Language Design and Renormalization

Ángel J. Gallego
Departament de Filologia Espanyola, Facultat de Filosofia i Lletres,
Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

Román Orús
Institute of Physics, Johannes Gutenberg University, 55099 Mainz, Germany

In this paper we consider some well-known facts in syntax from a physics perspective, which allows us to establish some remarkable equivalences. Specifically, we observe that the operation MERGE put forward by N. Chomsky in 1995 can be interpreted as a physical information coarse-graining. Thus, MERGE in linguistics entails information renormalization in physics, according to different time scales. We make this point mathematically formal in terms of language models, i.e., probability distributions over word sequences, widely used in natural language processing as well as other ambits. In this setting, MERGE corresponds to a 3-index probability tensor implementing a coarse-graining, akin to a probabilistic context-free grammar. The probability vectors of meaningful sentences are naturally given by stochastic tensor networks (TN) that are mostly loop-free, such as Tree Tensor Networks and Matrix Product States. These structures have short-ranged correlations in the syntactic distance by construction and, because of the peculiarities of human language, they are extremely efficient to manipulate computationally. We also propose how to obtain such language models from probability distributions of certain TN quantum states, which we show to be efficiently preparable by a quantum computer. Moreover, using tools from quantum information and entanglement theory, we use these quantum states to prove classical lower bounds on the perplexity of the probability distribution for a set of words in a sentence. Implications of these results are discussed in the ambits of theoretical and computational linguistics, artificial intelligence, programming languages, RNA and protein sequencing, quantum many-body systems, and beyond. Our work shows how many of the key linguistic ideas from the last century fit perfectly with know physical concepts linked to renormalization and how, as a consequence, many concepts in computational linguistics also match perfectly with well-known physical conceptions.

I. INTRODUCTION

Linguistics can be defined as "the scientific study of language, and its form, meaning, and context" [1]. The field itself is a broad science, sometimes even a philosophy, embracing interdisciplinary ideas from a wide variety of contexts: syntax, mathematics, computer science, neuroscience... all in all, there is no common agreement concerning why human language is as it is, or even about its basic defining properties. From the point of view of Artificial Intelligence (AI), for instance, one is worried about developing accurate algorithms for speech and text recognition/prediction [2]. Additionally, the generative approach led by Noam Chomsky tries to understand the linguistic capacity from a biological perspective, as part of human cognition. As Chomsky *et al.* observe [3], the point of departure is Descartes' observation that, among all animal species, only humans seem to have a language ability [4]. Work on comparative cognition has endorsed this insight: only humans appear to possess a mental grammar – an "I-language," where the "I" stands for *intensional, internal*, and *individual* – that allows us to create infinitely many meaningful expressions from a finite stock of discrete units [5, 6] Within the generative models, the Minimalist Program [7] tries to attribute the properties of human language to what Chomsky [8] calls the "third factor", namely "to language-independent principles of data processing, structural architecture, and computational efficiency" [8]. This picture is not different from the general study of organic systems, and D'Arcy Thompson's and Alan Turing's works on form and morphogenesis can be seen as an example [9]. In this framework, Chomsky proposed a basic operation, called MERGE, to build up linguistic structures [10]. In MERGE, two syntactic objects X and Y are combined to form a new syntactic unit K, i.e.,

$$MERGE : X, Y \longrightarrow K = \{X, Y\}, \tag{1}$$

where the brackets mean that the information in K is obtained from that in X and Y. The operation can be applied recursively, thus having the ability to create different generations of units.

In parallel to this, physics aims to understand how the universe behaves. Some of its subfields search for the fundamental mathematical laws of the building blocks of Nature, such as high-energy physics and quantum gravity. However, the knowledge of such fundamental rules (the so-called *reductionism*) does not imply a priori the knowledge of the emergent laws for aggregates of many fundamental entities (the so-called *emergentism* [11]). Typical examples of this are condensed matter and solid-state physics, where the knowledge of the rules governing the fundamental entities at a short length scale (such as atoms and molecules described by Schrödinger's equation) does not imply, at least directly, the knowledge of the rules governing the collective behavior of aggregates at a large length scale (such as phase diagrams of matter). In famous words of P. Anderson, "more is different" [12].

The key concept in the above discussion is that of *emergence*: the collective properties of aggregates of systems may be, because of different reasons, very different from the ones of the individual systems themselves. The mathematical formalization of this paradigm in physics is achieved by the so-called Renormalization Group (RG), or simply *renormalization* [13]. Originally developed (mainly) by K. Wilson and L. Kadanoff, renormalization is a strategy for dealing with problems at different physical scales. This scale is typically a length, energy or time scale, and allows for different descriptions of the problem as it changes. For instance, going from short to long length scales corresponds intuitively to "zooming out" the system, effectively going from, e.g., a description of individual atoms (short scale) to a description of a solid with $O(10^{23})$ interacting atoms (long scale). At its root, a renormalization transformation is built in two steps: first, one keeps the most relevant information degrees of freedom to describe the system at a new scale discarding the ones believed not to be relevant, and second, one implements a rescaling of the physical variable and operators/functions in order to maintain the original picture. Physics is full of successful applications of renormalization in different ambits, from condensed matter physics [14] to quantum field theory [15] and quantum information [16] (each one with its own peculiarities), to the point that it has become one of the basic pillars in our current understanding of the laws of Nature. Physical theories that cannot be renormalized, are considered wrong or incomplete.

Having said all this, our aim with this paper is twofold. On the one hand, we establish equivalences between physics and linguistics in the context of the Minimalist Program (for linguistics) and emergence (for physics) which, once mathematically formalized, turn out to have important consequences in ambits as diverse as AI, theoretical linguistics, computer science, RNA / protein sequencing, quantum many-body systems, and beyond. On the other hand, we strengthen the relation between physics and linguistics, where language is the system to be understood using the machinery of physics, and of information theory in particular.

Let us be more specific: here we observe that MERGE can be understood physically as a type of information coarse-graining according to time scales. Roughly speaking, the linguistic information in sequences of words (short time scale) gets renormalized by successive MERGE operations up to meaningful sentences (long time scale) [57]. This simple observation, somehow trivial from a physics' perspective, turns out to have deep consequences. In particular we show that *language models* (i.e., probability distributions over word sequences) [17], widely used in AI applications, admit a natural description in terms of Tensor Networks (TN) [18]. For instance, the simplest MERGE corresponds to a 3-index tensor of components $M_{\alpha\beta\gamma}$ accounting for a probability distribution of three variables α, β and γ. And this is nothing but a *Probabilistic (or Weighted) Context-Free Grammar* (PCFG), in a way to be made precise later. Probabilities of meaningful sentences with a given syntax tree are naturally given in this framework by *(mostly) loop-free TNs* which, on top, admit a correlated factorization when it comes to specific calculations. Such mathematical structures have a number of nice properties which make them particularly amenable to manipulations of their information content, as we shall explain. Moreover, the TN structure and the particularities of PCFGs allow for the description of the probability distributions in terms of some *TN quantum states*. Such an exotic description using quantum mechanics is only to be understood at the practical level, but it happens to provide a useful connection between computational linguistics and quantum information and computation, opening the door to unprecedented results and developments in language processing algorithms. As examples of this, we show how such states can be built efficiently on a quantum computer, and prove lower bounds on the *perplexity* of the probability distribution of a set of words in a sentence [58] by using mathematical tools borrowed from from the theory of quantum many-body entanglement [20]. We envisage important consequences of our results in machine learning and AI. For instance, one can use the full machinery of TNs and quantum information to validate, simulate, assess, and improve current language models. Moreover, the fact that such probabilistic models can be fed into a quantum computer means that we have, in fact, a quantum algorithm that allows perfect random sampling of language, which is impossible with classical computing. All in all, and together with other implications, we propose that our physical picture is in fact related to the conjectured "perfect design and economy" of language in Chomsky's Minimalist Program, as well as to the (also conjectured) efficient processing of linguistic information in the human brain [21].

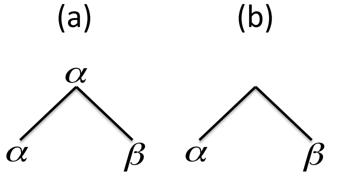

FIG. 1: MERGE operation, taking two lexical elements α and β, and projecting them into a new one, namely K, with label α. The fact that the label is also α means that the element resulting from the projection has the syntactic properties of α (the "head" of the syntactic object). (b) A label-free representation of the application of MERGE, compatible with the recent claim [19] that labels should be dispensed with.

The structure of this paper is as follows. In Sec.II we introduce our basic equivalence, namely, that MERGE in linguistics entails information-renormalization in physics.

