Approximate Minimization of Weighted Tree Automata

Borja Balle* Guillaume Rabusseau†
Mila and DIRO
Université de Montréal

September 15, 2019

Abstract

This paper studies the following *approximate minimization* problem: given a minimal weighted tree automaton A with n states recognizing a weighted tree language f, can we construct a smaller automaton $\hat{A}$ with $\hat{n} < n$ states recognizing a language $\hat{f}$ that is a good approximation of f? The corresponding problem for weighted automata on strings was recently studied by Balle et al. [16, 15], where the authors introduced a new canonical form for weighted automata called *singular value automata* inspired by spectral methods, and showed that truncating such canonical form yields a solution for the problem satisfying a certain approximation criteria. In this paper we take a similar approach and show that in the tree case one can obtain an analogous canonical form that we call *singular value tree automata*, and use it to study the approximate minimization problem for weighted tree automata. We first establish the existence of this canonical form for weighted tree automata and then provide bounds on the quality of the resulting approximation method based on truncation. We also study the problem of computing the canonical form given a minimal weighted tree automaton and show that in the tree case this task is considerably more complicated than in the string case. In particular, computing the canonical form reduces to solving a system of polynomial equations. By further reducing this problem to the computation of *generalized partition functions* for weighted tree automata, we propose and analyze two methods for computing the canonical form based on iterative methods: fixed point iteration and Newton's method. Our analysis of Newton's method unveils a connection between iterative methods and sequences of sets of trees satisfying a certain technical condition that might be of independent interest.

1 Introduction

Weighted tree automata (WTA) are a natural and powerful generalization of weighted (string) automata (WA) first considered in [19]. In the same way that weighted automata extend deterministic finite automata recognizing regular languages to a more general class of models that can recognize regular stochastic and weighted languages, weighted tree automata extend classical tree automata and closely related context-free grammars to the weighted case. The expressive power of WTA and their relation to other unweighted and weighted models, including probabilistic and weighted context-free grammars, have been thoroughly studied in the literature (see [34] and references there-in). Broadly speaking, results about the expressive power of classes of automata come in two flavors. One class of results provides a qualitative understanding of the separations between classes of languages that can and cannot be recognized by certain types of machines. The other class of results is more quantitative in nature: it studies the expressiveness of objects within a class as a function of their complexity, typically measured by the number of states required to capture a certain phenomenon.

A fundamental quantitative result about the expressiveness of WA with weights over a field is provided by Fliess' theorem [32], which can be seen as an generalization of the classic Myhill–Nerode theorem to the weighted case. This result exactly characterizes those weighted languages that can be recognized by a WA with n states as those whose corresponding infinite Hankel matrix has rank n. Fliess' theorem also provides a convenient characterization of minimality for WA: an automaton with n states is minimal if and only if the Hankel matrix of the language it recognizes has rank n. The analog of Fliess' theorem for WTA is due to Bozapalidis and Louscou-Bozapalidou [21]. As in the WA case, the theorem of Bozapalidis and Louscou-Bozapalidou provides information about the realizability of weighted tree languages and

*Work done while at Lancaster University.
†Canada CIFAR AI chair

minimality of WTA. The main difference is that, in the tree-case, the Hankel matrix of Fliess', whose rows and columns are both indexed by finite strings, is replaced by a generalized Hankel matrix where rows are indexed by context trees and columns are indexed by normal trees (see Section 3 for a precise definition).

Beyond its pure language-theoretic interest, Fliess' theorem for WA also has found a number of applications in machine learning. The starting point of these developments stems from the realization that constructive proofs of Fliess' theorem can be interpreted as algorithms for recovering a WA from (a finite sub-block of) the Hankel matrix corresponding to a weighted language. Implicitly, this idea was first used by Bergadano and Varricchio [18] to obtain algorithms for learning WA with weights on a field[1] in the exact learning with queries model introduced by Angluin and others [4]. More recently, a detailed analysis of noise-tolerant algorithms for recovering a WA from a (noisy) Hankel matrix have lead to PAC learning results in the sense of Valiant [51] (see also Kearns et al. [37]) for stochastic WA and other classes of WA with weights over the reals [8, 10]. A survey of this line of work can be found in [14].

The cornerstone needed to make WA learning algorithms robust to noise is to compute spectral decompositions of Hankel matrices and leverage the inherent ability of spectral representations to separate signal from noise in the learning process. This is a classic idea in machine learning and statistics that underlies techniques as fundamental as principal component analysis. However, it has only been during the last decade that spectral methods have been widely applied to models with latent variables, including WA [8, 10, 11], hidden Markov models [36, 49], and many others [44, 3, 1, 50, 43, 2], under the umbrella term of spectral learning. From a theoretical standpoint, the main interest in these algorithms resides in the fact that in most cases they provide the only known method for learning classes of practically relevant models with guaranteed polynomial running time and sample complexity.

In addition to their theoretical interest, spectral methods for learning WA and related models have been successfully applied to a variety of learning tasks involving real data. These tasks include important application domains, like reinforcement learning, natural language processing and time-series prediction. For example, in the context of reinforcement learning a number of spectral algorithms have been applied to model-based learning and planning with partial observations [20, 5, 12]. In natural language processing, spectral methods have been used for language modelling [10, 11], parsing [25, 41], and transcription and transliteration tasks [17, 7, 47, 45]. A puzzling fact, however, is that the practical success of spectral learning cannot be fully explained by existing theoretical analysis. Indeed, with notably few exceptions [13], existing theoretical analysis of spectral learning only work under the assumption that the data given to the learning algorithm was generated by an unknown automata of some form, and this is obviously not the case in most of the applications listed above.

Motivated by this apparent mismatch between the practical success of spectral learning and its theoretical guarantees, Balle et al. [15, 16] initiated a study of *approximate* minimization of weighted automata in the string case. These works stem from the realization that spectral learning algorithms can be interpreted as first computing a minimal WA that exactly explains the training data, and then trying to find a smaller WA that approximates this exact WA in order to obtain a model that generalizes to previously unseen data. Towards the goal of better understanding this approximation step, Balle et al. unveiled a connection between spectral learning algorithms and a novel canonical form for WA called *singular value automata* (SVA). The theory of SVA in the string case provides a way to implicitly compute the singular value decomposition of an infinite Hankel matrix, and yields bounds on the quality of approximate minimization algorithms for WA by SVA truncation. In particular, the approximation bounds obtained in [15, 16] can be seen as a first step towards quantifying the accuracy of spectral learning for WA from data not necessarily generated by an underlying automaton. Interestingly, similar bounds were also obtained independently in [40].

The present paper follows on the footsteps of [15, 16], and shows that most of the theory developed for SVA in the string case can be extended to the tree case. Our work provides a novel canonical form for WTA which we call *singular value tree automata* (SVTA). The existence of this canonical form follows from a spectral decomposition of the generalized (tree) Hankel matrix of Bozapalidis and Louscou-Bozapalidou. In the string case, computing the SVA form can be achieved by solving a system of linear equations. In the tree case, however, computing the SVTA form of a given WTA turns out to be a substantially more complicated problem, since it requires finding a solution to a system of *polynomial* equations. To solve the problem we propose two iterative algorithms that can be used to compute an SVTA form to arbitrary accuracy. Finally, we consider the approximate minimization problem for WTA and provide the first approximation bounds based on SVTA truncation. Again, the situation here is

[1] These type of WA are sometimes called *multiplicity automata* in the literature.

considerably more involved than in the string case, where the approximation error can be characterized in closed form. Instead, we obtain our bounds as a function of the size of the trees being considered by carefully analyzing the propagation of approximation errors through the computation performed by the truncated SVTA. From a learning point of view, our theory of SVTA computation and truncation provides an approximate minimization perspective on recent works on learning WTA and related models like weighted and probabilistic context-free grammars using spectral methods [9, 6, 24, 23].

After recalling the necessary background and preliminaries in Section 2, we proceed to present our contributions. These can be summarized as follows:

- *We introduce SVTA as a canonical form for WTA* in Section 3, thus generalizing the theory presented in [16] from strings to trees. We show under which conditions a WTA can be brought into SVTA canonical form and *we present an algorithm to compute an SVTA from such an WTA*. Similarly to [16], our approach to compute SVTA canonical forms reduces to the computation of the so-called *Gramian matrices* of the WTA.

- *We present an algorithm to approximate the Gramians of a WTA to an arbitrary precision* in Section 5. This algorithm leverages the fact that the problem of computing the Gramians can be reduced to the one of computing the *generalized partition function* of a WTA — a particular vector-valued series defined by the WTA.

- In Section 4, *we present two iterative algorithms to compute the generalized partition function of a WTA and analyze their convergence rate.* The first algorithm is based on a simple fixed point iteration and the second, which achieves a faster convergence rate, is based on Newton's method. While the proof of convergence and analysis of the fixed point method is relatively straightforward, the one of Newton's method unravels an elaborate construction which may be of independent interest. In particular, this construction gives combinatorial insights on the faster convergence rate obtained by Newton's method in this context and bares striking similarities with the work of Esparza et al. on *Newtonian program analysis* [29] for computing fixed points of system of equations over ω-continuous semi-rings (see Section 7 for a discussion of this connection).

- Leveraging the fact that the states of an SVTA are in one-to-one correspondence with the singular values of the Hankel matrix, *we propose a principled method for reducing the number of states of a WTA* to approximate a recognizable tree function by a model with a smaller size in Section 6. This method consists in first converting the WTA into SVTA canonical form before removing the states corresponding to the smaller singular values. *We also provide an analysis of the error introduced by this approximation scheme.*

Finally, we present our conclusions and future work in Section 7. Some of our results appeared in preliminary form in in the conference paper [46], where applications of approximate minimization of WTA to speed up parsing algorithms for weighted context-free grammars were considered. The SVTA theory developed in [46] is limited to WTA defined over binary rooted trees with leaves labeled by symbols from a finite alphabet. Here we extend this theory to WTA over rooted trees of arbitrary arity labeled with symbols from a finite ranked alphabet[2]. We also present improved analysis and faster algorithms for computing the SVTA form of a given WTA. In particular, using Newton's method for computing the Gramians was not considered in [46]. The approximation bounds presented in Section 6 are also new.

2 Background and Preliminaries

We start by introducing our notations and presenting the necessary background on linear and multilinear algebra before recalling the definitions of trees and weighted tree automata. The main notations used throughout the paper are summarized in Table 1 for convenience.

2.1 Linear and Multilinear Algebra

For any integer p, we denote by $[p]$ the set of integers from 1 to p. We use bold letters to denote vectors $\mathbf{v} \in \mathbb{R}^d$ and matrices $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$. Unless explicitly stated, all vectors are column vectors. We write $\mathbf{I}$ for the identity matrix and $\mathrm{diag}(a_1, \ldots, a_n)$ for a diagonal matrix with $a_1, \ldots, a_n$ in the diagonal. For a matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$, $i \in [d_1]$, and $j \in [d_2]$, we use $\mathbf{M}_{i,:}$ and $\mathbf{M}_{:,j}$ to denote the ith row and the jth

[2] This extension as well as the approximation bounds in Section 6 were initially presented in the PhD thesis [**?**].

column of $\mathbf{M}$ respectively. We will sometime use the notations $\mathbf{M}(i,j) = \mathbf{M}_{i,j}$, $\mathbf{M}(i,:) = \mathbf{M}_{i,:}$, etc. to avoid doubling indices. Given a matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ we denote by $\mathrm{vec}(\mathbf{M}) \in \mathbb{R}^{d_1 \cdot d_2}$ the vector obtained by concatenating the columns of $\mathbf{M}$ so that $\mathrm{vec}(\mathbf{M})_{(i-1)d_2+j} = \mathbf{M}_{i,j}$. Given two matrices $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ and $\mathbf{M}' \in \mathbb{R}^{d_1' \times d_2'}$ we denote their Kronecker product by $\mathbf{M} \otimes \mathbf{M}' \in \mathbb{R}^{d_1 d_1' \times d_2 d_2'}$, with entries given by $(\mathbf{M} \otimes \mathbf{M}')_{(i-1)d_1'+i',(j-1)d_2'+j'} = \mathbf{M}_{i,j}\mathbf{M}'_{i',j'}$, where $i \in [d_1]$, $j \in [d_2]$, $i' \in [d_1']$, and $j' \in [d_2']$. A *rank factorization* of a rank n matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ is an expression of the form $\mathbf{M} = \mathbf{Q}\mathbf{R}$ where $\mathbf{Q} \in \mathbb{R}^{d_1 \times n}$ and $\mathbf{R} \in \mathbb{R}^{n \times d_2}$ are full-rank matrices; i.e. $\mathrm{rank}(\mathbf{Q}) = \mathrm{rank}(\mathbf{R}) = \mathrm{rank}(\mathbf{M}) = n$.

Given a matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ of rank n, its *singular value decomposition* (SVD)[3] is a decomposition of the form $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{d_1 \times n}$, $\mathbf{D} \in \mathbb{R}^{n \times n}$, and $\mathbf{V} \in \mathbb{R}^{d_2 \times n}$ are such that: $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$, and $\mathbf{D} = \mathrm{diag}(\mathfrak{s}_1, \dots, \mathfrak{s}_n)$ with $\mathfrak{s}_1 \geq \cdots \geq \mathfrak{s}_n > 0$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are thus orthonormal and are called left and right *singular vectors* respectively, and the $\mathfrak{s}_i$ are its *singular values*. The SVD is unique (up to sign changes in associate singular vectors) whenever all inequalities between singular values are strict.

We now recall basic definitions of tensor algebra; more details can be found in [39]. A *tensor* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ can simply be seen as a multidimensional array $(\boldsymbol{\mathcal{T}}_{i_1,\dots,i_p} : i_n \in [d_n], n \in [p])$. We will mostly use hyper-cubic tensors in this work, i.e. $d_1 = \cdots = d_p = d$, and we denote the space of d-dimensional hyper-cubic tensors of order p by $(\mathbb{R}^d)^{\otimes p}$. The *mode-$n$* fibers of a tensor $\boldsymbol{\mathcal{T}} \in (\mathbb{R}^d)^{\otimes p}$ are the vectors obtained by fixing all indices except the nth one, e.g. the mode-1 fibers of $\boldsymbol{\mathcal{T}}$ are the vectors $\boldsymbol{\mathcal{T}}_{:,i_2,\dots,i_p} \in \mathbb{R}^d$ for $i_2, \dots, i_p \in [d]$. The *nth mode matricization* of $\boldsymbol{\mathcal{T}}$ is the matrix having the mode-n fibers of $\boldsymbol{\mathcal{T}}$ for columns and is denoted by $\mathbf{T}_{(n)} \in \mathbb{R}^{d \times d^{p-1}}$. The vectorization of a tensor is defined by $\mathrm{vec}(\boldsymbol{\mathcal{T}}) = \mathrm{vec}(\mathbf{T}_{(1)})$.

Given a tensor $\boldsymbol{\mathcal{T}} \in (\mathbb{R}^d)^{\otimes p}$ and matrices $\mathbf{M}_i \in \mathbb{R}^{d \times d_i}$ for $i \in [p]$, we define the tensor $\boldsymbol{\mathcal{T}}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_p) \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_p}$ whose entries are given by

$$\boldsymbol{\mathcal{T}}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_p)_{i_1,i_2,\dots,i_p} = \sum_{j_1,j_2,\dots,j_p} \boldsymbol{\mathcal{T}}_{j_1,j_2,\dots,j_p} (\mathbf{M}_1)_{j_1,i_1} (\mathbf{M}_2)_{j_2,i_2} \dots (\mathbf{M}_p)_{j_p,i_p} \ .$$

This operation corresponds to contracting $\boldsymbol{\mathcal{T}}$ with $\mathbf{M}_i$ across the ith mode of the tensor for each i. Observe that for the case of an order 2 tensor $\mathbf{T}$ (i.e. a matrix), we have $\mathbf{T}(\mathbf{M}_1, \mathbf{M}_2) = \mathbf{M}_1^\top \mathbf{T} \mathbf{M}_2$. In particular, if we take $\mathbf{M}_1$ to be the identity matrix we can identify the matrix $\mathbf{T}$ with the linear map $\mathbf{v} \mapsto \mathbf{T}(\mathbf{I}, \mathbf{v}) = \mathbf{T}\mathbf{v}$. Similarly, we can identify a tensor $\boldsymbol{\mathcal{T}} \in (\mathbb{R}^d)^{\otimes (p+1)}$ with the multilinear map $f : (\mathbb{R}^d)^p \to \mathbb{R}^d$ defined by $f(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p) = \boldsymbol{\mathcal{T}}(\mathbf{I}, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p)$. Given a tensor $\boldsymbol{\mathcal{T}} \in (\mathbb{R}^d)^{\otimes 3}$ and matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{R}^{d \times m}$ and $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3 \in \mathbb{R}^{m \times n}$ we have the following useful identity

$$(\boldsymbol{\mathcal{T}}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3))(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3) = \boldsymbol{\mathcal{T}}(\mathbf{A}_1\mathbf{B}_1, \mathbf{A}_2\mathbf{B}_2, \mathbf{A}_3\mathbf{B}_3)$$

which can easily be generalized to higher-order tensors.

2.2 Trees and Weighted Tree Automata

Trees on a ranked alphabet. We now introduce notations for describing trees generated by a weighted tree automaton; see Figure 1 for some illustrative examples.

A *ranked alphabet* is a tuple $\mathcal{F} = (\Sigma, \sharp)$ where Σ is a finite alphabet and $\sharp : \Sigma \to \mathbb{N}$ is an arity function. We denote by $\mathcal{F}_p = \{g \in \Sigma : \sharp g = p\}$ the set of symbols with arity p. Similarly, we will denote by $\mathcal{F}_{\leq p}$ (resp. $\mathcal{F}_{\geq p}$) the set of symbols with arity at most p (resp. at least p).

The set of trees $\mathfrak{T}_\mathcal{F}$ on a ranked alphabet $\mathcal{F}$ is the smallest set satisfying

- $\sigma \in \mathfrak{T}_\mathcal{F}$ for any $\sigma \in \mathcal{F}_0$,
- $g(t_1, \dots, t_p) \in \mathfrak{T}_\mathcal{F}$ for any $p \geq 1$, $g \in \mathcal{F}_p$ and $t_1, \dots, t_p \in \mathfrak{T}_\mathcal{F}$.

We will call symbols in $\mathcal{F}_0$ *leaf symbols* and symbols in $\mathcal{F}_{\geq 1}$ *internal symbols*. We will sometimes simply write $\mathfrak{T}$ instead of $\mathfrak{T}_\mathcal{F}$ when the ranked alphabet is clear from context. The *size* of a tree $t \in \mathfrak{T}_\mathcal{F}$ is denoted by $|t|$ and defined recursively by $|\sigma| = 1$ for $\sigma \in \mathcal{F}_0$, and $|g(t_1, \dots, t_p)| = 1 + |t_1| + \cdots + |t_p|$; that is, the number of nodes in the tree. Given a symbol g and a tree t we will denote by $|t|_g$ the number of nodes in t that are labeled with g. The *depth* of a tree $t \in \mathfrak{T}_\mathcal{F}$ is denoted by $\mathrm{depth}(t)$ and defined recursively by $\mathrm{depth}(\sigma) = 0$ for $\sigma \in \mathcal{F}_0$, and $\mathrm{depth}(g(t_1, \dots, t_p)) = 1 + \max\{\mathrm{depth}(t_1), \dots, \mathrm{depth}(t_p)\}$;

[3] To be more precise, this is a *compact* singular value decomposition, since the inner dimensions of the decomposition are all equal to the rank. In this paper we shall always use the term SVD to mean compact SVD.

