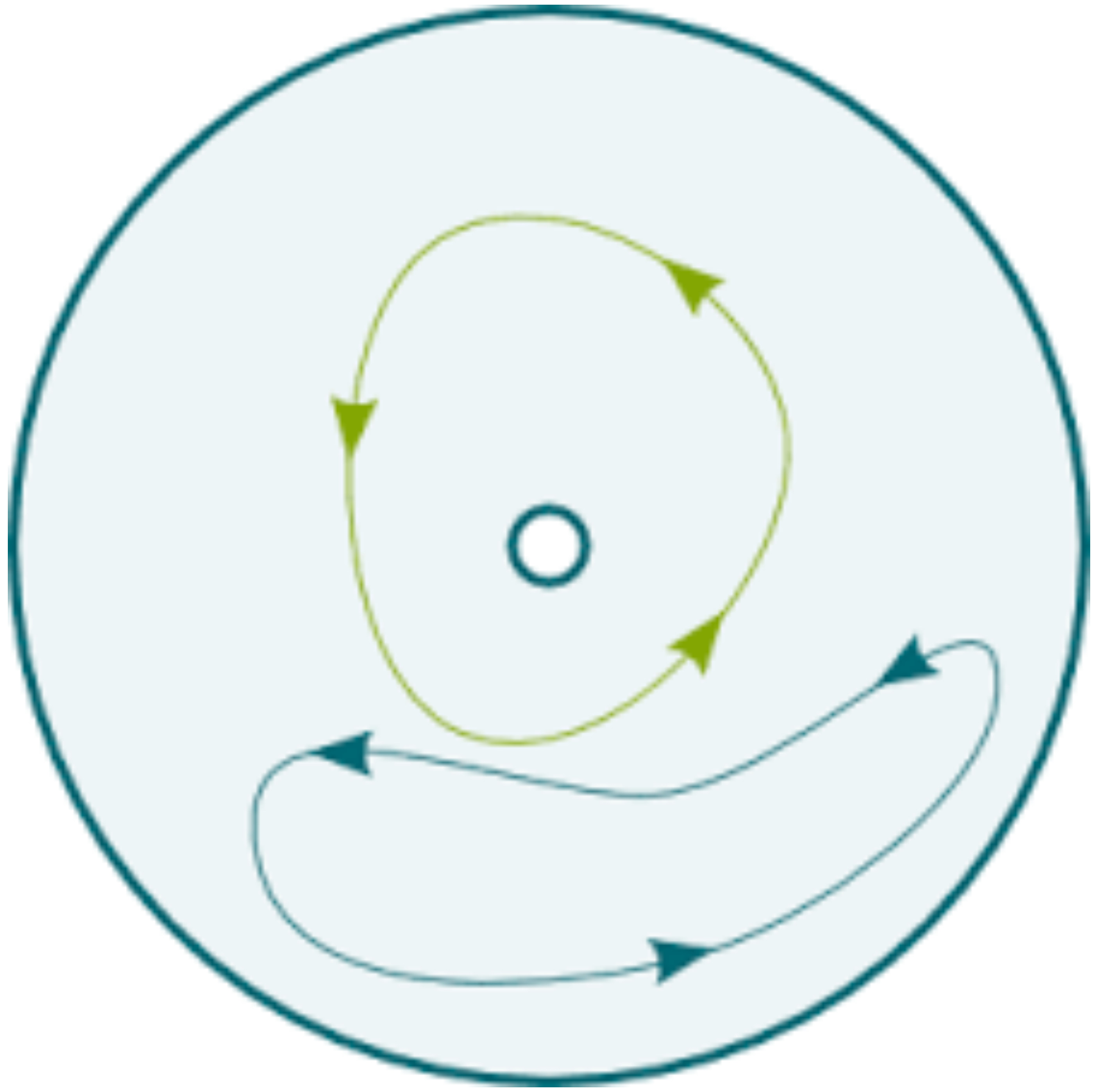


Image from http://www.science4all.org/article/homotopy-type-theory/

Analytic

Dyck Languages

- **Dyck Language:** Given a bipartite set of characters $(P, \bar{P})$, the Dyck language, $\mathscr{D}_P$, is defined by the set,

$$\mathscr{D}_P = \{x \in (P \cup \bar{P})^* \mid x \text{ is a well balanced set of parenthesis}\}.$$