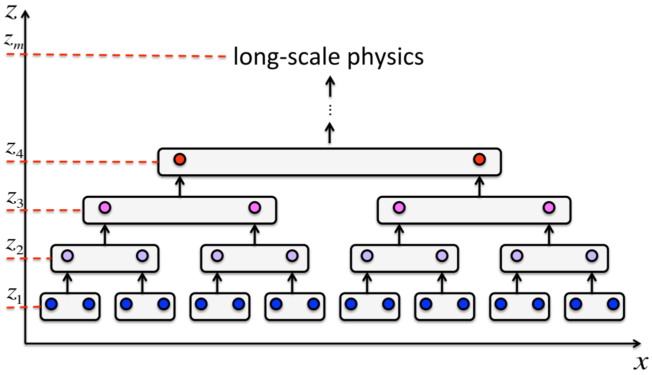

FIG. 2: (Color online) Pictorial representation of a renormalization process in real space for a $1d$ lattice. The horizontal axis is coordinate x (say, a space coordinate), and the vertical axis is coordinate z, which parametrizes the renormalization scale. Short renormalization scales (small z) amounts to a microscopic description of the system at small distances in x, whereas large renormalization scales (large z) amounts to a coarse-grained, macroscopic description of the relevant physics of the system at large distances in x. We codify short scales with "blue" and long scales with "red", following the intuition in physics that renormalization may take you from high energies (ultraviolet) to low energies (infrared). In our case, though, the colors have no special meaning and are just a convenient way of indicating the different scales $z_1, z_2, \ldots$, which are also shown for convenience. Formally, an RG step amounts to a coarse-graining followed by a rescaling of the lattice and associated operators/functions, which we implicitly assumed in the picture.

In Sec.III we explain a direct consequence of this: the quasi-loop-free TN structure of language models. We show how this applies to PCFGs and beyond, derive properties of such structures using tools from TNs, propose how to improve current probabilistic syntax-oriented language models, and establish the novel connection to TN quantum states. In Sec.IV we discuss some implications of our observations in different ambits. Finally, in Sec.V we wrap up our conclusions, include a table of the main equivalences discussed in the paper, and discuss future perspectives. We also include Appendix A with formalities for the readers with background on theoretical linguistics, and which allows us to find even more equivalences between linguistic and physical concepts, all of them linked to MERGE and renormalization. Overall, though, the paper is written assuming that the reader has mostly a physics & maths background, even though the style is highly heterogeneous, being this a consequence of the interdisciplinary nature of our results.

II. THE BASIC EQUIVALENCE

Our guiding principle is the following equivalence, which we write in the form of an equation:

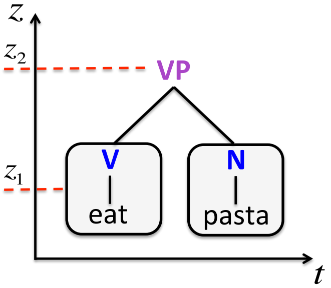

FIG. 3: (Color online) The linguistic MERGE operation, seen as a physical coarse-graining process. The horizontal axis is time t, and the vertical axis is the renormalization scale z. In this case, the operation takes a verb (eat) and a noun (pasta), and coarse-grains them into a verb phrase (eat pasta). At the scale z_2, all the relevant syntactic information is that the compound object is a verb phrase (VP). Unless stated otherwise, we assume that the basic building blocks are words together with their label, as shown in the grey boxes, though one could also interpret them more fundamentally as the result of a MERGE operation between a word and a set of lexical categories. For simplicity, we shall also assume here that no other information is carried by MERGE (such as genre, number, case, etc). In any case, this extra information can always be accounted for with minor trivial modifications of the scheme that we present here. The diagram provides the structure of *linguistic* correlations in the *physical* $\langle z, t \rangle$ plane.

$$\text{MERGE} = \text{Coarse-graining}$$

The left hand side of the above expression is a purely linguistic concept. MERGE is a basic operation in syntax, picking up a set of linguistic elements (such as lexical categories) and returning a new element describing the main features of the combination of the original, see Fig.1. On the other hand, the right hand side of the equation is a purely physical concept. A coarse-graining of information means the removal of superfluous degrees of freedom in order to describe a given physical system at a different scale (of, e.g., length, energy or time), see Fig.2. Combined with rescaling, it is the procedure entailing renormalization, by which the rules describing the macroscopic *emerge* from those describing the microscopic. The above equation establishes that both concepts are, in fact, the same basic idea but in different contexts. Chomsky's MERGE operation entails then the renormalization of linguistic information. Moreover, this renormalization accounts for different *time scales*.

To understand better why this is the case, see the example in Fig.3. In terms of information processing, the MERGE operation picks up two information units at a given time scale, namely

$$[_V \text{ eat}] \text{ and } [_N \text{ pasta}], \tag{2}$$

and keeps for the next time scale the most relevant information of their *coarse-grained* combination, i.e., that

$$[_{VP} \, [_V \text{ eat}] \, [_N \text{ pasta}]] \tag{3}$$

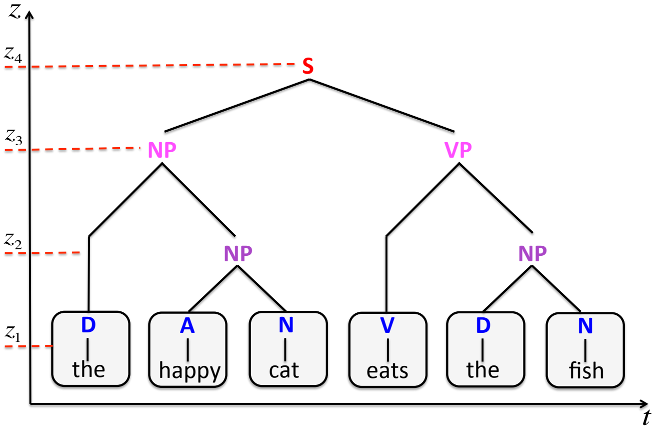

FIG. 4: (Color online) Syntax tree for "The happy cat eats the fish", seen as a renormalization process. The flow in z goes from the individual words, to the sentence, labeled by S. The different labels correspond to the different types of syntagmas (Noun Phrase, Verb Phrase, and so on). A rescaling of the time variable at every scale is also implicitly assumed.

is a verb phrase (i.e., lexical category VP), where we used bracketed phrase markers to represent the syntax tree. The operation MERGE is non-associative, as corresponds in general to a coarse-graining. Moreover, as linguists know very well, grammar rules for individual words are not the same as those governing more complex syntagmas such as noun and verb phrases. So, we have two different descriptions of a system at different scales, and with different linguistic information units. The physical variable with different scales must be time, since language is spoken and thought as time-ordered sequences of concepts, being written language just a graphical representation of this, see the more complex example in Fig.4.

This observation is ubiquitous in syntax and, when seen from the perspective of physics, entails the renormalization of linguistic information at different time scales. Consider for instance syntax trees (or parse trees, as known in computational linguistics) like the one in Fig.4. Such analysis are of the kind linguist use to describe how different elements (words) come together in order to produce a meaningful (active or passive) sentence, and have since long been widely used in the study of language. In practice, such syntax trees are nothing but the concatenation of several MERGE operations at different scales [59]: from words to other sintagmas, from these sintagmas to more complex sintagmas... and finally up to a sentence. According to our basic equivalence, the syntax tree that one obtains from such analysis is nothing but the graphical representation of the renormalization of the (linguistic) information of a sentence. This is, how the information in different words comes together, hyerarchically at different time scales, up to an emergent meaningful sentence that we can interpret semantically.

Moreover, the syntax tree also encodes the *structure of physical correlations at different time scales* in the sentence. More precisely, because of the local nature of MERGE, *correlations in a sentence are (essentially) built locally at different time scales.* Of course, it could be possible that other potentially-necessary operations in syntax, different from MERGE, introduce other dependencies (e.g., long-range movement). But still, it should be possible to codify them pictorically in the syntax tree. Worst-case scenario, such extra elements would introduce some loop in the tree. But even in such a case, the renormalization picture still holds, as we shall show in explicit examples.

Importantly, this observation is completely general, and therefore *must hold for any reasonable model of language following the Minimlalist Program.* In particular, theoretical linguistic models trying to account for the observed rules of grammar, as well as probabilistic language models in artificial intelligence accounting for the chances of finding a given sentence in a corpus, should somehow encompass the renormalization of linguistic information. This, in turn, has deep implications in the structure of correlations in syntax. As we shall see in the next section, a direct consequence of our observation is a natural description of probabilistic language models in terms of quasi-loop-free TNs [18] accounting for different time scales.

III. TENSOR NETWORKS AND PROBABILISTIC LANGUAGE MODELS

We now consider the implications of our basic equivalence for language models, i.e., probability distributions over sequences of words [17]. Such models produce probabilities $p_{w_1,\ldots,w_n}$ for a sequence of n words, represented by the random variables $w_1,\ldots,w_n$, and are widely used in several technological ambits such as speech recognition, machine translation, text prediction, and so forth. In practice, such probabilities are obtained by *training* the model (i.e., computing the frequencies of sequences) with very large corpuses of text. Here we focus on the general constraints that renormalization imposes on the structure of these probability distributions. As we shall see, a very natural description in terms of TNs just pops out, linking directly to Probabilistic Context-Free Grammars (PCFG), but not necessarily restricted to them only.

