On the Statistical Query Complexity of Learning Semiautomata: a Random Walk Approach

George Giapitzakis GGIAPITZ@UWATERLOO.CA
University of Waterloo

Kimon Fountoulakis KIMON.FOUNTOULAKIS@UWATERLOO.CA
University of Waterloo

Eshaan Nichani ESHNICH@PRINCETON.EDU
Princeton University

Jason D. Lee JASONDLEE@BERKELEY.EDU
University of California, Berkeley

Editors: Steve Hanneke and Tor Lattimore

Abstract

Semiautomata form a rich class of sequence-processing algorithms with applications in natural language processing, robotics, computational biology, and data mining. We establish the first Statistical Query hardness result for semiautomata under the uniform distribution over input words and initial states. We show that Statistical Query hardness can be established when both the alphabet size and input length are polynomial in the number of states. Unlike the case of deterministic finite automata, where hardness typically arises through the hardness of the language they recognize (e.g., parity), our result is derived solely from the internal state-transition structure of semiautomata. Our analysis reduces the task of distinguishing the final states of two semiautomata to studying the behavior of a random walk on the group $S_N \times S_N$. By applying tools from Fourier analysis and the representation theory of the symmetric group, we obtain tight spectral gap bounds, demonstrating that after a polynomial number of steps in the number of states, distinct semiautomata become nearly uncorrelated, yielding the desired hardness result.

Keywords: automata learning, semiautomata, statistical query learning, lower bounds

1. Introduction

We study the computational complexity of learning semiautomata with N states in the Statistical Query (SQ) model. Automata are among the most basic models of computation and arise across formal methods (Clarke et al., 2001; Isberner, 2015; Holzmann, 2011), natural language processing (Mohri, 1997; Maletti, 2017; Koskenniemi, 1983; Mohri, 1996), robotics and control systems (Kress-Gazit et al., 2009; Chatzilygeroudis et al., 2020), computational biology (Sakakibara, 2005; Tsafnat et al., 2011; de la Higuera, 2010), and data mining (Laxman and Sastry, 2006). On the learning-theoretic side, the SQ model captures a broad class of noise-tolerant algorithms (Kearns and Vazirani, 1994), and its modern relevance is highlighted by its connections to the power of gradient-based methods (Abbe et al., 2021).

Motivation: structural vs. language-based hardness. We investigate hardness results for semiautomata in the SQ model. Prior work on SQ hardness has focused primarily on deterministic finite automata (DFAs), that is, semiautomata with a designated initial state and a set of accepting states.

While this distinction may seem subtle at first, it is crucial from a learnability perspective. For DFAs, hardness arises from the complexity of the languages they recognize. Typically, SQ hardness is established by embedding the computation of a hard function, such as parity (Blum et al., 1994), into a (random) DFA. Moreover, hardness results often rely on an adversarial input distribution (Angluin et al., 2010), instead of the more natural uniform distribution. In contrast, semiautomata are purely state-transition systems. As a result, SQ hardness for semiautomata must come exclusively from the structure of the transitions themselves, namely, from families of transition functions whose induced state-evolution statistics remain pairwise close under the uniform distribution over the input and initial state.

Contributions. We construct a family of $N!$ semiautomata with N states and an alphabet of size $\Omega(N^3 \ln N)$ which are nearly uncorrelated after processing inputs of length $\Omega(N^2 \ln N)$, yielding a statistical dimension of $N!$. Consequently, any SQ learner for that class under the uniform distribution over the input and initial state must either (i) make a super-polynomial number of queries or (ii) use a super-polynomially small tolerance. Our analysis introduces a representation-theoretic toolkit for automata learning: we tie semiautomata processing of random words to the mixing of a random walk on the product of the symmetric group $S_N \times S_N$, and identify a specific irreducible representation that controls indistinguishability.

We propose a randomized construction such that any pair of semiautomata from a randomly generated family of size $N!$ is indistinguishable with high probability under the uniform distribution over input words and starting states. In our random construction, the letters of the alphabet are matched to transpositions and assigned to each semiautomaton by fair coin flips. We reduce the problem of distinguishing two semiautomata that read the same random input to tracking a single coupled random walk on $S_N \times S_N$. In our construction, the probability that two semiautomata agree after starting from the same random initial state and processing T symbols has the form

$$P_{\text{agree}}(T) = \frac{1}{N} + \text{error}(T, N) \quad \text{with} \quad |\,\text{error}(T, N)| \le \left(1 - \frac{1}{2N}\right)^T,$$

with probability at least $1 - \exp(-N \ln N)$. Thus, the absolute error in the agreement beyond the baseline $1/N$ decays exponentially fast as a function of T. Taking $T \ge \mathcal{O}(N^2 \ln N)$ drives the absolute error to $1/N!$, so any two semiautomata are statistically indistinguishable on polynomial-length inputs. Because we build a family of $N!$ nearly uncorrelated semiautomata, the statistical dimension is $N!$, yielding the desired SQ lower bound. Unlike previous hardness results for DFAs, which rely on reductions to hard function classes such as parity (Angluin et al., 2010), our hardness result is inherently structural. We model the behavior of semiautomata processing a random word as a random walk on the group $S_N \times S_N$ and analyze its mixing properties by building off the representation-theoretic framework for random walks on S_N developed by Diaconis and Shahshahani (1981). We summarize our main contribution with the following informal statement of the hardness result:

Theorem 1 (Informal version of Theorem 16) *The class of semiautomata with N states operating on an alphabet of size $\mathcal{O}(N^3 \ln N)$ and processing inputs of length $\mathcal{O}(N^2 \ln N)$ is hard to learn in the Statistical Query model under the uniform distribution over input words and initial states*

Finally, as a result of independent interest, we prove that the mixing time for the random walk of our random family construction is tight. In particular, the best-case mixing time for our random

construction is $T = \Omega(N^2 \ln N)$. In other words, any smaller choice of T cannot guarantee the indistinguishability bound we obtain. This follows naturally as a byproduct of our proof strategy for the main result.

Organization. The rest of the paper is organized as follows. In Section 2, we provide a review of the relevant literature on the learnability of automata. Section 3 establishes the notation and necessary mathematical preliminaries. The core of our technical approach is detailed in Section 4, where we connect the problem of distinguishing semiautomata to the mixing properties of a random walk on the product group $S_N \times S_N$. In Section 5, we leverage this connection to present a randomized construction of a large family of nearly uncorrelated semiautomata. Finally, in Section 6, we use this construction to establish a high Statistical Query dimension, proving our main hardness result.

2. Related work

Learnability of automata is a foundational problem in computational learning theory, with a rich history spanning several decades (Trakhtenbrot and Barzdin, 1973; Kearns and Valiant, 1994; Angluin et al., 2010; Wang et al., 2025). To the best of our knowledge, no learnability results are currently known for the class of semiautomata. In contrast, the literature on DFA learnability is extensive. In this section, we restrict our review to results on DFA learning within the SQ model. For related work outside the SQ model, we refer the reader to Appendix A.

2.1. The Statistical Query model

The SQ model, introduced by Kearns (1998), formalizes a large and natural class of algorithms that learn by requesting statistical properties of the data distribution rather than inspecting individual examples. A key motivation for the model is its inherent robustness to noise; any SQ-learnable class is also PAC-learnable (Kearns, 1998). The computational complexity of learning within this model was elegantly characterized by Blum et al. (1994) using Fourier analysis, who introduced the concept of the SQ dimension to provide a combinatorial measure for proving lower bounds. This model has been successfully used to prove the hardness of learning for many important concept classes, including noisy linear threshold functions (Blum and Frieze, 1996), halfspaces under Gaussian marginals (Diakonikolas et al., 2020; Hsu et al., 2022), high-dimensional Gaussians (Diakonikolas et al., 2017), and single-neuron networks (Goel et al., 2020), with some problems remaining open (Feldman, 2014). The relevance of the SQ model has been underscored by work connecting it to other constrained learning models, such as those with memory or communication limits (Steinhardt et al., 2016), and, most notably, to the capabilities of modern deep learning algorithms. Abbe et al. (2021) showed that learning with stochastic gradient descent (SGD) is equivalent in power to SQ learning when using large mini-batches or low-precision gradients, suggesting our hardness result has direct implications for a wide range of practical algorithms. For a comprehensive overview of the SQ model see the survey by Reyzin (2020). Our work contributes a new, SQ hardness result for a classic concept class (semiautomata), but does so by introducing novel representation-theoretic tools to establish a high SQ dimension.

2.2. Related DFA hardness results

One of the simplest SQ hardness results can be established from the work of Blum et al. (1994) on the hardness of the parity function. In particular, the existence of a DFA with $2N + 1$ states that can compute the parity of a fixed subset of any binary string of length N implies that the class of $2N+1$-state DFAs is hard to learn using Statistical Queries under the uniform distribution on inputs of length N.

On a related note, Angluin et al. (2010) demonstrate that random DNF formulas, random log-depth decision trees, and random DFAs (where both transitions and accepting states are chosen uniformly at random) cannot be weakly learned with a polynomial number of statistical queries if the distribution over the input is chosen adversarially. The paper argues that even if a structure (such as a DNF formula or a DFA) is chosen randomly, an adversary can construct a "bad" input distribution that makes the learning problem hard. The core technique for establishing hardness is to show that, with high probability, a hard-to-learn parity function can be embedded into the random structure. The learnability of random DFAs using Statistical Queries under the uniform input distribution was explicitly posed as an open problem by Angluin et al. (2010). Similarly, Fish and Reyzin (2017) conjectured that learning random DFAs is hard in the PAC model. Although our work considers a slightly different random semiautomata model, with transitions generated by random transpositions, we hope it can serve as a first step toward addressing these open problems.

The last negative result, which is also quite relevant to our work, is that of Wang et al. (2025). The paper establishes a computational hardness result for learning the class of k-fold composition functions in the SQ model. The k-fold composition task requires a model to compute the result of an interleaved composition of k permutations provided in the input context and k hidden, parametric permutations. This model can be interpreted as a restricted subclass of semiautomata with time-inhomogeneous transitions. The proof in Wang et al. (2025) relies heavily on the specific structure of the hypothesis class and mostly employs recursive algebraic calculations and combinatorial facts. Our result complements that of Wang et al. (2025) and is obtained by framing the problem as a random walk on the product of symmetric groups and by leveraging tools from Fourier analysis and representation theory.

3. Notation and preliminaries

Throughout the text, we use boldface to denote vectors. We reserve the notation $\boldsymbol{e_i}$ to denote the i-th standard basis vector. For $n \in \mathbb{N} := \{1, 2, \dots\}$ we use the notation $[n]$ to refer to the set $\{1, 2, \dots, n\}$. For a vector $\boldsymbol{x} \in \mathbb{C}^n$ we denote by $\|\boldsymbol{x}\| := \sqrt{\sum_{i=1}^n |\boldsymbol{x}_i|^2}$ its Euclidean norm. For matrices A and B, $A \otimes B$ denotes their Kronecker product and $A \oplus B$ their direct sum. For a matrix A, we denote by $\|A\|_2 := \sup_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|$ its spectral norm. For a finite set S, we denote by $\mathcal{U}(S)$ the uniform distribution over its elements.

3.1. Abstract algebra

We direct unfamiliar readers to Appendix B, which provides a self-contained overview of the necessary results from the representation theory of finite groups, and constitutes the foundation for our analysis. For a group G, we denote by id its identity element. When G and H are groups, $G \times H$ denotes their direct product. We denote the symmetric group on N elements by S_N. For a permutation

$g \in S_N$, we denote by $\mathrm{fix}(g)$ the set of fixed points of g. The trivial and standard representations of S_N are denoted by triv and std, respectively. For a representation ρ and a function $f : G \to \mathbb{C}$, the Fourier transform of f with respect to ρ is given by $\rho(f) := \sum_{g \in G} f(g)\rho(g)$. For representations ρ and σ, $\rho \otimes \sigma$ denotes their tensor product.

3.2. Semiautomata

A semiautomaton $\mathcal{A}$ is a triplet $(\mathcal{Q}, \Sigma, \delta)$, where

- $\mathcal{Q}$ is a non-empty set of states.
- Σ is the input alphabet.
- $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is the transition function.

We denote by $\mathcal{A}(\Sigma, \mathcal{Q})$ the set of semiautomata with alphabet Σ and state space $\mathcal{Q}$. Throughout this work, we will only consider semiautomata where both Σ and $\mathcal{Q}$ are finite, and we denote by $N = |\mathcal{Q}|$ the number of states. The set of all finite words over the alphabet Σ is denoted by Σ^*. For each symbol $a \in \Sigma$, we define the state-transition function $\delta_a : \mathcal{Q} \to \mathcal{Q}$ by $\delta_a(X) = \delta(X, a)$. Given a semiautomaton $\mathcal{A}$, we define the function $f_{\delta_\mathcal{A}} : \Sigma^* \times \mathcal{Q} \to \mathcal{Q}$, which assigns to each word and initial state the corresponding final state. Specifically, for a word $\mathrm{w}_T := a_1 \dots a_T$ and an initial state X, the final state is given by

$$f_{\delta_\mathcal{A}}(\mathrm{w}_T, X) := (\delta_{a_T} \circ \delta_{a_{T-1}} \circ \cdots \circ \delta_{a_1})(X).$$

3.3. The Statistical Query model

The Statistical Query (SQ) model, introduced by Kearns (1998), is a restricted learning model where the learner accesses information about a dataset through statistical queries rather than individual examples. The learner interacts with a statistical query oracle whose definition is given below.

Definition 2 (Statistical Query oracle) *Let $\mathcal{C}$ be a concept class of functions mapping $\mathcal{X}$ to $\mathcal{Y}$, and let $f^* : \mathcal{X} \to \mathcal{Y}$ be a ground-truth concept in $\mathcal{C}$. Let D be a distribution over the input space $\mathcal{X}$. An SQ oracle, denoted by $\mathrm{STAT}(f^*, D)$, accepts a query (h, τ), where $h : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$ is a bounded statistical query function (statistic), and $\tau > 0$ is the tolerance. The oracle returns a value v such that:*

$$|v - \mathbb{E}_{x \sim D}[h(x, f^*(x))]| \leq \tau.$$

We say that an SQ learner *learns a target $f^* \in \mathcal{C}$ up to error $\varepsilon > 0$ under D* if, given access to $\mathrm{STAT}(f^*, D)$, it outputs a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ such that

$$\mathbb{P}_{x \sim D}\big[h(x) \neq f^*(x)\big] \leq \varepsilon.$$

We say that the learner learns $\mathcal{C}$ under D if it learns every $f^* \in \mathcal{C}$ up to arbitrarily small error.