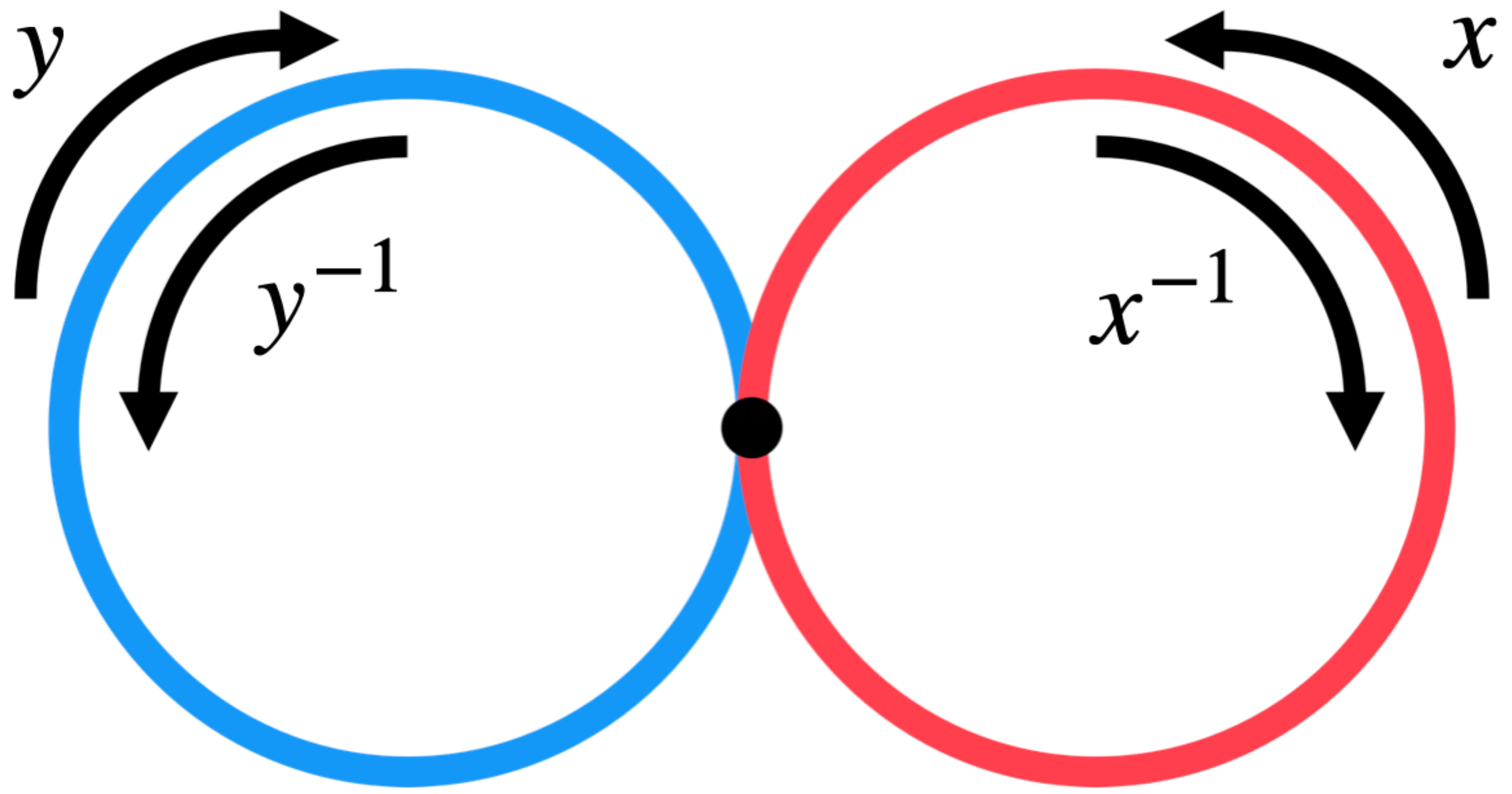
We write $\mathscr{D}_n$ is short hand for a Dyck language with n bracket types.