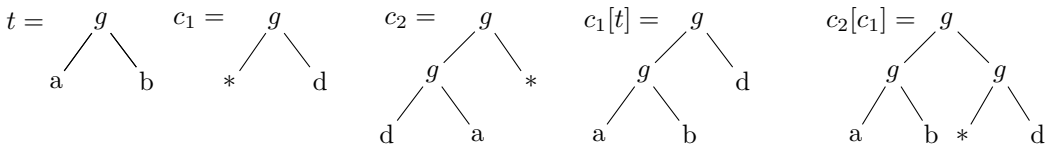

Figure 1: Examples of trees ($t, c_1[t] \in \mathfrak{T}_{\mathcal{F}}$) and contexts ($c_1, c_2, c_2[c_1] \in \mathfrak{C}_{\mathcal{F}}$) on the ranked alphabet $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_2$ where $\mathcal{F}_0 = \{a, b, d\}$ and $\mathcal{F}_2 = \{g\}$. With our notations: $c_1[t] = g(g(a, b), d)$, $|c_1[t]| = 5$, $\text{depth}(c_1[t]) = 2$, $\langle t \rangle = ab$, $\text{drop}(c_2[c_1]) = 2$

that is, the distance from the root of the tree to the farthest leaf. We will denote by $\mathfrak{T}^{\leq k} = \{t \in \mathfrak{T}_{\mathcal{F}} \mid \text{depth}(t) \leq k\}$ (resp. $\mathfrak{T}^{k}, \mathfrak{T}^{\geq k}$) the set of trees of depth at most (resp. equal to, at least) k.

Let $\mathcal{F}' = (\Sigma \cup \{*\}, \sharp')$, where $*$ is a symbol of arity 0 *not* in Σ. The set of *context trees* is the set $\mathfrak{C}_{\mathcal{F}} = \{c \in \mathfrak{T}_{\mathcal{F}'} : |c|_* = 1\}$; that is, a context $c \in \mathfrak{C}_{\mathcal{F}}$ is a tree in $\mathfrak{T}_{\mathcal{F}'}$ in which the symbol $*$ occurs exactly in one leaf. Note that given a context $c = g(t_1, \ldots, t_p) \in \mathfrak{C}_{\mathcal{F}}$ with $g \in \mathcal{F}_p$, $t_1, \ldots t_p \in \mathfrak{T}_{\mathcal{F}'}$ the symbol $*$ can only appear in one of the t_i's. The *drop* of a context $c \in \mathfrak{C}_{\mathcal{F}}$ is the distance between the root and the leaf labeled with $*$ in c, which can be defined recursively as $\text{drop}(*) = 0$, $\text{drop}(g(t_1, \ldots, t_i, c, t_{i+1}, \ldots, t_p)) = \text{drop}(c) + 1$ for any $g \in \mathcal{F}_{p+1}$, $t_1, \ldots, t_p \in \mathfrak{T}_{\mathcal{F}}$.

We usually think of the leaf with the symbol $*$ in a context as a placeholder where the root of another tree or another context can be inserted. Accordingly, given $t \in \mathfrak{T}_{\mathcal{F}}$ and $c \in \mathfrak{C}_{\mathcal{F}}$, we define $c[t] \in \mathfrak{T}_{\mathcal{F}}$ as the tree obtained by replacing the occurrence of $*$ in c with t. Similarly, given $c, c' \in \mathfrak{C}_{\mathcal{F}}$ we can obtain a new context tree $c[c']$ by replacing the occurrence of $*$ in c with c' (see Figure 1). We will denote by $\tau \subset t$ the fact that τ is a subtree of t (i.e. that there exists a context c such that $t = c[\tau]$).

Weighted tree automaton. A *weighted tree automaton* (WTA) A over trees on a ranked alphabet $\mathcal{F}$ is a tuple $(\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ where n is the number of states, $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the initial weight vector, $\boldsymbol{\omega}^\sigma \in \mathbb{R}^n$ is the final weight vector associated with the leaf symbol σ for each $\sigma \in \mathcal{F}_0$, and for any symbol g of arity $p \geq 1$, $\boldsymbol{\mathcal{A}}^g \in (\mathbb{R}^n)^{\otimes(p+1)}$ is the transition tensor of order $p+1$ associated with the internal symbol g. A WTA A *computes* a function $f_A : \mathfrak{T}_{\mathcal{F}} \to \mathbb{R}$ assigning to each tree $t \in \mathfrak{T}_{\mathcal{F}}$ the scalar computed as $f_A(t) = \boldsymbol{\alpha}^\top \boldsymbol{\omega}_A(t)$, where $\boldsymbol{\omega}_A(t) \in \mathbb{R}^n$ is obtained recursively as $\boldsymbol{\omega}_A(\sigma) = \boldsymbol{\omega}^\sigma$, and

$$\boldsymbol{\omega}_A(g(t_1, \ldots, t_p)) = \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}_A(t_1), \ldots, \boldsymbol{\omega}_A(t_p)).$$

In many cases we will just write $\boldsymbol{\omega}(t)$ when the automaton A is clear from the context.

An arbitrary function $f : \mathfrak{T}_{\mathcal{F}} \to \mathbb{R}$ is called *recognizable* (or rational) if there exists a WTA A such that $f = f_A$. The number of states of the smallest such WTA is the *rank* of f — we shall set $\text{rank}(f) = \infty$ if f is not recognizable. A WTA A with n states such that $f_A = f$ and $n = \text{rank}(f)$ is called *minimal*.

To conclude this section, we briefly discuss how the definition of WTA in terms of tensor algebra relates to equivalent definitions of WTA. First, in the seminal paper of Berstel and Reutenauer [19], weighted tree automata are defined in terms of multilinear maps acting on an $\mathbb{F}$-vector space for some (commutative) field $\mathbb{F}$ (making their definition basis independent). We focus here on the specific case of $\mathbb{F} = \mathbb{R}$ and (for reasons that will appear clearly in the sequel) we choose a strictly equivalent basis dependent definition (i.e. each multilinear map is expressed in some fixed basis)

The formalism we use can also be directly related to the classical definition of WTA in terms of run semantics (see e.g. [34, Section 3.2]). Formally, a WTA over a ranked alphabet $\mathcal{F}$ can be defined as a tuple $\mathcal{A} = \langle Q, \mathbb{S}, \delta, \lambda \rangle$ where Q is a finite set of states, $\mathbb{S}$ is a semi-ring, δ is a transition map, mapping transitions $(q_1 \ldots q_p, g, q)$ into $\mathbb{S}$ where $q_1, \ldots, q_p, q \in Q$ and $g \in \mathcal{F}_p$, and $\lambda \in \mathbb{S}^Q$ maps each state to its initial weight. Informally, a run of $\mathcal{A}$ on a tree $t \in \mathfrak{T}_{\mathcal{F}}$ is a labeling of the nodes of t with states in Q built using transitions in δ. The weight of a run is the product of the weights of all these transitions along with the initial weight in λ corresponding to the state labeling the root of t (using product in $\mathbb{S}$). The weight of the tree, i.e. the value computed by $\mathcal{A}$ on t, is then given by the sum of the weights of all possible runs of $\mathcal{A}$ on t (using addition in $\mathbb{S}$). To see that this definition is equivalent to ours, one can identify the number of states n with $|Q|$, $\boldsymbol{\alpha}$ with λ, and the non-zero entries of each transition tensor $\boldsymbol{\mathcal{A}}^g$ and leaf vectors $\boldsymbol{\omega}^\sigma$ for $g \in \mathcal{F}_{\geq 1}, \sigma \in \mathcal{F}_0$ with the corresponding weights given by the transition map δ.

Lastly, WTA are closely related to weighted and probabilistic context free grammars. Indeed, it is well known that the set of derivation trees of a context-free grammar forms a regular tree language, that is a language that can be recognized by a (unweighted) tree automaton. Weighted and probabilistic context-free grammars (WCFG and PCFG) are quantitative extensions of context-free grammars [22][48][26] and

$[p]$	set of integers from 1 to p
$\lvert S\rvert,\ \lvert a\rvert$	cardinal of set S, absolute value of a
$a, \mathbf{v}, \mathbf{A}, \boldsymbol{\mathcal{T}}$	scalar, vector, matrix and tensor
$\mathbf{I}, \mathbf{M}^{-1}$	identity matrix, inverse of a matrix
$\mathrm{vec}(\mathbf{M})$	vectorization of a matrix
$\mathbf{T}_{(1)}, (\boldsymbol{\mathcal{A}}+\boldsymbol{\mathcal{B}})_{(1)}$	matricization of a tensor
$\boldsymbol{\mathcal{T}}(\mathbf{A},\mathbf{B},\mathbf{C}),\ \boldsymbol{\mathcal{T}}(\mathbf{x},\mathbf{y},\mathbf{z})$	3rd order tensor multiplied by matrices/vectors:
$\mathbf{u}\otimes\mathbf{v}, \mathbf{A}\otimes\mathbf{B}$	Kronecker product (for vectors or matrices)
$\mathbf{J}_{F,\mathbf{v}}$	Jacobian of F at $\mathbf{v}$
$\mathfrak{s}_1, \mathfrak{s}_2, \ldots$	singular values of the Hankel matrix
$\mathcal{F} = (\Sigma, \sharp)$	ranked alphabet
$\mathcal{F}_p,\ \mathcal{F}_{\geq p}$	symbols of arity p (resp. greater than p)
$\mathfrak{T}_{\mathcal{F}},\ \mathfrak{C}_{\mathcal{F}}$ (or simply $\mathfrak{T},\ \mathfrak{C}$)	set of trees and contexts on $\mathcal{F}$
$*$	placeholder symbol in contexts (also the empty context)
$\lvert t\rvert,\ \mathrm{depth}(t),\ \mathrm{drop}(c)$	size and depth of a tree and drop of a context
$\mathfrak{T}^{\leq k}, \mathfrak{T}^{k}, \mathfrak{T}^{\geq k}$	set of trees of depth at most / equal to / at least k
$\mathfrak{C}^{\leq k}, \mathfrak{C}^{k}, \mathfrak{C}^{\geq k}$	set of contexts of drop at most / equal to / at least k
$(\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{T}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$	Weighted Tree Automaton with n states
$\boldsymbol{\omega}_A : \mathfrak{T} \to \mathbb{R}^n$	tree mapping induced by a WTA A with n states
$\boldsymbol{\alpha}_A : \mathfrak{C} \to \mathbb{R}^n,\ \boldsymbol{\Xi}_A : \mathfrak{C} \to \mathbb{R}^{n\times n}$	context mappings induced by a WTA A with n state
$\mathbf{G}_{\mathfrak{C}}, \mathbf{G}_{\mathfrak{T}}$	Gramian matrices of a WTA
$\mathfrak{z} = \mathfrak{z}_A = \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}_A(t)$	generalized partition function of a WTA

Table 1: Summary of notations

the connection between WCFG/PCFG and WTA is of a similar nature: any weighted tree language induced by a WCFG on derivation trees can be computed by a WTA. A proof of this result can be found in [28, Corollary 8.7] and the interested reader can refer to e.g. [33] for more details. In [46], we leveraged this equivalence between PCFG and WTA to use our approximate minimization method to reduce the size of a WTA obtained from a PCFG learned on a real world natural language corpus.

3 Singular Value Tree Automaton

This section develops the fundamentals of *singular value tree automata* (SVTA), which provide a novel canonical form for WTA inspired by spectral theory of linear operators. Following the development for the string case, we first prove the existence of the canonical form by exploiting the duality between minimal WTA and rank factorizations of the corresponding Hankel matrix. Next we show how the computation of the SVTA form of a given WTA reduces to the computation of a pair of finite Gramian matrices arising from rank factorizations of Hankel matrices. The computation of these Gramians in the tree case turns out to be significantly more involved than in the string case, and is deferred to Section 5.

3.1 Rank Factorizations of Hankel Matrices

We start by recalling a crucial observation about WTAs: there exist more than one WTA computing the same function — in fact, there exist infinitely many. An important construction along these lines is the *conjugate* of a WTA A with n states by an invertible matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$. If $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$, its conjugate by $\mathbf{Q}$ is

$$A^{\mathbf{Q}} = (\mathbb{R}^n, \mathbf{Q}^\top\boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g(\mathbf{Q}^{-\top}, \mathbf{Q}, \ldots, \mathbf{Q})\}_{g\in\mathcal{F}_{\geq 1}}, \{\mathbf{Q}^{-1}\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0}) \tag{1}$$

where $\mathbf{Q}^{-\top} = (\mathbf{Q}^\top)^{-1}$ denotes the inverse of the transpose. To prove that $f_A = f_{A^{\mathbf{Q}}}$ one applies an induction argument on $\mathrm{depth}(t)$ to show that $\boldsymbol{\omega}_{A^{\mathbf{Q}}}(t) = \mathbf{Q}^{-1}\boldsymbol{\omega}_A(t)$ for every $t \in \mathfrak{T}_{\mathcal{F}}$. The claim is obvious

for trees of zero depth $\sigma \in \Sigma$, and for $t = g(t_1, \ldots, t_p)$ we have

$$\begin{aligned}\boldsymbol{\omega}_{A^{\mathbf{Q}}}(g(t_1,\ldots,t_k)) &= (\boldsymbol{\mathcal{A}}^g(\mathbf{Q}^{-\top},\mathbf{Q},\ldots,\mathbf{Q}))(\mathbf{I},\boldsymbol{\omega}_{A^{\mathbf{Q}}}(t_1),\ldots,\boldsymbol{\omega}_{A^{\mathbf{Q}}}(t_p)) \\ &= (\boldsymbol{\mathcal{A}}^g(\mathbf{Q}^{-\top},\mathbf{Q},\ldots,\mathbf{Q}))(\mathbf{I},\mathbf{Q}^{-1}\boldsymbol{\omega}_A(t_1),\ldots,\mathbf{Q}^{-1}\boldsymbol{\omega}_A(t_p)) \\ &= \boldsymbol{\mathcal{A}}^g(\mathbf{Q}^{-\top},\boldsymbol{\omega}_A(t_1),\ldots,\boldsymbol{\omega}_A(t_p)) \\ &= \mathbf{Q}^{-1}\boldsymbol{\mathcal{A}}^g(\mathbf{I},\boldsymbol{\omega}_A(t_1),\ldots,\boldsymbol{\omega}_A(t_p)) = \mathbf{Q}^{-1}\boldsymbol{\omega}_A(t)\ ,\end{aligned}$$

where we just used some simple rules of tensor algebra.

Given any $f : \mathfrak{T} \to \mathbb{R}$ we define its *Hankel matrix* as the bi-infinite matrix $\mathbf{H}_f \in \mathbb{R}^{\mathfrak{C}\times\mathfrak{T}}$ with rows indexed by contexts, columns indexed by trees, and whose entries are given by $\mathbf{H}_f(c,t) = f(c[t])$. Note that given a tree $t' \in \mathfrak{T}$ there are exactly $|t'|$ different ways of splitting $t' = c[t]$ with $c \in \mathfrak{C}$ and $t \in \mathfrak{T}$. This implies that $\mathbf{H}_f$ is a highly redundant representation for f, and it turns out that this redundancy is the key to proving the following fundamental result about recognizable tree functions.

Theorem 1 ([21])**.** *For any $f : \mathfrak{T}_{\mathcal{F}} \to \mathbb{R}$ we have* $\mathrm{rank}(f) = \mathrm{rank}(\mathbf{H}_f)$.

The theorem above implies that the rank of $\mathbf{H}_f$ is finite if and only if f is recognizable. When the rank of $\mathbf{H}_f$ is indeed finite — say $\mathrm{rank}(\mathbf{H}_f) = n$ — one can find two rank n matrices $\mathbf{P} \in \mathbb{R}^{\mathfrak{C}\times n}$, $\mathbf{S} \in \mathbb{R}^{n\times\mathfrak{T}}$ such that $\mathbf{H}_f = \mathbf{PS}$. In this case we say that $\mathbf{P}$ and $\mathbf{S}$ give a *rank factorization* of $\mathbf{H}_f$. We shall now refine Theorem 1 by showing that when f is recognizable, the set of all possible rank factorizations of $\mathbf{H}_f$ is in direct correspondence with the set of minimal WTA computing f.

The first step is to show that any minimal WTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ computing f induces a rank factorization $\mathbf{H}_f = \mathbf{P}_A\mathbf{S}_A$. We build $\mathbf{S}_A \in \mathbb{R}^{n\times\mathfrak{T}}$ by setting the column corresponding to a tree t to $\mathbf{S}_A(:,t) = \boldsymbol{\omega}_A(t)$. In order to define $\mathbf{P}_A$ we need to introduce a new mapping $\boldsymbol{\Xi}_A : \mathfrak{C} \to \mathbb{R}^{n\times n}$ assigning a matrix to every context as follows: $\boldsymbol{\Xi}_A(*) = \mathbf{I}$ and for any context $c = g(t_1,\ldots,t_{i-1},c',t_{i+1},\ldots,t_p)$ where $p \geq 1$, $g \in \mathcal{F}_p$, $t_j \in \mathfrak{T}$ for $j \neq i$ and $c' \in \mathfrak{C}$

$$\boldsymbol{\Xi}_A(c) = \boldsymbol{\mathcal{A}}^g(\mathbf{I},\boldsymbol{\omega}_A(t_1),\ldots,\boldsymbol{\omega}_A(t_{i-1}),\boldsymbol{\Xi}_A(c'),\boldsymbol{\omega}_A(t_{i+1}),\ldots,\boldsymbol{\omega}_A(t_p)). \tag{2}$$

If we now define $\boldsymbol{\alpha}_A : \mathfrak{C} \to \mathbb{R}^n$ as $\boldsymbol{\alpha}_A(c)^\top = \boldsymbol{\alpha}^\top\boldsymbol{\Xi}_A(c)$, we can set the row of $\mathbf{P}_A$ corresponding to c to be $\mathbf{P}_A(c,:) = \boldsymbol{\alpha}_A(c)^\top$. With these definitions one can easily show by induction on $\mathrm{drop}(c)$ that $\boldsymbol{\Xi}_A(c)\boldsymbol{\omega}_A(t) = \boldsymbol{\omega}_A(c[t])$ for any $c \in \mathfrak{C}$ and $t \in \mathfrak{T}$. Then it is immediate to check that $\mathbf{H}_f = \mathbf{P}_A\mathbf{S}_A$:

$$\begin{aligned}\sum_{i=1}^{n}\mathbf{P}_A(c,i)\mathbf{S}_A(i,t) &= \boldsymbol{\alpha}_A(c)^\top\boldsymbol{\omega}_A(t) = \boldsymbol{\alpha}^\top\boldsymbol{\Xi}_A(c)\boldsymbol{\omega}_A(t) \\ &= \boldsymbol{\alpha}^\top\boldsymbol{\omega}_A(c[t]) = f_A(c[t]) = \mathbf{H}_f(c,t)\ .\end{aligned} \tag{3}$$

As before, we will sometimes just write $\boldsymbol{\Xi}(c)$ and $\boldsymbol{\alpha}(c)$ when A is clear from the context. We can now state the main result of this section, which generalizes similar results in [10, 15] for weighted automata on strings.