A. The MERGE tensor

To begin with, let us consider the probability distribution that two given linguistic elements α and β (e.g., two words) merge into a new element γ (e.g., some other syntagma). This probability distribution $M([\alpha,\beta]\to\gamma) = M(\alpha\cap\beta\cap\gamma)$ can in fact be described by a probability map M,

$$M : V_{in_1}\otimes V_{in_2} \longrightarrow V_{out}, \tag{4}$$

with V_{in_1}, V_{in_2} and V_{out} the input and output vector spaces. The coefficients of this map are given by a 3-index

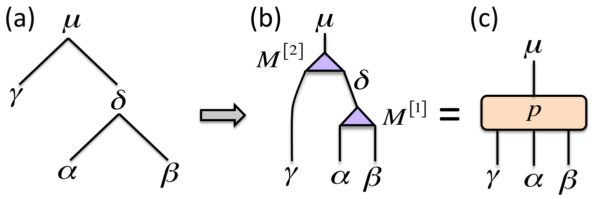

FIG. 5: (Color online) (a) Two concatenated MERGE operations, where we write different greek letters for all the possible lexical variables. For language models, this structure can be represented by the tensor network in (b), where $M^{[1]}$ and $M^{[2]}$ are two different MERGE probability tensors (see text). The contraction of the tensor network gives the probability tensor $p_{\mu\gamma\alpha\beta}$, see Eq.(7). In this picture we used a diagrammatic notation for tensors and their contractions, see text.

probability tensor $M_{\alpha\beta\gamma}$. The entries of this tensor are the probabilities of merging α and β (the linguistic input of MERGE) into γ (the linguistic output of MERGE). Physically, the tensor coarse-grains the variables α and β, at a given time scale, and retains the fundamental degrees of freedom of the common object at a different time scale. The result of this coarse-graining is variable γ.

The tensor $M_{\alpha\beta\gamma}$ obeys the usual normalization condition for probabilities,

$$\sum_{\alpha,\beta,\gamma} M_{\alpha\beta\gamma} = 1, \tag{5}$$

i.e., the sum of all the probabilities is equal to 1. One can also compute residual probability distributions in the usual way, i.e., by summing up over the variables that are discarded. For instance, one could have

$$M'_{\gamma} = \sum_{\alpha,\beta} M_{\alpha\beta\gamma}, \tag{6}$$

with M'_{γ} the residual probability distribution of obtaining γ as the output of MERGE, no matter the input.

From a linguistic point of view, the tensor $M_{\alpha\beta\gamma}$ is the implementation, at a mathematical level, of the MERGE operation for a probabilistic language model. If the same tensor is to be used everywhere in a syntactic structure, then this is nothing but the realization of a PCFG [22], i.e., a Context-Free Grammar with probabilities assigned to its merging rules. From the perspective of physics, though, this tensor coarse-grains degrees of freedom α and β at a given time scale into a new degree of freedom γ at a different time scale. From a mathematical perspective, this tensor describes the probability of obtaining the information codified in γ from the information in α and β. Regardless of the interpretation, this object constitutes the fundamental LEGO® brick of probabilistic language models following the Minimalist Program.

B. Syntactic tensor networks

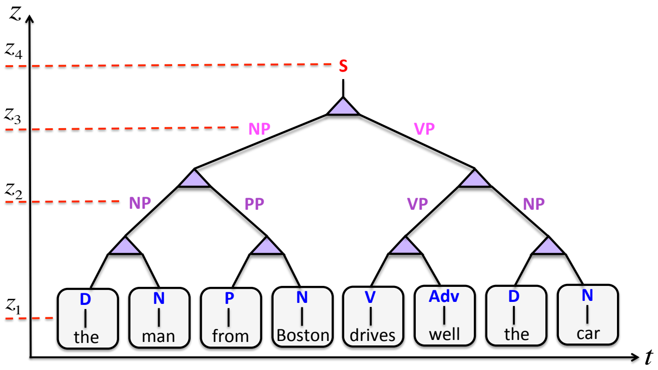

FIG. 6: (Color online) Syntactic TN for the sentence "The man from Boston drives well the car", where we included also the t and z axis, as well as the different renormalization scales. Linguistic information is naturally encoded in the TN at every possible scale. The contraction of the TN gives the probability of this sentence. In this particular example, the TN is a (binary) Tree Tensor Network.

Next, we notice that the structure of a syntax tree maps directly into a *tensor network* (TN) [18] for the probability distribution $p_{w_1,\ldots,w_n}$ of the sentence. Specifically, every syntactic MERGE$^{[i]}$ corresponds to a 3-index tensor $M^{[i]}_{\alpha\beta\gamma}$, with i simply a label to identify individual tensors, which could in principle be different. Now let us consider the case in which a variable μ is the result of merging δ and γ, with δ itself being the result of merging α and β. In such a case, following the usual mathematical treatment of probabilities, one has that the probability of obtaining μ from α, β and γ (i.e., no matter the value of δ) is given by the expression

$$p_{\mu\gamma\alpha\beta} = \sum_{\delta} M^{[2]}_{\mu\delta\gamma}\ M^{[1]}_{\delta\alpha\beta}, \tag{7}$$

i.e., we sum over all the possible intermediate events represented by δ. This admits a very intuitive diagrammatic representation, see Fig.5. In that figure, every tensor is a shape and every index is a line. Open indices, i.e., those over which there is no sum, are just "free" lines, whereas sums over all the possible values of a common index between tensors are represented by lines connecting the tensors. Such sums are called *contractions*, i.e., in this example we just contracted index δ. These type of structures, where one has a set of tensors whose indices are contracted according to some network pattern, are called *tensor networks* (TN) [18], and always admit a convenient diagrammatic representation as in Fig.5. With this in mind, we arrive to the conclusion that syntax trees of sentences map into TNs of MERGE tensors $M^{[i]}_{\alpha\beta\gamma}$ at the level of probabilistic language models. We call such structures *syntactic TNs*.

Let us be more precise: if the syntax tree does not have long-range dependencies (i.e., it is made only of MERGEs), then the TN is loop-free and corresponds generically to a Tree Tensor Network (TTN) [23], see

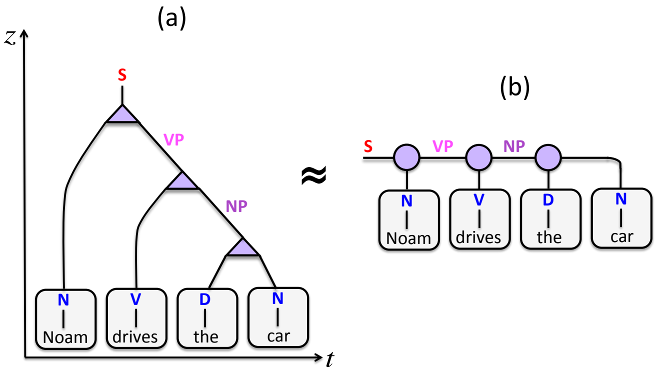

FIG. 7: (Color online) Syntactic TN for the sentence “Noam drives the car”. The Tree Tensor Network in (a) can be understood as a Matrix Product State, as shown in (b).

Fig.6. If the MERGEs are sequential in time, then the TN is in fact a special case of TTN called Matrix Product State (MPS), see Fig.7 [24]. These two types of structures appear quite often in the study of strongly correlated classical and quantum lattice systems in one spatial dimension [18, 23, 24] as well as in tensor calculus [25], and their properties are very well known by physicists and mathematicians. Moreover, if the syntax tree has some long-range dependency (e.g., movement, agree, c-command...), then this introduces some extra index in the network, correlating variables at different positions, and therefore introducing some loop in the diagram. To be precise, such extra index *correlates* the (perhaps distant) probability distributions for such variables, and can normally be casted into redefined tensors in order to keep the overall tree structure, as shown in the figure. As an example, this is in fact the case of the so-called CHAINS, which we mentioned in the introduction (Sec.I), and where a lexical object is intrinsically interpreted in different contexts of a sentence but only externalized in one of them, see Fig.8 for an example. More intricate cases, such as those involving a concatenation of chains (the so-called *successive cyclicity*), can also be accounted for similarly, see Fig.9 for an example. At any rate, though, the number of loops in the TN is always quite small, as long as the syntax tree is based on a Phrase-Structure (Constituency) Grammar [26], such as PCFGs. For the sake of clarity we restrict our explanation to these grammars. Other plausible situations, such as those arising in Dependency Grammars [27], will be briefly discussed in Sec.III F.

The syntactic structure of a sentence implies, therefore, that correlations in its probability distribution are orchestrated according to a (mostly loop-free) TN of MERGE tensors, which organize the degrees of freedom according to different time scales.