- *Observation:* $\mathscr{D}_k \approx$ the word problem language for a free group with k generators.
- *Observation:* A free group with k generators is the homotopy group for a k-petal. The word problem asks which closed curves are equivalent to the identity.

Analytic

Dyck Languages

- $\mathscr{D}_2 \approx$ in the homotopy group of a figure 8. We expect it to be embedded as a simply-connected component in some topological space. i.e., paths can be contracted to a point.



This is a 2-petal – a figure 8.

Empirical

Empirical
Goals

- As we saw earlier, previous work has shown that there is a deep connection between learnability and the topological complexity of the dataset.
- From this, one could say there is a hierarchy of datasets, each harder to learn than the last due to the topological complexity of the dataset.
- Formal languages also have hierarchy and complexity, and broadly speaking we would like to see how much these perspectives overlap.

Empirical

Datasets

- Random strings of length up to 50 of each of the following languages[1]:
  - $\mathscr{D}_1$ and $\mathscr{D}_2$ languages
  - Tomita Languages:
    - *Tomita 1:* $1*$;
    - *Tomita 2:* $(10)*$;
    - *Tomita 3:* All strings without $1^{2\ell+1} + 0^{2k+1}$ as a substring;
    - All strings without 000 as a substring.

[1] Datasets From *On the Ability and Limitations of Transformers to Recognize Formal Languages.*

Empirical

Dataset Motivation

- $\mathscr{D}_1$ and $\mathscr{D}_2$ languages are fundamental to context-free languages per the Chomsky-Schützenberger Theorem.
- Paraphrased this theorem says every context-free language is basically a version of a $\mathscr{D}_k$ language intersected with a regular language.

Image from https://en.wikipedia.org/wiki/Dyck_language

Empirical

Dataset Motivation

- In contrast to the more ‘complex’ Dyck languages, the Tomita languages are simple, allowing us to compare topological complexity across coarse classes in the Chomsky Hierarchy.
- Further, their minimal automata are known, allowing us to test the possibility that homology can capture a finer-grained notion of complexity.