Theorem 2. *Let $f : \mathfrak{T} \to \mathbb{R}$ be recognizable. If $\mathbf{H}_f = \mathbf{PS}$ is a rank factorization, then there exists a minimal WTA A computing f such that $\mathbf{P}_A = \mathbf{P}$ and $\mathbf{S}_A = \mathbf{S}$.*

Proof. Let $n = \mathrm{rank}(f)$. Let $B = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{B}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ be an arbitrary minimal WTA computing f. Suppose B induces the rank factorization $\mathbf{H}_f = \mathbf{P}'\mathbf{S}'$. Since the columns of both $\mathbf{P}$ and $\mathbf{P}'$ are basis for the column-span of $\mathbf{H}_f$, there exists a change of basis $\mathbf{Q} \in \mathbb{R}^{n\times n}$ between $\mathbf{P}$ and $\mathbf{P}'$. That is, $\mathbf{Q}$ is an invertible matrix such that $\mathbf{P}'\mathbf{Q} = \mathbf{P}$. Furthermore, since $\mathbf{P}'\mathbf{S}' = \mathbf{H}_f = \mathbf{PS} = \mathbf{P}'\mathbf{QS}$ and $\mathbf{P}'$ has full column rank, we must have $\mathbf{S}' = \mathbf{QS}$, or equivalently, $\mathbf{Q}^{-1}\mathbf{S}' = \mathbf{S}$. Thus, we let $A = B^{\mathbf{Q}}$, which immediately satisfies $f_A = f_B = f$. It remains to show that A induces the rank factorization $\mathbf{H}_f = \mathbf{PS}$. Note that when proving the equivalence $f_A = f_B$ we already showed $\boldsymbol{\omega}_A(t) = \mathbf{Q}^{-1}\boldsymbol{\omega}_B(t)$, thus $\mathbf{S}_A = \mathbf{Q}^{-1}\mathbf{S}' = \mathbf{S}$. To show $\mathbf{P}_A = \mathbf{P}'\mathbf{Q}$ we need to show that for any $c \in \mathfrak{C}$ we have $\boldsymbol{\alpha}_A(c)^\top = \boldsymbol{\alpha}_B(c)^\top\mathbf{Q}$. This will immediately follow if we show that $\boldsymbol{\Xi}_A(c) = \mathbf{Q}^{-1}\boldsymbol{\Xi}_B(c)\mathbf{Q}$. If we proceed by induction on $\mathrm{drop}(c)$, we see that the case $c = *$ is immediate. For $c = g(c',t_1,\ldots,t_p)$ where $c' \in \mathfrak{C}$, $p \geq 0$, $g \in \mathcal{F}_{p+1}$ and $t_1,\ldots,t_p \in \mathfrak{T}$, we get

$$\begin{aligned}\boldsymbol{\Xi}_A(g(c',t_1,\ldots,t_p)) &= \boldsymbol{\mathcal{A}}^g(\mathbf{I},\boldsymbol{\Xi}_A(c'),\boldsymbol{\omega}_A(t_1),\ldots,\boldsymbol{\omega}_A(t_p)) \\ &= (\boldsymbol{\mathcal{B}}^g(\mathbf{Q}^{-\top},\mathbf{Q},\ldots,\mathbf{Q}))(\mathbf{I},\mathbf{Q}^{-1}\boldsymbol{\Xi}_B(c')\mathbf{Q},\mathbf{Q}^{-1}\boldsymbol{\omega}_B(t_1),\ldots,\mathbf{Q}^{-1}\boldsymbol{\omega}_B(t_p)) \\ &= \boldsymbol{\mathcal{B}}^g(\mathbf{Q}^{-\top},\boldsymbol{\Xi}_B(c')\mathbf{Q},\boldsymbol{\omega}_B(t_1),\ldots,\boldsymbol{\omega}_B(t_p)) \\ &= \mathbf{Q}^{-1}\boldsymbol{\mathcal{B}}^g(\mathbf{I},\boldsymbol{\Xi}_B(c'),\boldsymbol{\omega}_B(t_1),\ldots,\boldsymbol{\omega}_B(t_p))\mathbf{Q} = \mathbf{Q}^{-1}\boldsymbol{\Xi}_B(c)\mathbf{Q}\ .\end{aligned}$$

Applying the same argument mutatis mutandis for contexts of the form $c = g(t_1, \ldots, t_{l-1}, c', t_l, \ldots, t_p)$ completes the proof. □

3.2 SVTA: Definition and Existence

Suppose $f : \mathfrak{T} \to \mathbb{R}$ is a rank n recognizable function whose Hankel matrix admits a reduced singular value decomposition $\mathbf{H}_f = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. Then we have that $\mathbf{P} = \mathbf{U}\mathbf{D}^{1/2}$ and $\mathbf{S} = \mathbf{D}^{1/2}\mathbf{V}^\top$ is a rank decomposition for $\mathbf{H}_f$, and by Theorem 2 there exists some minimal WTA A with $f_A = f$, $\mathbf{P}_A = \mathbf{U}\mathbf{D}^{1/2}$ and $\mathbf{S}_A = \mathbf{D}^{1/2}\mathbf{V}^\top$. We call such A a *singular value tree automaton* (SVTA) for f. However, these are not defined for every recognizable function f, because the fact that the columns of $\mathbf{U}$ and $\mathbf{V}$ must be unitary vectors (i.e. $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$) imposes some restrictions on which infinite Hankel matrices $\mathbf{H}_f$ admit an SVD — this phenomenon is related to the distinction between compact and non-compact operators in functional analysis. Our next theorem gives a sufficient condition for the existence of an SVD of $\mathbf{H}_f$.

We say that a function $f : \mathfrak{T}_\mathcal{F} \to \mathbb{R}$ is *strongly convergent* if the series $\sum_{t \in \mathfrak{T}_\mathcal{F}} |t||f(t)|$ converges. To see the intuitive meaning of this condition, assume that f is a probability distribution over trees in $\mathfrak{T}$. In this case, strong convergence is equivalent to saying that the expected size of trees generated from the distribution f is finite. It turns out that strong convergence of f is a sufficient condition to guarantee the existence of an SVD for $\mathbf{H}_f$.

Theorem 3. *If $f : \mathfrak{T}_\Sigma \to \mathbb{R}$ is recognizable and strongly convergent, then $\mathbf{H}_f$ admits a singular value decomposition.*

Proof. The result will follow if we show that $\mathbf{H}_f$ is the matrix of a compact operator between Hilbert spaces [35, Theorem 4.3.5]. We start by defining the Hilbert spaces of square-summable series indexed by trees and contexts. Given two functions $g, g' : \mathfrak{T}_\Sigma \to \mathbb{R}$ we define their inner product to be $\langle g, g' \rangle_\mathfrak{T} = \sum_{t \in \mathfrak{T}_\Sigma} g(t)g'(t)$ (whenever the sum converges). Let $\|g\|_\mathfrak{T} = \sqrt{\langle g, g \rangle_\mathfrak{T}}$ be the induced norm. We denote by $\ell^2_\mathfrak{T}$ be the real vector space of functions $\{g : \mathfrak{T} \to \mathbb{R} \mid \|g\|_\mathfrak{T} < \infty\}$, which is a separable Hilbert space because the set $\mathfrak{T}$ is countable. Similarly, given functions $g, g' : \mathfrak{C}_\Sigma \to \mathbb{R}$ we define an inner product $\langle g, g' \rangle_\mathfrak{C} = \sum_{c \in \mathfrak{C}_\Sigma} g(c)g'(c)$, a norm $\|g\|_\mathfrak{C} = \sqrt{\langle g, g \rangle_\mathfrak{C}}$, and a separable Hilbert space $\ell^2_\mathfrak{C} = \{g : \mathfrak{C} \to \mathbb{R} \mid \|g\|_\mathfrak{C} < \infty\}$. With this notation it is possible to see that $\mathbf{H}_f$ is the matrix under the standard basis on $\ell^2_\mathfrak{T}$ and $\ell^2_\mathfrak{C}$ of the operator $H_f : \ell^2_\mathfrak{T} \to \ell^2_\mathfrak{C}$ given by $(H_f g)(c) = \sum_{t \in \mathfrak{T}_\Sigma} f(c[t])g(t)$. Since f is recognizable, $\mathbf{H}_f$ is a finite-rank matrix and therefore H_f is a finite-rank operator. Thus, to show the compactness of H_f it only remains to show that H_f is bounded.

Given $f \in \ell^2_\mathfrak{T}$ and $c \in \mathfrak{C}_\Sigma$ we define a new function $f_c \in \ell^2_\mathfrak{T}$ given by $f_c(t) = f(c[t])$ for $t \in \mathfrak{T}_\Sigma$. Now let $g \in \ell^2_\mathfrak{T}$ with $\|g\|_\mathfrak{T} = 1$ and recall that H_f is bounded if $\|H_f g\|_\mathfrak{C} < \infty$ for every $g \in \ell^2_\mathfrak{T}$ with $\|g\|_\mathfrak{T} = 1$. To show that H_f is bounded observe that

$$
\begin{aligned}
\|H_f g\|^2_\mathfrak{C} &= \sum_{c \in \mathfrak{C}_\Sigma} (H_f g)(c)^2 = \sum_{c \in \mathfrak{C}_\Sigma} \left(\sum_{t \in \mathfrak{T}_\Sigma} f(c[t])g(t) \right)^2 \\
&= \sum_{c \in \mathfrak{C}_\Sigma} \langle f_c, g \rangle^2_\mathfrak{T} \leq \|g\|^2_\mathfrak{T} \sum_{c \in \mathfrak{C}_\Sigma} \|f_c\|^2_\mathfrak{T} \\
&= \sum_{c \in \mathfrak{C}_\Sigma} \sum_{t \in \mathfrak{T}_\Sigma} f_c(t)^2 = \sum_{c \in \mathfrak{C}_\Sigma} \sum_{t \in \mathfrak{T}_\Sigma} f(c[t])^2 \\
&= \sum_{t \in \mathfrak{T}_\Sigma} |t| f(t)^2 \leq \sup_{t \in \mathfrak{T}_\Sigma} |f(t)| \cdot \sum_{t \in \mathfrak{T}_\Sigma} |t||f(t)| \\
&< \infty \ ,
\end{aligned}
$$

where we used the Cauchy–Schwarz inequality, and the fact that $\sup_{t \in \mathfrak{T}_\Sigma} |f(t)|$ is bounded when f is strongly convergent. □

Together, Theorems 2 and 3 imply that every recognizable strongly convergent $f : \mathfrak{T} \to \mathbb{R}$ can be represented by an SVTA A. Since the singular value decomposition admits strong uniqueness properties (eg. if all the singular values are distinct, the decomposition is unique up to sign changes in corresponding pairs of singular vectors), the SVTA representation inherits these same uniqueness properties, and thus allows us to consider it as a *canonical representation*. Next we address the question of computing this canonical representation starting from an arbitrary WTA.

3.3 Gramian Matrices of WTA

The definition of SVTA suggests that computing the canonical form is equivalent to computing the SVD of the infinite Hankel matrix $\mathbf{H}_f$. From a purely algebraic perspective, the proof of Theorem 2 shows that if we know both the rank factorization induced by the SVD of $\mathbf{H}_f$ and the rank factorization induced by an arbitrary minimal WTA, then computing the SVTA amounts to finding the corresponding change of basis. However, this approach works with infinite matrices and it is not immediate how to convert it into an effective algorithm. In the string case, [15, 16] reduces this computation to a problem about finite matrices by using appropriately defined Gramian matrices. The same strategy also works in the tree case, with some caveats that will be discussed at the end of this section. In the next section, we will show how to compute an SVTA canonical form when given access to the Gramian matrices of an arbitrary minimal WTA.

Suppose A is a minimal WTA. The Gramian matrices of A, defined in terms of the rank factorization $\mathbf{H}_f = \mathbf{P}_A\mathbf{S}_A$ induced by A, are given by $\mathbf{G}_{\mathfrak{C}} = \mathbf{P}_A^\top\mathbf{P}_A$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{S}_A\mathbf{S}_A^\top$. Observe that if A has n states, then both these Gramians are $n \times n$ matrices. Unfolding these definitions we observe that the Gramian matrices satisfy the following:

$$\mathbf{G}_{\mathfrak{C}} = \mathbf{P}_A^\top\mathbf{P}_A = \sum_{c\in\mathfrak{C}} \boldsymbol{\alpha}_A(c)\boldsymbol{\alpha}_A(c)^\top \quad \text{and} \quad \mathbf{G}_{\mathfrak{T}} = \mathbf{S}_A\mathbf{S}_A^\top = \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}_A(t)\boldsymbol{\omega}_A(t)^\top \ .$$

An equivalent characterization of these infinite series in terms of fixed point equations can be obtained as follows. Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ be a strongly convergent WTA of dimension n computing a function f.

Theorem 4. *The Gramian matrices* $\mathbf{G}_{\mathfrak{T}} = \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^\top$ *and* $\mathbf{G}_{\mathfrak{C}} = \sum_{c\in\mathfrak{C}} \boldsymbol{\alpha}(c)\boldsymbol{\alpha}(c)^\top$ *satisfy the following fixed point equations:*

$$\mathbf{G}_{\mathfrak{T}} = \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}^\sigma\boldsymbol{\omega}^{\sigma\top} + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \mathbf{A}^g_{(1)}(\underbrace{\mathbf{G}_{\mathfrak{T}}\otimes\cdots\otimes\mathbf{G}_{\mathfrak{T}}}_{p \text{ times}})\mathbf{A}^{g\top}_{(1)}, \tag{4}$$

$$\mathbf{G}_{\mathfrak{C}} = \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{i=1}^{p} \mathbf{A}^g_{(i+1)}(\mathbf{G}_{\mathfrak{C}}\otimes\underbrace{\mathbf{G}_{\mathfrak{T}}\otimes\cdots\otimes\mathbf{G}_{\mathfrak{T}}}_{p-1 \text{ times}})\mathbf{A}^{g\top}_{(i+1)}. \tag{5}$$

Proof. Using the fact that any tree t of depth greater than 1 can be written as $g(t_1,\ldots,t_p)$ for some $p \geq 1$, $g \in \mathcal{F}_p$ and $t_1,\ldots,t_p \in \mathfrak{T}$ we have

$$\begin{aligned}
\mathbf{G}_{\mathfrak{T}} &= \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^\top = \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \sum_{t\in\mathfrak{T}\,:\,\mathrm{depth}(t)\geq 1} \boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^\top \\
&= \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{t_1,\ldots,t_p\in\mathfrak{T}} \boldsymbol{\omega}(g(t_1,\ldots,t_p))\boldsymbol{\omega}(g(t_1,\ldots,t_p))^\top \\
&= \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{t_1,\ldots,t_p\in\mathfrak{T}} \boldsymbol{\mathcal{A}}^g(\mathbf{I},\boldsymbol{\omega}(t_1),\ldots,\boldsymbol{\omega}(t_p))\boldsymbol{\mathcal{A}}^g(\mathbf{I},\boldsymbol{\omega}(t_1),\ldots,\boldsymbol{\omega}(t_p))^\top \\
&= \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \mathbf{A}^g_{(1)} \sum_{t_1,\ldots,t_p\in\mathfrak{T}} (\boldsymbol{\omega}(t_1)\otimes\cdots\otimes\boldsymbol{\omega}(t_p))(\boldsymbol{\omega}(t_1)\otimes\cdots\otimes\boldsymbol{\omega}(t_p))^\top \mathbf{A}^{g\top}_{(1)} \\
&= \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma\boldsymbol{\omega}_\sigma^\top + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \mathbf{A}^g_{(1)}(\underbrace{\mathbf{G}_{\mathfrak{T}}\otimes\cdots\otimes\mathbf{G}_{\mathfrak{T}}}_{p \text{ times}})\mathbf{A}^{g\top}_{(1)}.
\end{aligned}$$

To derive a fixed point equation for the Gramian matrices for contexts we use the fact that any context $c \in \mathfrak{C}$ of drop greater than 1 can be written as $c'[g(t_1,\ldots,t_{i-1},*,t_i,\ldots,t_{p-1})]$ for some $c' \in \mathfrak{C}$,

$p \geq 1$, $g \in \mathcal{F}_p$ and $t_1, \ldots, t_{p-1} \in \mathfrak{T}$. Using the notation $\mathbf{v}^{\circ 2} = \mathbf{v}\mathbf{v}^\top$ for any vector $\mathbf{v}$ we have

$$\begin{aligned}
\mathbf{G}_{\mathfrak{C}} &= \sum_{c \in \mathfrak{C}} \boldsymbol{\alpha}(c)\boldsymbol{\alpha}(c)^\top = \boldsymbol{\alpha}(*)\boldsymbol{\alpha}(*)^\top + \sum_{c \in \mathfrak{C} \,:\, \mathrm{drop}(c) \geq 1} \boldsymbol{\alpha}(c)\boldsymbol{\alpha}(c)^\top \\
&= \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \sum_{\substack{p \geq 1, \; g \in \mathcal{F}_p, \; c \in \mathfrak{C}, \\ t_1, \ldots, t_{p-1} \in \mathfrak{T}}} \sum_{i=1}^{p} \boldsymbol{\alpha}(c[g(t_1, \ldots, t_{i-1}, *, t_i, \ldots, t_{p-1})])^{\circ 2} \\
&= \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \sum_{\substack{p \geq 1, \; g \in \mathcal{F}_p, \; c \in \mathfrak{C}, \\ t_1, \ldots, t_{p-1} \in \mathfrak{T}}} \sum_{i=1}^{p} \left(\boldsymbol{\mathcal{A}}^g(\boldsymbol{\alpha}(c), \boldsymbol{\omega}(t_1), \ldots, \boldsymbol{\omega}(t_{i-1}), \mathbf{I}, \boldsymbol{\omega}(t_i), \ldots, \boldsymbol{\omega}(t_{p-1})) \right)^{\circ 2} \\
&= \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \sum_{p \geq 1, \; g \in \mathcal{F}_p} \sum_{i=1}^{p} \sum_{\substack{c \in \mathfrak{C}, \\ t_1, \ldots, t_{p-1} \in \mathfrak{T}}} \left(\mathbf{A}^g_{(i+1)}(\boldsymbol{\alpha}(c) \otimes \boldsymbol{\omega}(t_1) \otimes \cdots \otimes \boldsymbol{\omega}(t_{p-1})) \right)^{\circ 2} \\
&= \boldsymbol{\alpha}\boldsymbol{\alpha}^\top + \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=1}^{p} \mathbf{A}^g_{(i+1)} (\mathbf{G}_{\mathfrak{C}} \otimes \underbrace{\mathbf{G}_{\mathfrak{T}} \otimes \cdots \otimes \mathbf{G}_{\mathfrak{T}}}_{p-1 \text{ times}}) {\mathbf{A}^g_{(i+1)}}^\top. \qquad \square
\end{aligned}$$

Treating $\mathbf{G}_{\mathfrak{T}}$ as an unknown, we see that (4) is a polynomial equation of degree $P = \max\{p : \mathcal{F}_p \neq \emptyset\}$ in the entries of $\mathbf{G}_{\mathfrak{T}}$. The coefficients of these equations depend only on the weights of the WTA A. Equation (5) provides a polynomial relationship between the entries of $\mathbf{G}_{\mathfrak{T}}$ and $\mathbf{G}_{\mathfrak{C}}$. In this case, given $\mathbf{G}_{\mathfrak{T}}$ the resulting equation in the coefficients of $\mathbf{G}_{\mathfrak{C}}$ is linear. This behavior is qualitatively different to the one observed for the corresponding fixed-point equations that characterize the Gramians in the string case (cf. [16, Theorem 20]). In the tree case, the Gramian equations simplify to two independent systems of linear equations when $P = 1$, which corresponds to the case where there is no branching, i.e. the WTA recognize a language over strings. In general, however, the Gramian equations do not admit a closed-form solution in the tree case. How to solve these fixed-point equations using iterative algorithms will be discussed in Section 5.

3.4 Computing the Singular Value WTA

Now we show that if we are given an arbitrary minimal WTA A admitting an SVTA representation, then we can transform A into the corresponding SVTA efficiently[4] provided that we have an oracle for computing the Gramian matrices of A. In other words, given a representation of $\mathbf{H}_f$ as a WTA, we can compute its SVD *without* the need to operate on infinite matrices. The key observation is to reduce the computation of the SVD of $\mathbf{H}_f$ to the computation of spectral properties of the Gramians $\mathbf{G}_{\mathfrak{C}} = \mathbf{P}^\top\mathbf{P}$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{S}\mathbf{S}^\top$ associated with the rank factorization $\mathbf{H}_f = \mathbf{P}\mathbf{S}$ induced by some minimal WTA computing f. This is captured by the following proposition.