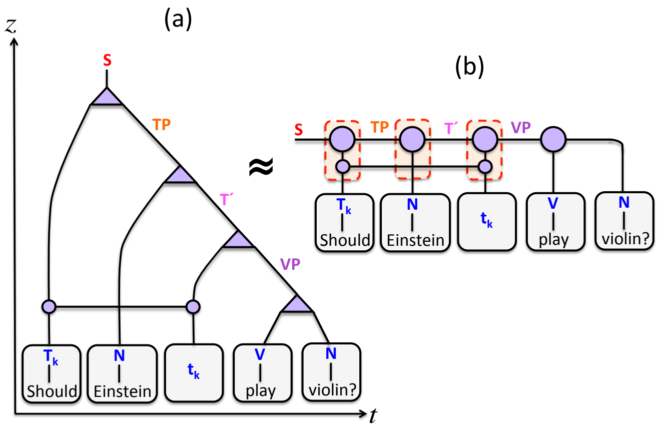

FIG. 8: (Color online) Syntactic TN for the sentence “Should Einstein play violin?”, as an example of syntactic movement. The element “Should” is created at the position of t_k but externalized at the position of T_k (hence it “moved”). At the level of the TN, this can easily be accounted for by an extra correlation between these two positions, i.e., an extra link between them (and perhaps two new tensors, as shown in the figure). This introduces a loop in the TN. However, as shown in (b), it is possible to redefine the overall structure as a loop-free TN with tensors as those shown in the dotted red boxes, and reshaped (or fused) tensor indices (i.e., whenever there are two indices together, fuse them into a single big index).

C. Properties

Let us now enumerate some important properties of the probability structures that we just found, which come out naturally from their TN description. Some of them were already mentioned briefly, but we revisit them again for clarity:

1. *Locally-built syntactic correlations at every scale*

Correlations in the probability distribution are built locally at every renormalization time scale by MERGE. Distant parts of the sentence become correlated at long time scales (i.e., up in the syntax tree), whereas those that are close become correlated at short time scales (i.e., down in the syntax tree). This locality implies a nice feature of loop-free syntax trees: for a sentence with n words, there are always exactly $n-1$ merged objects. Translated into syntactic TNs, this means that if the TN has n indices at the first renormalization scale z_1 (i.e., those corresponding to the words in the sentence), then there will be exactly $n-1$ indices on the whole at higher renormalization scales $z_m, m > 1$. This can be easily checked by inspection in all the figures with loop-free syntax trees and syntactic TNs of this paper. The consequence is that to specify the full syntax of a typical

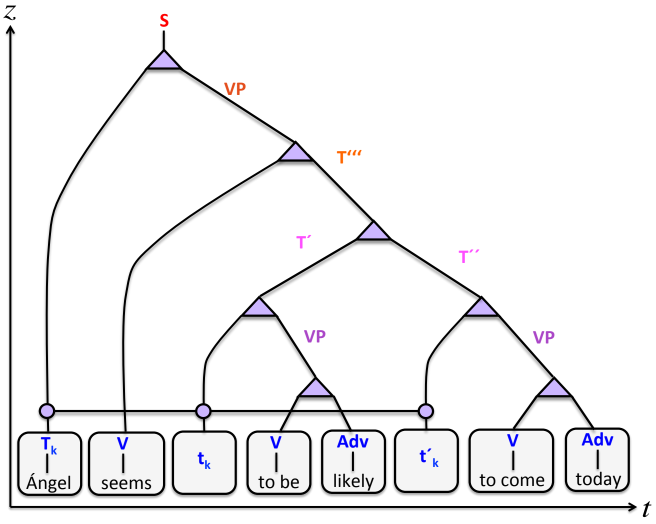

FIG. 9: (Color online) Syntactic TN for the sentence "Ángel seems to be likely to come today", as an example of concatenation of chains, or successive cyclicity. The syntactic information of the element "Ángel" is at different places leaving traces $t_k, t'_k, \ldots$, but externalized at only one position T_k. At the level of the TN this can be easily accounted for by an extra index correlating different sites, similarly as in Fig.8.

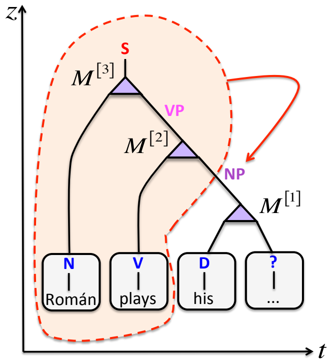

FIG. 10: (Color online) Syntactic TN for the sentence "Román plays his ...", where the last word is unspecified. The syntactic environment inside the dashed area forces the upper index of tensor $M^{[1]}$ to be NP. The first index of $M^{[1]}$ is forced to be the determiner "his". This constraints the probability of finding a given word at the last place of the sentence: whatever it is, it needs to merge with a determiner to become a noun phrase. There are not too many options: the word needs to be a noun. Notice that this is fully determined by the immediate neighbourhood in the sentence (the determiner), as well as the syntactic environment (the dashed region).

sentence of n words, one requires on the whole $2n-1$ units of syntactic information. In the case of having TNs with loops, as in the case of long-range movement in Fig.8 and Fig.9, the index creating the loop establishes a correlation between distant positions in the sentence, some of them with only syntactic information and no word present (the so called *traces*). In such cases it is clear that the number of required syntactic information units is larger than $2n-1$, though not much larger.

2. *Very efficient computations via correlated factorization*

The TN, when contracted from up towards down, reproduces the different probability distributions of the linguistic variables at every renormalization time scale. In other words, the TN encodes the probabilities of the relevant degrees of freedom *at all possible time scales*. Moreover, it is possible to obtain the residual probability of *any* of the variables just by contracting all the rest. Quite importantly, in syntactic TNs one does not even need to perform any tensor contraction since, once the sentence is fixed or partially fixed, there is a *correlated factorization* of the whole TN because of the way human language turns out to be, which we explain in what follows.

A well-known fact in grammar is that the output of a MERGE operation is always uniquely determined by its input. This is, given two objects being merged, there is only one possible output, no matter the context. This is a simple observation about how human language seems to work: the human brain does not merge an adjective A and a noun N into an object that sometimes behaves like a noun phrase NP, and sometimes like an adjectival phrase AP. Instead the combined object behaves *always* like a noun phrase NP. So, given the input of MERGE, its output becomes fixed uniquely [60].

This turns out to have an important consequence for us: it means that once the sentence is given, or partially given, then the TN factorizes in a correlated way. To see why this is so, notice that if the output of MERGE is always uniquely determined by its input, then all the indices in the syntactic TN *become fixed once the indices at the shortest time scale are fixed*, i.e., once a specific sentence is given. Because of this, the probability of a specific sentence actually factors out in terms of correlated probabilities and no TN contraction is needed at all. The overall correct syntactic structure of the sentence is the global, non-local property that correlates all the probabilities amongst themselves. Moreover, the residual probability of, say, finding a specific word in a sentence that is partially given, can be easily computed using one MERGE tensor only, which contains information about both the immediate neighborhood of the word, as well as about the overall syntactic neighborhood, see Fig.10. This is a very remarkable property that has its roots in the peculiarities of human language. In particular, it implies that the calculation of probabilities is *extremely* efficient, and that if the correct syntactic structure of a sentence is fully or partially known, then the *statistical perplexities* of reduced probability distributions are remarkably low, as we shall discuss in more detail in the forthcoming sections.

For a given sentence, therefore, the formalism produces a correlated structure of 3-index tensors linking all possible renormalization scales, see Fig.10. For example, the overall probability of, e.g., the 4-word sentence "Román plays his guitar" (an actual possibility in Fig.10) reads

$$p_{w_1^*,w_2^* w_3^*,w_4^*} = M^{[3]}_{w_1^*,VP,S} M^{[2]}_{w_2^*,NP,VP} M^{[1]}_{w_3^*,w_4^*,NP}, \qquad (8)$$

where $w_1^*, ..., w_4^*$ are the fixed words of the sentence, and no tensor contraction is needed at all. The above equation is a correlated product of coefficients from 3-index probability distributions, which encode *all* the syntactic information of the sentence at all time scales. The effect of this is more dramatic when it comes to residual probabilities: consider for instance predicting the word "drank" in the sentence "The man John met yesterday drank japanese whisky". A 3-gram model [28] (a rather common option in speech recognition) would give a probability distribution such as

$$p_{w_4^*,w_5^*,w_6} \quad 3-\text{gram model}, \qquad (9)$$

i.e., correlating the word w_6 only to "met" and "yesterday". The predictive power of this distribution is thus not very good, because there is no use whatsoever of the syntactic information from the rest of the sentence. However, in our TN description, the residual probability distribution, as shown in Fig.11, is given by

$$M^{[6]}_{w_6,NP,VP} \quad \text{Syntactic TN model}, \qquad (10)$$

which includes all the relevant syntactic information of the environment needed to predict w_6 in the sentence. In other words, having $[_{NP}$ $[_A$ japanese] $[_N$ whisky]], the rest of the sentence imposes that whatever goes in w_6 needs to combine together with this NP necessarily into a verb phrase VP. To put it simply, the marginal probability distribution is governed by this question: with whom do I merge to form what, as constrained by the rest of the sentence? In hindsight, this description includes all the relevant syntactic information required to predict the word exactly at that point.