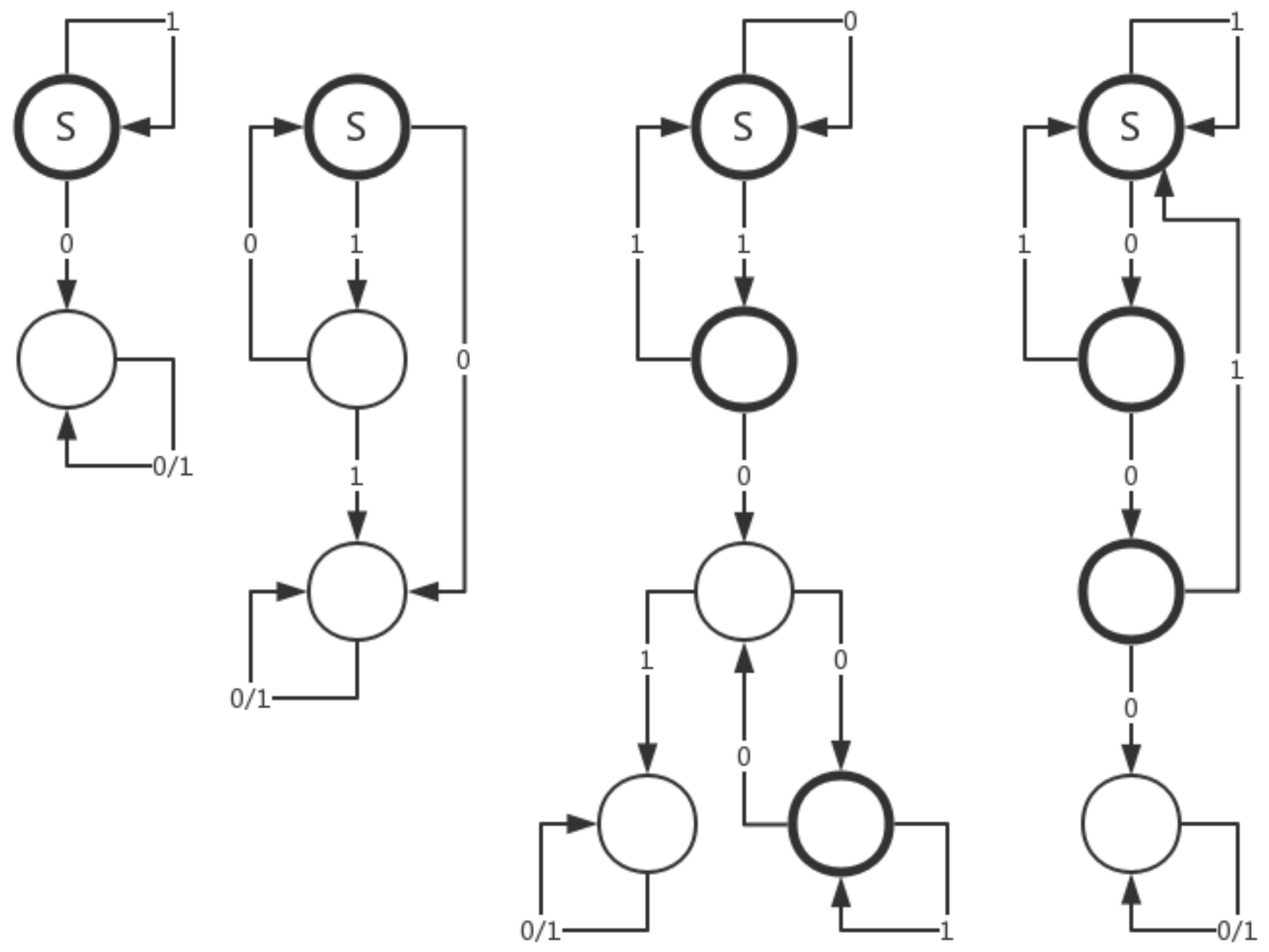