Proposition 5. *Let $f : \mathfrak{T} \to \mathbb{R}$ be a recognizable function such that its Hankel matrix $\mathbf{H}_f$ admits an SVD. Let A be a minimal WTA with n states computing f and inducing the rank factorization $\mathbf{H}_f = \mathbf{P}\mathbf{S}$, and let $\mathbf{G}_{\mathfrak{C}} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n \times n}$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{S}\mathbf{S}^\top \in \mathbb{R}^{n \times n}$ be the corresponding Gramian matrices.*

Let $\mathbf{L}_{\mathfrak{C}}, \mathbf{L}_{\mathfrak{T}} \in \mathbb{R}^{n \times n}$ be such that $\mathbf{G}_{\mathfrak{C}} = \mathbf{L}_{\mathfrak{C}}^\top\mathbf{L}_{\mathfrak{C}}$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{L}_{\mathfrak{T}}^\top\mathbf{L}_{\mathfrak{T}}$ and let $\mathbf{L}_{\mathfrak{C}}\mathbf{L}_{\mathfrak{T}}^\top = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be a singular value decomposition.

Then, the conjugate WTA $A^{\mathbf{Q}}$ where $\mathbf{Q} = \mathbf{L}_{\mathfrak{C}}^{-1}\mathbf{U}\mathbf{D}^{1/2}$ is an SVTA computing f.

Proof. First observe that since the Gramian matrices are symmetric positive semi-definite there exist matrices $\mathbf{L}_{\mathfrak{C}}, \mathbf{L}_{\mathfrak{T}} \in \mathbb{R}^{n \times n}$ such that $\mathbf{G}_{\mathfrak{C}} = \mathbf{L}_{\mathfrak{C}}^\top\mathbf{L}_{\mathfrak{C}}$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{L}_{\mathfrak{T}}^\top\mathbf{L}_{\mathfrak{T}}$ (one can use Cholesy factorization or eigendecomposition to obtain such matrices). Moreover, since A is a minimal WTA the Gramian matrices are of full rank and $\mathbf{L}_{\mathfrak{C}}$ and $\mathbf{L}_{\mathfrak{T}}$ are non-singular.

The conjugate WTA $A^{\mathbf{Q}}$ induces the factorization $\mathbf{H}_f = \mathbf{P}'\mathbf{S}'$ with $\mathbf{P}' = \mathbf{P}\mathbf{Q}$ and $\mathbf{S}' = \mathbf{Q}^{-1}\mathbf{S}$. It is easy to check that $\mathbf{Q}^{-1} = \mathbf{D}^{1/2}\mathbf{V}^\top\mathbf{L}_{\mathfrak{T}}^{-\top}$, thus $\mathbf{P}' = \tilde{\mathbf{U}}\mathbf{D}^{1/2}$ and $\mathbf{S}' = \mathbf{D}^{1/2}\tilde{\mathbf{V}}^\top$ with $\tilde{\mathbf{U}} = \mathbf{P}\mathbf{L}_{\mathfrak{C}}^{-1}\mathbf{U}$ and $\tilde{\mathbf{V}}^\top = \mathbf{V}^\top\mathbf{L}_{\mathfrak{T}}^{-\top}\mathbf{S}$. Hence, to show that $A^{\mathbf{Q}}$ is an SVTA it suffices to show that $\mathbf{H}_f = \mathbf{P}'\mathbf{S}' = \tilde{\mathbf{U}}\mathbf{D}\tilde{\mathbf{V}}^\top$ is an SVD which boils down to checking that $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ are column-wise orthogonal matrices. Indeed, we have

$$\tilde{\mathbf{U}}^\top\tilde{\mathbf{U}} = \mathbf{U}^\top\mathbf{L}_{\mathfrak{C}}^{-\top}\mathbf{P}^\top\mathbf{P}\mathbf{L}_{\mathfrak{C}}^{-1}\mathbf{U} = \mathbf{U}^\top\mathbf{L}_{\mathfrak{C}}^{-\top}\mathbf{G}_{\mathfrak{C}}\mathbf{L}_{\mathfrak{C}}^{-1}\mathbf{U} = \mathbf{U}^\top\mathbf{U} = \mathbf{I}$$

[4] If the WTA given to the algorithm is not minimal, a pre-processing step can be used to minimize the input using the algorithm from e.g. [38].

and

$$\tilde{\mathbf{V}}^\top \tilde{\mathbf{V}} = \mathbf{V}^\top \mathbf{L}_{\mathfrak{T}}^{-\top} \mathbf{S}\mathbf{S}^\top \mathbf{L}_{\mathfrak{T}}^{-1} \mathbf{V} = \mathbf{V}^\top \mathbf{L}_{\mathfrak{T}}^{-\top} \mathbf{G}_{\mathfrak{T}} \mathbf{L}_{\mathfrak{T}}^{-1} \mathbf{V} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}$$

where we used the fact that the matrices $\mathbf{U}$ and $\mathbf{V}$ are orthogonal. □

Algorithm 1 summarizes the overall procedure to construct the SVTA corresponding to a minimal WTA computing a strongly convergent function. Note that the algorithm depends on an oracle for computing the Gramian matrices $\mathbf{G}_{\mathfrak{T}}$ and $\mathbf{G}_{\mathfrak{C}}$.

Algorithm 1 `ComputeSVTA`

Input: A strongly convergent minimal WTA A
Output: The corresponding SVTA

1: $\mathbf{G}_{\mathfrak{C}}, \mathbf{G}_{\mathfrak{T}} \leftarrow$ `GramMatrices`(A)
2: Compute $\mathbf{L}_{\mathfrak{C}}, \mathbf{L}_{\mathfrak{T}} \in \mathbb{R}^{n \times n}$ such that $\mathbf{G}_{\mathfrak{C}} = \mathbf{L}_{\mathfrak{C}}^\top \mathbf{L}_{\mathfrak{C}}$ and $\mathbf{G}_{\mathfrak{T}} = \mathbf{L}_{\mathfrak{T}}^\top \mathbf{L}_{\mathfrak{T}}$ (using e.g. Cholesky factorizations or eigendecompositions)
3: Let $\mathbf{L}_{\mathfrak{C}} \mathbf{L}_{\mathfrak{T}}^\top = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be an SVD
4: **return** $A^{\mathbf{Q}}$ where $\mathbf{Q} = \mathbf{L}_{\mathfrak{C}}^{-1} \mathbf{U} \mathbf{D}^{1/2}$

In order to implement an efficient oracle to compute the Gramian matrices of a minimal WTA A we will further reduce the computation to the solution of a more general problem: compute the generalized partition function of a WTA. This problem is tackled in the next section.

4 Computing Generalized Partition Functions of WTA

Recall that any WTA $(\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{T}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ induces a mapping $\boldsymbol{\omega} : \mathfrak{T}_{\mathcal{F}} \to \mathbb{R}^n$. In this section, we tackle the problem of computing the sum $\sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}(t) \in \mathbb{R}^n$ for a given WTA (assuming that this series converges). We will denote the limit of this series by $\boldsymbol{\jmath}_A$, or simply $\boldsymbol{\jmath}$ if the WTA is clear from context, and we will refer to this quantity as the *generalized partition function*[5] of the WTA A. We will show in the next section that computing the Gramian matrices of a WTA can be reduced to the problem of computing a generalized partition function.

While it is not possible to obtain a closed-form expression for the sum $\sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}(t)$ in general, we will show that it can be efficiently approximated to an arbitrary precision using either a fixed-point iterative method or Newton's method (the latter providing a faster convergence rate, potentially at the cost of an increased computational complexity for each iteration).

Suppose $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{T}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ is a minimal WTA computing an absolutely convergent function f. We start by showing that the series $\sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}(t)$ converges.

Proposition 6. *If A is a minimal WTA computing an absolutely convergent function f (i.e. $\sum_t |f(t)| < \infty$), then the series $\sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}_A(t)$ converges and we denote its limit by $\boldsymbol{\jmath}_A$.*

Proof. Let n be the rank of f. Since A is minimal, there exist n contexts $c_1, \ldots, c_n \in \mathfrak{C}$ such that $(\boldsymbol{\Xi}(c_i)^\top \boldsymbol{\alpha})_{i=1,\ldots,n}$ is a basis of $\mathbb{R}^n$. For any $i \in [n]$, we have

$$\sum_t \left| \boldsymbol{\alpha}^\top \boldsymbol{\Xi}(c_i) \boldsymbol{\omega}(t) \right| = \sum_t |f(c_i[t])| \leq \sum_t |f(t)| < \infty \ .$$

Hence for each i the series $\sum_t \boldsymbol{\alpha}^\top \boldsymbol{\Xi}(c_i) \boldsymbol{\omega}(t)$ is absolutely convergent, thus this series is also convergent, which shows the claim of the proposition. □

We now show that $\boldsymbol{\jmath}$ is a fixed point of the polynomial map $F : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$F(\mathbf{v}) = \sum_{\sigma \in \mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \mathbf{v}, \ldots, \mathbf{v}). \tag{6}$$

This is the fundamental observation from which we will derive the two algorithms for computing $\boldsymbol{\jmath}$.

[5] We use the adjective *generalized* to emphasize the facts that (i) $\boldsymbol{\jmath}_A$ is a vector (whereas the term partition function in statistical physics usually refers to a scalar) and (ii) the components of $\boldsymbol{\jmath}_A$ do not have a probabilistic interpretation for arbitrary WTA (whereas it is the case for stochastic WTA and related models such as (latent) probabilistic context-free grammars [30]).

Theorem 7. *For any integer $k \geq 0$, we have $F^{k+1}(\mathbf{0}) = \sum_{t \in \mathfrak{T}^{\leq k}} \boldsymbol{\omega}(t)$ where $\mathfrak{T}^{\leq k} = \{t \in \mathfrak{T}_{\mathcal{F}} \mid \text{depth}(t) \leq k\}$ is the set of trees of depth at most k.*

Consequently, the generalized partition function $\boldsymbol{\mathfrak{z}} = \sum_{t \in \mathfrak{T}} \boldsymbol{\omega}(t) \in \mathbb{R}^n$ is a fixed point of F and $\mathbf{0}$ is in the basin of attraction of $\boldsymbol{\mathfrak{z}}$.

Proof. We proceed by induction on k. We have $F(\mathbf{0}) = \sum_{\sigma \in \mathcal{F}_0} \boldsymbol{\omega}_\sigma = \sum_{t \in \mathfrak{T}:\text{depth}(t) \leq 0} \boldsymbol{\omega}(t)$. Suppose that the claim holds for any integer up to k, we have

$$\begin{aligned}
F^{k+1}(\mathbf{0}) &= F(F^k(\mathbf{0})) \\
&= F\left(\sum_{t \in \mathfrak{T}^{\leq k}} \boldsymbol{\omega}(t)\right) \\
&= \sum_{\sigma \in \mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{t_1, \ldots, t_p \in \mathfrak{T}^{\leq k}} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}(t_1), \ldots, \boldsymbol{\omega}(t_p)) \\
&= \sum_{\sigma \in \mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{t : 1 \leq \text{depth}(t) \leq k+1} \boldsymbol{\omega}(t) \\
&= \sum_{t \in \mathfrak{T}^{\leq k+1}} \boldsymbol{\omega}(t).
\end{aligned}$$

For the second part of the theorem, we have $F^k(\mathbf{0}) = \sum_{t:\text{depth}(t)<k} \boldsymbol{\omega}(t)$ for any integer $k \geq 1$, from which it follows that $\lim_{k \to \infty} F^k(\mathbf{0}) = \sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}(t) = \boldsymbol{\mathfrak{z}}$, showing both claims by continuity of F. □

Fixed point iteration We are now ready to derive the two algorithms to approximate $\boldsymbol{\mathfrak{z}}$. We start with the straightforward fixed point iteration method:

$$\mathbf{f}_0 = F(\mathbf{0}) = \sum_{\sigma \in \mathcal{F}_0} \boldsymbol{\omega}_\sigma, \quad \mathbf{f}_{k+1} = F(\mathbf{f}_k) \quad \text{for} \quad k \geq 0. \tag{7}$$

From a classical result on fixed point theory, this iteration will converge linearly to $\boldsymbol{\mathfrak{z}}$ if the spectral radius of the Jacobian of F at $\boldsymbol{\mathfrak{z}}$ is less than 1:

Theorem 8. *Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ be a WTA and suppose that the series $\sum_{t \in \mathfrak{T}_{\mathcal{F}}} \boldsymbol{\omega}_A(t)$ converges (to the generalized partition function $\boldsymbol{\mathfrak{z}}$). Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be the polynomial map defined in Eq. (6).*

Then, the iteration defined in Eq. (7) is such that $\|\mathbf{f}_k - \boldsymbol{\mathfrak{z}}\|_2 \leq \mathcal{O}\left(\rho^k\right)$ for all $k \geq 1$, where ρ is the spectral radius of the Jacobian of F at $\boldsymbol{\mathfrak{z}}$. In particular, if $\rho < 1$ the sequence $(\mathbf{f}_k)_k$ converges linearly to $\boldsymbol{\mathfrak{z}}$.

Proof. The convergence of the fixed point iteration follows directly from Theorem 7 and the convergence rate follows from classical results from numerical analysis (see e.g. [42, Theorem 8.1.7]). □

We will see in the next section that the assumption on the spectral radius of the Jacobian in the previous theorem is always satisfied when computing the Gramian matrices of a *strongly convergent* WTA by estimating a generalized partition function. We now give an expression of the Jacobian of the mapping F at any point $\mathbf{v}$.

Proposition 9. *The Jacobian of F defined in Eq. (6) at a point $\mathbf{v} \in \mathbb{R}^n$, denoted by $\mathbf{J}_{F,\mathbf{v}} \in \mathbb{R}^{n \times n}$ (or simply $\mathbf{J}_{\mathbf{v}}$ if F is clear from context), is given by*

$$\mathbf{J}_{F,\mathbf{v}} = \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=0}^{p-1} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \underbrace{\mathbf{v}, \ldots, \mathbf{v}}_{i \text{ times}}, \mathbf{I}, \underbrace{\mathbf{v}, \ldots, \mathbf{v}}_{p-1-i \text{ times}}). \tag{8}$$

Proof. By expanding $F(\mathbf{v}+\mathbf{h})$ and isolating the terms that are linear in $\mathbf{h}$ we get

$$\begin{aligned} F(\mathbf{v}+\mathbf{h}) &= \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \mathbf{v}+\mathbf{h}, \ldots, \mathbf{v}+\mathbf{h}) \\ &= \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{i=0}^{p-1} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \underbrace{\mathbf{v},\ldots,\mathbf{v}}_{i \text{ times}}, \mathbf{h}, \underbrace{\mathbf{v},\ldots,\mathbf{v}}_{p-1-i \text{ times}}) + \varepsilon(\mathbf{h}) \\ &= \left(\sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{i=0}^{p-1} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \underbrace{\mathbf{v},\ldots,\mathbf{v}}_{i \text{ times}}, \mathbf{I}, \underbrace{\mathbf{v},\ldots,\mathbf{v}}_{p-1-i \text{ times}})\right)\mathbf{h} + \varepsilon(\mathbf{h}) \end{aligned}$$

where $\varepsilon(\mathbf{h})$ gather the terms that are at least quadratic in the components of $\mathbf{h}$. □

Newton's method In order to obtain a better convergence rate, we can use Newton's method to find the vanishing points of the map $G : \mathbf{v} \mapsto F(\mathbf{v}) - \mathbf{v}$. Starting with the initial guess $F(\mathbf{0}) = \sum_{\sigma\in\mathcal{F}_0}\boldsymbol{\omega}_\sigma$, the Newton iteration is defined by

$$\mathbf{n}_0 = \sum_{\sigma\in\mathcal{F}_0}\boldsymbol{\omega}_\sigma, \qquad \mathbf{n}_{k+1} = \mathbf{n}_k - (\mathbf{J}_{\mathbf{n}_k,G})^{-1}G(\mathbf{n}_k). \tag{9}$$

One can easily check that this can be rewritten as

$$\mathbf{n}_{k+1} = \mathbf{n}_k + (\mathbf{I} - \mathbf{J}_{\mathbf{n}_k,F})^{-1}(F(\mathbf{n}_k) - \mathbf{n}_k). \tag{10}$$

The convergence of this procedure is established in the following theorem. The proof of the theorem will occupy the remainder of this section.

Theorem 10. *Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ be a WTA and suppose that the series $\boldsymbol{\mathfrak{z}} = \sum_{t\in\mathfrak{T}_\mathcal{F}}\boldsymbol{\omega}_A(t)$ converges. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be the polynomial map defined in Eq. (6).*

Then, under the assumptions that

1. *the Jacobian $\mathbf{J}_{\boldsymbol{\mathfrak{z}},F}$ of F at the fixed point has spectral radius less than one,*
2. *the iteration defined in Eq. (9) is such that $\mathbf{J}_{\mathbf{n}_k,G}$ is invertible for all $k \geq 0$,*

the sequence $(\mathbf{n}_k)_k$ converges quadratically to $\boldsymbol{\mathfrak{z}}$, i.e. there exists $0 < \bar{\rho} < 1$ such that $\|\mathbf{n}_k - \boldsymbol{\mathfrak{z}}\|_2 \leq \mathcal{O}\left(\bar{\rho}^{2^k}\right)$.

The two algorithms for computing the generalized partition function of a strongly convergent WTA are summarized in Algorithms 2 and 3. From the convergence rates of these algorithms it is immediate to see that that $O(\log\log 1/\varepsilon)$ iteration of Newton's method suffice to compute the generalized partition function to a desired accuracy ε, while the fixed point iteration method requires $O(\log 1/\varepsilon)$ iterations to achieve the same accuracy. To make the comparison more precise, one also needs to understand the per-iteration complexity of each of these two methods. The computational complexity of the fixed point iteration is in $\mathcal{O}\left(|\mathcal{F}|n^P\right)$, where n is the number of states of the WTA and P is the maximal arity of symbols in $\mathcal{F}$, while the complexity of Newton's method is in $\mathcal{O}\left(|\mathcal{F}|n^P + n^3\right)$ (where the additional polynomial term in n comes from inverting the Jacobian of G). It is interesting to observe that (i) the complexity of computing the Jacobian is the same as the one of computing the mapping F and (ii) as soon as the alphabet $\mathcal{F}$ contains symbols of arity greater than 3, the two algorithms have the same asymptotic complexity. In particular, Newton's method is asymptotically more efficient than the fixed point iteration when $\mathcal{F}$ contains symbols of arity greater than 3.

Proof of Theorem 10 The remainder of this section will be devoted to proving the convergence of Newton's method starting at $\mathbf{n}_0$, the convergence rate in Theorem 10 then directly follows from classical results on Newton's method[6] (see e.g. [42, Section 8.1.10]). In addition, similarly to the case of the fixed point iteration, we will show in the next section that the assumption on the Jacobian of G in Theorem 10 is satisfied when computing the Gramian matrices of a *strongly convergent* WTA by estimating a generalized partition function.