Definition 3 (Statistical query hardness) *Statistical query hardness for $\mathcal{C}$ is established by showing that any SQ learner that learns $\mathcal{C}$ must, in the worst case over the choice of $f^* \in C$, either make a super-polynomial number of queries or use a tolerance τ such that $1/\tau$ is super-polynomial in the problem parameters. Equivalently, every SQ learner using polynomially many queries of inverse-polynomial tolerance fails to learn at least one target $f^* \in C$.*

4. The learning problem and random walk connections

The concept class $\mathcal{C}$ consists of all functions $f_{\delta_\mathcal{A}}$ that map an initial state and an input word to the corresponding final state according to some semiautomaton $\mathcal{A} \in A(\Sigma, \mathcal{Q})$, namely $\mathcal{C} = \{f_{\delta_\mathcal{A}} : \mathcal{A} \in \mathcal{A}(\Sigma, \mathcal{Q})\}$. With the notation of Section 3.3, $\mathcal{X} = \Sigma^* \times \mathcal{Q}$ and $\mathcal{Y} = \mathcal{Q}$. We investigate the hardness of learning $\mathcal{C}$ using statistical queries when the underlying distribution is uniform over input words of length T and initial states. We will show that when $|\Sigma|$ and T scale polynomially with the number of states N, learning $\mathcal{C}$ is hard.

4.1. Modeling single semiautomata as random walks

To prove our hardness result, we will need to construct a "hard" set: a large family of concepts from $\mathcal{C}$ that are nearly uncorrelated (details are presented in Section 6). We restrict our search for the hard set by considering semiautomata where the state-transition functions δ_a are permutations for each $a \in \Sigma$.[1] Under this restriction, we can model a single semiautomaton processing a uniformly random word as a random walk on the symmetric group S_N. Consider a semiautomaton $\mathcal{A}$ where all its state-transition functions δ_a are permutations from a set $\Gamma \subseteq S_N$.

1. *The Walk State:* The state of the random walk after t steps is the total permutation P_{w_t} computed so far. This is an element of the symmetric group S_N.

2. *The Starting State:* Before processing any symbols, the total permutation is the identity permutation, $\mathrm{id} \in S_N$, which maps every state in $\mathcal{Q}$ to itself.

3. *The Transition Rule:* Let $P_{\mathrm{w}_t} \in S_N$ be the permutation computed after processing the first t symbols of a (random) word $\mathrm{w}_T = a_1 a_2 \dots a_T$, and so, $P_{\mathrm{w}_t} = \delta_{a_t} \circ \cdots \circ \delta_{a_1}$. When the $(t+1)$-th symbol, a_{t+1}, is processed, the new walk state (total permutation) becomes $P_{\mathrm{w}_{t+1}} = \delta_{a_{t+1}} \circ P_{\mathrm{w}_t}$.

Under the uniform distribution assumption on the input word, the final state $f_{\delta_\mathcal{A}}(\mathrm{w}_T, X)$ coincides with $P_{\mathrm{w}_T}(X)$, for any initial state $X \in \mathcal{Q}$.

4.2. Modeling pairs of semiautomata as random walks

Our hardness result relies on evaluating a correlation measure between pairs of distinct semiautomata from the hard set. This requires bounding the probability that two such semiautomata end in the same final state when started from the same uniformly chosen input word and initial state. This can also be modeled as a (coupled) random walk on the product group $G = S_N \times S_N$. Let $\mathcal{A}$ and $\mathcal{A}'$ be two semiautomata (operating on the same alphabet Σ and state-space $\mathcal{Q}$) with permutation transition functions from a set $\Gamma \subseteq S_N$.

1. *The Walk Space:* To track both processes simultaneously, the walk state is a pair of permutations $(P_{\mathrm{w}_t}, P'_{\mathrm{w}_t})$, representing the total permutation computed by each semiautomaton after t steps. This state is an element of the product group $G = S_N \times S_N$.

2. *The Starting State:* Both walks start at the identity, so the starting state is $(\mathrm{id}, \mathrm{id}) \in G$.

1. In fact, our constructions will only require each δ_a to be either the identity or a transposition.

3. *The Transition Rule:* Let (P_t, P'_t) be the permutation computed after processing the first t symbols of a (random) word $\mathrm{w}_T = a_1 a_2 \dots a_T$. When the $(t+1)$-th symbol, a_{t+1}, is processed, the state of the joint walk transitions as follows:

$$(P_{\mathrm{w}_{t+1}}, P'_{\mathrm{w}_{t+1}}) = (\delta_{a_{t+1}} \circ P_{\mathrm{w}_t}, \delta'_{a_{t+1}} \circ P'_{\mathrm{w}_t}).$$

As before, the pair $(f_{\delta_{\mathcal{A}}}(\mathrm{w}_T, X), f_{\delta_{\mathcal{A}'}}(\mathrm{w}_T, X))$ coincides with $(P_{\mathrm{w}_T}(X), P'_{\mathrm{w}_T}(X))$ under the uniform distribution assumption on the input word, for any initial state $X \in \mathcal{Q}$.
The single-step probability distribution of this random walk is thus defined as follows:

Definition 4 (Single-step probability) *The probability distribution for a single step of the joint random walk defined in Section 4.2 is the function $T_{\mathrm{SA}} : G \to [0,1]$ given by:*

$$T_{\mathrm{SA}}(g) = \frac{|\{a \in \Sigma \mid (\delta_a, \delta'_a) = g\}|}{|\Sigma|} \quad \forall\, g \in G.$$

The support of T_{SA} is contained in $\Gamma \times \Gamma$.

As noted above, our hardness result requires a careful analysis of the probability of agreement for two semiautomata after T steps under a uniformly random input word and initial state. Namely, we are interested in the probability that for an initial state X_0 picked uniformly at random from $\mathcal{Q}$, after T steps of the random walk, the two states $P_{\mathrm{w}_T}(X_0)$ and $P'_{\mathrm{w}_T}(X_0)$ coincide. To calculate this, we borrow tools from Fourier analysis and representation theory of the symmetric group. As it turns out, this probability is controlled solely by the Fourier transform of T_{SA} with respect to the irreducible representation $\Pi_0 = \mathrm{std} \otimes \mathrm{std}$ of $S_N \times S_N$. The result, along with a proof outline, is presented in Theorem 5. The full proof relies on a technical lemma and is presented in Appendix C.

Theorem 5 (Agreement probability) *Consider the random walk corresponding to two semiautomata $\mathcal{A}$ and $\mathcal{A}'$ operating on the same alphabet Σ and state-space $\mathcal{Q}$, as described in Section 4.2. Let $N = |\mathcal{Q}|$ and let $T \in \mathbb{N}$ be the input length. Let $X_0 \sim \mathcal{U}(\mathcal{Q})$ be a (common) initial state picked uniformly at random. Denote by $P_{\mathrm{agree}} := P_{\mathrm{agree}}(T)$ the probability that after processing the same uniformly random word $\mathrm{w}_T \in \Sigma^*$, both semiautomata reach the same state. In terms of the random walk, this probability is given by*

$$P_{\mathrm{agree}}(T) = \mathbb{P}_{\mathrm{w}_T \sim \mathcal{U}(\Sigma)^{\otimes T}, X_0 \sim \mathcal{U}(\mathcal{Q})} \left(P_{\mathrm{w}_T}(X_0) = P'_{\mathrm{w}_T}(X_0) \right).$$

Then

$$P_{\mathrm{agree}} = \frac{1}{N} + \frac{1}{N} \boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v},$$

where M_{Π_0} is the Fourier transform of the single-step distribution T_{SA} with respect to the irreducible representation $\Pi_0 = \mathrm{std} \otimes \mathrm{std}$ of G and $\boldsymbol{v} = \sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}$.

Proof outline The proof begins by expressing the agreement probability, P_{agree}, as an expected value over the final states of the random walk on the product group $S_N \times S_N$. This is written as

$$P_{\mathrm{agree}} = \sum_{(g,h) \in G} P_T(g,h) \cdot \frac{|\operatorname{fix}(h^{-1}g)|}{N}, \tag{1}$$

where P_T denotes the distribution of the walk after T steps. The distribution P_T is then decomposed as a sum of the uniform distribution on $S_N \times S_N$ and an error term: $P_T(g,h) = P_U(g,h) + \mathrm{err}(g,h)$. Substituting this into Equation (1), we find that the uniform part of the sum evaluates cleanly to the baseline agreement probability of $1/N$. The remaining error term is then analyzed using the Plancherel formula from Fourier analysis (Theorem 36), and is given by the following sum over non-trivial irreducible representations:

$$\frac{1}{N} \cdot \frac{1}{|G|} \sum_{\Pi \neq \mathrm{triv}} d_\Pi \operatorname{Tr}\left(M_\Pi^T \Pi(f)^*\right),$$

where $M_\Pi = \Pi(P_T)$ and f is given by $f(g,h) = |\operatorname{fix}(h^{-1}g)|$. The key insight is that the Fourier transform of f is non-zero only for a single irreducible representation, $\Pi_0 = \mathrm{std} \otimes \mathrm{std}$ (Theorem 51). This collapses the sum above into a single term, yielding the final expression. ■

Theorem 5 is central to our analysis: it reduces the task of bounding P_{agree} for our constructions from having to account for all (exponentially many) irreducible representations of $S_N \times S_N$,[2] to considering just a single one. In the next section, we introduce a randomized construction used to generate a family of semiautomata and determine the required alphabet size $|\Sigma|$ and word length T that ensure $|P_{\mathrm{agree}} - 1/N| \leq 1/N!$, a bound necessary for establishing SQ-learning hardness.

5. Randomized construction of the hard set

In this section, we present a randomized construction of a family of semiautomata that will form the basis of our hard set, which will be used to derive the statistical query lower bound. In what follows, we assume that all semiautomata operate on the same, fixed set of N states. The construction depends on a parameter k that controls the size of the alphabet. We construct M semiautomata operating on a common alphabet consisting of k labeled copies of every transposition in S_N. For each semiautomaton i and symbol τ, we flip an independent fair coin to decide whether the state-transition function corresponding to τ acts as the identity or τ itself. Thus, each semiautomaton is defined by a random mask of swap and identity symbols. The formal definition is given below:

Definition 6 (Randomized (k, M)-shuffle family) *We construct a (random) family of semiautomata operating on the same alphabet with the following properties:*

1. *Let $\Sigma_1 = \{\tau_1, \ldots, \tau_{\binom{N}{2}}\}$ be an enumeration of all transpositions in S_N. The alphabet Σ_k is the disjoint union of k copies of Σ_1. The size is $|\Sigma_k| = k\binom{N}{2}$.*
2. *A family $\mathcal{F} = \{\delta_1, \ldots, \delta_M\}$ is generated by assigning to each symbol (transposition) $\tau \in \Sigma_k$ and each semiautomaton δ_i an independent random variable $b_i(\tau) \in \{0,1\}$ with $\mathbb{P}(b_i(\tau) = 1) = 1/2$. Each transition function $\delta_i : \mathcal{Q} \times \Sigma_k \to \mathcal{Q}$ is given by*

$$\delta_i(X,\tau) = \begin{cases} \tau(X) & \text{if } b_i(\tau) = 1 \\ X & \text{if } b_i(\tau) = 0 \end{cases} \quad \text{for all } X \in \mathcal{Q} \text{ and } \tau \in \Sigma_k.$$

2. The group $S_N \times S_N$ has $p(N)^2$ irreducible representations, where $p(N)$ is the partition function (c.f. Theorem 43). The partition function grows exponentially (Hardy and Ramanujan, 1918) and hence there are exponentially many irreducible representations.

For simplicity, we overload our notation to identify semiautomata by their transition functions.

We refer to the resulting family $\mathcal{F}$ as a randomized (k, M)-shuffle family.

Note that while Theorem 6 forces the alphabet to consist of transpositions, this is done only to discharge notation and enhance the readability of our proofs. Indeed, it is easy to see that the construction can work with any alphabet, so long as its size is $k\binom{N}{2}$, simply by appropriately remapping the alphabet characters to transpositions.

The main goal of this section is to find appropriate values for the alphabet parameter k and the input word length T such that, with high probability, any two semiautomata from a randomized $(k, N!)$-shuffle family $\mathcal{F}$ satisfy $P_{\text{agree}} \leq 1/N + 1/N!$. Given the form of P_{agree} derived in Theorem 5, the first step is to show that, with high probability, the spectral norm of M_{Π_0} is bounded away from 1 for any choice of distinct semiautomata from $\mathcal{F}$. This is done in Theorem 7, which we state and provide a proof outline here. The full proof is deferred to Appendix D.

Lemma 7 *Let $N \geq 4$ and consider a randomized (k, M)-shuffle family $\mathcal{F}$ with $M = N!$ and*

$$k \geq \frac{16(3N+1)}{3(N-1)}\left[N\ln N + \ln\binom{N!}{2} + 2\ln(N-1)\right].$$

Then any two distinct semiautomata $\delta_i, \delta_j \in \mathcal{F}$ satisfy $\left\|M^{i,j}_{\Pi_0}\right\|_2 \leq 1 - \frac{1}{2N}$ with probability at least $1 - \exp(-N\ln N)$ where $\Pi_0 = \text{std} \otimes \text{std}$ and $M^{i,j}_{\Pi_0}$ denotes the Fourier transform of the single-step probability distribution for the joint walk according to δ_i and δ_j.

Proof outline The proof follows a two-step "expectation-concentration" argument. First, we calculate the expectation of the Fourier transform matrix, $\mathbb{E}[M^{i,j}_{\Pi_0}]$, by averaging over all random choices in our semiautomaton construction. The canonical way by which randomness is introduced in the construction (through the fair coin toss), along with the use of transpositions as state-transition functions, ensures that the expectation has a manageable form. Namely, by applying a generalization of Schur's Lemma (Theorem 38), we show that the expected matrix is similar to a direct sum of scaled identity matrices, and that its spectral norm is $1 - \frac{1}{N-1}$. Second, we show that the matrix for a randomized $(k, N!)$-shuffle family concentrates around this expectation. We use the Matrix Bernstein inequality (Theorem 52) to prove that the deviation $\|M^{i,j}_{\Pi_0} - \mathbb{E}[M^{i,j}_{\Pi_0}]\|_2$ is very small with high probability, provided the alphabet size parameter k is sufficiently large. By combining the bound on the expectation's norm with the high-probability bound on the deviation via the triangle inequality, we arrive at the final desired bound of $1 - \frac{1}{2N}$. ■

Combining Theorem 7 with Theorem 5, we can directly derive an upper bound on P_{agree} that converges to $1/N$ exponentially in T.

Lemma 8 *For $M = N!$ and $N \geq 4$, if we choose the alphabet parameter*

$$k \geq \frac{16(3N+1)}{3(N-1)}\left[N\ln N + \ln\binom{N!}{2} + 2\ln(N-1)\right],$$

then any pair of semiautomata (δ_i, δ_j) *from a randomized* (k, M)*-shuffle family* $\mathcal{F}$ *satisfies*

$$\left| P_{\text{agree}} - \frac{1}{N} \right| \leq \left(1 - \frac{1}{2N}\right)^T$$

with probability at least $1 - \exp(-N \ln N)$.

Lastly, we show how to choose T to achieve the desired bound.