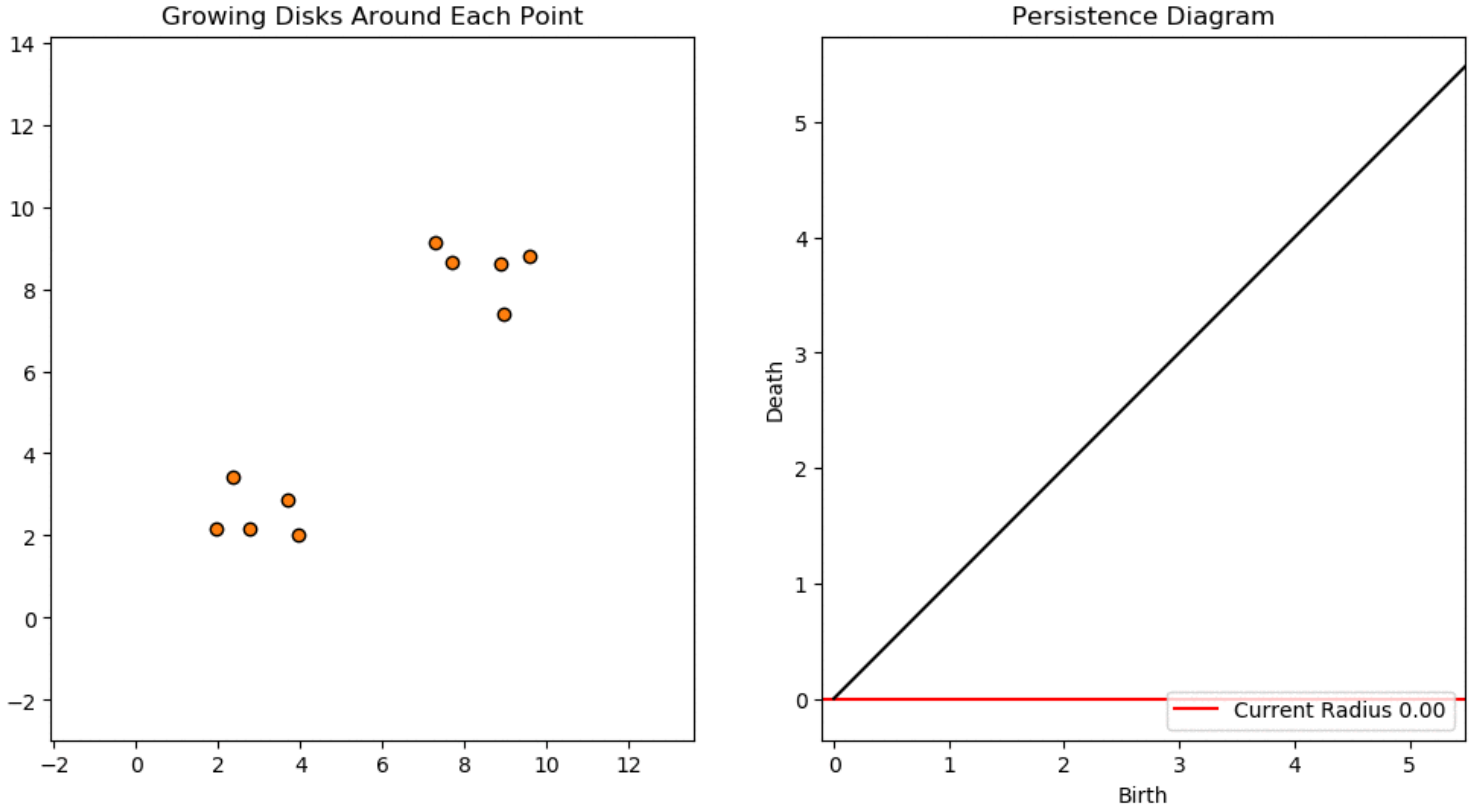