[6] Note that the quadratic rate of convergence of Newton's method relies on the invertibility of the Jacobian $\mathbf{J}_{\boldsymbol{\mathfrak{z}},G}$ of G at the solution, which is guaranteed by assumption 1 in Theorem 10

Algorithm 2 `Generalized Partition Function - Fixed Point`

Input: A strongly convergent minimal WTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$
Output: Generalized partition function $\boldsymbol{\mathfrak{z}}_A = \sum_{t\in\mathfrak{T}_\mathcal{F}} \boldsymbol{\omega}_A(t)$

$\mathbf{f} \leftarrow \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}^\sigma \in \mathbb{R}^n$ // Initialization
repeat
  $\mathbf{f} \leftarrow \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \mathbf{f}, \ldots, \mathbf{f})$ // $\mathbf{f}_{k+1} = F(\mathbf{f}_k)$
until convergence
return $\mathbf{f}$

Algorithm 3 `Generalized Partition Function - Newton`

Input: A strongly convergent minimal WTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$
Output: Generalized partition function $\boldsymbol{\mathfrak{z}}_A = \sum_{t\in\mathfrak{T}_\mathcal{F}} \boldsymbol{\omega}_A(t)$

$\mathbf{n} \leftarrow \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}^\sigma \in \mathbb{R}^n$ // Initialization
repeat
  $\mathbf{f} \leftarrow \sum_{\sigma\in\mathcal{F}_0} \boldsymbol{\omega}_\sigma + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \mathbf{n}, \ldots, \mathbf{n})$ // $\mathbf{f} = F(\mathbf{n})$
  $\mathbf{M} \leftarrow \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{i=0}^{p-1} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \underbrace{\mathbf{n}, \ldots, \mathbf{n}}_{i \text{ times}}, \mathbf{I}, \underbrace{\mathbf{n}, \ldots, \mathbf{n}}_{p-1-i \text{ times}})$ // $\mathbf{M} = \mathbf{J}_{F,\mathbf{n}}$
  $\mathbf{n} \leftarrow \mathbf{n} + (\mathbf{I} - \mathbf{M})^{-1}(\mathbf{f} - \mathbf{n})$ // $\mathbf{n}_{new} = \mathbf{n} + (\mathbf{I} - \mathbf{J}_{F,\mathbf{n}})^{-1}(F(\mathbf{n}) - \mathbf{n})$
until convergence
return $\mathbf{n}$

Let us first recall that the convergence of the fixed point iteration defined in Eq. (7) follows from the fact that $\mathbf{f}_k = F^{k+1}(\mathbf{0}) = \sum_{t\in\mathfrak{T}\leq k} \boldsymbol{\omega}(t)$ (see Theorem 7). Intuitively, the fixed point iteration can be seen as a naive method to compute $\boldsymbol{\mathfrak{z}}$ where, at each iteration k, the set of trees of depth k is added to the current estimate:

$$\boldsymbol{\mathfrak{z}} = \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}(t) = \underbrace{\sum_{t:\ \mathrm{depth}(t)=0} \boldsymbol{\omega}(t)}_{\mathbf{f}_0} + \underbrace{\sum_{t:\ \mathrm{depth}(t)=1} \boldsymbol{\omega}(t)}_{\mathbf{f}_1-\mathbf{f}_0} + \underbrace{\sum_{t:\ \mathrm{depth}(t)=2} \boldsymbol{\omega}(t)}_{\mathbf{f}_2-\mathbf{f}_1} + \ \ldots.$$

This decomposition in terms of a telescoping series shows that the fixed point iteration method only adds *a finite set of trees* to the current estimate at every step. In contrast, we will show that Newton's method corresponds to adding an *infinite* number of trees at each iteration. This observation provides an intuitive explanation for the faster convergence of Newton's method.

We start by introducing the notion of contexts built on a subset of $\mathfrak{T}$, as well as two fundamental operations on subsets of trees.

Definition 11. *Given a set of trees $S \subset \mathfrak{T}$, the set of S-contexts $\mathfrak{C}(S) \subset \mathfrak{C}$ is the smallest set satisfying*

(i) $* \in \mathfrak{C}(S)$

(ii) $g(t_1, \ldots, t_{i-1}, c, t_{i+1}, \ldots, t_p) \in \mathfrak{C}(S)$ *for all* $c \in \mathfrak{C}(S)$, $p \geq 1$, $g \in \mathcal{F}_p$, *and* $t_1, \ldots, t_p \in S$.

Definition 12. *Given any set of contexts $C \subset \mathfrak{C}$ and any set of trees $S \subset \mathfrak{T}$, we define*

$$C[S] = \{c[t] \mid c \in C,\ t \in S\} \qquad \text{and} \qquad \mathcal{F}(S) = \{g(t_1, \ldots, t_p) \mid p \geq 0,\ g \in \mathcal{F}_p,\ t_1, \ldots, t_p \in S\}.$$

One can easily check that $\mathfrak{C} = \mathfrak{C}(\mathfrak{T})$ and $\mathcal{F}_0 \subset \mathcal{F}(S)$ for any $S \subset \mathfrak{T}$. Before diving further into the convergence of Newton's method, let us observe that the set of trees can be expressed using the operator $\mathcal{F}$ with $\mathfrak{T}_\mathcal{F} = \cup_{k\geq 0}\mathcal{F}^k(\emptyset)$. This definition of the set of trees is intrinsically related to the mapping F defined in Eq. (6) and to the fixed point iteration in the sense that $\mathbf{f}_k = F^k(\mathbf{0}) = \sum_{t\in\mathcal{F}^k(\emptyset)} \boldsymbol{\omega}(t)$ for all $k \geq 0$.

We are now ready to show how Newton iteration can be expressed, similarly to the fixed point iteration, as successive sums over growing subsets of $\mathfrak{T}$. We consider the sequence $(S_k)_k$ of subsets of $\mathfrak{T}$ defined by

$$S_0 = \mathcal{F}_0, \qquad S_{k+1} = \mathfrak{C}(S_k)[S_k] \text{ for all } k \geq 0. \tag{11}$$

We will prove the convergence of Newton's method by showing that $\mathfrak{T}_\mathcal{F} = \cup_{k\geq 0} S_k$ and $\mathbf{n}_k = \sum_{t\in S_k} \boldsymbol{\omega}(t)$ for all $k \geq 0$. Hence, whereas the fixed point iteration can be interpreted as successive summations over

finite sets of trees, Newton's method corresponds to successive summations over *infinite* sets of trees. For example, S_1 is the set of all trees whose internal nodes all have at most one child that is not a leaf[7].

We first prove two fundamental properties of S-contexts in the following proposition.

Proposition 13. *Let $S \subset \mathfrak{T}$ be* closed under subtree*, i.e. such that for any tree $t \in S$, all the subtrees of t belong to S, and let $c \in \mathfrak{C}(S)$. The following hold*

(i) any subtree τ of c is either a tree in S or a context in $\mathfrak{C}(S)$,

(ii) if $c = c_1[c_2]$ for some contexts $c_1, c_2 \in \mathfrak{C}$, then $c_1 \in \mathfrak{C}(S)$.

In particular, the proposition holds whenever S is any element $S_k \subset \mathfrak{T}$ of the sequence defined in Equation (11).

Proof. The comment about the sequence $(S_k)_k$ follows from the fact that each S_k is closed under subtree, which can be shown by induction on k.
(i) We proceed by induction on $\mathrm{drop}(c)$. If $c = *$ then $\tau = c \in \mathfrak{C}(S)$ by definition of $\mathfrak{C}(S)$. Suppose now that the result holds for any context in $\mathfrak{C}(S)$ of drop less than k and let c be a context of drop $k+1$. Then, by definition of $\mathfrak{C}(S)$, $c = g(t_1, \dots, t_{i-1}, c', t_{i+1}, \dots, t_p)$ for some $p \geq 1$, $g \in \mathcal{F}_p$, $t_1, \dots, t_p \in S$ and some $c' \in \mathfrak{C}(S)$ of drop k. Let $\tau \subset c$ be a subtree of c. Then, one of three cases can occur:

- either $\tau = c \in \mathfrak{C}(S)$,
- or $\tau \subset t_i \in S$ for some i, in which case $\tau \in S$ since S is closed under subtree,
- or τ is a subtree of c', in which case the result holds by the induction hypothesis.

(ii) We proceed by induction on $\mathrm{drop}(c_1)$. The result is trivial for $c_1 = *$. Suppose now that $c_1 = g(t_1, \dots, t_{i-1}, c', t_{i+1}, \dots, t_p)$ for some $p \geq 1$, $g \in \mathcal{F}_p$, $t_1, \dots, t_p \in S$ and some $c' \in \mathfrak{C}$. Then $c'[c_2]$ is a subtree of c, hence it belongs to $\mathfrak{C}(S)$ by (i), from which $c' \in \mathfrak{C}(S)$ follows using the induction hypothesis, and we thus have $c_1 \in \mathfrak{C}(S)$ by definition of $\mathfrak{C}(S)$. □

We now show several properties of the sequence $(S_k)_k$, including the crucial fact that $\mathfrak{T} = \cup_{k \geq 0} S_k$.

Proposition 14. *Let $k \geq 1$. The following hold:*

(i) $S_k \subset S_{k+1}$,

(ii) $t \in S_k$ if and only if any subtree of t belongs to S_k,

(iii) $S_k \subset \mathcal{F}(S_k)$,

(iv) $\cup_{k \geq 0} S_k = \mathfrak{T}$.

Proof. (i) This directly follows from the fact that $* \in \mathfrak{C}(S_k)$.
(ii) We proceed by induction on k. The case $k = 1$ is immediate. Suppose that the result holds for integers up to k and let $t \in S_{k+1}$. By definition of S_{k+1} we have $t = c[t']$ for some $c \in \mathfrak{C}(S_k)$ and some $t' \in S_k$. Let τ be a subtree of t. One of three cases can occur:

- τ is a subtree of t', in which case $\tau \in S_k$ by the induction hypothesis (since $t' \in S_k$) and $\tau \in S_{k+1}$ by (i),
- t' is a subtree of τ, in which case $\tau = c'[t']$ for some context c' which is a subtree of c, hence $\tau \in \mathfrak{C}(S_k)[S_k] = S_{k+1}$ since $c' \in \mathfrak{C}(S_k)$ by Proposition 13.(i),
- τ is a subtree of c which is not a context, in which case $\tau \in S_k$ by Proposition 13.(i), hence $\tau \in S_{k+1}$ by (i).

[7] It is worth mentioning that the trees in S_k almost coincides with the set of trees of *dimension* at most k, using the notion of dimension of a tree introduced in [29]. The difference between the two resides in the treatment of arity one symbols. For example, if f is a symbol of arity one and a a symbol of arity 0, the tree $f(f(f(a)))$ has dimension zero but belongs to the set S_3. If there are no symbols of arity one in $\mathcal{F}$, then S_k is exactly the set of trees of dimension at most k.

(iii) Let $t \in S_k$. Suppose first that $t \in \mathcal{F}_0$; then $t \in \mathcal{F}(S_k)$ since $\mathcal{F}_0 \subset \mathcal{F}(S)$ for any $S \subset \mathfrak{T}$. Now if $t \notin \mathcal{F}_0$, we can write $t = g(t_1, \dots, t_p)$ for some $p \geq 1$, $g \in \mathcal{F}_p$ and $t_1, \dots, t_p \in \mathfrak{T}$. It then follows from (ii) that $t_i \in S_k$ for all i (since they are all subtrees of $t \in S_k$) and thus $t \in \mathcal{F}(S_k)$.
(iv) We show that for any tree t, there exists a k such that $t \in S_k$, from which the result follows. We proceed by induction on $|t|$. If $t = \sigma$ then $t \in \mathcal{F}_0 = S_0$. Now suppose $t = g(t_1, \dots, t_p)$ for some $p \geq 1$, $g \in \mathcal{F}_p$, $t_1, \dots, t_p \in \mathfrak{T}$. By induction hypothesis and using (i) we have $t_1, \dots, t_p \in S_k$ for some integer k. We can write $t = c[t_1]$ with $c = g(*, t_2, \dots, t_p)$. We have $c \in \mathfrak{C}(S_k)$ (by definition of $\mathfrak{C}(S_k)$ and using $t_2, \dots, t_p \in S_k$). Hence $t = c[t_1] \in \mathfrak{C}[S_k](S_k) = S_{k+1}$ (using $t_1 \in S_k$). □

It remains to show that $\mathbf{n}_k = \sum_{t \in S_k} \boldsymbol{\omega}(t)$ for all $k \geq 0$. We start by showing how the operations introduced in Definition 12 relate to the operator F and to its Jacobian.

Proposition 15. *Let $\mathbf{v} = \sum_{t \in S} \boldsymbol{\omega}(t)$ for some $S \subset \mathfrak{T}$. Then the mapping F defined in Eq. (6) and its Jacobian around $\mathbf{v}$ (see Eq. (8)) satisfy*

(i) $F(\mathbf{v}) = \sum_{t \in \mathcal{F}(S)} \boldsymbol{\omega}(t)$,

(ii) $(\mathbf{J}_{\mathbf{v},F})^k = \sum_{c \in \mathfrak{C}(S) \cap \mathfrak{C}^k} \boldsymbol{\Xi}(c)$ *for all $k \geq 1$, where $\mathfrak{C}^k$ denote the set of contexts of drop k.*

Proof. (i) directly follows from the definitions of F and $\mathcal{F}(S)$. For (ii), we proceed by induction on k. The case $k = 0$ is immediate since $\mathfrak{C}^0 = \{*\}$ and $\boldsymbol{\Xi}(*) = \mathbf{I}$. Suppose the result holds for integers up to k. First observe that

$$\begin{aligned}
\mathbf{J}_{\mathbf{v},F} &= \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=0}^{p-1} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \underbrace{\mathbf{v}, \dots, \mathbf{v}}_{i \text{ times}}, \mathbf{I}, \underbrace{\mathbf{v}, \dots, \mathbf{v}}_{p-1-i \text{ times}}) \\
&= \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=0}^{p-1} \sum_{t_1, \dots, t_{p-1} \in S} \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}(t_1), \dots, \boldsymbol{\omega}(t_i), \mathbf{I}, \boldsymbol{\omega}(t_{i+1}), \dots, \boldsymbol{\omega}(t_{p-1})) \\
&= \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=0}^{p-1} \sum_{t_1, \dots, t_{p-1} \in S} \boldsymbol{\Xi}(g(t_1, \dots, t_i, *, t_{i+1}, \dots, t_{p-1})) \\
&= \sum_{c \in \mathfrak{C}(S) \cap \mathfrak{C}^1} \boldsymbol{\Xi}(c).
\end{aligned}$$

We then have

$$\begin{aligned}
(\mathbf{J}_{\mathbf{v},F})^{k+1} &= \mathbf{J}_{\mathbf{v},F} (\mathbf{J}_{\mathbf{v},F})^k \\
&= \sum_{c_1 \in \mathfrak{C}(S) \cap \mathfrak{C}^1} \boldsymbol{\Xi}(c_1) \sum_{c_2 \in \mathfrak{C}(S) \cap \mathfrak{C}^k} \boldsymbol{\Xi}(c_2) \\
&= \sum_{c_1 \in \mathfrak{C}(S) \cap \mathfrak{C}^1} \sum_{c_2 \in \mathfrak{C}(S) \cap \mathfrak{C}^k} \boldsymbol{\Xi}(c_1[c_2]) \\
&= \sum_{c \in \mathfrak{C}(S) \cap \mathfrak{C}^{k+1}} \boldsymbol{\Xi}(c)
\end{aligned}$$

where we used the induction hypothesis for the second equality and the last equality comes from the fact that, for any family of contexts $C \subset \mathfrak{C}$, the map $(c_1, c_2) \mapsto c_1[c_2]$ is an isomorphism between $C \cap \mathfrak{C}^1 \times C \cap \mathfrak{C}^k$ and $C \cap \mathfrak{C}^{k+1}$. □

Finally, we will need the following technical lemma.

Lemma 16. *For any $k \geq 1$, we have*

$$\mathfrak{C}(S_k) \times (\mathcal{F}(S_k) \setminus S_k) \;\cong\; \mathfrak{C}(S_k)[S_k] \setminus S_k$$

and the map $\phi \colon (c, t) \mapsto c[t]$ is an isomorphism between the two sets.

Proof. We first show that ϕ is injective. Let $(c, t), (c', t') \in \mathfrak{C}(S_k) \times (\mathcal{F}(S_k) \setminus S_k)$ be such that $c[t] = c'[t']$. First observe that t' cannot be a subtree of c: indeed since c is in $\mathfrak{C}(S_k)$, all of its subtrees are in S_k by Proposition 13.(i). Hence we have that either t is a subtree of t', or t' is a subtree of t. Suppose

the latter (a symmetric argument can be used for the former). We have $t = g(t_1, \dots, t_p)$ for some $p \geq 0$, $g \in \mathcal{F}_p$, $t_1, \dots, t_p \in S_k$ by definition of $\mathcal{F}(S_k)$. Since $t \notin S_k$ and $\mathcal{F}_0 \subset S_k$ we must have $p \geq 1$. Furthermore, by Proposition 14.(ii), t' cannot be a subtree of one of the $t_i \in S_k$ since it does not belong to S_k, hence the only way for t' to be a subtree of t is if $t' = t$, from which $c' = c$ follows.

We now show that ϕ is surjective. Let $t = c[t'] \in \mathfrak{C}(S_k)[S_k] \setminus S_k$ with $(c, t') \in \mathfrak{C}(S_k) \times S_k$. We show that $t = c'[s]$ for some $c' \in \mathfrak{C}(S_k)$ and some $s \in \mathcal{F}(S_k) \setminus S_k$. Let τ be the largest subtree of t satisfying both $\tau \in S_k$ and t' is a subtree of τ. Since $t \notin S_k$ we have $\tau \neq t$ and we can write $t = c'[g(t_1, \dots, t_{i-1}, \tau, t_{i+1}, \dots, t_p)]$ for some $p \geq 1$, $g \in \mathcal{F}_p$, $t_1, \dots, t_p \in \mathfrak{T}$ and some context c'. Since $t_1, \dots, t_p$ are subtrees of c they all belong to S_k by Proposition 13.(i), hence $s = g(t_1, \dots, t_{i-1}, \tau, t_{i+1}, \dots, t_p) \in \mathcal{F}(S_k)$ and moreover $s \notin S_k$ since τ is maximal. It remains to show that $c' \in \mathfrak{C}(S_k)$, which follows from observing that $c = c'[c'']$ for some context c'' and applying Proposition 13.(ii). □

We can now prove the convergence of Newton's method.

Theorem 17. *Under the assumption of Theorem 10, the Newton sequence* $(\mathbf{n}_k)_k$ *defined in Eq. (9) is such that*

$$\mathbf{n}_k = \sum_{t \in S_k} \boldsymbol{\omega}(t) \quad \text{for all } k \geq 0 \ .$$

Consequently, $\lim_{k \to \infty} \mathbf{n}_k = \boldsymbol{\mathfrak{z}}$.