Theorem 9 *For* $T \geq 2N \ln(N!)$ *and* k *and* M *as given in Theorem 8, we have that any pair of distinct semiautomata* (δ_i, δ_j) *from a randomized* (k, M)*-shuffle family* $\mathcal{F}$ *satisfies* $|P_{\text{agree}} - 1/N| \leq 1/N!$ *with probability at least* $1 - \exp(-N \ln N)$.

The proofs of Theorem 8 and Theorem 9 are deferred to Appendix D. Notice that to achieve the desired bound, both k and T can be chosen to be polynomial in N. This is expressed in the following remark:

Remark 10 *Since* $\ln(N!) = \mathcal{O}(N \ln N)$*, the high probability result of Theorem 9 holds for a choice of* $T = \mathcal{O}(N^2 \ln N)$ *and* $k = \mathcal{O}(N \ln N)$*. In that case, the total alphabet size is* $\mathcal{O}(N^3 \ln N)$.

Note that choosing $b_i(\tau)$ as a $\text{Ber}(1/2)$ random variable is optimal. Intuitively, biasing the transitions toward either the identity or transpositions makes pairs of semiautomata easier to distinguish. As a result, a larger alphabet and longer inputs are needed to achieve indistinguishability. This is formalized in the following remark:

Remark 11 *In the randomized construction of Theorem 6, defining each* $b_i(\tau)$ *as an independent* $\text{Ber}(p)$ *random variable with* $p \in (0, 1)$ *requires an alphabet of size at least* $\mathcal{O}\left(\frac{N^3 \ln N}{p^2(1-p)^2}\right)$ *and an input word length of at least* $\mathcal{O}\left(\frac{N^2 \ln N}{p(1-p)}\right)$*. Both quantities are minimized when* $p = 1/2$*, showing that a fair coin flip is the most efficient choice for establishing hardness.*

As a result of independent interest, in Theorem 12 we prove that the choice of T above is asymptotically optimal. This essentially shows that to achieve the desired bound on P_{agree} with high probability, we must choose $T = \Omega(N^2 \ln N)$. In terms of the random walk, this shows that the best-case mixing time of the walk corresponding to the randomized (k, M)-shuffle family with k and M chosen as in Theorem 8 is $\Omega(N^2 \ln N)$. The proof of Theorem 12 follows easily by the analysis carried out in the proof of Theorem 7 and is deferred to Appendix E.

Theorem 12 (Tightness of mixing time) *Let* $N \geq 5$*, and let* k *and* M *be as in Theorem 8. For any pair of distinct semiautomata* (δ_i, δ_j) *from a randomized* (k, M)*-shuffle family to satisfy* $|P_{\text{agree}} - 1/N| \leq 1/N!$ *with probability* $1 - \exp(-N \ln N)$*, the input word length must be at least* $T = \Omega(N^2 \ln N)$.

6. Statistical Query hardness

This section is organized into two parts, with the goal of establishing the hardness of learning semiautomata in the Statistical Query framework. In Section 6.1, we introduce the necessary notions of pairwise correlation and SQ dimension, together with a lower bound theorem for SQ learning. These results are stated in a general setting for concept classes whose functions take finitely many values, thereby extending earlier results from the binary case. In Section 6.2, we apply those results to obtain our main hardness result for semiautomata.

6.1. General results

Throughout this section, we assume a general concept class $\mathcal{C}$ consisting of functions $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are abstract spaces with $|\mathcal{Y}| < \infty$. Likewise, we assume an arbitrary distribution D over $\mathcal{X}$. The definitions and results we present are straightforward extensions of the well-known binary case $\mathcal{Y} = \{0, 1\}$.

Definition 13 (Pairwise correlation) *Let $\mathcal{F}$ be a family of functions $f : \mathcal{X} \to \mathcal{Y}$. Let D be a distribution over $\mathcal{X}$. When $\mathcal{Y}$ is a finite set, the correlation between two distinct functions $f_i, f_j \in \mathcal{F}$ under D is defined in terms of the agreement probability as:*

$$\chi(f_i, f_j) = \mathbb{P}_{x\sim D}[f_i(x) = f_j(x)] - \frac{1}{|\mathcal{Y}|}.$$

SQ lower bounds are typically proven by constructing a large family of functions that are nearly uncorrelated, formalized by the concept of SQ dimension.

Definition 14 (SQ dimension) *Let $\mathcal{C}$ be a concept class over a distribution D. The statistical dimension $\mathrm{SQDim}_{\mathcal{C}}^{D}$ is the largest integer d such that there exists a subset $\{f_1, \ldots, f_d\} \subseteq \mathcal{C}$ of nearly uncorrelated functions satisfying $|\chi(f_i, f_j)| \leq 1/d$ for all $i, j \in [d]$ with $i \neq j$.*

The following theorem, whose proof is given in Appendix F, establishes a lower bound on the number of queries a learner must make in the worst case in terms of the SQ dimension:

Theorem 15 (SQ lower bound) *Let $\mathcal{C}$ be a concept class and suppose $\mathrm{SQDim}_{\mathcal{C}}^{D} \geq d$. Then any SQ learner using tolerance $\tau > 0$ requires, in the worst case, at least*

$$q \geq \frac{(d-1)(d\tau^2 - |\mathcal{Y}|)}{2d(|\mathcal{Y}| - 1)}$$

queries to learn the ground-truth concept f^*.

6.2. The case of semiautomata

We are now prepared to establish our main hardness result for the class of semiautomata. Recall that the concept class is defined as $\mathcal{C} = \{f_{\delta_{\mathcal{A}}} : \mathcal{A} \in \mathcal{A}(\Sigma, \mathcal{Q})\}$ and that the underlying distribution D_T on $\Sigma^* \times \mathcal{Q}$ is uniform over all states in $\mathcal{Q}$ and all words of length T in Σ^*. Denoting by $N = |\mathcal{Q}|$ the number of states, our hardness result can be stated as follows:

Theorem 16 *Suppose the alphabet size and word length satisfy* $|\Sigma| = \Omega(N^3 \ln N)$ *and* $T = \Omega(N^2 \ln N)$*, respectively. Then for all* $c > 0$*, any SQ learner for* $\mathcal{C}$ *under the distribution* D_T *must, in the worst case, either make* $\omega(N^c)$ *queries or use a tolerance of* $o(N^{-c})$*. Hence, learning* $\mathcal{C}$ *with statistical queries is hard.*

Proof By Theorem 10, a randomized $(k, N!)$-shuffle family $\mathcal{F} = \{\delta_1, \ldots, \delta_{N!}\}$ with $k = \mathcal{O}(N \ln N)$ satisfies $|P_{\text{agree}} - 1/N| \leq 1/N!$ for any two distinct semiautomata. In terms of pairwise correlation, this translates to $|\chi(\delta_i, \delta_j)| \leq 1/N!$ for all $i \neq j$. We have thus shown that the SQ dimension of $\mathcal{C}$ is at least $N!$. The result follows from Theorem 15. ■

7. Conclusion and future work

We have shown that learning semiautomata with N states and alphabet size $\Omega(N^3 \ln N)$ is computationally hard in the SQ model under the uniform distribution over initial states and inputs of length $\Omega(N^2 \ln N)$. Unlike previous hardness results for Deterministic Finite Automata (DFAs), which typically rely on embedding complex languages such as parity, our result derives hardness exclusively from the internal state-transition structure. Using a novel link between semiautomata distinguishability and mixing properties of random walks on $S_N \times S_N$, we demonstrated that a randomized family of $N!$ semiautomata becomes statistically uncorrelated after a polynomial number of steps, yielding an SQ dimension of $N!$. Consequently, we prove that any SQ learner attempting to learn this class must inevitably require either a super-polynomial number of queries or a super-polynomially small tolerance.

Future work. In terms of future work, we identify the following interesting research directions:

1. **Studying hardness for different parameter regimes:** Our hardness results are established for semiautomata with an alphabet size of $\Omega(N^3 \ln N)$ and input length of $\Omega(N^2 \ln N)$. Our analysis reveals that the requirement for the alphabet parameter k to scale as $\mathcal{O}(N \ln N)$ is an artifact of the probabilistic method used: to ensure every pair within a family of size $N!$ is statistically indistinguishable, we must employ a union bound over $\binom{N!}{2}$ pairs. This introduces a logarithmic dependence on the family size that drives up the alphabet complexity.

   Consequently, reducing the alphabet size likely necessitates a fundamental shift in proof strategy, moving away from concentration of measure toward constructions where the norm of the single-step Fourier transform, M_{Π_0}, is controlled directly. This motivates several interesting open questions:

   - **Minimal alphabet size:** What is the smallest alphabet size sufficient for SQ hardness? Answering this may require masking constructions (i.e., schemes that assign to each alphabet symbol a binary mask indicating whether that symbol acts as the identity) based on combinatorial designs or deterministic codes, or extending the analysis to transition functions beyond simple transpositions.
   - **Parameter tradeoffs:** There may exist a computational tradeoff where reducing the alphabet size necessitates a corresponding increase in input length to maintain hardness.

- **Efficient learnability:** Conversely, identifying regimes where both alphabet size and input length are sufficiently restricted could yield positive results, potentially making the class efficiently learnable.

2. **Tightness of SQ-hardness theorem:** Theorem 16 shows that any SQ learner must either make a super-polynomial number of queries or operate with super-polynomially small tolerance. A natural question is whether there exist algorithms that realize these extremes individually: one incurring the query cost and another incurring the tolerance cost.

3. **Learnability in neural architectures:** Our results indicate that conventional gradient-based training of expressive models such as Transformers may not suffice to learn the general class of semiautomata from a uniform data distribution, even though Transformers are known to be capable of efficiently simulating semiautomata (Liu et al., 2023). This motivates an investigation into alternative training paradigms that could circumvent this limitation, like curriculum learning or reinforcement learning.

Acknowledgments

K. Fountoulakis would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [RGPIN-2019-04067, DGECR-2019-00147].

G. Giapitzakis would like to acknowledge the support of the Onassis Foundation - Scholarship ID: F ZU 020-1/2024-2025.

In preparing this paper, we made use of Google's Gemini 2.5 Pro. The randomized construction in Section 5, which is used in Theorem 9, was initially suggested by the model, prompting us to draw a connection with random walks on groups. This connection naturally led to the introduction of tools from group and representation theory that form the theoretical backbone of our results.

References

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the Power of Differentiable Learning versus PAC and SQ Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 24340–24351. Curran Associates, Inc., 2021.

Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978.

Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

Dana Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, 1988.

Dana Angluin, David Eisenstat, Leonid (Aryeh) Kontorovich, and Lev Reyzin. Lower Bounds on Learning Random Structures with Statistical Queries. In *Algorithmic Learning Theory*, pages 194–208, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

Dana Angluin, James Aspnes, Sarah Eisenstat, and Aryeh Kontorovich. On the Learnability of Shuffle Ideals. *Journal of Machine Learning Research*, 14(11):1513–1531, 2013.

Idan Attias, Lev Reyzin, Nathan Srebro, and Gal Vardi. On the Hardness of Learning Regular Expressions, 2025. arXiv: 2510.04834.

José L. Balcázar, Josep Díaz, Ricard Gavaldà, and Osamu Watanabe. The query complexity of learning DFA. *New Generation Computing*, 12(4):337–358, 1994.

Borja Balle. *Learning finite-state machines: statistical and algorithmic aspects*. PhD thesis, Universitat Politècnica de Catalunya, 2013.

Borja Balle, Jorge Castro, and Ricard Gavaldà. Learning probabilistic automata: A study in state distinguishability. *Theoretical Computer Science*, 473:46–60, 2013.

A. Blum and A. Frieze. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, page 330. IEEE Computer Society, 1996.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 253–262. Association for Computing Machinery, 1994.

Benedikt Bollig, Peter Habermehl, Martin Leucker, and Benjamin Monmege. A Fresh Approach to Learning Register Automata. In *Developments in Language Theory*, pages 118–130. Springer Berlin Heidelberg, 2013.

Rafael C. Carrasco and Jose Oncina. Learning stochastic regular grammars by means of a state merging method. In *Grammatical Inference and Applications*, pages 139–152. Springer Berlin Heidelberg, 1994.

Carrasco, Rafael C. and Oncina, Jose. Learning deterministic regular grammars from stochastic samples in polynomial time. *RAIRO-Theor. Inf. Appl.*, 33(1):1–19, 1999.

Konstantinos Chatzilygeroudis, Vassilis Vassiliades, Freek Stulp, Sylvain Calinon, and Jean-Baptiste Mouret. A Survey on Policy Search Algorithms for Learning Robot Controllers in a Handful of Trials. *IEEE Transactions on Robotics*, 36(2):328–347, 2020.

Alexander Clark and Franck Thollard. PAC-learnability of Probabilistic Deterministic Finite State Automata. *Journal of Machine Learning Research*, 5:473–497, 2004.

Edmund Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT Press, 2001.

Corinna Cortes, Leonid Kontorovich, and Mehryar Mohri. Learning Languages with Rational Kernels. In *Learning Theory*, pages 349–364. Springer Berlin Heidelberg, 2007.

Colin de la Higuera. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

Persi Diaconis and Mehrdad Shahshahani. Generating a random permutation with random transpositions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(2):159–179, 1981.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '17, pages 73–84. IEEE, 2017.

Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-Optimal SQ Lower Bounds for Agnostically Learning Halfspaces and ReLUs under Gaussian Marginals. In *Advances in Neural Information Processing Systems*, volume 33, pages 13586–13596. Curran Associates, Inc., 2020.

Funda Ergün, S. Ravi Kumar, and Ronitt Rubinfeld. On learning bounded-width branching programs. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, COLT '95, pages 361–368. Association for Computing Machinery, 1995.

Vitaly Feldman. Open Problem: The Statistical Query Complexity of Learning Sparse Halfspaces. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 1283–1289. PMLR, 2014.

Benjamin Fish and Lev Reyzin. Open Problem: Meeting Times for Learning Random Automata. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 8–11. PMLR, 2017.

W. Fulton and J. Harris. *Representation Theory: A First Course*. Graduate Texts in Mathematics. Springer New York, 2013.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial Lower Bounds for Learning One-Layer Neural Networks using Gradient Descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3587–3596. PMLR, 2020.

E Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

G. H. Hardy and S. Ramanujan. Asymptotic Formulaæ in Combinatory Analysis. *Proceedings of the London Mathematical Society*, s2-17(1):75–115, 1918.

Gerard Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 1st edition, 2011.

Daniel J Hsu, Clayton H Sanford, Rocco Servedio, and Emmanouil Vasileios Vlatakis-Gkaragkounis. Near-Optimal Statistical Query Lower Bounds for Agnostically Learning Intersections of Halfspaces with Gaussian Marginals. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 283–312. PMLR, 2022.

Malte Isberner. *Foundations of Active Automata Learning: An Algorithmic Perspective*. PhD thesis, TU Dortmund, 2015.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45 (6):983–1006, 1998.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

Leonid Kontorovich, Corinna Cortes, and Mehryar Mohri. Learning Linearly Separable Languages. In *Algorithmic Learning Theory*, pages 288–303. Springer Berlin Heidelberg, 2006.