Image from https://github.com/LC-John/RNN2DFA

Empirical

Setup



Roughly speaking, persistent homology discovers the number of connected components, loops etc., by building growing balls of increasingly large radii around each point, and tracking when features like loops are *born* and when they *die*.

 https://towardsdatascience.com/persistent-homology-with-examples-1974d4b9c3d0

Empirical

Setup

- Unfortunately, computing persistent homology in high dimensions is computationally expensive.
- As such, we need to embed the data into a lower dimensional space to be able to compute topological complexity.
- We do this with locally linear embeddings (LLEs), as LLEs with low reconstruction error approximately preserve dataset homology.

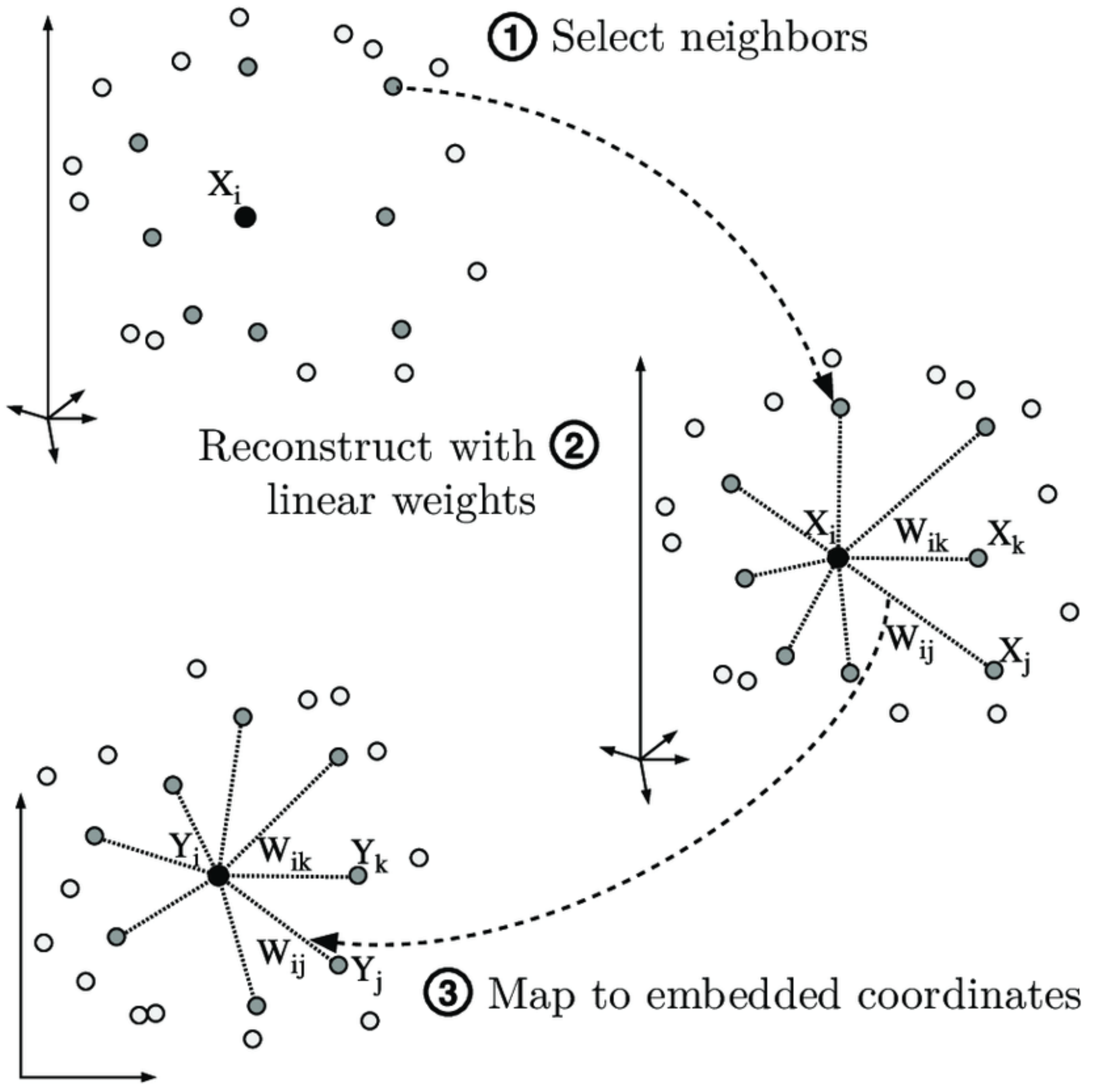
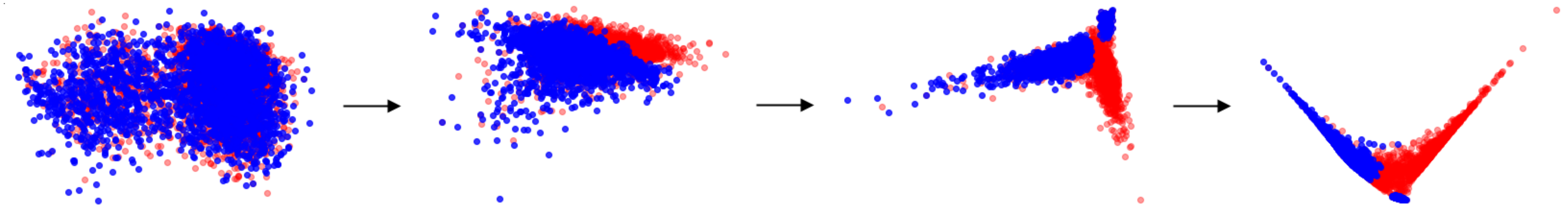


Diagram from *Memory Organization for Invariant Object Recognition and Categorization*

Empirical

Process

1. Generate a dataset of random strings from a formal language of choice.
2. Train a neural network on the dataset X.
3. For each hidden layer, $\mathbf{h}_i$, of the network,
   - Embed $(\mathbf{h}_i \circ \cdots \circ \mathbf{h}_1)(X)$ using LLE to get $\tilde{X}_i$.
   - Run persistent homology on $\tilde{X}$ to get persistence diagram $\mathscr{I} := \left\{[b_t, d_t]\right\}_t$;
   - Calculate $\beta_k^{(\epsilon)} = \left| \left\{[b_t, d_t] \in \mathscr{I} : |b_t - d_t| \geq \epsilon\right\} \right|$ for $k = 0,1$ and $\epsilon = 0.015, 0.025, 0.05$;
   - Compute $\omega(\tilde{X}_i) = \sum_j \beta_j \left(\tilde{X}_i\right)$.