From the above derivations, it is clear that all probabilities can be computed very efficiently, and exactly, from the TN. To be more specific, the fact that the structures are mostly loop-free implies that the calculation of probabilities, which amounts to the contraction of the tensors in the TN, can be done in polynomial time in the number of words, i.e., $O(\text{poly}(n))$ [18, 23, 24]. From the perspective of complexity theory, this is a consequence of the fact that the contraction of a loop-free TN is a problem in the complexity class P [29] [61]. But in the case of syntactic TNs like the ones described here, the situation is even better because of the correlated factorization explained above, which implies that no contraction of tensors needs to be done at all. The calculation of the probability of a given sentence amounts, simply, to determining the relevant syntax tree for a sentence and then multipliying the corresponding MERGE coefficients. For a sentence with n words, it is easy to see that both steps have a computational cost of $O(n)$, and therefore the overall cost is also $O(n)$. Therefore, the renormalization structure imposed by MERGE implies a *very* economical manipulation of the linguistic information in terms of computational resources such as time, memory, and energy.

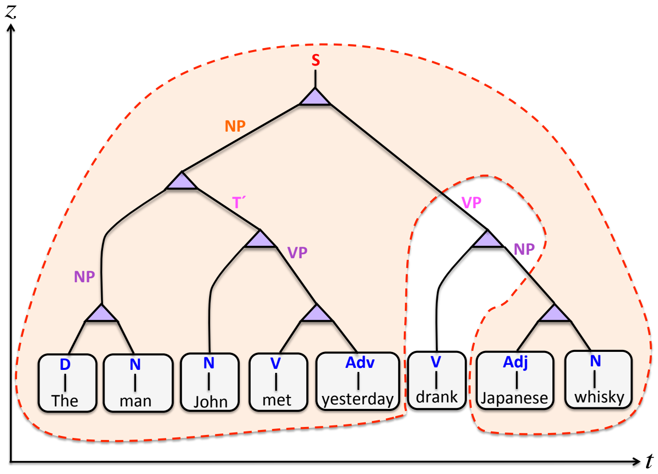

FIG. 11: (Color online) Syntactic TN for the sentence "The man John met yesterday drank Japanese whisky". The full syntactic environment of the word "drank" is highlighted in the dashed region, and determines the probability distribution of finding a specific lexical element at that place.

3. *Short syntactic correlations*

The two-point correlations in the probability distributions depend on the sentence (specifically, the syntax tree) and the renormalization time scale chosen to compute the correlations. This is also a well-known property of loop-free TNs, and in our case it means that the correlation between two words in the sentence decays exponentially fast with their *separation distance in the syntax tree*, which may be equal to the actual separation distance in the sentence or not.

Mathematically, this means the following: consider the two-point correlation function

$$C(i,j) \equiv \langle f(w_i) f'(w_j) \rangle - \langle f(w_i) \rangle \langle f'(w_j) \rangle, \qquad (11)$$

with

$$\begin{aligned} \langle f(w_i) f'(w_j) \rangle &= \sum_{w_1,\cdots,w_n} f(w_i) f'(w_j) \, p_{w_1,\ldots,w_n} \\ \langle f(w_i) \rangle &= \sum_{w_1,\cdots,w_n} f(w_i) \, p_{w_1,\ldots,w_n} \\ \langle f'(w_j) \rangle &= \sum_{w_1,\cdots,w_n} f'(w_j) \, p_{w_1,\ldots,w_n}, \end{aligned} \qquad (12)$$

and $f(w_i), f'(w_j)$ some functions of the variables w_i, w_j. We could think of these variables as those representing words at times i and j, but they could also be the variables for other (renormalized) syntagmas at a longer time scale (i.e., somewhere up in the tree). It is possible to prove mathematically [18, 23, 24] that this correlation function decays asymptotically as

$$C(i,j) \approx e^{-d(i,j)/\tau} \quad \text{for} \quad d(i,j) \gg \tau, \tag{13}$$

with $d(i,j)$ the size of the path between w_i and w_j in the syntax tree, and τ a sentence-dependent (finite) correlation time, see Fig.12 and Fig.13. As is well known from the theory of TNs, parameter τ does not depend on the choice of functions $f(w_i)$ and $f'(w_j)$, so it depends *only* on the type of sentence and the MERGE probabilities. This conclusion also holds if the TN has a small number of loops. Importantly, the quantity $d(i,j)$ can depend a lot on the type of syntax tree that one has. Consider for instance the two examples "Noam drives the car", and "The man from Boston drives well the car", with syntax trees as in Fig.12 and Fig.13. In the first case, Fig.12, the syntax tree is purely sequential, and therefore the TN for the probability distribution is a Matrix Product State [24]. In such a case it is clear that the distance $d(i,j)$ between two words is the actual separation distance in the sentence, i.e., $d(i,j) = |j-i|$. However, in the second case, Fig.13, the syntax tree is a binary tree, and therefore the corresponding TN is a Tree Tensor Network [23]. In such a case, the path along the tree between two words in the sentence *necessarily goes also along the vertical axis*, and one can prove that it is given by $d(i,j) \approx \log_2 |j-i|$, again with $|j-i|$ the separation in the sentence, and where $\approx$ means that it is correct up to some possible additive constant term [62]. Therefore, in cases such as the one in Fig.12 ("Noam drives the car"), the correlation function between two words will behave like

$$C(i,j) \approx e^{-|j-i|/\tau} \quad \text{for} \quad |j-i| \gg \tau, \tag{14}$$

whereas in cases such as Fig.13 ("The man from Boston drives well the car") it will behave like

$$C(i,j) \approx e^{-(\log_2 |j-i|)/\tau} \approx \frac{1}{|j-i|^{1/\tau}} \quad \text{for} \quad |j-i| \gg \tau. \tag{15}$$

In both cases the correlation falls down towards zero with the separation distance $|j-i|$ in the sentence, but in the first case it decreases exponetially fast, whereas in the second it is polynomially fast, and therefore much slower than in the first case. Notice, however, that at the level of renormalized objects up in the syntax tree, the two situations are completely equivalent, see Fig.13. This means that the correlation functions for the second sentence (Fig.13), but at some longer time scales (i.e., at the level of renormalized syntagmas up in the tree), decay exactly in the same way as those in the first sentence (see Fig.12).

Three remarks are in order. First, in intermediate situations between those described by the two examples above we expect also an intermediate regime between the two limiting cases from Eq.(14) and Eq.(15), but always obeying Eq.(13) asymptotically. Second, notice that the correlation time τ measures roughly how fast these correlations decay: the shorter τ is, the faster they decay towards zero. And third, notice that Eqs.(13), (14) and (15) essentially imply that language, at least within this description, has always very short correlations *within the syntax tree*, which does not necessarily imply short correlations in the separation distance within a sentence, as shown in Eq.(15). Similar conclusions apply as well in the case of having a small number of loops in the network, e.g. in linguistic chains, or in situations such as

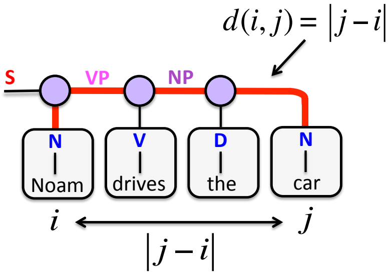

FIG. 12: (Color online) For a TN structure such as the one for "Noam drives the car", the syntactic distance $d(i,j)$ is the same as the time separation distance, i.e., $d(i,j) = |j-i|$. This is because the structure of correlations can be written as a Matrix Product State. Two-point correlation functions in this type of sentences decay exponentially fast in the time separation $|j-i|$, as explained in the text. The syntactic path between i and j is shown with a red thick line.

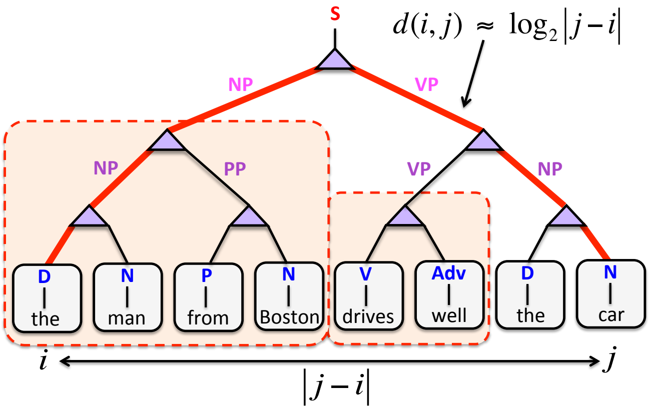

FIG. 13: (Color online) For a TN structure such as the one for "The man from Boston drives well the car", the syntactic distance $d(i,j)$ is *not* the time distance, but rather its logarithm, i.e., $d(i,j) \approx \log_2 |j-i|$. This is so because the syntactic path between positions i and j *goes also through the renormalization scale*. Consequently, there are two-point correlation functions for these types of sentences which can decay polynomially fast towards zero in the time separation $|j-i|$, hence much slower than in the case of Fig.12 (see text). The path between i and j is shown with a red thick line. Notice, however, that at the level of the renormalized syntagmas in the red dotted boxes, the structure is exactly the same as the one in Fig.12.

the German language, where a word correlated with the beginning of the sentence is actually sent to the end.