Leonid (Aryeh) Kontorovich and Boaz Nadler. Universal Kernel-Based Learning with Applications to Regular Languages. *Journal of Machine Learning Research*, 10(39):1095–1129, 2009.

Leonid (Aryeh) Kontorovich, Corinna Cortes, and Mehryar Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223–236, 2008.

Kimmo Koskenniemi. Two-level model for morphological analysis. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'83, pages 683–685. Morgan Kaufmann Publishers Inc., 1983.

Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

Loes Kruger, Bharat Garhewal, and Frits Vaandrager. Lower Bounds for Active Automata Learning. In *Proceedings of 16th edition of the International Conference on Grammatical Inference*, volume 217 of *Proceedings of Machine Learning Research*, pages 157–180. PMLR, 2023.

Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *Grammatical Inference*, pages 1–12. Springer Berlin Heidelberg, 1998.

S. Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, 2005.

Srivatsan Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers Learn Shortcuts to Automata. In *The Eleventh International Conference on Learning Representations*, 2023.

Andreas Maletti. Survey: Finite-state technology in natural language processing. *Theoretical Computer Science*, 679:2–17, 2017.

Mehryar Mohri. On some applications of finite-state automata theory to natural language processing. *Natural Language Engineering*, 2(1):61–80, 1996.

Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

Nick Palmer and Paul W. Goldberg. PAC-Learnability of Probabilistic Deterministic Finite State Automata in Terms of Variation Distance. In *Algorithmic Learning Theory*, pages 157–170, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

Rajesh Parekh and Vasant Honavar. Learning DFA from simple examples. In *Algorithmic Learning Theory*, pages 116–131. Springer Berlin Heidelberg, 1997.

Leonard Pitt and Manfred K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the ACM*, 40(1):95–142, 1993.

Lev Reyzin. Statistical Queries and Statistical Algorithms: Foundations and Applications, 2020.

Dana Ron, Yoram Singer, and Naftali Tishby. On the learnability and usage of acyclic probabilistic finite automata. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, COLT '95, pages 31–40. Association for Computing Machinery, 1995.

Yasubumi Sakakibara. Grammatical inference in bioinformatics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1051–1062, 2005.

J.P. Serre. *Linear Representations of Finite Groups*. Graduate texts in mathematics. Springer-Verlag, 1996.

Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, Communication, and Statistical Queries. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1490–1516. PMLR, 2016.

Kaito Suzuki, Diptarama Hendrian, Ryo Yoshinaka, and Ayumi Shinohara. Query Learning Algorithm for Symbolic Weighted Finite Automata. In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pages 202–216. PMLR, 2021.

Balázs Szörényi. Characterizing Statistical Query Learning: Simplified Notions and Proofs. In *Algorithmic Learning Theory*, pages 186–200. Springer Berlin Heidelberg, 2009.

B. A. Trakhtenbrot and J. M. Barzdin. *Finite Automata: Behavior and Synthesis*. North-Holland Publishing Company, 1973.

Joel A. Tropp. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Guy Tsafnat, Jaron Schaeffer, Andrew Clayphan, Jon R. Iredell, Sally R. Partridge, and Enrico Coiera. Computational inference of grammars for larger-than-gene structures from annotated gene sequences. *Bioinformatics*, 27(6):791–796, 2011.

Marcell Vazquez-Chanlatte, Karim Elmaaroufi, Stefan Witwicki, Matei Zaharia, and Sanjit A. Seshia. L*LM: Learning Automata from Demonstrations, Examples, and Natural Language. In *Proceedings of the International Conference on Neuro-symbolic Systems*, volume 288 of *Proceedings of Machine Learning Research*, pages 543–569. PMLR, 2025.

Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D Lee, and Denny Wu. Learning Compositional Functions with Transformers from Easy-to-Hard Data. In *Proceedings of Thirty Eighth Conference on Learning Theory*, Proceedings of Machine Learning Research, pages 5632–5711. PMLR, 2025.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5247–5256. PMLR, 2018.

Wilson Wu. Learning Deterministic Finite Automata from Confidence Oracles, 2023. arXiv: 2311.10963.

Kevin Zhou. Query Learning Bounds for Advice and Nominal Automata. In *Automated Technology for Verification and Analysis*, pages 257–278. Springer Nature Switzerland, 2025.

Appendix A. More on related work

In this section, we review a broad range of DFA learnability results outside the statistical query model. These include results in the PAC model, learning with margin-based guarantees, and query-based learning. We cover both positive and negative results.

A.1. Learnability within the PAC model

Positive results. One of the first positive results was given in (Clark and Thollard, 2004; Palmer and Goldberg, 2005), where, with structural assumptions, the PAC-learnability of probabilistic DFAs is proved. Another approach restricts the distribution of examples; for instance, Parekh and Honavar (1997) showed that "simple" DFAs are PAC-learnable under the universal distribution. Early work by Ron et al. (1995) showed that acyclic Probabilistic Deterministic Finite Automata (PDFAs) were learnable. This was followed by a series of results on learning general PDFAs using state-merging techniques (Carrasco and Oncina, 1994; Carrasco, Rafael C. and Oncina, Jose, 1999; Lang et al., 1998). A key theoretical breakthrough came by Clark and Thollard (2004), which proved the PAC-learnability of PDFAs by parameterizing the complexity by the "distinguishability" of the automaton's states. This line of research culminated in the work of Balle et al. (2013) (and the corresponding thesis (Balle, 2013)), which rigorously characterized state-merging algorithms. They proved that PDFAs are efficiently learnable, but crucially, that the complexity must depend polynomially on the inverse of the distinguishability parameter. Finally, the work by Ergün et al. (1995) investigates the PAC-learnability of deterministic finite automata (DFAs), focusing on the subclass of bounded-width branching programs. It shows that width-2 branching programs can be efficiently learned (both distribution-free and properly under the uniform distribution), while learning width-3 branching programs is at least as hard as learning disjunctive normal forms.

Negative results. The first hardness results for DFA were established in the works of Angluin (1978); Gold (1978); Pitt and Warmuth (1993) under standard complexity assumptions. Subsequent works by Kearns and Valiant (1994) and Kearns and Vazirani (1994) established representation-independent hardness results for finite automata within the PAC model by utilizing "bad" distributions over the input so that any efficient learning algorithm that achieves a polynomial advantage over random guessing will break various cryptographic hardness assumptions. We remark that establishing a hardness result for the PAC-learnability of random DFAs, i.e., DFAs with transition functions and accepting states chosen uniformly at random, remains an open problem (Fish and Reyzin, 2017).

A.2. Learnability with margin-based guarantees

A different line of work from the state-merging approach considers margin-based guarantees, a framework weaker than PAC learning (Kontorovich et al., 2006; Cortes et al., 2007; Kontorovich et al., 2008; Kontorovich and Nadler, 2009). This approach embeds piecewise-testable languages and other discrete concepts into high-dimensional spaces using kernel methods and identifies languages via hyperplanes.

A.3. Learnability based on queries

The seminal L^* algorithm by Angluin (1987) showed that DFAs are efficiently learnable in polynomial time using membership and equivalence queries. This foundational result established a fundamental dichotomy: DFAs are easy to learn with an interactive teacher but hard to learn from random examples alone. The power of queries has been extensively studied by Angluin (1988), with research exploring the query complexity of learning (Balcázar et al., 1994) and establishing lower bounds on the number of queries required (Kruger et al., 2023). This line of work has been extended to more complex automata models, such as register automata (Bollig et al., 2013), symbolic weighted automata (Suzuki et al., 2021), and nominal DFAs (Zhou, 2025), and has been surveyed in works like (Isberner, 2015). More recently, researchers have explored using modern tools like Recurrent Neural Networks (Weiss et al., 2018) and Large Language Models (Vazquez-Chanlatte et al., 2025) to act as oracles. Other works have demonstrated learnability with different powerful oracles, such as those that provide confidence scores (Wu, 2023). Another approach has been to restrict the class of learnable automata, leading to positive SQ results for specific subclasses such as shuffle ideals (Angluin et al., 2013). On a related note, recent work by Attias et al. (2025) establishes hardness results for learning regular expressions when the learner is permitted to make membership queries in addition to sampling random labeled examples. Their results apply to both arbitrary input distributions and the uniform distribution, and are based on standard computational hardness assumptions.

Appendix B. Abstract algebra background

This section provides a concise overview of the essential concepts from group theory and representation theory that are used in our proofs. It is divided into two parts: in Appendix B.1 we present foundational definitions and results from both theories, while in Appendix B.2 we focus specifically on the representation theory of the symmetric group S_N, which plays a central role in our analysis.

B.1. Group and representation theory basics

To enhance readability, this section is further divided into two subsections: in Appendix B.1.1 we present an overview of elementary definitions and facts from group theory, whereas in Appendix B.1.2 we provide a concise summary of key definitions and results from the representation theory of groups. The results presented in Appendix B.1.1 are standard and can be found in any introductory text on abstract algebra, such as Lang (2005). The results in Appendix B.1.2 are likewise standard within the representation theory literature and can be found, for instance, in Serre (1996).

B.1.1. GROUP THEORY OVERVIEW

We begin by defining the notion of a group:

Definition 17 (Group) *A* group *is a non-empty set G endowed with a binary operation $\star : G \times G \to G$ such that the following three axioms are satisfied:*

1. ***Associativity:*** *For all $a, b, c \in G$ one has $(a \star b) \star c = a \star (b \star c)$.*

2. ***Identity element:*** *There exists an element* $\mathrm{id} \in G$ *such that for all elements* $g \in G$ *one has* $g \star \mathrm{id} = \mathrm{id} \star g = g$.

3. ***Inverse element:*** *For each element* $g \in G$ *there exists an element* $h \in G$ *such that* $g \star h = h \star g = \mathrm{id}$.

From the above definition, it follows that id is unique and that for each $g \in G$, the element $h \in G$ that satisfies Item 3 is unique. Hence, we may write g^{-1} to refer to that element. Furthermore, when it is clear from the context, for $g, h \in G$ we will write gh instead of $g \star h$ and also use the notation $g^n := g \star \cdots \star g$. The *order* of G, denoted by $|G|$, is the size of the underlying set G. The *order of an element* $g \in G$, denoted by $|g|$, is the smallest integer $n \in \mathbb{N}$ such that $g^n = \mathrm{id}$. If no such n exists, we say that g has infinite order. The order of every element $g \in G$ divides the order of the group G (Lagrange's theorem, Proposition 2.2 in Lang (2005)), and in particular $|g| \leq |G|$. The latter shows that when G is finite, every element has finite order and that $g^{|G|} = \mathrm{id}$. We can now define the notion of a homomorphism, which is a mapping between groups that respects their structure.

Definition 18 *Let* $(G, \star)$ *and* $(H, *)$ *be groups. A* homomorphism *is a mapping* $f : G \to H$ *such that for all* $g_1, g_2 \in G$ *one has*

$$f(g_1 \star g_2) = f(g_1) * f(g_2).$$

From the definition, it follows that f maps the identity of G to the identity of H, namely $f(\mathrm{id}_G) = \mathrm{id}_H$.[3] When f is bijective, we say that it is an *isomorphism* and the groups G and H are called isomorphic. In that case we write $G \simeq H$. From an algebraic perspective, isomorphic groups are essentially the same group. An isomorphism $f : G \to G$ is called an *automorphism* of G.

Example 1 *Here we present two key examples, which we will encounter frequently in this work:*

i) The set of all permutations of $1, 2, \ldots, N$*, equipped with the operation of function composition, where applying* $\sigma \circ \pi$ *means first applying* π *followed by* σ*, forms a group, denoted by* S_N*. Its order is* $|S_N| = N!$.

ii) Let V *be a vector space over a field* $\mathbb{F}$*. The set of automorphisms of* V*, i.e., the set of all bijective linear transformations* $V \to V$*, endowed with the function composition operation, is a group, denoted by* $\mathrm{GL}(V)$*. When* V *is a finite-dimensional vector space of dimension* $d < \infty$*, the group* $\mathrm{GL}(V)$ *is isomorphic to the group* $\mathrm{GL}(d, \mathbb{F})$ *of invertible* $d \times d$ *matrices over* $\mathbb{F}$ *with the operation of matrix multiplication. The isomorphism maps each automorphism of* V *to its matrix representation, allowing us to view elements of* $\mathrm{GL}(V)$ *as* $d \times d$ *invertible matrices with entries in* $\mathbb{F}$.

Next, we define the direct product of two groups as follows:

Definition 19 (Direct product) *Let* $(G, \star)$ *and* $(H, *)$ *be groups. We endow the Cartesian product* $G \times H$ *with the binary operation* $\circ : (G \times H) \times (G \times H) \to G \times H$ *defined component-wise:*

$$(g_1, h_1) \circ (g_2, h_2) = (g_1 \star g_2, h_1 * h_2).$$

3. To see why, write $f(\mathrm{id}_G) = f(\mathrm{id}_G\, \mathrm{id}_G) = f(\mathrm{id}_G)^2$ and pre-multiply by $f(\mathrm{id}_G)^{-1}$.

The resulting structure $(G \times H, \circ)$ *satisfies the group axioms and is called the* direct product *of* G *and* H*, denoted by* $G \times H$*.*

Lastly, we conclude this section by discussing conjugacy, a key property when studying group representations.

Definition 20 (Conjugacy) *Let* G *be a group and let* $g, h \in G$*. The elements* g *and* h *are called* conjugate *if there exists* $a \in G$ *such that* $h = aga^{-1}$*. This is an equivalence relation whose equivalence classes are called* conjugacy classes*.*

B.1.2. REPRESENTATION THEORY OVERVIEW

At a high level, representation theory provides a powerful framework for studying groups by translating their abstract algebraic structure into the concrete language of linear algebra. We begin by presenting the definition of a representation:

Definition 21 (Representation) *Let* G *be a group. A* representation *of* G *over a vector space* V *over some field* $\mathbb{F}$ *is a homomorphism* $\rho : G \to \mathrm{GL}(V)$*. The dimension of the representation, denoted by* d_ρ*, is the dimension of the vector space* V*.*

From this point forward, we will assume that all groups are finite, all representations are finite-dimensional, and that $\mathbb{F} = \mathbb{C}$. These assumptions hold in all cases relevant to our work. By the discussion in Example 1, we can therefore view the image of each $g \in G$ under ρ, i.e. $\rho(g)$, as an invertible $d_\rho \times d_\rho$ complex matrix. From the definition, we immediately get that $\rho(\mathrm{id}) = I_{d_\rho}$ and that $\rho(g^{-1}) = \rho(g)^{-1}$. Furthermore, it can be shown that $\rho(g)$ can always be chosen to be *unitary* (c.f. Section 1.3 in Serre (1996)), in which case $\rho(g^{-1}) = \rho(g)^*$.