Proof. We proceed by induction on k. The case $k = 0$ is immediate. Suppose that the result holds for integers up to k. We have

$$\begin{aligned}
\mathbf{n}_{k+1} &= \mathbf{n}_k + (\mathbf{I} - \mathbf{J}_{\mathbf{n}_k, F})^{-1}(F(\mathbf{n}_k) - \mathbf{n}_k) \\
&= \mathbf{n}_k + \sum_{i \geq 0} (\mathbf{J}_{\mathbf{n}_k, F})^i (F(\mathbf{n}_k) - \mathbf{n}_k) \\
&= \mathbf{n}_k + \sum_{i \geq 0} \sum_{c \in \mathfrak{C}(S_k) \cap \mathfrak{C}^i} \boldsymbol{\Xi}[c](F(\mathbf{n}_k) - \mathbf{n}_k) \\
&= \mathbf{n}_k + \sum_{c \in \mathfrak{C}(S_k)} \boldsymbol{\Xi}[c](F(\mathbf{n}_k) - \mathbf{n}_k).
\end{aligned}$$

where we used the fact that the spectral radius of $(\mathbf{I} - \mathbf{J}_{\mathbf{n}_k, F})$ is less than one (see assumption 1 in Theorem 10) and Proposition 15.(ii). Now, using Proposition 15.(i), the fact that $S_k \subset \mathcal{F}(S_k)$ (see Proposition 14.(iii)) and the induction hypothesis, we have $F(\mathbf{n}_k) - \mathbf{n}_k = \sum_{t \in \mathcal{F}(S_k) \setminus S_k} \boldsymbol{\omega}(t)$, hence

$$\mathbf{n}_{k+1} = \sum_{t \in S_k} \boldsymbol{\omega}(t) + \sum_{c \in \mathfrak{C}(S_k)} \sum_{t \in \mathcal{F}(S_k) \setminus S_k} \boldsymbol{\omega}(c[t]).$$

The result then follows by observing that Lemma 16 implies

$$\sum_{c \in \mathfrak{C}(S_k)} \sum_{t \in \mathcal{F}(S_k) \setminus S_k} \boldsymbol{\omega}(c[t]) = \sum_{t \in \mathfrak{C}(S_k)[S_k] \setminus S_k} \boldsymbol{\omega}(t) = \sum_{t \in S_{k+1} \setminus S_k} \boldsymbol{\omega}(t).$$

We thus showed that $\mathbf{n}_k = \sum_{t \in S_k} \boldsymbol{\omega}(t)$ for all k and $\lim_{k \to \infty} \mathbf{n}_k = \boldsymbol{\mathfrak{z}}$ follows by Proposition 14.(iv). □

5 Computation of the Gramian Matrices

We now show how the results obtained in the previous section can be leveraged to design an efficient algorithm to approximate the Gramian matrices of a strongly convergent WTA. Let $f : \mathfrak{T} \to \mathbb{R}$ be a strongly convergent recognizable function and recall that, given the rank factorization $\mathbf{H}_f = \mathbf{P}_A \mathbf{S}_A$ induced by some minimal WTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{T}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ computing f, the Gramian matrices satisfy

$$\mathbf{G}_{\mathfrak{C}} = \mathbf{P}_A^\top \mathbf{P}_A = \sum_{c \in \mathfrak{C}} \boldsymbol{\alpha}_A(c) \boldsymbol{\alpha}_A(c)^\top \quad \text{and} \quad \mathbf{G}_{\mathfrak{T}} = \mathbf{S}_A \mathbf{S}_A^\top = \sum_{t \in \mathfrak{T}} \boldsymbol{\omega}_A(t) \boldsymbol{\omega}_A(t)^\top.$$

We will show that the vectorization of the Gramian matrix $\mathbf{G}_{\mathfrak{T}}$ is the generalized partition function of the product WTA $A^{\otimes} = (\mathbb{R}^{n^2}, \tilde{\boldsymbol{\alpha}}, \{\tilde{\boldsymbol{\mathcal{A}}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\tilde{\boldsymbol{\omega}}^\sigma\}_{\sigma \in \mathcal{F}_0})$ defined by

- $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \otimes \boldsymbol{\alpha}$,

- $\tilde{\boldsymbol{\mathcal{A}}}^g = \boldsymbol{\mathcal{A}}^g \otimes \boldsymbol{\mathcal{A}}^g \in \mathbb{R}^{n^2 \times \cdots \times n^2}$ for all $p \geq 1$, $g \in \mathcal{F}_p$ and
- $\tilde{\boldsymbol{\omega}}^\sigma = \boldsymbol{\omega}^\sigma \otimes \boldsymbol{\omega}^\sigma$ for all $\sigma \in \mathcal{F}_0$.

One can easily check that $A^{\otimes}$ computes the function $f_{A^{\otimes}} \colon t \mapsto f(t)^2$ (see e.g. [19]). For the sake of brevity let $\boldsymbol{\omega} = \boldsymbol{\omega}_A$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}_A$, $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}_{A^{\otimes}}$ and $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_{A^{\otimes}}$. Let also $\tilde{\mathbf{z}} = \sum_{t \in \mathfrak{T}} \tilde{\boldsymbol{\omega}}(t)$. Now, since $\mathbf{G}_{\mathfrak{T}} = \sum_{t \in \mathfrak{T}} \boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^\top$ and $\tilde{\boldsymbol{\omega}}(t) = \mathrm{vec}(\boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^\top)$, we have $\tilde{\mathbf{z}} = \mathrm{vec}(\mathbf{G}_{\mathfrak{T}})$ and computing the Gramian matrix $\mathbf{G}_{\mathfrak{T}}$ boils down to computing the vector $\tilde{\mathbf{z}}$, which can be performed efficiently using either the fixed point or the Newton iterations derived in the previous section. In order to apply Theorem 7 and Theorem 10, we need to show that the assumptions on the Jacobian of the polynomial map F in these theorems hold in this setting; this is done in the following theorem.

Theorem 18. *Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma \in \mathcal{F}_0})$ be a minimal WTA computing a strongly convergent function f and let $A^{\otimes} = (\mathbb{R}^{n^2}, \tilde{\boldsymbol{\alpha}}, \{\tilde{\boldsymbol{\mathcal{A}}}^g\}_{g \in \mathcal{F}_{\geq 1}}, \{\tilde{\boldsymbol{\omega}}^\sigma\}_{\sigma \in \mathcal{F}_0})$ where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \otimes \boldsymbol{\alpha}$, $\tilde{\boldsymbol{\mathcal{A}}}^g = \boldsymbol{\mathcal{A}}^g \otimes \boldsymbol{\mathcal{A}}^g \in \mathbb{R}^{n^2 \times \cdots \times n^2}$ for all $p \geq 1$, $g \in \mathcal{F}_p$ and $\tilde{\boldsymbol{\omega}}^\sigma = \boldsymbol{\omega}^\sigma \otimes \boldsymbol{\omega}^\sigma$ for all $\sigma \in \mathcal{F}_0$.*

We denote by $\tilde{F}$ the mapping defined in Eq. (6) for the WTA $A^{\otimes}$, i.e.

$$\tilde{F} \colon \mathbf{v} \mapsto \sum_{\sigma \in \mathcal{F}_0} \tilde{\boldsymbol{\omega}}_\sigma + \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \mathbf{v}, \ldots, \mathbf{v}). \tag{12}$$

Then, for any set of trees $S \subset \mathfrak{T}$, the spectral radius of the Jacobian $\mathbf{J}_{\tilde{F},\mathbf{s}}$ of $\tilde{F}$ at $\mathbf{s} = \sum_{t \in S} \tilde{\boldsymbol{\omega}}(t)$ is strictly smaller than 1 and, consequently, the matrix $\mathbf{I} - \mathbf{J}_{\tilde{F},\mathbf{s}}$ is invertible.

Proof. Let $\mathbf{E} = \mathbf{J}_{\tilde{F},\mathbf{s}}$. Since A is minimal, there exist trees $t_1, \ldots, t_n \in \mathfrak{T}$ and contexts $c_1, \ldots, c_n \in \mathfrak{C}$ such that both $\{\boldsymbol{\omega}(t_i)\}_{i \in [n]}$ and $\{\boldsymbol{\alpha}(c_i)\}_{i \in [n]}$ are sets of linear independent vectors in $\mathbb{R}^n$ [9]. Therefore, the sets $\{\boldsymbol{\omega}(t_i) \otimes \boldsymbol{\omega}(t_j)\}_{i,j \in [n]}$ and $\{\boldsymbol{\alpha}(c_i) \otimes \boldsymbol{\alpha}(c_j)\}_{i,j \in [n]}$ are sets of linear independent vectors in $\mathbb{R}^{n^2}$. Let $\mathbf{v} \in \mathbb{R}^{n^2}$ be an eigenvector of $\mathbf{E}$ with eigenvalue $\lambda \neq 0$, and let $\mathbf{v} = \sum_{i,j \in [n]} \beta_{i,j}(\boldsymbol{\omega}(t_i) \otimes \boldsymbol{\omega}(t_j))$ be its expression in terms of the basis $(\boldsymbol{\omega}(t_i) \otimes \boldsymbol{\omega}(t_j))_{i,j=1,\ldots,n}$. For any $i, j \in [n]$, the vector $\mathbf{u} = \boldsymbol{\alpha}(c_i) \otimes \boldsymbol{\alpha}(c_j)$ is such that

$$\lim_{k \to \infty} \mathbf{u}^\top \mathbf{E}^k \mathbf{v} \leq \lim_{k \to \infty} |\mathbf{u}^\top \mathbf{E}^k \mathbf{v}| \leq \sum_{i,j \in [n]} |\beta_{i,j}| \lim_{k \to \infty} |\mathbf{u}^\top \mathbf{E}^k (\boldsymbol{\omega}(t_i) \otimes \boldsymbol{\omega}(t_j))| = 0 \ ,$$

where we used Lemma 19 (see below) in the last equality. Since this is true for any vector $\mathbf{u}$ in the basis $(\boldsymbol{\alpha}(c_i) \otimes \boldsymbol{\alpha}(c_j))_{i,j=1,\ldots,n}$ of $\mathbb{R}^{n^2}$, we have $\lim_{k \to \infty} \mathbf{E}^k \mathbf{v} = \lim_{k \to \infty} \lambda^k \mathbf{v} = \mathbf{0}$, hence $|\lambda| < 1$. This reasoning holds for any eigenvalue of $\mathbf{E}$, hence $\rho(\mathbf{E}) < 1$. □

Lemma 19. *Under the hypothesis (and using the notations) of Theorem 18, for any $c_1, c_2 \in \mathfrak{C}$, any $t_1, t_2 \in \mathfrak{T}$ and any $S \subset \mathfrak{T}$ we have*

$$\lim_{k \to \infty} |(\boldsymbol{\alpha}(c_1) \otimes \boldsymbol{\alpha}(c_2))^\top (\mathbf{J}_{\tilde{F},\mathbf{s}})^k (\boldsymbol{\omega}(t_1) \otimes \boldsymbol{\omega}(t_2))| = 0$$

where $\mathbf{s} = \sum_{t \in S} \tilde{\boldsymbol{\omega}}(t)$.

Proof. Let $\mathbf{E} = \mathbf{J}_{\tilde{F},\mathbf{s}}$ and let $\tilde{\boldsymbol{\Xi}} : \mathfrak{C} \to \mathbb{R}^{n^2 \times n^2}$ be the context mapping associated with the WTA $A^{\otimes}$; i.e. $\tilde{\boldsymbol{\Xi}} = \boldsymbol{\Xi}_{A^{\otimes}}$. We start by proving by induction on $\mathrm{drop}(c)$ that $\tilde{\boldsymbol{\Xi}}(c) = \boldsymbol{\Xi}(c) \otimes \boldsymbol{\Xi}(c)$ for all $c \in \mathfrak{C}$. Let $\mathfrak{C}^k$ denote the set of contexts $c \in \mathfrak{C}$ with $\mathrm{drop}(c) = k$. The statement is trivial for $c \in \mathfrak{C}^0$. Assume the statement is true for all naturals up to $k-1$ and let $c = g(c', t_1, \ldots, t_p) \in \mathfrak{C}^d$ for some $p \geq 0$, $g \in \mathcal{F}_{p+1}$, $t_1, \ldots, t_p \in \mathfrak{T}$ and $c' \in \mathfrak{C}^{k-1}$. Then using our induction hypothesis and the fact that $\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) \otimes \boldsymbol{\omega}(t)$ for any tree t, we have

$$\begin{aligned}
\tilde{\boldsymbol{\Xi}}(c) &= \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}_{n^2}, \tilde{\boldsymbol{\Xi}}(c'), \tilde{\boldsymbol{\omega}}(t_1), \ldots, \tilde{\boldsymbol{\omega}}(t_p)) \\
&= \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}_{n^2}, \boldsymbol{\Xi}(c') \otimes \boldsymbol{\Xi}(c'), \boldsymbol{\omega}(t_1) \otimes \boldsymbol{\omega}(t_1), \ldots, \boldsymbol{\omega}(t_p) \otimes \boldsymbol{\omega}(t_p)) \\
&= \boldsymbol{\mathcal{A}}^g(\mathbf{I}_n, \boldsymbol{\Xi}(c'), \boldsymbol{\omega}(t_1), \ldots, \boldsymbol{\omega}(t_p)) \otimes \boldsymbol{\mathcal{A}}^g(\mathbf{I}_n, \boldsymbol{\Xi}(c'), \boldsymbol{\omega}(t_1), \ldots, \boldsymbol{\omega}(t_p)) \\
&= \boldsymbol{\Xi}(c) \otimes \boldsymbol{\Xi}(c).
\end{aligned}$$

The case $c = g(t_1, \ldots, t_{i-1}, c', t_i, \ldots, t_p)$ for $i > 1$ follows from an identical argument.

Using the fact that $\mathbf{E}^k = \sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} \tilde{\boldsymbol{\Xi}}(c)$ (see Proposition 15) and writing $d_c = \min(\mathrm{drop}(c_1), \mathrm{drop}(c_2))$ and $d_t = \min(\mathrm{depth}(t_1), \mathrm{depth}(t_2))$, we have

$$\begin{aligned}
\left|(\boldsymbol{\alpha}(c_1)\otimes\boldsymbol{\alpha}(c_2))^\top \mathbf{E}^k (\boldsymbol{\omega}(t_1)\otimes\boldsymbol{\omega}(t_2))\right| &= \left|\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} (\boldsymbol{\alpha}(c_1)\otimes\boldsymbol{\alpha}(c_2))^\top \tilde{\boldsymbol{\Xi}}(c)(\boldsymbol{\omega}(t_1)\otimes\boldsymbol{\omega}(t_2))\right| \\
&= \left|\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} (\boldsymbol{\alpha}(c_1)\otimes\boldsymbol{\alpha}(c_2))^\top (\boldsymbol{\Xi}(c)\otimes\boldsymbol{\Xi}(c))(\boldsymbol{\omega}(t_1)\otimes\boldsymbol{\omega}(t_2))\right| \\
&= \left|\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} (\boldsymbol{\alpha}(c_1)^\top \boldsymbol{\Xi}(c)\boldsymbol{\omega}(t_1)) \cdot (\boldsymbol{\alpha}(c_2)^\top \boldsymbol{\Xi}(c)\boldsymbol{\omega}(t_2))\right| \\
&= \left|\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} f(c_1[c[t_1]])f(c_2[c[t_2]])\right| \\
&\le \left(\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} |f(c_1[c[t_1]])|\right)\left(\sum_{c\in\mathfrak{C}(S)\cap\mathfrak{C}^k} |f(c_2[c[t_2]])|\right) \\
&\le \left(\sum_{c\in\mathfrak{C}^k} |f(c_1[c[t_1]])|\right)\left(\sum_{c\in\mathfrak{C}^k} |f(c_2[c[t_2]])|\right) \\
&\le \left(\sum_{t\in\mathfrak{T}^{\ge d_c+d_t+k}} |t||f(t)|\right)^2
\end{aligned}$$

which tends to 0 with $k\to\infty$ since f is strongly convergent. To prove the last inequality, check that any tree of the form $t' = c[c'[t]]$ satisfies $\mathrm{depth}(t') \ge \mathrm{drop}(c) + \mathrm{drop}(c') + \mathrm{depth}(t)$, and that for fixed $c\in\mathfrak{C}$ and $t, t'\in\mathfrak{T}$ we have $|\{c'\in\mathfrak{C} : c[c'[t]] = t'\}| \le |t'|$ (indeed, a factorization $t' = c[c'[t]]$ is fixed once the root of t is chosen in t', which can be done in at most $|t'|$ different ways).

□

It follows that we can approximate the Gramian matrix $\mathbf{G}_\mathfrak{T}$ to an arbitrary precision by computing the generalized partition function of $A^\otimes$, benefiting from the convergence rates given in Theorems 8 and 10. Though we could derive a similar iterative algorithm to compute $\mathbf{G}_\mathfrak{C}$, it turns out that knowledge of $\tilde{\mathbf{z}} = \mathrm{vec}(\mathbf{G}_\mathfrak{T})$ provides an alternative procedure to obtain $\mathbf{G}_\mathfrak{C}$. As before, we have $\mathbf{G}_\mathfrak{C} = \mathbf{P}^\top\mathbf{P} = \sum_{c\in\mathfrak{C}} \boldsymbol{\alpha}(c)\boldsymbol{\alpha}(c)^\top$ and $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}(c)\otimes\boldsymbol{\alpha}(c)$ for all $c\in\mathfrak{C}$, hence $\mathbf{q} \triangleq \mathrm{vec}(\mathbf{G}_\mathfrak{C}) = \sum_{c\in\mathfrak{C}} \tilde{\boldsymbol{\alpha}}(c)$. First observe that the Jacobian of the mapping $\tilde{F}$, defined in Eq. (12), at the generalized partition function $\tilde{\mathbf{z}}$ is given by

$$\mathbf{J}_{\tilde{F},\tilde{\mathbf{z}}} = \sum_{p\ge1}\sum_{g\in\mathcal{F}_p}\sum_{i=0}^{p-1} \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \underbrace{\tilde{\mathbf{z}},\dots,\tilde{\mathbf{z}}}_{i \text{ times}}, \mathbf{I}, \underbrace{\tilde{\mathbf{z}},\dots,\tilde{\mathbf{z}}}_{p-1-i \text{ times}}), \tag{13}$$

which only depends on the tensors $\{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\ge1}}$ and $\tilde{\mathbf{z}}$. We can thus use the expression $\tilde{\boldsymbol{\alpha}}(c) = \boldsymbol{\Xi}_{A^\otimes}(c)^\top\tilde{\boldsymbol{\alpha}}$ to get

$$\mathbf{q}^\top = \sum_{c\in\mathfrak{C}} \tilde{\boldsymbol{\alpha}}^\top \boldsymbol{\Xi}_{A^\otimes}(c) = \tilde{\boldsymbol{\alpha}}^\top \sum_{k\ge0} (\mathbf{J}_{\tilde{F},\tilde{\mathbf{z}}})^k = \tilde{\boldsymbol{\alpha}}^\top (\mathbf{I} - \mathbf{J}_{\tilde{F},\tilde{\mathbf{z}}})^{-1},$$

where we used the facts that $(\mathbf{J}_{\tilde{F},\tilde{\mathbf{z}}})^k = \sum_{c\in\mathfrak{C}:\mathrm{drop}(c)=k} \boldsymbol{\Xi}_{A^\otimes}(c)$ (which follows from Proposition 15) and that the spectral radius of $\mathbf{J}_{\tilde{F},\tilde{\mathbf{z}}}$ is strictly less than 1 (see Theorem 18).

Algorithm 4 summarizes the overall approximation procedure for the Gramian matrices, which can be done to an arbitrary precision. There, $\mathrm{reshape}(\cdot, n\times n)$ is an operation that takes an n^2-dimensional vector and returns the $n\times n$ matrix whose first column contains the first n entries in the vector and so on. Theoretical guarantees on the convergence rate of this algorithm are summarized in the following theorem.

Theorem 20. *There exist $\rho, \bar{\rho} \in (0,1)$ such that, after k iterations of the iterative method used to compute the generalized partition function in line 7 of Algorithm 4, the approximations $\hat{\mathbf{G}}_{\mathfrak{C}}$ and $\hat{\mathbf{G}}_{\mathfrak{T}}$ satisfy*

$$\|\mathbf{G}_{\mathfrak{C}} - \hat{\mathbf{G}}_{\mathfrak{C}}\|_F \leq \mathcal{O}(\rho^k) \text{ and } \|\mathbf{G}_{\mathfrak{T}} - \hat{\mathbf{G}}_{\mathfrak{T}}\|_F \leq \mathcal{O}(\rho^k)$$

when using the fixed-point iteration method, and

$$\|\mathbf{G}_{\mathfrak{C}} - \hat{\mathbf{G}}_{\mathfrak{C}}\|_F \leq \mathcal{O}(\bar{\rho}^{2^k}) \text{ and } \|\mathbf{G}_{\mathfrak{T}} - \hat{\mathbf{G}}_{\mathfrak{T}}\|_F \leq \mathcal{O}(\bar{\rho}^{2^k})$$

when using Newton's method.