This figure illustrates the simplification of topological structure of a Dyck-2 language at four intermediate layers of our trained feedforward neural network. Blue dots are in the language and red dots are not. The Dyck-2 words are of length 1-50 so this is a projection of high dimensional Euclidean spaces to two dimensions for visualization purposes, using locally linear embedding (LLE) for the projections. Note that at the final layer displayed, the dataset is nearly linearly separable as a result of the nonlinear network layers' transformations.

Empirical

Results

	Scale	$\omega(\mathcal{X})$	$\omega(h_1(\mathcal{X}))$	$\omega(h_2(\mathcal{X}))$	$\omega(h_3(\mathcal{X}))$	$\omega(h_4(\mathcal{X}))$	$\omega(h_5(\mathcal{X}))$	$\omega(h_6(\mathcal{X}))$
$\mathcal{D}_1$ Dataset	0.015	374	539	775	427	284	68	32
	0.025	100	47	115	75	55	19	12
	0.050	17	3	9	10	8	4	3
$\mathcal{D}_2$ Dataset	0.015	558	537	495	298	79	47	32
	0.025	104	51	56	54	25	15	12
	0.050	18	9	11	9	5	4	5
Tomita 1	0.015	48	110	182	287	238	81	70
	0.025	18	36	105	176	134	34	47
	0.050	2	2	34	35	19		10
Tomita 2	0.015	60	143	168	242	12	99	38
	0.025	24	47	90	144	5	37	27
	0.050	9	9	37	30	3	10	14
Tomita 3	0.015	44	109	828	825	654	583	197
	0.025	7	11	224	145	55	56	24
	0.050	0	3	8	7	4	5	3
Tomita 4	0.015	331	591	595	825	233	60	45
	0.025	102	118	56	131	22	10	12
	0.050	25	12	6	9	0	2	3

The coarse complexity of the datasets (according to the Chomsky hierarchy) is correctly expressed in these measurements at $\epsilon := 0.015$, However automata state complexity is not, as the Tomita 4 language has smaller minimal state-complexity than the Tomita 3 language.