Example 2 *The trivial representation, denoted by* triv*, maps each group element to the* 1×1 *identity matrix. Hence,* $d_{\mathrm{triv}} = 1$*.*

We can define the notion of an isomorphism between two representations as follows:

Definition 22 (Isomorphic representations) *Two representations* $(\rho, V), (\sigma, V')$ *of a group* G *are called isomorphic if there exists a linear isomorphism* $\tau : V \to V'$ *such that* $\tau \circ \rho(g) = \rho'(g) \circ \tau$ *for all* $g \in G$*. In that case, we write* $\rho \simeq \sigma$*.*

Next, we define the concept of irreducible representations. Maschke's theorem (Theorem 2 in Serre (1996)) guarantees that every representation is isomorphic to a direct sum of irreducible representations, allowing us to restrict our study to only those representations that are irreducible.

Definition 23 (Irreducible representation) *A representation* $\rho : G \to \mathrm{GL}(V)$ *over a vector space* V *is called* irreducible *if the only subspaces of* V *that are left invariant by every* $\rho(g)$ *are* $\{0\}$ *and* V*. In symbols,* ρ *is irreducible if the following holds for any subspace* $W \subseteq V$*:*

$$\rho(g)(W) \subseteq W \text{ for all } g \in G \implies W = \{0\} \text{ or } W = V.$$

We use the abbreviation irreps *to refer to the irreducible representations. The set of all irreducible unitary representations of* G *is denoted by* $\hat{G}$*.*

Next, we define the direct sum of two representations.

Definition 24 (Direct sum of representations) *Let $\rho_1 : G \to \mathrm{GL}(V_1)$ and $\rho_2 : G \to \mathrm{GL}(V_2)$ be two representations of G. Their direct sum is defined as the representation $\rho_1 \oplus \rho_2 : G \to \mathrm{GL}(V_1 \oplus V_2)$ and its action is given by*

$$(\rho_1 \oplus \rho_2)(g) = \begin{bmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{bmatrix}$$

for all $g \in G$. The definition trivially extends to a finite number of representations.

In what follows, we characterize the irreducible representations of the direct product via the irreducible representations of the individual groups. To do so, we need to define the tensor product representation:

Definition 25 (Tensor product of representations) *Let G and H be two groups and let $\rho : G \to \mathrm{GL}(V_1)$ and $\sigma : H \to \mathrm{GL}(V_2)$ be representations of G and H, respectively. We define the* tensor product representation *$\rho \otimes \sigma$ of the direct product $G \times H$ over $V_1 \otimes V_2$ as the representation*

$$\begin{aligned} \rho \otimes \sigma : G \times H &\to \mathrm{GL}(V_1 \otimes V_2) \\ (g, h) &\mapsto \rho(g) \otimes \sigma(h) \end{aligned}$$

The dimension of $\rho \otimes \sigma$ is $d_{\rho\otimes\sigma} = \dim(V_1 \otimes V_2) = \dim(V_1)\dim(V_2) = d_\rho d_\sigma$.

For finite groups, the action of the tensor product representation $\rho\otimes\sigma$ on an element $(g, h) \in G\times H$ can be realized as the Kronecker product of the corresponding matrices $\rho(g)$ and $\sigma(h)$. As such, standard Kronecker product properties apply.
The following theorem yields a complete characterization of the irreducible representations of $G \times H$.

Theorem 26 (Theorem 10 in Serre (1996)) *Every irreducible (unitary) representation of $G \times H$ is isomorphic to a tensor product of irreducible (unitary) representations of G and H. Symbolically, if $\Pi = G \times H$ we have*

$$\hat{\Pi} \simeq \left\{ \rho \otimes \sigma : \rho \in \hat{G}, \sigma \in \hat{H} \right\}.$$

We now move to the study of characters, an essential tool in representation theory that captures key information about a representation.

Definition 27 (Character) *Let $\rho : G \to \mathrm{GL}(V)$ be a representation of a group G over a vector space V. Define the* character *of ρ as the mapping $\chi_\rho : G \to \mathbb{C}$ given by $\chi_\rho(g) = \mathrm{Tr}[\rho(g)]$ for each $g \in G$.*

Definition 28 (Class function) *A function $f : G \to \mathbb{C}$ that is constant on every conjugacy class of G is called a* class function.

From the cyclic property of the trace, we immediately get the following:

Remark 29 *The character of any representation is a class function.*

By Theorem 22 we can also deduce the following:

Remark 30 *The characters of two isomorphic representations of G are equal. Namely, if $\rho \simeq \sigma$ then $\chi_\rho(g) = \chi_\sigma(g)$ for all $g \in G$.*

Proof Since $\rho \simeq \sigma$, there exists an invertible matrix P such that $\sigma(g) = P\rho(g)P^{-1}$ for all $g \in G$. Hence,

$$\chi_\sigma(g) = \mathrm{Tr}(\sigma(g)) = \mathrm{Tr}(P\rho(g)P^{-1}) = \mathrm{Tr}(P^{-1}P\rho(g)) = \mathrm{Tr}(\rho(g)) = \chi_\rho(g).$$

■

For two complex-valued functions $f : G \to \mathbb{C}$ and $h : G \to \mathbb{C}$ we define a scalar product as

$$\langle f, h\rangle = \frac{1}{|G|}\sum_{g\in G} f(g)\overline{h(g)}. \tag{2}$$

It can be shown that Equation (2) defines an inner product when restricted to the set of class functions. In particular, for the case of characters, we have the following theorem:

Theorem 31 (Orthogonality relations, Theorem 3 in Serre (1996)) *If χ is the character of an irreducible representation of G, then $\langle \chi, \chi\rangle = 1$. Furthermore, if χ, χ' are the characters of two non-isomorphic irreducible representations of G, then $\langle \chi, \chi'\rangle = 0$.*

Next, we present Schur's orthogonality relations, also known as the "Great Orthogonality Theorem", which provides a powerful tool for simplifying several calculations in our proofs:

Theorem 32 (Great Orthogonality Theorem, Corollaries 1-3 in Serre (1996)) *Let ρ and σ be two non-isomorphic irreducible unitary representations ρ and σ of G. Let $\rho(g)_{ij}$ and $\sigma(g)_{kl}$ denote matrix elements of $\rho(g)$ and $\sigma(g)$, respectively. Then*

$$\sum_{g\in G} \rho(g)^*_{ij}\sigma(g)_{kl} = 0.$$

For matrices of the same irreducible unitary representation ρ, the relation is:

$$\sum_{g\in G} \rho(g)^*_{ij}\rho(g)_{kl} = \frac{|G|}{d_\rho}\delta_{il}\delta_{jk},$$

where δ denotes the Kronecker delta and A^* the conjugate transpose of a matrix A.

The strength of representation theory lies in its ability to provide a clean analogue of Fourier analysis when working on groups. This perspective proves especially valuable when analyzing random processes on groups. We provide the definition below:

Definition 33 (Fourier transform) *Let G be a group and let $\rho : G \to \mathrm{GL}(V)$ be a representation. The* Fourier transform *of a function $f : G \to \mathbb{C}$ is a matrix valued function acting on G given by*

$$\rho(f) = \sum_{g \in G} f(g)\rho(g).$$

The convolution of two functions acting on the same group is defined as follows:

Definition 34 (Convolution) *Let G be a group and let $f_1 : G \to \mathbb{C}$ and $f_2 : G \to \mathbb{C}$ be functions acting on G. The convolution $f_1 * f_2 : G \to \mathbb{C}$ is given by*

$$(f_1 * f_2)(g) = \sum_{h \in G} f_2(gh^{-1}) f_1(h).$$

The following textbook result gives the Fourier transform of the convolution of two functions in terms of their individual Fourier transforms:

Lemma 35 (Lemma 1 in Diaconis and Shahshahani (1981)) *Let G be a group and let f_1, f_2 be functions acting on G. Then for any representation ρ we have*

$$\rho(f_1 * f_2) = \rho(f_1)\rho(f_2).$$

By induction, Theorem 35 generalizes for the convolution of more than two functions. Concluding this section, we present a series of standard lemmas that we use in our analysis:

Lemma 36 (Plancherel formula, Exercise 3.32 in Fulton and Harris (2013)) *Let G be a group and $f, h : G \to \mathbb{C}$. Then*

$$\sum_{g \in G} f(g)\overline{h(g)} = \frac{1}{|G|} \sum_{\rho \in \hat{G}} d_\rho \operatorname{Tr}[\rho(f)\rho(h)^*].$$

The sum of the right-hand side is taken over all irreducible unitary representations of G.

Lemma 37 (Schur's lemma, Proposition 4 in Serre (1996)) *Let $\rho_1 : G \to \mathrm{GL}(V_1)$ and $\rho_2 : G \to \mathrm{GL}(V_2)$ be irreducible representations of G, and let M be a matrix that $\rho_2(g)M = M\rho_1(g)$ for all $g \in G$. Then:*

1. *If ρ_1 and ρ_2 are not isomorphic, we have $M = 0$*
2. *If $V_1 = V_2$ and $\rho_1 = \rho_2$, then $M = cI$ for some $c \in \mathbb{C}$.*

Schur's Lemma gives rise to a useful corollary in the case where a representation $\rho : G \to \mathrm{GL}(V)$ can be decomposed into a direct sum of non-isomorphic irreps $(\rho_1, V_1), \ldots, (\rho_k, V_k)$, i.e., $\rho = \rho_1 \oplus \cdots \oplus \rho_k$ and $\rho_i \not\cong \rho_j$ for all $i \neq j$:

Corollary 38 (Schur's Lemma for reducible representations) *Let $\rho : G \to \mathrm{GL}(V)$ be a representation of G that decomposes as a direct sum of non-isomorphic irreps $(\rho_1, V_1), \dots, (\rho_k, V_k)$, and let M be a matrix such that $\rho(g)M = M\rho(g)$ for all $g \in G$. Then*

$$M = \begin{bmatrix} c_1 I_{d_{\rho_1}} & 0 & \dots & 0 \\ 0 & c_2 I_{d_{\rho_2}} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & c_k I_{d_{\rho_k}} \end{bmatrix}$$

for some constants $c_1, \dots, c_k \in \mathbb{C}$.

Proof Recall that by Theorem 24 we have

$$\rho(g) = \begin{bmatrix} \rho_1(g) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_k(g) \end{bmatrix}$$

The condition $\rho(g)M = M\rho(g)$ thus translates to

$$\begin{bmatrix} M_{11} & \dots & M_{1k} \\ \vdots & \ddots & \vdots \\ M_{k1} & \dots & M_{kk} \end{bmatrix} \begin{bmatrix} \rho_1(g) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_k(g) \end{bmatrix} = \begin{bmatrix} \rho_1(g) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \rho_k(g) \end{bmatrix} \begin{bmatrix} M_{11} & \dots & M_{1k} \\ \vdots & \ddots & \vdots \\ M_{k1} & \dots & M_{kk} \end{bmatrix}$$

where M_{ij} are $d_{\rho_i} \times d_{\rho_j}$ blocks of M. This gives $M_{ij}\rho_j(g) = \rho_i(g)M_{ij}$ for all $i, j \in [k]$. The result trivially follows by applying Schur's Lemma (Theorem 37). ■

Lemma 39 (Lemma 5 in Diaconis and Shahshahani (1981)) *Let G be a group and ρ an irreducible representation of G. Let $f : G \to \mathbb{C}$ be a class function, i.e., it is constant on each conjugacy class. Let f_i be the value of f on the i-th conjugacy class, n_i the cardinality of the i-th conjugacy class, and χ_i the value of the character of ρ on the i-th conjugacy class. Then*

$$\rho(f) = CI \quad \textit{where} \quad C = \frac{1}{d_\rho} \sum_i f_i n_i \chi_i.$$

The sum is taken over distinct conjugacy classes.

The last result, which we prove, characterizes the Fourier transforms of the uniform probability distribution on a group G with respect to irreducible representations.

Lemma 40 *Let G be a group and $U : G \to \mathbb{C}$ be the uniform probability distribution over G, namely $U(g) = 1/|G|$ for all $g \in G$. Then*

$$\rho(U) = \begin{cases} 1 & \text{if } \rho = \mathrm{triv} \\ 0_{d_\rho \times d_\rho} & \text{if } \rho \in \hat{G} \setminus \{\mathrm{triv}\}. \end{cases}$$

Proof The case $\rho = \mathrm{triv}$ follows by the definition of the trivial representation. For an irrep $\rho \neq \mathrm{triv}$ we use Schur's lemma (Theorem 37). Notice that for all $g \in G$ we have

$$\rho(U)\rho(g) = \rho(g)\rho(U) = \frac{1}{|G|}\sum_{g \in G}\rho(g).$$

Hence, Schur's lemma asserts that $\rho(U) = cI_{d_\rho}$ for some $c \in \mathbb{C}$. Taking the trace of both sides, we get

$$\frac{1}{|G|}\sum_{g \in G}\chi_\rho(g) = cd_\rho.$$

The left-hand side of the above expression is precisely the product of the characters of ρ and triv, $\langle \chi_\rho, \chi_{\mathrm{triv}} \rangle$. By the orthogonality relations of characters, $\langle \chi_\rho, \chi_{\mathrm{triv}} \rangle = 0$ and so $c = 0$. This concludes the proof. ■

B.2. Representations of the symmetric group

Although the results of the previous section apply to any finite group, this work exclusively focuses on the symmetric group S_N and the product group $S_N \times S_N$. Fortunately, the representation theory of symmetric groups is well studied, and we can draw on a wealth of established results. In this section, we summarize the key definitions and facts about the representations of S_N. Most of the results presented as "Facts" are taken from Chapter 4 of Fulton and Harris (2013). Alongside each statement, we provide a reference to the corresponding passage. Recall the definition of the symmetric group from Example 1:

Definition 41 (Symmetric group) *The set S_N of all permutations of $[N]$ (i.e., bijections $f : [N] \to [N]$), equipped with function composition, forms a group of order $N!$ called the* symmetric group *on N elements.*

It can be shown that irreducible representations of S_N are in one-to-one correspondence with the partitions of N. A *partition* of a positive integer N is a tuple $(\lambda_1, \ldots, \lambda_k) \in \mathbb{N}^k$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 1$ and $\lambda_1 + \cdots + \lambda_k = N$. We can now characterize the irreps of S_N as follows:

Fact 42 (p. 44 in Fulton and Harris (2013)) *Each irrep of S_N corresponds to a partition of N. Conversely, each partition of N corresponds to an irrep of S_N. The trivial representation of S_N corresponds to the partition (N).*

The *standard representation* of S_N, denoted by std, is the representation corresponding to the partition $(N-1, 1)$. In what follows, we will occasionally denote irreps by their corresponding partitions. For example, we may write $(N-1, 1)$ instead of std.
If we let $p(N)$ denote the number of partitions of $N \in \mathbb{N}$, Theorem 42 and Theorem 26 give the following:

Remark 43 *The number of irreducible representations is $p(N)$ for S_N and $p(N)^2$ for $S_N \times S_N$.*

The following fact states that irreducible representations of S_N can be defined over the reals.