Proof. The result for the Gramian matrix $\mathbf{G}_{\mathfrak{T}}$ directly follows from applying Theorems 8 and 10, whose assumptions are satisfied by Theorem 18. We now show how the error in the approximation of $\mathbf{G}_{\mathfrak{T}} = \text{reshape}(\mathbf{s}, n \times n)$ affects the approximation of $\mathbf{q} = (\boldsymbol{\alpha}^{\otimes})^\top (\mathbf{I} - \mathbf{E})^{-1} = \text{vec}(\mathbf{G}_{\mathfrak{C}})$. Let $\hat{\mathbf{s}} \in \mathbb{R}^{n^2}$ be such that $\|\mathbf{s} - \hat{\mathbf{s}}\| \leq \varepsilon$, let

$$\hat{\mathbf{E}} = \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} \sum_{i=0}^{p-1} \tilde{\boldsymbol{\mathcal{A}}}^g (\mathbf{I}, \underbrace{\hat{\mathbf{s}}, \ldots, \hat{\mathbf{s}}}_{i \text{ times}}, \mathbf{I}, \underbrace{\hat{\mathbf{s}}, \ldots, \hat{\mathbf{s}}}_{p-1-i \text{ times}})$$

and let $\mathbf{q} = \tilde{\boldsymbol{\alpha}}^\top (\mathbf{I} - \hat{\mathbf{E}})^{-1}$. We first bound the distance between $\mathbf{E}$ and $\hat{\mathbf{E}}$. For any vector $\mathbf{v}$ we denote by $\mathbf{v}^{\otimes k} = \mathbf{v} \otimes \mathbf{v} \otimes \cdots \otimes \mathbf{v}$ (k times) its kth Kronecker power. Using the fact that for any symbol g of arity p we have $\|\tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \mathbf{s}, \ldots, \mathbf{s}, \mathbf{I}, \mathbf{s}, \ldots, \mathbf{s}) - \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \hat{\mathbf{s}}, \ldots, \hat{\mathbf{s}}, \mathbf{I}, \hat{\mathbf{s}}, \ldots, \hat{\mathbf{s}})\|_F = \|\mathbf{A}(\mathbf{s}^{\otimes p-1} - \hat{\mathbf{s}}^{\otimes p-1})\|_F$ where $\mathbf{A}$ is the matricization of $\tilde{\boldsymbol{\mathcal{A}}}^g$ obtained by mapping the two modes multiplied by $\mathbf{I}$ to rows and the remaining modes to columns, we obtain

$$\begin{aligned}
\|\mathbf{E} - \hat{\mathbf{E}}\|_F &\leq \sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} p \|\tilde{\boldsymbol{\mathcal{A}}}^g\|_F \|\mathbf{s}^{\otimes p-1} - \hat{\mathbf{s}}^{\otimes p-1}\| \\
&\leq \left(\sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} p \|\tilde{\boldsymbol{\mathcal{A}}}^g\|_F (\|\mathbf{s}\| + \|\hat{\mathbf{s}}\|)^{p-2} \right) \|\mathbf{s} - \hat{\mathbf{s}}\| \\
&\leq \left(\sum_{p \geq 1} \sum_{g \in \mathcal{F}_p} p \|\tilde{\boldsymbol{\mathcal{A}}}^g\|_F (2\|\mathbf{s}\| + \varepsilon)^{p-2} \right) \varepsilon \\
&= \mathcal{O}(\varepsilon)
\end{aligned}$$

as $\varepsilon \to 0$, where we used the inequality $\|\mathbf{x}^{\otimes k} - \mathbf{y}^{\otimes k}\| \leq (\|\mathbf{x}\| + \|\mathbf{y}\|)^{k-1} \|\mathbf{x} - \mathbf{y}\|$ for any $k \geq 1$.

Let $\delta = \|\mathbf{E} - \hat{\mathbf{E}}\|$ and let $\mathfrak{s}$ be the smallest nonzero eigenvalue of the matrix $\mathbf{I} - \mathbf{E}$. It follows from [27, Equation 7.2] that if $\delta < \mathfrak{s}$ then $\|(\mathbf{I} - \mathbf{E})^{-1} - (\mathbf{I} - \hat{\mathbf{E}})^{-1}\| \leq \delta / (\mathfrak{s}(\mathfrak{s} - \delta))$. Since $\delta = \mathcal{O}(\varepsilon)$ from our previous bound, the condition $\delta \leq \mathfrak{s}/2$ will be eventually satisfied as $\varepsilon \to 0$, in which case we can conclude that

$$\|\mathbf{G}_{\mathfrak{C}} - \hat{\mathbf{G}}_{\mathfrak{C}}\|_F = \|\mathbf{q} - \hat{\mathbf{q}}\| \leq \|(\mathbf{I} - \mathbf{E})^{-1} - (\mathbf{I} - \hat{\mathbf{E}})^{-1}\| \|\tilde{\boldsymbol{\alpha}}\| \leq \frac{2\delta}{\mathfrak{s}^2} \|\tilde{\boldsymbol{\alpha}}\| = \mathcal{O}(\varepsilon).$$

□

6 Approximate Minimization of Weighted Tree Automata

We now turn back to the problem of approximate minimization: given a WTA A with n states, can we find a WTA $\hat{A}$ with $\hat{n} < n$ states that is a good approximation of A, that is, such that f_A is close to $f_{\hat{A}}$? We propose a principled way to approach this problem that consists in removing states from the SVTA computing f_A and analyze the approximation error induced by this truncation.

6.1 SVTA Truncation

Let f be a recognizable function and let A be a WTA computing f. Recall that the value $f_A(c[t])$ is given by the inner product $\langle \boldsymbol{\alpha}_A(c), \boldsymbol{\omega}_A(t) \rangle = \sum_i (\boldsymbol{\alpha}_A(c))_i (\boldsymbol{\omega}_A(t))_i$. Thus, $(\boldsymbol{\alpha}_A(c))_i$ and $(\boldsymbol{\omega}_A(t))_i$ quantify the influence of state i in the computation of $f_A(c[t])$. By extension, given the rank factorization $\mathbf{H}_f = \mathbf{P}_A \mathbf{S}_A$, one can use $\|\mathbf{P}_A(:, i)\|$ and $\|\mathbf{S}_A(i, :)\|$ to measure the overall influence of state i in f_A. Since our goal

Algorithm 4 `GramMatrices`

Input: A strongly convergent minimal WTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$

Output: Gramian matrices $\hat{\mathbf{G}}_{\mathfrak{C}} \simeq \sum_{c\in\mathfrak{C}} \boldsymbol{\alpha}_A(c)\boldsymbol{\alpha}_A(c)^\top$ and $\hat{\mathbf{G}}_{\mathfrak{T}} \simeq \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}_A(t)\boldsymbol{\omega}_A(t)^\top$

1: // *Initialization of the parameters of the product WTA* $A^\otimes$
2: $\tilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha} \otimes \boldsymbol{\alpha}$
3: $\tilde{\boldsymbol{\mathcal{A}}}^g \leftarrow \boldsymbol{\mathcal{A}}^g \otimes \boldsymbol{\mathcal{A}}^g \in (\mathbb{R}^{n^2})^{\otimes(\sharp g+1)}$ for each $g \in \mathcal{F}_{\geq 1}$.
4: $\tilde{\boldsymbol{\omega}}^\sigma \leftarrow \boldsymbol{\omega}^\sigma \otimes \boldsymbol{\omega}^\sigma \in \mathbb{R}^{n^2}$ for each $\sigma \in \mathcal{F}_0$.
5: // *Iterative method to approximate* $\mathbf{s} = \mathrm{vec}(\mathbf{G}_{\mathfrak{T}})$
6: Define the product WTA $A^\otimes = (\mathbb{R}^{n^2}, \tilde{\boldsymbol{\alpha}}, \{\tilde{\boldsymbol{\mathcal{A}}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\tilde{\boldsymbol{\omega}}^\sigma\}_{\sigma\in\mathcal{F}_0})$
7: Compute the generalized partition function $\mathbf{s} \simeq \sum_{t\in\mathfrak{T}} \boldsymbol{\omega}_{A^\otimes}(t)$ of $A^\otimes$ using Algorithm 2 or 3.
8: // *Approximation of* $\mathbf{q} = \mathrm{vec}(\mathbf{G}_{\mathfrak{C}})$
9: $\mathbf{E} \leftarrow \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{i=0}^{p-1} \tilde{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \underbrace{\mathbf{s},\dots,\mathbf{s}}_{i \text{ times}}, \mathbf{I}, \underbrace{\mathbf{s},\dots,\mathbf{s}}_{p-1-i \text{ times}})$
10: $\mathbf{q} \leftarrow (\boldsymbol{\alpha}\otimes\boldsymbol{\alpha})^\top (\mathbf{I} - \mathbf{E})^{-1}$
11: $\hat{\mathbf{G}}_{\mathfrak{T}} \leftarrow \mathrm{reshape}(\mathbf{s}, n \times n)$
12: $\hat{\mathbf{G}}_{\mathfrak{C}} \leftarrow \mathrm{reshape}(\mathbf{q}, n \times n)$
13: **return** $\hat{\mathbf{G}}_{\mathfrak{C}}, \hat{\mathbf{G}}_{\mathfrak{T}}$

is to approximate a given WTA by a smaller WTA obtained by removing some states in the original one, we shall proceed by removing those states with overall less influence on the computation of f. But because there are infinitely many WTAs computing f, we need to first fix a particular representation for f before we can remove the less influential states. In particular, we seek a representation where each state is decoupled as much as possible from each other state, and where there is a clear ranking of states in terms of overall influence: the SVTA canonical form is thus a perfect candidate. Indeed, let $\tilde{A}$ be the SVTA computing f and let $\mathbf{H}_f = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the SVD of $\mathbf{H}_f$. If $\mathrm{rank}(f) = n$, then $\tilde{A}$ has n states and for every $i \in [n]$ the ith state contributes to $\mathbf{H}_f$ by generating the ith left and right singular vectors weighted by $\sqrt{\mathfrak{s}_i}$, where $\mathfrak{s}_i = \mathbf{D}_{i,i}$ is the ith singular value. Thus, if we want to obtain a good approximation $\hat{f}$ to f with $\hat{n}$ states, we can take the WTA $\hat{A}$ obtained by removing the last $n - \hat{n}$ states from $\tilde{A}$, which corresponds to removing from f the contribution of the smallest singular values of $\mathbf{H}_f$. We call such $\hat{A}$ an *SVTA truncation*.

6.2 Approximation Error of an SVTA Truncation

In this section, we analyze the approximation error induced by the truncation of an SVTA. Given an SVTA $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$, its truncation to $\hat{n}$ states is the automaton

$$\hat{A} = (\mathbb{R}^{\hat{n}}, \boldsymbol{\Pi}\boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g(\boldsymbol{\Pi}^\top,\dots,\boldsymbol{\Pi}^\top)\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\Pi}\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$$

where $\boldsymbol{\Pi} = \begin{bmatrix}\mathbf{I}_{\hat{n}} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{\hat{n}\times n}$ is the projection matrix that removes the states associated with the $n - \hat{n}$ smallest singular values of the Hankel matrix.

Intuitively, the states associated with the smaller singular values are the ones with the less influence on the Hankel matrix, thus they should also be the states having the less effect on the computation of the SVTA. The following theorem supports this intuition by showing a fundamental relation between the singular values of the Hankel matrix of a recognizable function f and the parameters of the SVTA computing it.

Proposition 21. *Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ be an SVTA with n states computing a function f and let $\mathfrak{s}_1 \geq \mathfrak{s}_2 \geq \cdots \geq \mathfrak{s}_n$ be the singular values of the Hankel matrix $\mathbf{H}_f$.*

Then, for all indices $i, i_1, \dots, i_{p+1} \in [n]$, the following hold:

- $|\boldsymbol{\omega}(t)_i| \leq \sqrt{\mathfrak{s}_i}$ *for any* $t \in \mathfrak{T}$,
- $|\boldsymbol{\alpha}(c)_i| \leq \sqrt{\mathfrak{s}_i}$ *for any* $c \in \mathfrak{C}$, *and*
- $|\boldsymbol{\mathcal{A}}^g_{i_1\dots i_{p+1}}| \leq \min_{k\in[p+1]} \dfrac{\mathfrak{s}_{i_k}}{\sqrt{\mathfrak{s}_{i_1}\cdots\mathfrak{s}_{i_{p+1}}}}$ *for any* $p \geq 1$, $g \in \mathcal{F}_p$.

Proof. For the first point, let $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the SVD of $\mathbf{H}_f$. Since A is an SVTA we have

$$\boldsymbol{\omega}(t)_i^2 = (\mathbf{S}_{i,t})^2 = (\mathbf{D}^{1/2}\mathbf{V}^\top)_{i,t}^2 = \mathfrak{s}_i(\mathbf{V}_{t,i})^2$$

and since the rows of $\mathbf{V}$ are orthonormal we have $(\mathbf{V}_{t,i})^2 \leq 1$. The inequality for contexts is proved similarly by reasoning on the rows of $\mathbf{U}\mathbf{D}^{1/2}$.

The third point is a direct consequence of the fixed point equations for $\mathbf{G}_{\mathfrak{T}}$ and $\mathbf{G}_{\mathfrak{C}}$ given in Theorem 4. Indeed, since A is an SVTA we have $(\mathbf{G}_{\mathfrak{T}})_{i,i} = (\mathbf{G}_{\mathfrak{C}})_{i,i} = \mathfrak{s}_i$, it is then easy to check that the fixed point equations imply

$$\mathfrak{s}_i = \sum_{\sigma\in\Sigma}(\boldsymbol{\omega}^\sigma)_i^2 + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{j_1,\ldots,j_p=1}^{n}(\boldsymbol{\mathcal{A}}^g_{i,j_1,\ldots,j_p})^2\mathfrak{s}_{j_1}\mathfrak{s}_{j_2}\ldots\mathfrak{s}_{j_p}$$

and

$$\mathfrak{s}_i = \boldsymbol{\alpha}_i^2 + \sum_{p\geq 1}\sum_{g\in\mathcal{F}_p}\sum_{j_1,\ldots,j_p=1}^{n}((\boldsymbol{\mathcal{A}}^g_{j_1,i,j_2,\ldots,j_p})^2 + (\boldsymbol{\mathcal{A}}^g_{j_1,j_2,i,j_3,\ldots,j_p})^2 + \cdots + (\boldsymbol{\mathcal{A}}^g_{j_1,\ldots,j_p,i})^2)\mathfrak{s}_{j_1}\ldots\mathfrak{s}_{j_p}$$

for all $i \in [n]$. The result follows from observing that all the summands in the two equations are positive. □

Two important properties of SVTAs follow from this proposition. First, the fact that $|\boldsymbol{\omega}(t)_i| \leq \sqrt{\mathfrak{s}_i}$ implies that the weights associated with states corresponding to small singular values are small. Second, this proposition gives us some intuition on how the states of an SVTA interact with each other. To see this, let $g \in \mathcal{F}_2$ be a symbol of arity 2 and let $\mathbf{M} = \boldsymbol{\mathcal{A}}^g(\boldsymbol{\alpha}, \mathbf{I}, \mathbf{I})$. Then for a tree $t = g(t_1, t_2) \in \mathfrak{T}$ we have $f_A(t) = \boldsymbol{\omega}(t_1)^\top\mathbf{M}\boldsymbol{\omega}(t_2)$. Using the previous proposition one can show[8] that

$$|\mathbf{M}_{ij}| \leq n\sqrt{\frac{\min\{\mathfrak{s}_i, \mathfrak{s}_j\}}{\max\{\mathfrak{s}_i, \mathfrak{s}_j\}}},$$

which tells us that two states corresponding to singular values far away from each other have very little interaction in the computations of the automata.

Proposition 21 is key to proving the following theorem, which is the main result of this section. It shows how the approximation error induced by the truncation of an SVTA is impacted by the magnitudes of the singular values associated with the removed states.

Theorem 22. *Let $P = \max_{g\in\Sigma}\sharp g$ be the maximum arity of symbols in $\mathcal{F} = (\Sigma, \sharp)$. Let $f : \mathfrak{T}_{\mathcal{F}} \to \mathbb{R}$ be a function computed by an SVTA with n states and let $\hat{f}$ be the function computed by its truncation to $\hat{n}$ states. Then, for any tree $t \in \mathfrak{T}$ we have $|f(t) - \hat{f}(t)| \leq n^{P|t|}\mathfrak{s}_{\hat{n}+1}$.*

Proof. The proof of this theorem is postponed to the end of this section. □

Since $\mathfrak{s}_{\hat{n}+1} > \mathfrak{s}_{\hat{n}+2} > \cdots > \mathfrak{s}_n$, this theorem shows that the smaller the singular values associated with the removed states are, the better will be the approximation. As a direct consequence, the error introduced by the truncation grows with the number of states removed. The dependence on the size of the trees comes from the propagation of the error during the contractions of the tensor $\hat{\boldsymbol{\mathcal{T}}}$ of the truncated SVTA. One can easily use this theorem to show that, when the singular value $\mathfrak{s}_{\hat{n}+1}$ is small, the truncated SVTA will achieve low approximation error on all trees up to some maximal size: for any $\varepsilon > \mathfrak{s}_{\hat{n}+1}$ and any tree $t \in \mathfrak{T}$ of size at most $\frac{\log\varepsilon - \log\mathfrak{s}_{\hat{n}+1}}{P\log n}$, we have $|f(t) - \hat{f}(t)| \leq \varepsilon$.

The decay of singular values can be very slow in the worst case, but in practice it is not unusual to observe an exponential decay on the tail. Assuming such an exponential decay of the form $\mathfrak{s}_i = C\theta^i$ for some $0 < \theta < 1$, the bound above on the size of the trees for which $|f(t) - \hat{f}(t)| < \varepsilon$ specializes to

$$\frac{\log(\varepsilon) + (\hat{n}+1)\log(1/\theta) - \log(C)}{P\log n}.$$

It is interesting to observe that the dependence of this bound on the number of total/removed states is $\mathcal{O}(\hat{n}/\log(n))$.

[8] Indeed, we have $|\mathbf{M}_{i,j}| = |(\boldsymbol{\mathcal{A}}^g(\boldsymbol{\alpha}, \mathbf{I}, \mathbf{I}))_{i,j}| \leq \sum_{k=1}^n |\boldsymbol{\mathcal{A}}^g_{k,i,j}|\,|\boldsymbol{\alpha}_k|$, hence it follows from Proposition 21 that $|\mathbf{M}_{i,j}| \leq \sum_{k=1}^n \min\{\sqrt{\mathfrak{s}_i/(\mathfrak{s}_j\mathfrak{s}_k)}, \sqrt{\mathfrak{s}_j/(\mathfrak{s}_i\mathfrak{s}_k)}\}\sqrt{\mathfrak{s}_k} = n\cdot\min\{\sqrt{\mathfrak{s}_i/\mathfrak{s}_j}, \sqrt{\mathfrak{s}_j/\mathfrak{s}_i}\}$.