4. *Positivity*

By construction, the syntactic TNs presented here are such that all the tensors are non-negative, i.e., they are made entirely of non-negative coefficients. This is because of the stochastic nature of the MERGE tensor, which has been defined in terms of probabilities. These are sometimes called *stochastic tensor networks.* It is well known that such positivity restriction on the coefficients of the tensors is in fact very stringent [31] and usually implies a very large dimension for the vector spaces of the coarse-grained variables in the TN. We may therefore expect that a TN description of the overall probability distribution in terms of non-negative tensors lowers down this dimension, thus making the representation computationally more efficient. The price to pay, however, is that we loose the interpretation of the MERGE tensor as a tensor of probabilities. Still, a non-positive TN may be computationally more convenient in some situations.

D. Refinement levels in practice

The TN structure of MERGE tensors that we just described admits different levels of refinement, when it comes to determining the actual probability of a sentence in a given language model. A practical evaluation of such probabilities, once a parsed corpus (a *Penn TreeBank*) is given, proceeds as follows:

(i) First, one does a frequency count of all the words, and computes the probability of being some lexical category (N, V, etc) conditioned to being a certain word. This probability distribution corresponds, formally, to a MERGE operation at an initial time scale z_0 between a set of words and a set of lexical categories, as mentioned briefly in the caption of Fig.3. In practice, though, it can be accounted for by a 2-index probability matrix $\widetilde{M}_{\alpha\beta}$, with the first index referring to a particular word, and the second to its lexical category.

(ii) Second, one considers every sentence in the corpus and the respective syntax tree, and computes the probabilities corresponding to the coefficients of the MERGE tensors. This is done by counting the frequency of how many times two given lexical elements merge into a given object. Quite importantly, there are (at least) four different levels of refinement of the computed tensors, depending on their position in the $\langle z,t\rangle$ plane and the structure of the syntax tree. In increasing order of refinement, these are:

1. One single MERGE tensor M for all possible positions in the $\langle z,t\rangle$ plane.
2. One MERGE tensor $M^{[z]}$ for each possible renormalization scale, each one for all possible positions in t at the corresponding scale.
3. One MERGE tensor $M^{[z,t]}$ for each possible position in the $\langle z,t\rangle$ plane.
4. One MERGE tensor $M^{[T,z,t]}$ for each possible position in the $\langle z,t\rangle$ plane, and for each possible syntax tree T.

The more refined the information included in the computed MERGE tensors, the more accurate is the probability distribution, and therefore the better is the language model. The first of the refinement levels described above corresponds to the probabilistic language models provided by PCFGs [22]. These models are known to work reasonably in some circumstances, although on average not as good as, say, N-gram models [28]. But this is understandable, because one does not expect a priori the same MERGE tensor at all the possible positions in the $\langle z,t\rangle$ plane. Importantly there are still three more levels of refinement, which should account for better models. The second level drops the assumption of "ancestor-free" (akin "scale invariance" in physical jargon), so that the tensors may depend on the scale z. The third level drops, additionally, also the assumption of "place invariance" (akin "translation invariance" in physical jargon), so that the tensors may also depend on the variable t. Finally, the fourth level of refinement drops the assumption of the MERGE tensors being tree-independent. In principle, the four refinement levels are computable from a TreeBank, implying increasing level of precision for the language model. As for the computational cost of retrieving the MERGE tensors, in the first three levels it should be $O(M\bar{n})$ both for time and memory, with $\bar{n}$ the average number of words per sentence in a corpus containing M sentences. In the fourth case, however, the time cost is also the same but the memory cost may be larger since, for a large text, we expect to find roughly all possible syntax trees for every sentence length, which for n words is in turn given by the $(n-1)$th Catalan number,

$$C_{(n-1)} = \frac{(2(n-1))!}{n!(n-1)!} \approx \frac{4^n}{\sqrt{\pi}n^{3/2}}\left(1 + O\left(\frac{1}{n}\right)\right), \quad (16)$$

where the approximation is in the limit $n \gg 1$, and therefore scales exponentially. However, typical sentences in human language do not usually imply a dramatically-large number of words (we elaborate more on this in Sec.IV), and therefore the number of different syntax trees to be stored in memory may not be as large in practice as the above number.

Once the MERGE tensors have been computed from the TreeBank, the numerical probability for a sentence of n given words can be obtained in a two-step process:

(i) First, compute the possible syntax trees of the sentence (there may more than one valid tree in ambiguous cases).

(ii) Second, evaluate the probability for each tree following the correlated factorization procedure explained in the previous section, according to the four refinement levels mentioned above. The overall probability is the sum of probabilities for each valid syntax tree.

As the probabilities are computed, it is possible to calculate standard benchmark measures of language models, such as the so-called *perplexity* $\mathcal{P}$,

$$\mathcal{P} = 2^{H(p)} = 2^{-\sum_{\{w\}} p_{w_1,\cdots,w_n} \log_2 p_{w_1,\cdots,w_n}}, \tag{17}$$

with $H(p)$ the Shannon entropy of the probability distribution. The lower the perplexity, the more peaked is the distribution and thus the better it predicts the sample. So, the better the language model, the lower its perplexity, at least a priori. In our case we also expect the perplexity to decrease substantially as the refinement of the coefficients of the MERGE tensors increases, according to the four refinement levels mentioned above. Moreover, the perplexity also goes down with the precission of the probabilities in our MERGE tensors. We prove these points in the following section, using at some steps a novel reformulation of language models in terms of quantum states.

E. Language model quantum states

Let us now define the following quantum state:

$$|\Psi(T_n)\rangle = \frac{1}{Z(T_n)^{\frac{1}{2}}} \sum_{w_1,\ldots,w_n} \left(p_{w_1,\cdots,w_n}\right)^{\frac{1}{2}} |w_1,\ldots,w_n\rangle, \tag{18}$$

with $p_{w_1,\cdots,w_n}$ the probability of a sentence with words $w_1,\cdots,w_n$ and syntax tree T_n, and $\{|w_1,\ldots,w_n\rangle\}$ an orthonormal (tensor product) basis of some Hilbert space for n parties, each party corresponding to the position of a word in the sentence. The dividing normalization factor $Z(T_n)$ is actually the partition function of the probability distribution, i.e.,

$$\langle\Psi(T_n)|\Psi(T_n)\rangle = \frac{1}{Z(T_n)} \sum_{w_1,\ldots,w_n} p_{w_1,\cdots,w_n} = 1. \tag{19}$$

We call the state in Eq.(18) a *language model quantum state.*

Because of the correlated factorization of syntactic TNs explained in previous sections, one can see easily that these language model quantum states admit a TN representation of their coefficients, i.e., they are really *TN states* in the strict quantum-mechanical sense. The TN structure of the coefficient $(p_{w_1,\cdots,w_n})^{\frac{1}{2}}$ is simply given by the same one as for the probability distribution $p_{w_1,\cdots,w_n}$ (the syntactic TN), but replacing every coefficient of a MERGE tensor by its square root. More specifically, it is the same TN but with 3-index tensors $A^{[i]}$ of coefficients

$$A^{[i]}_{\alpha\beta\gamma} \equiv \left(M^{[i]}_{\alpha\beta\gamma}\right)^{\frac{1}{2}}, \tag{20}$$

again with i simply label for the different tensors. This simple prescription is a direct consequence of the correlated factorization of the syntactic TN. Notice also that these tensors obey the condition

$$\sum_{\alpha,\beta} A^{[i]}_{\alpha\beta\gamma} \left(A^{[i]}_{\alpha\beta\gamma'}\right)^* = \left(\sum_{\alpha,\beta} M^{[i]}_{\alpha\beta\gamma}\right) \delta_{\gamma\gamma'} = p^{[i]}_{\gamma}\delta_{\gamma\gamma'}, \tag{21}$$

with $p^{[i]}_\gamma$ the probability of merging at position i any two given lexical objects into γ, and $\delta_{\gamma\gamma'}$ the Kronecker delta, see Fig.14.

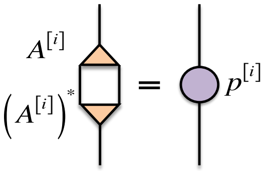

FIG. 14: (Color online) TN diagram for Eq.(21). The matrix on the right hand side is diagonal, and with entries $p^{[i]}_\gamma\delta_{\gamma\gamma'}$.

The language TN quantum state that we just defined is interesting for a number of reasons. Some of them are described in what follows.

1. *Truly random sampling*

First, notice that if this quantum state becomes (somehow) experimentally available in an actual quantum system, then it can be used to do *truly random sampling* of the probability distribution of sentences with that particular syntax tree. For comparison, all classical samplings are based on pseudo-random number generators, which are known to induce errors in the long run for, e.g., Monte Carlo methods. The state could also be useful, for instance, to find the most-likely sentences in a language model, and things alike.