Fact 44 (p. 46 in Fulton and Harris (2013)) *Each irreducible representation of S_N can be defined over the rationals. In particular, we can define every irreducible representation ρ of S_N such that $\rho(g)$ is an orthogonal matrix for every $g \in S_N$.*

The dimension of an irrep can be determined solely by the corresponding partition, as shown in the following fact:

Fact 45 (Equation 4.11 in Fulton and Harris (2013)) *Let $\rho = (\lambda_1, \dots, \lambda_k)$ be an irrep of S_N. Then*

$$d_\rho = \frac{N!}{l_1! \cdots l_k!} \prod_{i<j} (l_i - l_j),$$

where $l_i = \lambda_i + k - i$.

Another quantity of interest is the *character ratio* of transpositions. For an irrep ρ, it is defined as $r(\rho) = \chi_\rho(\tau)/d_\rho$, where $\chi_\rho(\tau)$ is the character of ρ evaluated at any transposition.[4] The following fact gives a closed-form expression for $r(\rho)$ that only depends on the corresponding partition:

Fact 46 (Lemma 7 in Diaconis and Shahshahani (1981)) *Let $\rho = (\lambda_1, \dots, \lambda_k)$ be an irrep of S_N. The character ratio of transpositions is given by*

$$r(\rho) = \frac{1}{N(N-1)} \sum_{j=1}^{k} \big[(\lambda_j - j)(\lambda_j - j + 1) - j(j-1)\big].$$

Example 3 *Using Theorem 46 we can obtain the following:*

i) It follows by the definition of the trivial representation that $r(\mathrm{triv}) = 1$. It is easy to see that the expression given in Theorem 46 gives the same result.

ii) For the standard representation $\mathrm{std} = (N-1, 1)$, we obtain $r(\mathrm{std}) = \frac{N-3}{N-1}$.

We now introduce the permutation representation of S_N, a useful non-irreducible representation of S_N.

Definition 47 (Permutation representation) *Let $\{\boldsymbol{e_1}, \dots, \boldsymbol{e_N}\}$ be the standard basis of $\mathbb{C}^N$. The* permutation representation[5] *of S_N, denoted by perm, is the homomorphism $\rho : S_N \to \mathrm{GL}(\mathbb{C}^N)$ defined by*

$$\rho(g)(\boldsymbol{e_i}) = \boldsymbol{e_{g(i)}}$$

for all $g \in S_N$ and $i \in [N]$.

We leave the verification that perm is indeed a valid representation of S_N as an (easy) exercise. In essence, the permutation representation of an element $g \in S_N$ is the linear map that acts by permuting the standard basis vectors of $\mathbb{C}^N$ according to g. The permutation representation is closely related to the number of fixed points of the elements of S_N. This is made precise in Theorem 49, following the definition of fixed points.

4. Note that this is well-defined since transpositions form a conjugacy class.
5. In some textbooks it may also be referred to as the defining representation of S_N.

Definition 48 *Let $g \in S_N$ be a permutation. We say that $i \in [N]$ is a* fixed point *of g if $g(i) = i$. The set of fixed points of g is denoted by $\mathrm{fix}(g) = \{i \in [N] : g(i) = i\}$.*

Lemma 49 *The character of* perm *on any $g \in S_N$ is equal to the number of fixed points of g. Namely, $\chi_{\mathrm{perm}}(g) = |\mathrm{fix}(g)|$ for all $g \in S_N$.*

Proof Observe that, by definition, when expressed in the standard basis, $\rho(g)$ is a permutation matrix. The diagonal element in the (i,i)-th coordinate is equal to 1 *if and only if* $\rho(g)(\boldsymbol{e}_i) = \boldsymbol{e_i}$, which happens *if and only if* $g(i) = i$, or equivalently $i \in \mathrm{fix}(g)$. Hence,

$$\chi_{\mathrm{perm}}(g) = \sum_{i=1}^{N} \rho(g)_{ii} = |\mathrm{fix}(g)|,$$

which concludes the proof. ■

Since perm is not irreducible, Maschke's theorem guarantees that it is isomorphic to a direct sum of irreps. The decomposition is given in the following fact:

Fact 50 (p. 55 in Fulton and Harris (2013)) *The permutation decomposition of S_N is isomorphic to the direct sum of the trivial and the standard representation of S_N, namely* $\mathrm{perm} \simeq \mathrm{triv} \oplus \mathrm{std}$. *Consequently,*

$$\chi_{\mathrm{perm}}(g) = \chi_{\mathrm{triv}}(g) + \chi_{\mathrm{std}}(g) = 1 + \chi_{\mathrm{std}}(g)$$

for all $g \in S_N$.

Appendix C. Proof of Theorem 5

In this section, we provide the full proof of Theorem 5. The proof relies on a technical lemma regarding the Fourier transform of the function on $S_N \times S_N$ given by $(g,h) \mapsto |\mathrm{fix}(h^{-1}g)|$, which we state and prove first.

Lemma 51 (Fourier transform of fix**)** *Let $G = S_N \times S_N$ and $f : G \to \mathbb{C}$ be given by $f(g,h) = |\mathrm{fix}(h^{-1}g)|$. Then for any non-trivial irreducible representation Π of G, we have*

$$\Pi(f) = \begin{cases} \frac{(N!)^2}{N-1} P_{\mathrm{diag}} & \text{if } \Pi = \mathrm{std} \otimes \mathrm{std} \\ 0_{d_\Pi \times d_\Pi} & \text{otherwise} \end{cases}$$

where P_{diag} is the orthogonal projection onto the diagonal subspace $\mathrm{span}\left\{\sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}\right\}$.

Proof By Theorem 26, we can write $\Pi = \rho \otimes \sigma$ with at least one of the irreps $\rho, \sigma \in \hat{S}_N$ being non-trivial. The Fourier transform of f is thus given by

$$\Pi(f) = \sum_{g,h \in S_N} |\mathrm{fix}(h^{-1}g)| \rho(g) \otimes \sigma(h)$$

Performing the change of variables $g' = h^{-1}g$, the above sum becomes

$$\begin{aligned}\Pi(f) &= \sum_{g,h\in S_N} |\operatorname{fix}(g)|\rho(hg)\otimes\sigma(h) \\ &= \sum_{g,h\in S_N} |\operatorname{fix}(g)|(\rho(h)\otimes\sigma(h))(\rho(g)\otimes I_{d_\sigma}) \\ &= \sum_{g\in S_N} |\operatorname{fix}(g)| \left\{\sum_{h\in S_N} \rho(h)\otimes\sigma(h)\right\}(\rho(g)\otimes I_{d_\sigma}) \\ &= \left(\sum_{h\in S_N} \rho(h)\otimes\sigma(h)\right)\left(\sum_{g\in S_N} |\operatorname{fix}(g)|\rho(g)\right)\otimes I_{d_\sigma}\end{aligned}$$

Since $|\operatorname{fix}(\cdot)|$ is the character of the permutation representation of S_N (Theorem 49), and hence a class function on S_N, by Theorem 39, the second sum is equal to CI_{d_ρ} where

$$C = \frac{1}{d_\rho}\sum_{g\in S_N} |\operatorname{fix}(g)|\chi_\rho(g) = \frac{N!}{d_\rho}(\langle\chi_{\text{triv}},\chi_\rho\rangle + \langle\chi_{\text{std}},\chi_\rho\rangle).$$

The last equality follows from the fact that the permutation representation decomposes as the direct sum of the trivial representation and the standard representation (Theorem 50). In particular, character orthogonality (Theorem 31) yields $C \neq 0$ if and only if $\rho \in \{\text{triv}, \text{std}\}$.

For the first sum, let

$$M = \sum_{h\in S_N} \rho(h)\otimes\sigma(h).$$

Indexing M by the indices of the products we get

$$M_{(i,j),(k,l)} = \sum_{h\in S_N} \rho(h)_{ij}\sigma(h)_{kl} = \sum_{h\in S_N} \rho(h)^\top_{ji}\sigma(h)_{kl}.$$

By the Great Orthogonality Theorem (Theorem 32) and the fact that ρ can be taken such that $\rho(h)$ is real for every $h \in S_N$ (Theorem 44), $M_{(i,j),(k,l)}$ vanishes, unless $\rho = \sigma$, in which case it works out to

$$M_{(i,j),(k,l)} = \frac{N!}{d_\rho}\delta_{jl}\delta_{ik}.$$

In matrix form, when $\rho = \sigma$

$$M = \frac{N!}{d_\rho}\sum_{i,j=1}^{d_\rho} E_{ij}\otimes E_{ij},$$

where $E_{ij} = \boldsymbol{e_i}\boldsymbol{e_j}^\top \in \mathbb{R}^{d_\rho\times d_\rho}$. Putting everything together, we obtain that $\Pi(f)$ does not vanish *if and only if* $\rho = \sigma$ and $\rho \in \{\text{triv}, \text{std}\}$. If $\rho = \sigma = \text{triv}$ then $\Pi = \text{triv}$, and so the only non-trivial irrep Π for which the Fourier transform does not vanish is $\Pi = \text{std}\otimes\text{std}$. In that case, combining the above calculations with the fact that $d_{\text{std}\otimes\text{std}} = d_{\text{std}}^2 = (N-1)^2$, we obtain

$$\Pi(f) = \frac{(N!)^2}{(N-1)^2}\sum_{i,j=1}^{N-1} E_{ij}\otimes E_{ij}.$$

To see why this is equal to the required expression, let $\boldsymbol{v} = \sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}$ and observe that

$$\frac{1}{N-1}\sum_{i,j=1}^{N-1} E_{ij} \otimes E_{ij} = \frac{1}{\|\boldsymbol{v}\|^2}\sum_{i,j=1}^{N-1} (\boldsymbol{e_i}\boldsymbol{e_j}^\top) \otimes (\boldsymbol{e_i}\boldsymbol{e_j}^\top) = \frac{1}{\|\boldsymbol{v}\|^2}\sum_{i,j=1}^{N-1} (\boldsymbol{e_i} \otimes \boldsymbol{e_i})(\boldsymbol{e_j} \otimes \boldsymbol{e_j})^\top$$
$$= \frac{1}{\|\boldsymbol{v}\|^2}\left(\sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}\right)\left(\sum_{j=1}^{N-1} \boldsymbol{e_j} \otimes \boldsymbol{e_j}\right)^\top = \frac{1}{\|\boldsymbol{v}\|^2}\boldsymbol{v}\boldsymbol{v}^\top.$$

The latter expression is precisely equal to P_{diag}, the orthogonal projection matrix onto $\text{span}\{\boldsymbol{v}\}$, thus concluding the proof. ■

We are now ready to prove Theorem 5, which we restate for convenience:

Theorem 5 (Agreement probability) *Consider the random walk corresponding to two semiautomata $\mathcal{A}$ and $\mathcal{A}'$ operating on the same alphabet Σ and state-space $\mathcal{Q}$, as described in Section 4.2. Let $N = |\mathcal{Q}|$ and let $T \in \mathbb{N}$ be the input length. Let $X_0 \sim \mathcal{U}(\mathcal{Q})$ be a (common) initial state picked uniformly at random. Denote by $P_{\text{agree}} := P_{\text{agree}}(T)$ the probability that after processing the same uniformly random word $\mathrm{w}_T \in \Sigma^*$, both semiautomata reach the same state. In terms of the random walk, this probability is given by*

$$P_{\text{agree}}(T) = \mathbb{P}_{\mathrm{w}_T \sim \mathcal{U}(\Sigma)^{\otimes T}, X_0 \sim \mathcal{U}(\mathcal{Q})}\left(P_{\mathrm{w}_T}(X_0) = P'_{\mathrm{w}_T}(X_0)\right).$$

Then

$$P_{\text{agree}} = \frac{1}{N} + \frac{1}{N}\boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v},$$

where M_{Π_0} is the Fourier transform of the single-step distribution T_{SA} with respect to the irreducible representation $\Pi_0 = \text{std} \otimes \text{std}$ of G and $\boldsymbol{v} = \sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}$.

Proof The analysis of P_{agree} requires understanding the convergence of the joint distribution of $(P_{\mathrm{w}_T}, P'_{\mathrm{w}_T})$ on the product group $G = S_N \times S_N$. We condition on the final state $(g, h) \in G$ of the random walk after T steps. Let $P_T(g,h) = \mathbb{P}_{\mathrm{w}_T}(P_{\mathrm{w}_T} = g, P'_{\mathrm{w}_T} = h)$. The number of states on which two permutations g and h agree is the number of fixed points of $h^{-1}g$, denoted $\text{fix}(h^{-1}g)$. Thus, we write:

$$\begin{aligned} P_{\text{agree}} &= \sum_{(g,h)\in G} \mathbb{P}(P_{\mathrm{w}_T} = g, P'_{\mathrm{w}_T} = h) \cdot \mathbb{P}_{X\sim\mathcal{U}(\mathcal{Q})}(g(X) = h(X) \mid P_{\mathrm{w}_T} = g, P'_{\mathrm{w}_T} = h) \\ &= \sum_{(g,h)\in G} P_T(g,h) \cdot \frac{|\{X \in \mathcal{Q} \mid g(X) = h(X)\}|}{N} \\ &= \sum_{(g,h)\in G} P_T(g,h) \cdot \frac{|\,\text{fix}(h^{-1}g)|}{N}. \end{aligned}$$

The distribution P_T of the random walk on G can be written in terms of the uniform distribution over the elements of G, $P_U(g,h) = 1/|G|$ and a residual term $\text{err}(g,h)$, namely $P_T(g,h) =$

$P_U(g,h) + \mathrm{err}(g,h)$, where $\mathrm{err}(g,h) = P_T(g,h) - P_U(g,h)$. Therefore, we get

$$\begin{aligned} P_{\text{agree}} &= \frac{1}{N} \sum_{(g,h)\in G} \left(\frac{1}{(N!)^2} + \mathrm{err}(g,h) \right) \cdot \mathrm{fix}(h^{-1}g) \\ &= \underbrace{\frac{1}{N \cdot (N!)^2} \sum_{(g,h)\in G} \mathrm{fix}(h^{-1}g)}_{\text{Main Term}} + \underbrace{\frac{1}{N} \sum_{(g,h)\in G} (P_T(g,h) - P_U(g,h)) \cdot \mathrm{fix}(h^{-1}g)}_{\text{Error Term}}. \end{aligned} \tag{3}$$

For the main term of Equation (3) notice that for fixed $g \in S_N$ we have

$$\sum_{h\in S_N} \mathrm{fix}(h^{-1}g) = \sum_{h\in S_N} \mathrm{fix}(h)$$

since the map $h \mapsto h^{-1}g$ is a bijection. Hence, we have

$$\text{Main Term} = \frac{1}{N \cdot (N!)^2} \left(\sum_{g\in S_N} \mathrm{fix}(g) \right)^2 = \frac{1}{N \cdot (N!)^2} \cdot (N!)^2 = \frac{1}{N}, \tag{4}$$

where the second-to-last equality follows from the fact that

$$\begin{aligned} \sum_{g\in S_N} \mathrm{fix}(g) &= \sum_{g\in S_N} \sum_{X\in\mathcal{Q}} \mathbb{1}_{\{g(X)=X\}} = \sum_{X\in\mathcal{Q}} \sum_{g\in S_N} \mathbb{1}_{\{g(X)=X\}} \\ &= \sum_{X\in\mathcal{Q}} (N-1)! = N \cdot (N-1)! = N!. \end{aligned}$$