Empirical

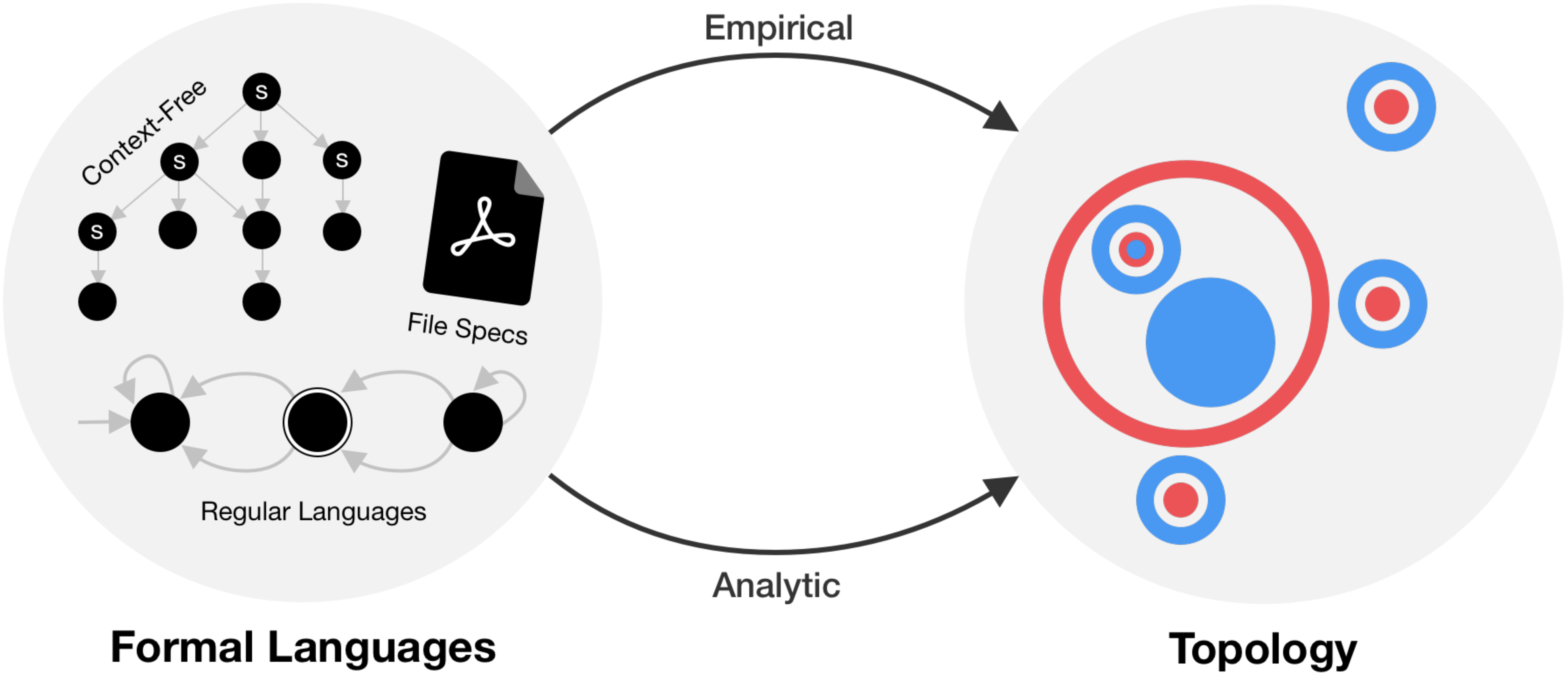
Interpretation

- These results are empirical — the datasets are not all encompassing and randomly generated.
- Further neither LLE nor persistent homology performs well with outliers.
- All considered, these results are promising, as the coarse relationships between classes represented in the Chomsky Hierarchy take shape.

Conclusion

Summary

We are exploring both directions in parallel. Each approach provides different insights about the relationship between formal languages, topology and neural networks.



Future Work

Empirical

- Perform more exhaustive experimentation with more advanced design to account for the disclaimers mentioned on the last slide.
- Extend these insights towards other architectures.
- Apply this complexity to measure generative models.

Analytic

- Study relationship between Chomsky-Schützenberger Theorem and virtually free groups.
- Every context-free language is the word problem for a finitely presented group if and only if the group is virtually free (i.e., an infinite subgroup with finite number of cosets or finite factor group if the subgroup is normal).

Acknowledgments

- Motivated by DARPA SafeDocs Program
  - `https://www.darpa.mil/program/safe-documents`
  - Dr. Sergey Bratus, Program Manager
- Joshua Ackerman, Dartmouth CS PhD student (coauthor)
- Prime Contractor, SRI - Natarajan Shankar, PI
- SafeDocs AI/ML Interest Group
  - Letitia Li, BAE Systems
  - Michael Robinson, American University
  - Mikael Vejdemo-Johansson, Special Circumstances
  - Walt Woods, Galois

Details of Our Work

- J. Ackerman and G. Cybenko, "Formal Languages, Deep Learning, Topology and Algebraic Word Problems", to appear in Proceedings of LangSec May 2021 (IEEE).
- J. Ackerman and G. Cybenko, "A Survey of Neural Networks and Formal Languages." arXiv preprint arXiv:2006.01338 (2020).

Background Material

- Epstein, David BA. Word processing in groups. CRC Press, 1992.
- Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. "Automata theory, languages, and computation." International Edition 24.2 (2006).
- Edelsbrunner, Herbert, and John Harer. "Persistent homology — a survey." Contemporary mathematics 453 (2008): 257-282.
- Dold, Albrecht. Lectures on algebraic topology. Springer Science & Business Media, 2012.
- Guss, W. H., and Salakhutdinov, R. (2018). On characterizing the capacity of neural networks using algebraic topology. arXiv preprint arXiv:1802.04443.

Questions?

joshua.m.ackerman.gr@dartmouth.edu | gvc@dartmouth.edu

End.