2. *Language model quantum circuit*

Second, the state can, in fact, be created by a *quantum circuit* with as many two-body gates as A-tensors. The procedure is sketched in Fig.15: starting from the shortest renormalization scale z_1, one reshapes the indices of the A-tensors as a matrix and performs a QR decomposition [32], as shown in the figure. Since the A-tensors are real and positive, the matrix Q is orthogonal, i.e., $Q^TQ = \mathbb{I}$. Reshaping back Q as a 3-index tensor provides an isometric tensor, which we keep at the particular sites of the network at that renormalization scale. Matrices R, however, are contracted with the A-tensors at the next renormalization scale z_2, see Fig.15. The resulting tensors, call them B, are then also QR-decomposed, where

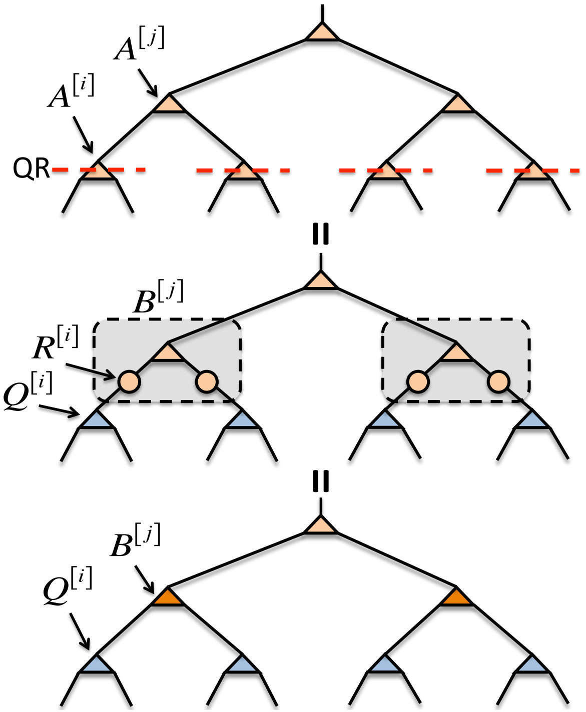

FIG. 15: (Color online) Iterative procedure to get the quantum circuit producing a language model quantum state for a given syntax tree (see text). The red dashed lines in the upper diagram correspond to QR decompositions. The process is iterated at every scale, until reaching the top.

the Qs define again isometries, which we keep in the network, and the Rs are contracted with the A-tensors at the next renormalization scale. By iterating this process up to the top level, one gets a TN of *isommetric* 3-index tensors $Q^{[i]}$, and a quantum state $|\Omega\rangle$ at the very top carrying non-local information about the probability of the whole sentence. In particular, since tensors $Q^{[i]}$ are isommetries, one has that

$$\langle\Psi(T_n)|\Psi(T_n)\rangle = \frac{1}{Z(T_n)}\langle\Omega|\Omega\rangle = 1, \tag{22}$$

(where the last equality follows from the normalization of the state), and therefore

$$\langle\Omega|\Omega\rangle = Z(T_n) = \sum_{w_1,\ldots,w_n} p_{w_1,\cdots,w_n}, \tag{23}$$

which means that the norm of the quantum state $|\Omega\rangle$ is the overall probability of having an n-word sentence (whichever) with syntax tree T_n in the language model. This global information just moved up to the top level of the TN and, importantly, we constructed it locally at every renormalization scale by a sequence of QR decompositions, therefore very efficiently (notice that we *never* needed to compute each one of the terms $p_{w_1,\cdots,w_n}$ individually!) [63]. Connecting to the usual developments in quantum-mechanical TN states, this is an example of an isometric TTN state [23]. Finally, in order to promote this structure to a quantum circuit, we simply notice that an isometric tensor can be understood as a two-body unitary gate, where one of the indices is fixed to some ancillary state $|0\rangle$ [16], see Fig.16. The resulting diagram is nothing but the picture of the quantum circuit producing the desired quantum state. The conclusion is that if the MERGE tensors are given, then one could in principle produce these quantum states efficiently in a quantum computer or a quantum simulator. Last but not least: the description above has been for TNs without loops, but it can be generalized to other situations. In case of having a small number of loops in the network (e.g. in CHAINS), there is also a similar procedure as the one indicated here by playing with several tensor decompositions (QR, Singular Value Decomposition, etc), always sending the non-unitary parts upwards in the syntactic network.

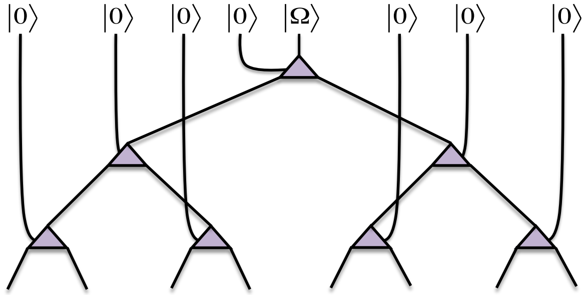

FIG. 16: (Color online) Quantum circuit of 2-body gates producing a language model quantum state for a given syntax tree. Ancillary degrees of freedom are fixed to the quantum state $|0\rangle$. The state $|\Omega\rangle$ at the top may be produced from $|0\rangle$ by some extra 1-body gate, and its squared norm codifies the overall probability of the tree.

3. Lower bounds on perplexity from entanglement

Our third property concerns the perplexity $\mathcal{P}$ of a language model, which was defined in Eq.(17). For a given sentence, it turns out that we can give lower bounds on the perplexity of a given subset of words, using tools from quantum information theory, as we show next.

Let us start by considering a sentence with n words, and a subset of $n' < n$ contiguous words within the sentence. These are a *block* of n' words. The question we want to answer now is: how much is the entanglement of this block of n' words in a given quantum state $|\Psi(T_n)\rangle$ for a syntax tree T_n? Following the usual procedure for bipartite entanglement, we get first the reduced density matrix of the block,

$$\rho(n') = \mathrm{tr}_{n-n'}|\Psi(T_n)\rangle\langle\Psi(T_n)|, \tag{24}$$

with $\mathrm{tr}_{n-n'}(\cdot)$ the partial trace over the rest of the system (the *environment*). As shown in the diagrams of Fig.17,

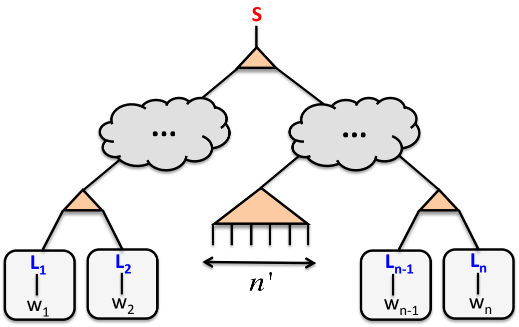

FIG. 17: (Color online) Subset of n' contiguous words in an arbitrary sentence, as described by the language quantum state in Eq.(18). The clouds indicate some arbitrary piece of an arbitrary syntactic tree.

this can be achieved by "cutting" out the relevant sub-tree linking the n' words from the rest of the sentence. After the appropriate contractions, this reduced density matrix can always be written as

$$\rho(n') = \frac{1}{Z(T_n)} W X W^\dagger \tag{25}$$

with W some rectangular matrix amounting for the contraction of the sub-tree for the block, and X a square matrix whose rank is the number of lexical categories N_l in our grammar, being this also the rank of $\rho(n')$, see Fig.18. It is easy to see, moreover, that in fact matrix X is diagonal,

$$X_{\alpha\alpha'} \propto p(n-n')_\alpha \delta_{\alpha\alpha'}, \tag{26}$$

with $p(n-n')_\alpha$ the overall probability of the string of $n-n'$ words merging into lexical category α, no matter the words in the string. One can also see that the (unnormalized) eigenvectors of $\rho(n')$ are given by

$$(v_\alpha)_\omega = \left(W^\dagger\right)_{w\alpha}, \tag{27}$$

with $(v_\alpha)_\omega$ the ω-coefficient of the αth eigenvector, and eigenvalues λ_α given by

$$\lambda_\alpha = p(n')_\alpha \; p(n-n')_\alpha, \tag{28}$$

with $p(n-n')_\alpha$ as described above, and similarly for $p(n')_\alpha$ but for the set of n' words, see Fig.19. Using Eq.(28), one can get the *entanglement entropy* $S(\rho(n'))$ and the *single-copy entanglement* $E_1(\rho(n'))$ of the block of n' words [33], which are given respectively by

$$\begin{aligned} S(\rho(n')) &= -\sum_\alpha \lambda_\alpha \log_2 \lambda_\alpha \\ E_1(\rho(n')) &= -\log_2\left(\max_\alpha \lambda_\alpha\right). \end{aligned} \tag{29}$$

The above entanglement measures obey the chain of inequalities

$$E_1(\rho(n')) \leq S(\rho(n')) \leq \log_2 N_l, \tag{30}$$

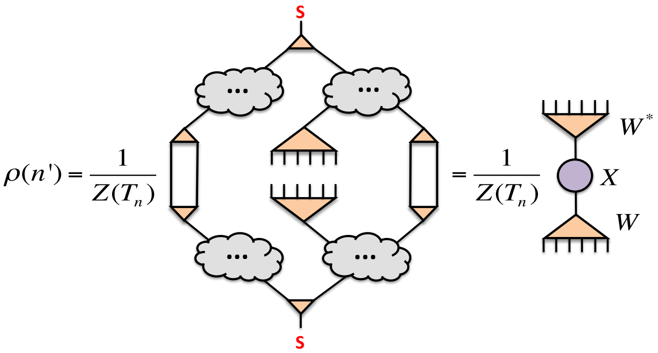

FIG. 18: (Color online) Reduced density matrix of a block of contiguous n' words in the language state of Eq.(18).

which implies that the entanglement of the block can never be too large, since the number of lexical categories N_l in a typical grammar for human language is usually quite small.