We now turn our attention to the error term of Equation (3), given by

$$\frac{1}{N} \sum_{(g,h)\in G} (P_T(g,h) - P_U(g,h)) \cdot \mathrm{fix}(h^{-1}g).$$

By letting $f(g,h) = \mathrm{fix}(h^{-1}g)$ and using the Plancherel formula (Theorem 36), the error term is equal to:

$$\frac{1}{N} \cdot \frac{1}{|G|} \sum_{\Pi\in\hat{G}} d_\Pi \operatorname{Tr}\left(\Pi(\mathrm{err})\Pi(f)^*\right). \tag{5}$$

Since the Fourier transform is linear and $\Pi(P_U) = 0$ for any non-trivial irrep Π (Theorem 40), and $\Pi(P_T) = M_\Pi^T$ where M_Π is the Fourier transform of the single-step distribution (Theorem 35), the expression simplifies to a sum over non-trivial irreps:

$$\frac{1}{N} \cdot \frac{1}{|G|} \sum_{\Pi\neq\mathrm{triv}} d_\Pi \operatorname{Tr}\left(M_\Pi^T \Pi(f)^*\right).$$

Finally, using Theorem 51, we see that all terms of the sum vanish except for the contribution of the irrep $\mathrm{std} \otimes \mathrm{std}$, which gives

$$\begin{aligned} \text{Error Term} &= \frac{1}{N \cdot (N!)^2} \cdot \frac{(N!)^2}{N-1} \cdot (N-1)^2 \operatorname{Tr}(M_{\Pi_0}^T P_{\mathrm{diag}}) \\ &= \frac{N-1}{N} \cdot \frac{1}{\|\boldsymbol{v}\|^2} \operatorname{Tr}\left(M_{\Pi_0}^T \boldsymbol{v}\boldsymbol{v}^\top\right) \\ &= \frac{1}{N} \boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v}, \end{aligned} \tag{6}$$

where the last equality follows from the cyclic property of the trace and the fact that $\|\boldsymbol{v}\|^2 = N-1$. Substituting Equation (4) and Equation (6) into Equation (3) we obtain

$$P_{\text{agree}} = \frac{1}{N} + \frac{1}{N}\boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v},$$

as required. ■

Appendix D. Proofs from Section 5

In this section, we provide detailed proofs of the key results presented in Section 5: Theorem 7, Theorem 8, and Theorem 9. For the proof of Theorem 8, we make use of the following version of the Matrix Bernstein inequality for Hermitian matrices:

Lemma 52 (Matrix Bernstein, Theorem 6.6.1 in Tropp (2015)) *Let $Z_1, \ldots, Z_m$ be independent random $d \times d$ Hermitian matrices with $\mathbb{E}[Z_i] = 0$ and $\|Z_i\|_2 \leq B$ a.s. Let $v = \left\|\sum_{i=1}^m \mathbb{E}[Z_i^2]\right\|_2$ be the matrix variance parameter. Then for any $t > 0$:*

$$\mathbb{P}\left(\left\|\sum_{i=1}^m Z_i\right\|_2 > t\right) \leq d \cdot \exp\left(\frac{-t^2}{2v + \frac{2}{3}Bt}\right).$$

The statements of the results are restated below for convenience.

Lemma 53 *Let $N \geq 4$ and consider a randomized (k, M)-shuffle family $\mathcal{F}$ with $M = N!$ and*

$$k \geq \frac{16(3N+1)}{3(N-1)}\left[N \ln N + \ln\binom{N!}{2} + 2\ln(N-1)\right].$$

Then any two distinct semiautomata $\delta_i, \delta_j \in \mathcal{F}$ satisfy $\left\|M_{\Pi_0}^{i,j}\right\|_2 \leq 1 - \frac{1}{2N}$ with probability at least $1 - \exp(-N \ln N)$ where $\Pi_0 = \text{std} \otimes \text{std}$ and $M_{\Pi_0}^{i,j}$ denotes the Fourier transform of the single-step probability distribution for the joint walk according to δ_i and δ_j.

Proof Our proof can be summarized in two key steps: first, we compute the operator norm of the expectation of the Fourier transform of the single-step probability distribution; subsequently, we use concentration arguments to show that for the particular choice of k and M, the norm of the actual operator concentrates around this norm.

Let $\rho = \text{std}$ and take $\delta_i, \delta_j \in \mathcal{F}$ with $i \neq j$. To avoid cluttering, we denote the state-transition function for each character $\tau \in \Sigma_k$ by $\delta_i(\tau)$. The expected Fourier transform of the single-step

probability distribution for the joint walk according to δ_i and δ_j is given by

$$\begin{aligned}
\mathbb{E}[M^{i,j}_{\Pi_0}] &= \frac{1}{|\Sigma_k|}\sum_{\tau\in\Sigma_k}\mathbb{E}[\rho(\delta_i(\tau))\otimes\rho(\delta_j(\tau))] \\
&= \frac{1}{k|\Sigma_1|}\sum_{j=1}^{k}\sum_{\tau\in\Sigma_1}\frac{1}{4}\left(\rho(\tau)\otimes\rho(\tau)+\rho(\tau)\otimes I_{d_\rho}+I_{d_\rho}\otimes\rho(\tau)+I_{d_\rho}\otimes I_{d_\rho}\right) \\
&= \frac{1}{4|\Sigma_1|}\Bigg[\left(\sum_{\tau\in\Sigma_1}\rho(\tau)\otimes\rho(\tau)\right)+\left(\sum_{\tau\in\Sigma_1}\rho(\tau)\right)\otimes I_{d_\rho}+I_{d_\rho}\otimes\left(\sum_{\tau\in\Sigma_1}\rho(\tau)\right) \\
&\qquad\qquad +|\Sigma_1|I_{d_\rho^2}\Bigg] \\
&= \frac{1}{4|\Sigma_1|}\left\{\left(\sum_{\tau\in\Sigma_1}\rho(\tau)\otimes\rho(\tau)\right)+2c_\rho I_{d_\rho^2}+|\Sigma_1|I_{d_\rho^2}\right\} \qquad (7)
\end{aligned}$$

where $c_\rho = |\Sigma_1|r(\rho)$ is given by an application of Theorem 39. Next, let

$$S = \sum_{\tau\in\Sigma_1}\rho(\tau)\otimes\rho(\tau)$$

and consider $\pi(g) = \rho(g)\otimes\rho(g)$ for all $g\in S_N$. By Exercise 4.19 in Fulton and Harris (2013), for $N\geq 4$, π is a (non-irreducible) representation of S_N that decomposes as a direct sum of non-isomorphic irreps

$$\pi \simeq \mathrm{triv}\oplus\mathrm{std}\oplus(N-2,2)\oplus(N-2,1,1) \qquad (8)$$

where we identify representations by their corresponding partition. A short calculation shows that for all $g\in S_N$ we have

$$\pi(g)S\pi(g)^{-1} = \sum_{\tau\in\Sigma_1}\rho(g\tau g^{-1})\otimes\rho(g\tau g^{-1}),$$

and since the set of transpositions is a conjugacy class (Theorem 20), the sum above is equal to S. Hence, $\pi(g)S = S\pi(g)$ for all $g\in S_N$, and by the generalization of Schur's Lemma given in Theorem 38, we obtain that (under a change of basis) S has a block diagonal form. In particular, we have that[6]

$$S = c_{\mathrm{triv}}I_{d_{\mathrm{triv}}}\oplus c_{\mathrm{std}}I_{d_{\mathrm{std}}}\oplus c_{(N-2,2)}I_{d_{(N-2,2)}}\oplus c_{(N-2,1,1)}I_{d_{(N-2,1,1)}}.$$

The coefficients c_a can be calculated by restricting S to the underlying vector space V_a corresponding to each direct summand of Equation (8) and taking traces. Hence, for $a\in\{\mathrm{triv},\mathrm{std},(N-2,2),(N-2,1,1)\}$ we find

$$c_a d_a = \sum_{\tau\in\Sigma_1}\chi_a(\tau)$$

6. While Theorem 38 does not require a change of basis, in this case it is induced by the isomorphism between the representations in Equation (8). From this point onward, we assume that S is expressed in the new basis.

which, given that Σ_1 is the set of transpositions, simplifies to

$$c_a = |\Sigma_1| \cdot \frac{\chi_a(\tau)}{d_a} = |\Sigma_1| r(a)$$

where $\chi_a(\tau)$ is the character of a on transpositions. The eigenvalues of $\mathbb{E}[M^{i,j}_{\Pi_0}]$ are thus given by $\frac{1}{4|\Sigma_1|}(c_a + 2c_\rho + |\Sigma_1|)$ for $a \in \{\text{triv}, \text{std}, (N-2,2), (N-2,1,1)\}$. Substituting the values for c_a, we find that the eigenvalues are given by $\frac{1}{4}(r(a) + 2r(\rho) + 1)$. The value of the character ratios can be computed by invoking Theorem 46:

- For $a = \text{triv}$, we find $r(a) = 1$.
- For $a = \rho = \text{std}$, we find $r(a) = \frac{N-3}{N-1}$.
- For $a = (N-2, 2)$, we find $r(a) = \frac{N-4}{N}$.
- For $a = (N-2, 1, 1)$, we find $r(a) = \frac{N-5}{N-1}$.

By the calculations above, we have

$$\text{spec}(\mathbb{E}[M^{i,j}_{\Pi_0}]) = \left\{ \frac{N-2}{N-1}, \frac{2N-5}{2(N-1)}, \frac{N^2-3N+1}{N(N-1)}, \frac{N-3}{N-1} \right\}. \tag{9}$$

Since the expectation is Hermitian (representations can always be chosen to be unitary, see Appendix B.1.2), the operator norm of $\mathbb{E}[M^{i,j}_{\Pi_0}]$ is equal to its maximum absolute eigenvalue and hence

$$\|\mathbb{E}[M^{i,j}_{\Pi_0}]\|_2 = \frac{N-2}{N-1} = 1 - \frac{1}{N-1}. \tag{10}$$

For the last step of the proof, we will choose appropriate values for k and M and use the Matrix Bernstein inequality to derive a high probability bound on the operator norm $\|M^{i,j}_{\Pi_0}\|_2$ for a randomized (k, M)-shuffle family. Let $\mathcal{F}$ be such a family (the values k and M will be determined later) and fix $\delta_i, \delta_j \in \mathcal{F}$. From the triangle inequality, we get

$$\|M^{i,j}_{\Pi_0}\|_2 \leq \|\mathbb{E}[M^{i,j}_{\Pi_0}]\|_2 + \|M^{i,j}_{\Pi_0} - \mathbb{E}[M^{i,j}_{\Pi_0}]\|_2. \tag{11}$$

For every $\tau \in \Sigma_k$ we let

$$Z^{i,j}_\tau = \rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau)) - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]$$

and rewrite

$$M^{i,j}_{\Pi_0} - \mathbb{E}[M^{i,j}_{\Pi_0}] = \frac{1}{|\Sigma_k|} \sum_{\tau \in \Sigma_k} Z^{i,j}_\tau. \tag{12}$$

By construction, the $Z^{i,j}_\tau$ are independent, Hermitian,[7] zero-mean matrices that satisfy

$$\|Z^{i,j}_\tau\|_2 \leq \|\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))\|_2 + \|\mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]\|_2 \leq 2.$$

7. To see why, notice that since transpositions satisfy $\tau^2 = \text{id}$ we have $\delta_i(\tau)^2 = \delta_j(\tau)^2 = \text{id}$. Since ρ preserves the group operation and can be chosen to be unitary, $\rho(\delta_i(\tau))$ and $\rho(\delta_j(\tau))$ are Hermitian. The Kronecker product and the expectation of Hermitian (random) matrices are Hermitian.

We now analyze the variance parameter $v = \left\| \sum_{\tau \in \Sigma_k} \mathbb{E}\left[\left(Z_\tau^{i,j} \right)^2 \right] \right\|_2$. For every $\tau \in \Sigma_k$ we have

$$\begin{aligned}\mathbb{E}\left[\left(Z_\tau^{i,j} \right)^2 \right] &= \mathbb{E}[(\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau)))^2] - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]^2 \\ &= I_{d_\rho^2} - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]^2\end{aligned}$$

where the last equality follows from the fact that transpositions in S_N satisfy $\tau^2 = \mathrm{id}$ and hence $\delta_i(\tau)^2 = \mathrm{id}$ (regardless of the realization). Consequently,

$$\begin{aligned}(\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau)))^2 &= \rho(\delta_i(\tau))^2 \otimes \rho(\delta_j(\tau))^2 = \rho(\delta_i(\tau)^2) \otimes \rho(\delta_j(\tau)^2) \\ &= \rho(\mathrm{id}) \otimes \rho(\mathrm{id}) = I_{d_\rho^2}.\end{aligned}$$

By an application of Jensen's inequality we get $\|\mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]\|_2 \leq 1$, and since the expectation is Hermitian, we can deduce that $I_{d_\rho} - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]^2$ is positive semidefinite with eigenvalues bounded by 1. Hence,

$$\begin{aligned}v &= \left\| \sum_{\tau \in \Sigma_k} \mathbb{E}\left[\left(Z_\tau^{i,j} \right)^2 \right] \right\|_2 = \left\| \sum_{\tau \in \Sigma_k} (I_{d_\rho^2} - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]^2) \right\|_2 \\ &\leq \sum_{\tau \in \Sigma_k} \|I_{d_\rho^2} - \mathbb{E}[\rho(\delta_i(\tau)) \otimes \rho(\delta_j(\tau))]^2\|_2 \leq \sum_{\tau \in \Sigma_k} 1 = |\Sigma_k|.\end{aligned}$$

We call a randomized (k, M)-shuffle family $\mathcal{F}$ "bad" if there exist distinct semiautomata $\delta_i, \delta_j \in \mathcal{F}$ such that $\|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 > \frac{1}{2N}$, and "good" otherwise. Notice that by Equation (11), and the derivation in Equation (10), whenever $\mathcal{F}$ is a good family, every distinct pair of semiautomata $\delta_i, \delta_j \in \mathcal{F}$ satisfy

$$\|M_{\Pi_0}^{i,j}\|_2 \leq \|\mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 + \|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 < 1 - \frac{1}{N} + \frac{1}{2N} = 1 - \frac{1}{2N},$$

as required. Hence, it suffices to consider the probability of generating a bad family. By the union bound

$$\mathbb{P}(\mathcal{F} \text{ is bad}) \leq \sum_{\delta_i, \delta_j \in \mathcal{F}} \mathbb{P}\left(\|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 > \frac{1}{2N} \right).$$

Each term of the summation above can be upper-bounded by an application of the matrix Bernstein inequality. Indeed, by Equation (12) and matrix Bernstein we have

$$\begin{aligned}\mathbb{P}\left(\|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 > \frac{1}{2N} \right) &= \mathbb{P}\left(\left\| \frac{1}{|\Sigma_k|} \sum_{\tau \in \Sigma_k} Z_\tau^{i,j} \right\|_2 \geq \frac{1}{2N} \right) \\ &= \mathbb{P}\left(\left\| \sum_{\tau \in \Sigma_k} Z_\tau^{i,j} \right\|_2 \geq \frac{|\Sigma_k|}{2N} \right) \\ &\leq d_\rho^2 \cdot \exp\left(\frac{-\frac{|\Sigma_k|^2}{4N^2}}{2|\Sigma_k| + \frac{2}{3} \cdot 2 \cdot \frac{|\Sigma_k|}{2N}} \right).\end{aligned}$$

Since the right-hand side of the above inequality does not depend on δ_i and δ_j, we can bound

$$\mathbb{P}(\mathcal{F} \text{ is bad}) \leq \binom{N!}{2} \cdot d_\rho^2 \cdot \exp\left(\frac{-\frac{|\Sigma_k|^2}{4N^2}}{2|\Sigma_k| + \frac{2}{3} \cdot 2 \cdot \frac{|\Sigma_k|}{2N}}\right)$$

Substituting $d_\rho = N - 1$ and $|\Sigma_k| = k\binom{N}{2}$, and after some algebraic manipulations, we find that taking

$$k \geq \frac{16(3N+1)}{3(N-1)}\left[N \ln N + \ln\binom{N!}{2} + 2\ln(N-1)\right]$$

guarantees that the probability of $\mathcal{F}$ being bad is upper bounded by $\exp(-N \ln N)$, concluding the proof. ■

Lemma 54 *For $M = N!$ and $N \geq 4$, if we choose the alphabet parameter*

$$k \geq \frac{16(3N+1)}{3(N-1)}\left[N \ln N + \ln\binom{N!}{2} + 2\ln(N-1)\right],$$

then any pair of semiautomata (δ_i, δ_j) from a randomized (k, M)-shuffle family $\mathcal{F}$ satisfies

$$\left|P_{\text{agree}} - \frac{1}{N}\right| \leq \left(1 - \frac{1}{2N}\right)^T$$

with probability at least $1 - \exp(-N \ln N)$.