Proof of Theorem 22 Let $A = (\mathbb{R}^n, \boldsymbol{\alpha}, \{\boldsymbol{\mathcal{A}}^g\}_{g\in\mathcal{F}_{\geq 1}}, \{\boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$ be an SVTA with n states computing a function f and let $1 \leq \hat{n} < n$. We consider the WTA with n states

$$\hat{A} = (\mathbb{R}^n, \hat{\boldsymbol{\alpha}} = \boldsymbol{\Pi}\boldsymbol{\alpha}, \{\hat{\boldsymbol{\mathcal{A}}}^g = \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}, \dots, \boldsymbol{\Pi})\}_{g\in\mathcal{F}_{\geq 1}}, \{\hat{\boldsymbol{\omega}}^\sigma = \boldsymbol{\omega}^\sigma\}_{\sigma\in\mathcal{F}_0})$$

where $\boldsymbol{\Pi} \in \mathbb{R}^{n\times n}$ is the projection matrix defined by $\boldsymbol{\Pi}_{i,i} = 1$ if $i \leq \hat{n}$ and 0 otherwise. It is easy to check that the function $\hat{f}$ computed by the WTA $\hat{A}$ is equal to the one computed by the truncation of the SVTA A to $\hat{n}$ states. Note that for any tree t we have

$$|f(t) - \hat{f}(t)| = |\boldsymbol{\alpha}^\top \boldsymbol{\omega}(t) - \hat{\boldsymbol{\alpha}}^\top \hat{\boldsymbol{\omega}}(t)| = |\boldsymbol{\alpha}^\top (\boldsymbol{\omega}(t) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))|.$$

We start by bounding the magnitude of the components of the vectors $\hat{\boldsymbol{\omega}}(t)$ and $(\boldsymbol{\omega}(t) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))$ for any tree t in the following lemmas.

Lemma 23. *For any tree $t \in \mathfrak{T}$ and any $i \in [n]$ we have $|\hat{\boldsymbol{\omega}}(t)_i| \leq n^{|t|-1}\sqrt{\mathfrak{s}_i}$.*

Proof. We proceed by induction on the size of t. If $t = \sigma \in \mathcal{F}_0$ we have $|\hat{\boldsymbol{\omega}}(\sigma)_i| = |\boldsymbol{\omega}(\sigma)_i| \leq \sqrt{\mathfrak{s}_i}$ by Proposition 21. Suppose the result holds for trees of size at most m and let $t = g(t_1, \dots, t_p)$ be a tree of size $m+1$. We have

$$\begin{aligned}
|\hat{\boldsymbol{\omega}}(t)_i| &= |(\boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \dots, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_p)))_i| \\
&= \left| \sum_{j_1,\dots,j_p=1}^{n} \boldsymbol{\mathcal{A}}^g_{i,j_1,\dots,j_p} (\boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1))_{j_1} \dots (\boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_p))_{j_p} \right| \\
&\leq \sum_{j_1,\dots,j_p=1}^{n} \left|\boldsymbol{\mathcal{A}}^g_{i,j_1,\dots,j_p}\right| |\hat{\boldsymbol{\omega}}(t_1)_{j_1}| \dots |\hat{\boldsymbol{\omega}}(t_p)_{j_p}| \\
&\leq \sum_{j_1,\dots,j_p=1}^{n} \frac{\sqrt{\mathfrak{s}_i}}{\sqrt{\mathfrak{s}_{j_1} \dots \mathfrak{s}_{j_p}}} n^{|t_1|-1}\sqrt{\mathfrak{s}_{j_1}} \dots n^{|t_p|-1}\sqrt{\mathfrak{s}_{j_p}} \\
&= n^p n^{|t_1|-1} \dots n^{|t_p|-1}\sqrt{\mathfrak{s}_i} = n^{|t|-1}\sqrt{\mathfrak{s}_i}
\end{aligned}$$

where we used the induction hypothesis and Proposition 21 for the last inequality. □

Lemma 24. *Let $P = \max_{g\in\Sigma} \sharp g$ be the maximum arity of symbols in $\mathcal{F} = (\Sigma, \sharp)$. Then for any tree t and any $i \in [n]$ we have $|\boldsymbol{\omega}(t)_i - (\boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))_i| \leq n^{P(|t|-1)} \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}}$.*

Proof. If $i > \hat{n}$ then $|\boldsymbol{\omega}(t)_i - (\Pi\hat{\boldsymbol{\omega}}(t))_i| = |\boldsymbol{\omega}(t)_i| \leq \sqrt{\mathfrak{s}_i} \leq \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}}$ (where we used Proposition 21 for the first inequality). For $i \leq \hat{n}$ we proceed by induction on the size of t. If $t = \sigma \in \mathcal{F}_0$ we have $|\boldsymbol{\omega}(t)_i - (\Pi\hat{\boldsymbol{\omega}}(t))_i| = |\boldsymbol{\omega}(t)_i - \hat{\boldsymbol{\omega}}(t)_i| = 0$ because $\hat{\boldsymbol{\omega}}^\sigma = \boldsymbol{\omega}^\sigma$. Suppose the result holds for trees of size at most m and let $t = g(t_1, \dots, t_p)$ be a tree of size $m+1$. Since $i \leq \hat{n}$ we have $\boldsymbol{\omega}(t)_i - (\Pi\hat{\boldsymbol{\omega}}(t))_i = (\boldsymbol{\omega}(t) - \hat{\boldsymbol{\omega}}(t))_i$. First, we have

$$\begin{aligned}
\boldsymbol{\omega}(t) - \hat{\boldsymbol{\omega}}(t) &= \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}(t_1), \dots, \boldsymbol{\omega}(t_p)) - \hat{\boldsymbol{\mathcal{A}}}^g(\mathbf{I}, \hat{\boldsymbol{\omega}}(t_1), \dots, \hat{\boldsymbol{\omega}}(t_p)) \\
&= \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}(t_1), \dots, \boldsymbol{\omega}(t_p)) - \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \dots, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_p)) \\
&= \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\omega}(t_1) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \boldsymbol{\omega}(t_2), \dots, \boldsymbol{\omega}(t_p)) \\
&+ \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \boldsymbol{\omega}(t_2) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_2), \boldsymbol{\omega}(t_3), \dots, \boldsymbol{\omega}(t_p)) \\
&+ \dots + \boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \dots, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_{p-1}), \boldsymbol{\omega}(t_p) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_p)).
\end{aligned}$$

Now for any $k \in [p]$, using the induction hypothesis and the bounds $|\boldsymbol{\omega}(t)_i| \leq \sqrt{\mathfrak{s}_i}$, $|\boldsymbol{\mathcal{A}}^g_{i_1\dots i_{p+1}}| \leq$

$\min_{k\in[p+1]} \frac{\mathfrak{s}_{i_k}}{\sqrt{\mathfrak{s}_{i_1}\cdots\mathfrak{s}_{i_{p+1}}}}$ (from Proposition 21) and $|\hat{\boldsymbol{\omega}}(t)_i| \leq n^{|t|-1}\sqrt{\mathfrak{s}_i}$ (from the previous lemma) we get

$$
\begin{aligned}
&|\boldsymbol{\mathcal{A}}^g(\mathbf{I}, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_1), \ldots, \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_{k-1}), \boldsymbol{\omega}(t_k) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_k), \boldsymbol{\omega}(t_{k+1}), \ldots, \boldsymbol{\omega}(t_p))_i| \\
&\leq \sum_{j_1,\ldots,j_p=1}^{n} \left|\boldsymbol{\mathcal{A}}^g_{i,j_1,\ldots,j_p}\right| \left(\prod_{r=1}^{k-1} |(\boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_r))_{j_r}|\right) |(\boldsymbol{\omega}(t_k) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t_k))_{j_k}| \left(\prod_{r=k+1}^{p} |\boldsymbol{\omega}(t_r)_{j_r}|\right) \\
&\leq \sum_{j_1,\ldots,j_p=1}^{n} \frac{\sqrt{\mathfrak{s}_{j_k}}}{\sqrt{\mathfrak{s}_i \mathfrak{s}_{j_1}\cdots\mathfrak{s}_{j_{k-1}}\mathfrak{s}_{j_{k+1}}\cdots\mathfrak{s}_{j_p}}} \left(\prod_{r=1}^{k-1} n^{|t_r|-1}\sqrt{\mathfrak{s}_{j_r}}\right) n^{P(|t_k|-1)} \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_{j_k}}} \left(\prod_{r=k+1}^{p} \sqrt{\mathfrak{s}_{j_r}}\right) \\
&= \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}} n^p \left(\prod_{r=1}^{k-1} n^{|t_r|-1}\right) n^{P(|t_k|-1)} \\
&\leq \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}} n^{P|t_k|} \left(\prod_{r=1}^{k-1} n^{|t_r|-1}\right).
\end{aligned}
$$

It follows that

$$
|(\boldsymbol{\omega}(t) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))_i| \leq \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}} \sum_{k=1}^{p} n^{P|t_k|} \left(\prod_{r=1}^{k-1} n^{|t_r|-1}\right).
$$

Using the fact that $n \geq 2$, one can check by induction on p that

$$
\sum_{k=1}^{p} n^{P|t_k|} \left(\prod_{r=1}^{k-1} n^{|t_r|-1}\right) = \sum_{k=1}^{p} n^{P|t_k|+|t_{k-1}|+\cdots+|t_1|-k+1} \leq n^{P(|t_k|+|t_{k-1}|+\cdots+|t_1|)} = n^{P(|t|-1)},
$$

which concludes the proof. □

We can now combine the previous lemma with Proposition 21 to show Theorem 22. Indeed, for any tree t we have

$$
\begin{aligned}
|f(t) - \hat{f}(t)| &= |\boldsymbol{\alpha}^\top(\boldsymbol{\omega}(t) - \boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))| \\
&\leq \sum_{i=1}^{n} |\boldsymbol{\alpha}_i| \; |\boldsymbol{\omega}(t)_i - (\boldsymbol{\Pi}\hat{\boldsymbol{\omega}}(t))_i| \\
&\leq \sum_{i=1}^{n} \sqrt{\mathfrak{s}_i} \; n^{P(|t|-1)} \frac{\mathfrak{s}_{\hat{n}+1}}{\sqrt{\mathfrak{s}_i}} \\
&\leq \; n^{P|t|}\mathfrak{s}_{\hat{n}+1}.
\end{aligned}
$$

7 Conclusion

We described a technique for approximate minimization of WTA, yielding a model smaller than the original one which retains good approximation properties. We introduced a canonical form of WTA in which the states of the automaton are associated with singular values of the corresponding Hankel matrix. This canonical form allowed us to achieve approximate minimization in a principled way by removing states corresponding to small singular values of the Hankel matrix. We also provided theoretical approximation guarantees for this minimization scheme. Our main algorithm relies on a singular value decomposition of the infinite Hankel matrix induced by the WTA, and the main technical difficulty to extend the method proposed in [15] to the tree case resided in the computation of the Gramian matrices associated with the WTA. Even though these Gramian matrices do not have a closed form solution in the tree case, we proposed an efficient algorithm to approximate them to an arbitrary precision. This algorithm relies on an iterative procedure to compute the generalized partition function of a WTA, a problem which may be of independent interest. We considered two approaches for computing the generalized partition function, one based on a simple fixed point iteration and a second one based on Newton's method which allowed us to obtain faster convergence rates.

In future work, we will try to obtain better approximation guarantees of an SVTA truncation. In [16], Balle et al. obtained the following bound in the string case:

$$
\|f - \hat{f}\|_2^2 \leq \mathfrak{s}_{\hat{n}+1}^2 + \cdots + \mathfrak{s}_n^2.
$$

In comparison to the tree case (Theorem 22), this bound does not exhibit any dependency on the size of the strings. We conjecture that this bound should also hold in the tree case but, even though simulation studies suggest that this is the case, we did not yet manage to derive such a bound for WTA.

We also believe that the combinatorial construction unraveled in the proof of convergence of Newton's method in Section 4 may be of independent interest and is worth exploring further. In particular, we plan to revisit this construction in other contexts where the problem of computing a generalized partition function appears, for example when performing marginalization operations on probabilistic context free grammars or for reachability analysis of branching Markov decision processes [31]. As mentioned in the introduction, this construction bares a striking similarity with the technique developed in [29] for computing fixed points of system of equations over ω-continuous semi-rings. In this work, Esperza et al. proposes an extension of Newton's method for ω-continuous semi-rings and show that its iterates correspond to successive summations over (infinite) subset of derivation trees of bounded dimension. While the similarity with Theorem 17 is striking, the two methods are incomparable since our results extend beyond ω-continuous semi-rings while being restricted to the computation of generalized partition functions for weighted tree automata. Nonetheless, a more fundamental relation between Newton's method, the notion of dimension of derivation trees, and the duality between contexts and trees may be at play here, and potentially deserves further investigation.

Acknowledgments

Guillaume Rabusseau gratefully acknowledges the support of the Canadian Institute for Advanced Research (CIFAR CCAI program) and of the Natural Sciences and Engineering Research Council of Canada.

References

[1] A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. Liu. A spectral algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

[2] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[3] A. Anandkumar, D. Hsu, F. Huang, and S. M. Kakade. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pages 1052–1060, 2012.

[4] D. Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2), 1987.

[5] P.-L. Bacon, B. Balle, and D. Precup. Learning and planning with timing information in Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence*, pages 111–120, 2015.

[6] R. Bailly, X. Carreras, F. Luque, and A. Quattoni. Unsupervised spectral learning of WCFG as low-rank matrix completion. In *Conference on Empirical Methods in Natural Language Processing*, 2013.

[7] R. Bailly, X. Carreras, and A. Quattoni. Unsupervised spectral learning of finite state transducers. In *Conference on Uncertainty in Artificial Intelligence*, 2013.

[8] R. Bailly, F. Denis, and L. Ralaivola. Grammatical inference as a principal component analysis problem. In *International Conference on Machine Learning*, pages 33–40, 2009.

[9] R. Bailly, A. Habrard, and F. Denis. A spectral approach for probabilistic grammatical inference on trees. In *Algorithmic Learning Theory*, pages 74–88. Springer, 2010.

[10] B. Balle, X. Carreras, F. M. Luque, and A. Quattoni. Spectral learning of weighted automata. *Machine learning*, 96(1-2):33–63, 2014.

[11] B. Balle, W. Hamilton, and J. Pineau. Methods of moments for learning stochastic languages: Unified presentation and empirical comparison. In *International Conference on Machine Learning*, 2014.

[12] B. Balle and O.-A. Maillard. Spectral learning from a single trajectory under finite-state policies. In *International Conference on Machine Learning*, 2017.

[13] B. Balle and M. Mohri. Spectral learning of general weighted automata via constrained matrix completion. In *Advances in Neural Information Processing Systems*, pages 2159–2167, 2012.

[14] B. Balle and M. Mohri. Learning weighted automata. In *International Conference on Algebraic Informatics*, pages 1–21. Springer, 2015.

[15] B. Balle, P. Panangaden, and D. Precup. A canonical form for weighted automata and applications to approximate minimization. In *Logic in Computer Science (LICS), 2015 30th Annual ACM/IEEE Symposium on*, pages 701–712. IEEE, 2015.

[16] B. Balle, P. Panangaden, and D. Precup. Singular value automata and approximate minimization. *Mathematical Structures in Computer Science*, page 1–35, 2019.

[17] B. Balle, A. Quattoni, and X. Carreras. A spectral learning algorithm for finite state transducers. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011.

[18] F. Bergadano and S. Varricchio. Learning behaviors of automata from multiplicity and equivalence queries. In *International Conference on Algorithms and Complexity*, volume 778. Springer, 1994.

[19] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18(2):115–148, 1982.

[20] B. Boots, S. M. Siddiqi, and G. J. Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

[21] S. Bozapalidis and O. Louscou-Bozapalidou. The rank of a formal tree power series. *Theoretical Computer Science*, 27(1):211–215, 1983.

[22] N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In *Studies in Logic and the Foundations of Mathematics*, volume 26, pages 118–161. Elsevier, 1959.

[23] S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. Spectral learning of latent-variable PCFGs: Algorithms and sample complexity. *The Journal of Machine Learning Research*, 15(1):2399–2449, 2014.

[24] S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. H. Ungar. Experiments with spectral learning of latent-variable PCFGs. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 148–157, 2013.

[25] M. Collins and S. B. Cohen. Tensor decomposition for fast parsing with latent-variable PCFGs. In *Advances in Neural Information Processing Systems*, pages 2519–2527, 2012.

[26] M. Droste and H. Vogler. The Chomsky-Schützenberger theorem for quantitative context-free languages. *International Journal of Foundations of Computer Science*, 25(08):955–969, 2014.

[27] L. El Ghaoui. Inversion error, condition number, and approximate inverses of uncertain matrices. *Linear algebra and its applications*, 343:171–193, 2002.

[28] Z. Ésik and W. Kuich. Formal tree series. *BRICS Report Series (see also J. of Automata, Languages, and Combinatorics, 8(2):219–285, 2003)*, (21), 2002.

[29] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. *Journal of the ACM (JACM)*, 57(6):33, 2010.

[30] K. Etessami, A. Stewart, and M. Yannakakis. Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars. In *Symposium on Theory of Computing*, pages 579–588. ACM, 2012.

[31] K. Etessami, A. Stewart, and M. Yannakakis. Greatest fixed points of probabilistic min/max polynomial equations, and reachability for branching Markov decision processes. *Information and Computation*, 261:355–382, 2018.

[32] M. Fliess. Matrices de Hankel. *Journal de Mathématiques Pures et Appliquées*, 1974.

[33] Z. Fülöp and Z. Gazdag. Weighted languages recognizable by weighted tree automata. *Acta Cybernetica*, 23(3):867–886, 2018.

[34] Z. Fülöp and H. Vogler. Weighted tree automata and tree transducers. In *Handbook of Weighted Automata*, pages 313–403. Springer, 2009.

[35] T. Hsing and R. Eubank. *Theoretical foundations of functional data analysis, with an introduction to linear operators*. John Wiley & Sons, 2015.

[36] D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5), 2012.

[37] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Symposium on Theory of Computing*, pages 273–282. ACM, 1994.

[38] S. Kiefer, I. Marusic, and J. Worrell. *Minimisation of multiplicity tree automata*, pages 297–311. 2015.

[39] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[40] A. Kulesza, N. R. Rao, and S. Singh. Low-rank spectral learning. In *International Conference on Artificial Intelligence and Statistics*, pages 522–530, 2014.

[41] F. M. Luque, A. Quattoni, B. Balle, and X. Carreras. Spectral learning for non-deterministic dependency parsing. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2012.

[42] J. M. Ortega. *Numerical analysis: a second course*, volume 3. Siam, 1990.

[43] A. Parikh, L. Song, M. Ishteva, G. Teodoru, and E. Xing. A spectral algorithm for latent junction trees. In *Conference on Uncertainty in Artificial Intelligence*, 2012.

[44] A. P. Parikh, L. Song, and E. P. Xing. A spectral algorithm for latent tree graphical models. In *International Conference on Machine Learning*, pages 1065–1072, 2011.

[45] A. Quattoni, B. Balle, X. Carreras, and A. Globerson. Spectral regularization for max-margin sequence tagging. In *International Conference on Machine Learning*, 2014.

[46] G. Rabusseau, B. Balle, and S. B. Cohen. Low-Rank Approximation of Weighted Tree Automata. In *International Conference on Artificial Intelligence and Statistics*, pages 839–847, 2016.

[47] A. Recasens and A. Quattoni. Spectral learning of sequence taggers over continuous sequences. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2013.

[48] A. Salomaa and M. Soittola. *Automata-theoretic aspects of formal power series*. Springer-Verlag, 1978.

[49] S. M. Siddiqi, B. Boots, and G. Gordon. Reduced-rank hidden Markov models. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[50] L. Song, M. Ishteva, A. Parikh, E. Xing, and H. Park. Hierarchical tensor decomposition of latent tree graphical models. In *International Conference on Machine Learning*, 2013.

[51] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11), 1984.