Proof Let $\delta_i, \delta_j \in \mathcal{F}$ with $i \neq j$. By Theorem 5 we have

$$P_{\text{agree}} = \frac{1}{N} + \frac{1}{N}\boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v}$$

where $\boldsymbol{v} = \sum_{i=1}^{N-1} \boldsymbol{e_i} \otimes \boldsymbol{e_i}$. By Theorem 7, for the choice of k assumed, with probability $1 - \exp(-N \ln N)$ we have

$$\left|P_{\text{agree}} - \frac{1}{N}\right| = \frac{|\boldsymbol{v}^\top M_{\Pi_0}^T \boldsymbol{v}|}{N} \leq \frac{\|M_{\Pi_0}^T\|_2 \cdot \|\boldsymbol{v}\|^2}{N} \leq \left(1 - \frac{1}{2N}\right)^T \frac{N-1}{N} \leq \left(1 - \frac{1}{2N}\right)^T.$$

This concludes the proof. ■

Theorem 9 *For $T \geq 2N \ln(N!)$ and k and M as given in Theorem 8, we have that any pair of distinct semiautomata (δ_i, δ_j) from a randomized (k, M)-shuffle family $\mathcal{F}$ satisfies $|P_{\text{agree}} - 1/N| \leq 1/N!$ with probability at least $1 - \exp(-N \ln N)$.*

Proof From the bound of Theorem 8, the inequality $1 - x \leq e^{-x}$ which holds for all $x \in \mathbb{R}$, and the choice of $T \geq 2N \ln(N!)$, we get:

$$\left|P_{\text{agree}} - \frac{1}{N}\right| \leq \left(1 - \frac{1}{2N}\right)^T \leq \exp\left(-\frac{T}{2N}\right) \leq \exp\left(-\frac{2N \ln(N!)}{2N}\right) = \frac{1}{N!},$$

concluding the proof. ■

Appendix E. Proof of Theorem 12

In this section, we prove Theorem 12, which we restate for convenience:

Theorem 12 (Tightness of mixing time) *Let $N \geq 5$, and let k and M be as in Theorem 8. For any pair of distinct semiautomata (δ_i, δ_j) from a randomized (k, M)-shuffle family to satisfy $|P_{\text{agree}} - 1/N| \leq 1/N!$ with probability $1 - \exp(-N \ln N)$, the input word length must be at least $T = \Omega(N^2 \ln N)$.*

The proof requires deriving a lower bound on $|P_{\text{agree}} - 1/N|$ that decays exponentially in T, which is given by the following lemma:

Lemma 55 *For $M = N!$ and $N \geq 5$, if we choose the alphabet parameter*

$$k \geq \frac{16(3N+1)}{3(N-1)} \left[N \ln N + \ln \binom{N!}{2} + 2 \ln(N-1) \right],$$

then any pair of semiautomata (δ_i, δ_j) from a randomized (k, M)-shuffle family $\mathcal{F}$ satisfies

$$\left| P_{\text{agree}} - \frac{1}{N} \right| \geq \frac{1}{2} \left(1 - \frac{3}{N} \right)^T$$

with probability at least $1 - \exp(-N \ln N)$.

Proof From Equation (9), the minimum eigenvalue of the expected operator $\mathbb{E}[M_{\Pi_0}^{i,j}]$ is given by $\lambda_{\min}(\mathbb{E}[M_{\Pi_0}^{i,j}]) = \frac{N-3}{N-1}$. Furthermore, we have shown that, for k chosen as in the statement, with probability at least $1 - \exp(-N \ln N)$, the deviation $\|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2$ satisfies

$$\|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 \leq \frac{1}{2N}.$$

By Weyl's inequality, with probability at least $1 - \exp(-N \ln N)$, it holds

$$\lambda_{\min}(M_{\Pi_0}) \geq \lambda_{\min}(\mathbb{E}[M_{\Pi_0}^{i,j}]) - \|M_{\Pi_0}^{i,j} - \mathbb{E}[M_{\Pi_0}^{i,j}]\|_2 \geq \frac{N-3}{N-1} - \frac{1}{2N}. \tag{13}$$

For $N \geq 4$, the right-hand side of the above inequality is positive, and so $M_{\Pi_0}^{i,j}$ is positive definite, in which case

$$\boldsymbol{v}^\top \left(M_{\Pi}^{i,j} \right)^T \boldsymbol{v} \geq \left(\lambda_{\min}(M_{\Pi}^{i,j}) \right)^T \cdot \|\boldsymbol{v}\|^2$$

Substituting $\|\boldsymbol{v}\|^2 = N - 1$ and the lower bound for the minimum eigenvalue obtained in Equation (13), we obtain

$$\left| P_{\text{agree}} - \frac{1}{N} \right| = \frac{\boldsymbol{v}^\top \left(M_{\Pi}^{i,j} \right)^T \boldsymbol{v}^\top}{N} \geq \frac{N-1}{N} \left(\frac{N-3}{N-1} - \frac{1}{2N} \right)^T \geq \frac{1}{2} \left(1 - \frac{3}{N} \right)^T,$$

where the last inequality is valid for $N \geq 5$. This concludes the proof. ■

Using the lower bound established in Theorem 55, we are now ready to prove Theorem 12:

Proof of Theorem 12 By Theorem 55, T must satisfy

$$\frac{1}{2}\left(1-\frac{3}{N}\right)^T \leq \frac{1}{N!}.$$

Solving for T we obtain

$$T \geq \frac{\ln(N!/2)}{\ln\left(\frac{N}{N-3}\right)}.$$

The numerator is $\Theta(N \ln N)$ while a Taylor approximation on the denominator shows that for $N \gg 1$:

$$\ln\left(\frac{N}{N-3}\right) = -\ln\left(1-\frac{3}{N}\right) \approx \frac{3}{N}.$$

Thus, the denominator is $\Theta(1/N)$, which shows that $T = \Omega(N^2 \ln N)$. ■

Appendix F. Proof of Theorem 15

In this section, we prove Theorem 15, by generalizing the standard argument for the case $\mathcal{Y} = \{0,1\}$ (e.g. Theorem 2 in Szörényi (2009)). For convenience, we restate the theorem first:

Theorem 15 (SQ lower bound) *Let $\mathcal{C}$ be a concept class and suppose $\mathrm{SQDim}_{\mathcal{C}}^{D} \geq d$. Then any SQ learner using tolerance $\tau > 0$ requires, in the worst case, at least*

$$q \geq \frac{(d-1)(d\tau^2 - |\mathcal{Y}|)}{2d(|\mathcal{Y}|-1)}$$

queries to learn the ground-truth concept f^*.

Proof We represent each concept $f \in \mathcal{C}$ as a vector-valued function $\boldsymbol{u}_f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ given by $\boldsymbol{u}_f(x) = \boldsymbol{e}_{f(x)} - \bar{\boldsymbol{e}}$ where $\boldsymbol{e}_y$ is the standard basis vector corresponding to label y and $\bar{\boldsymbol{e}}$ is the vector with all entries equal to $1/|\mathcal{Y}|$. It is easy to verify that by the linearity of expectation, for any $f, g \in \mathcal{C}$ we have $\chi(f,g) = \langle \boldsymbol{u}_f, \boldsymbol{u}_g \rangle_D$ where the $L^2(\mathcal{X}, \mathbb{R}^{|\mathcal{Y}|})$ inner product $\langle \cdot, \cdot \rangle_D$ is defined as

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_D := \mathbb{E}_{x \sim D}\left[\langle \boldsymbol{u}(x), \boldsymbol{v}(x) \rangle\right].$$

Assume that $f_1, \ldots, f_d \in \mathcal{C}$ fulfill $|\chi(f_i, f_j)| \leq 1/d$ for all $i, j \in [d]$ with $i \neq j$. To discharge notation, we will write $\boldsymbol{u}_i$ instead of $\boldsymbol{u}_{f_i}$ to refer to the representation defined above. By the previous discussion, we have $|\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle_D| \leq 1/d$ for all $i, j \in [d]$ with $i \neq j$. We present an adversarial answering strategy for the oracle that guarantees that the learner can eliminate only a small number of concepts after every answer when the ground-truth concept f^* is one of the f_i's.

Let $h : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$ be an arbitrary query function used by the learner. For each candidate concept f_i, define

$$v_i := \mathbb{E}_{x \sim D}\left[h(x, f_i(x))\right].$$

The learner requests the value of the expectation

$$v^* := \mathbb{E}_{x\sim D}\left[h(x, f^*(x))\right]$$

and the oracle returns an answer $\hat{v}$ such that $|v^* - \hat{v}| \leq \tau$. Using this information, the learner can eliminate any f_i for which $|v_i - \hat{v}| > \tau$. We represent h by the vector-valued function $\boldsymbol{H} : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ where the y-th component of $\boldsymbol{H}(x)$ is equal to $h(x, y)$. Since $h(x, f_i(x)) = \langle \boldsymbol{H}(x), \boldsymbol{e}_{f_i(x)}\rangle$, we have:

$$v_i = \mathbb{E}_{x\sim D}[\langle \boldsymbol{H}(x), \boldsymbol{e}_{f_i(x)}\rangle] = \langle \boldsymbol{H}, \boldsymbol{e}_{f_i}\rangle_D.$$

Since $\boldsymbol{u}_i(x) = \boldsymbol{e}_{f_i(x)} - \bar{\boldsymbol{e}}$, we can rearrange this to write $\boldsymbol{e}_{f_i(x)} = \boldsymbol{u}_i(x) + \bar{\boldsymbol{e}}$. Substituting this into the previous expression for v_i:

$$v_i = \langle \boldsymbol{H}, \boldsymbol{u}_i + \bar{\boldsymbol{e}}\rangle_D = \langle \boldsymbol{H}, \boldsymbol{u}_i\rangle_D + \langle \boldsymbol{H}, \bar{\boldsymbol{e}}\rangle_D.$$

Consider the adversarial answering strategy where the oracle responds with $\hat{v} = \langle \boldsymbol{H}, \bar{\boldsymbol{e}}\rangle_D$. As such, the learner eliminates all concepts f_i for which $|\langle \boldsymbol{H}, \boldsymbol{u}_i\rangle_D| > \tau$. Under this answering strategy, we can count how many candidates the learner can eliminate with each query. Define $A^+ = \{i \in [d] : \langle \boldsymbol{H}, \boldsymbol{u}_i\rangle_D \geq \tau\}$ and $A^- = \{i \in [d] : \langle \boldsymbol{H}, \boldsymbol{u}_i\rangle_D \leq -\tau\}$, and notice that the number of eliminated candidates is precisely $|A^+| + |A^-|$. To upper bound the cardinality of A^+, we apply the Cauchy-Schwartz inequality to obtain:

$$\left\langle \boldsymbol{H}, \sum_{i\in A^+} \boldsymbol{u}_i \right\rangle_D^2 \leq \|\boldsymbol{H}\|_D^2 \cdot \left\| \sum_{i\in A^+} \boldsymbol{u}_i \right\|_D^2. \tag{14}$$

By the definition of A^+, the left-hand side of Equation (14) is at least $(|A^+|\tau)^2$. On the other hand, since $|h(x, y)| \leq 1$ we have

$$\|\boldsymbol{H}\|_D^2 = \mathbb{E}_{x\sim D}\left[\sum_{y\in\mathcal{Y}} h(x, y)^2\right] \leq |\mathcal{Y}|,$$

and

$$\left\| \sum_{i\in A^+} \boldsymbol{u}_i \right\|_D^2 = \sum_{i,j\in A^+} \langle \boldsymbol{u}_i, \boldsymbol{u}_j\rangle_D = \sum_{i\in A^+} \|\boldsymbol{u}_i\|_D^2 + \sum_{i\neq j} \langle \boldsymbol{u}_i, \boldsymbol{u}_j\rangle_D \leq |A^+|(1 - 1/|\mathcal{Y}|) + \frac{|A^+|^2}{d}$$

where in the last inequality we used the fact that $\|\boldsymbol{u}_i\|_D^2 = \chi(f_i, f_i) = 1 - 1/|\mathcal{Y}|$. Chaining the inequalities, we get

$$(|A^+|\tau)^2 \leq |\mathcal{Y}| \left(|A^+|(1 - 1/|\mathcal{Y}) + \frac{|A^+|^2}{d} \right).$$

Dividing by $|A^+|$ and rearranging yields:

$$|A^+| \leq \frac{d(|\mathcal{Y}| - 1)}{d\tau^2 - |\mathcal{Y}|}.$$

A similar procedure for A^- shows that the number of eliminated concepts after the query is at most $\frac{2d(|\mathcal{Y}|-1)}{d\tau^2-|\mathcal{Y}|}$ if the adversary returns $\langle \boldsymbol{H}, \bar{\boldsymbol{e}}\rangle_D$. Thus, the learner requires at least $\frac{(d-1)(d\tau^2-|\mathcal{Y}|)}{2d(|\mathcal{Y}|-1)}$ queries to identify the ground-truth concept f^*. ■