Next, we notice that the probability distribution p_ω for the n' words in the block is actually given by the diagonal elements of $\rho(n')$ in the basis of Eq.(18) restricted to the block, i.e.,

$$p_\omega = \rho(n')_{\omega\omega}. \tag{31}$$

One can check from the derivations above that this probability distribution and the one of the eigenvalues λ_α obey the majorization relation [34]

$$\vec{p} \prec \vec{\lambda}, \tag{32}$$

which implies

$$H(p_w) \geq S(\rho(n')), \tag{33}$$

i.e., the Shannon entropy of the reduced probability distribution for the block of n' words is larger than the entanglement entropy of the block. This relation, combined with Eq.(30), implies directly that

$$\mathcal{P} = 2^{H(p_w)} \geq 2^{S(\rho(n'))} \geq 2^{E_1(\rho(n'))}, \tag{34}$$

with $\mathcal{P}$ the perplexity of the distribution of the n' words. Combining this with Eq.(28) and Eq.(29), in the end we arrive to the following result:

$$\mathcal{P} \geq \min_\alpha \left(\frac{1}{p(n')_\alpha \; p(n-n')_\alpha} \right), \tag{35}$$

which is our main lower-bound for the perplexity of the probability distribution of the n' words.

Some remarks are in order. First, notice that Eq.(35) is a fully classical result, even though we used the machinery of quantum information theory to find it. Second, the inequality is giving us a fundamental lower bound on how well our language model can predict sentences, just because of its statistical nature. Third, we can roughly

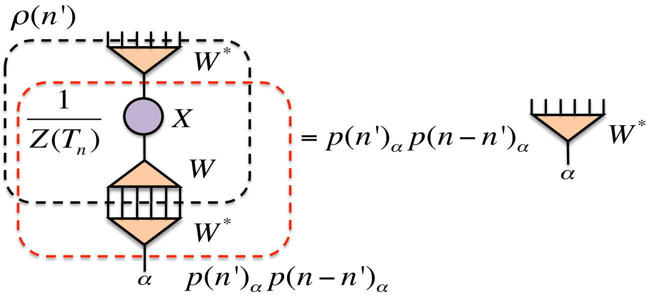

FIG. 19: (Color online) TN diagram for the eigenvalue equation of the reduced density matrix $\rho(n')$.

estimate the scaling of this lower bound: if $p_{\max}$ is the maximum merging probability over all MERGE tensors in the network, it is easy to see that

$$\mathcal{P} \gtrapprox \left(\frac{1}{p_{\max}}\right)^{n-1} \tag{36}$$

which implies, also roughly, that the perplexity gets worse (increases) exponentially fast with the number of words n in the sentence, but also that it improves (decreases) exponentially fast if the MERGE probabilities of the language model get more refined and accurate. This inequality shows clearly the route required in order to improve the performance of syntax-based probabilistic language models.

F. Arbitrary grammars and language models

We would like to conclude this section with a couple of words about other types of grammars, not necessarily context-free, as well as other language models. Importantly, the tensor network picture of language is not necessarily restricted to the cases that we presented above, and in fact can be used to describe the correlation structure of, essentially, any type of grammar and/or language model. For instance, the trees of *dependency grammars* [27], though not based on the MERGE operation, also admit a TN representation of their correlations when put as a probabilistic language model. We could even add long-range dependencies between the probability distributions in constituency grammars, as was shown for the case of chains in Fig.8, but which can in fact be generalized over the whole $\langle z, t\rangle$ plane, obtaining what is known in physics as a MERA-like tensor network [16], see Fig.20. As a matter of fact, it would be possible to model with TNs *any* grammatical correlation structure, even if not directly linked to human language. An example would be a syntactic structure based on an hypothetical MERGE operation with multiple outputs for a given input. Such structures would not have the property of "correlated factorization" discussed above, but most of the key properties that we mentioned would still hold, including those related to computational efficiency and short-range syntactic correlations.

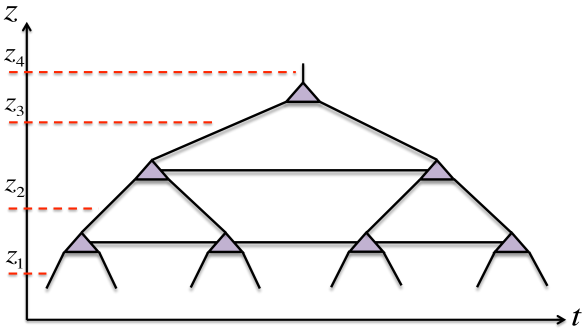

FIG. 20: (Color online) Possible MERA-like TN for some possible dependency grammar. Probability distributions (tensors) are correlated at every renormalization scale. The structure is no longer a tree if all possible dependencies are taken into account at every scale, as shown in the diagram.

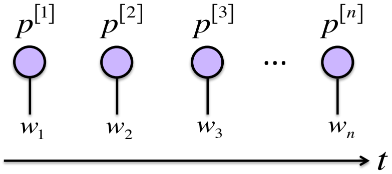

FIG. 21: (Color online) TN for a 1-gram language model. Only the time axis is relevant, and there is no correlation between the words $w_1, ..., w_n$. In physics, this is the analogue of the so-called mean-field theory approximation.

From a practical perspective, the so-called *N-gram models* [28], where the probability of observing a word is assumed to depend only on the history of the preceding $N-1$ words, also admit a similar description. For instance the case of 1-grams corresponds to the product probability distribution

$$p_{w_1,\ldots,w_n} = p^{[1]}_{w_1} \cdots p^{[n]}_{w_n}, \tag{37}$$

which can be represented by the TN diagram of Fig.21. Such a 1-gram TN does not include *any* correlation between the words. For comparison, similar separable TNs are also the ones used in the so-called *mean-field* approximation to strongly correlated systems, where correlations between different sites are discarded [35], and which is known to fail whenever correlations are important. For the case of more complicated N-grams, one can actually define an appropriate language model quantum state, i.e.,

$$|\Psi(N-\mathrm{gram})\rangle = \frac{1}{Z^{\frac{1}{2}}} \sum_{\alpha \in \mathrm{N-gram}} (p_\alpha)^{\frac{1}{2}} |\alpha\rangle, \tag{38}$$

with α an index running over all possible N-grams, p_α their probabilities, $|\alpha\rangle$ a set of orthonormal states, one for every N-gram (which is rather easy to construct), and Z the partition function of the distribution. Once such a state is available, one can do similar things as for the

TN language models discussed previously, such as truly random sampling, and so forth.

IV. IMPLICATIONS

Our "renormalization picture" of syntax and the results presented above demand for a necessary and detailed discussion about its implications, which extend into different ambits. In what follows we elaborate on some of them, taking a somehow more phylosophical perspective than in the previous sections, though well-grounded in our rigorous observations so far.

A. Legitimacy of language models

The first practical implication, as we have already hinted in the previous sections, is that legitimate language models (of any kind) should be compatible with the coarse-graining picture that we presented. From a generic perspective, one should expect a language model to reproduce the way humans seem to organize correlations in sentences, and from our perspective, this is given by the organization of coarse-grained information at different time scales. Concerning the field of artificial intelligence, we thus believe that a good starting point to obtain better language-processing algorithms, is to include also this organization of linguistic information according to time scales. This is in fact partially achieved already by the so-called "syntactic language models" [36]. The same applies to theoretical models of language in theoretical linguistics [64]. Notice, importantly, that in this work we never hypothesized about what is the fundamental theory of grammar behind the known properties of MERGE. Questions such as "why a noun and an adjective merge into a noun phrase?", or "why is the output of MERGE uniquely determined by its input?", are beyond the scope of this work. In other words: we observed how correlations in language get organized, explained this organization using the tools of physics, and exploited the consequences. And that is everything we did. We never discussed where these correlations could come from, or why they are as they are. In any case, and this is the point that we wish to make here, models attempting to explain this, either computational or theoretical, should encompass the picture presented here to be legitimate, since our observations are general.

B. Universality of language

Given the renormalization structure and the properties of TN language models, one can predict universal quantities, i.e., numbers that should be the same, no matter the language, and which only depend on the correlation structure of syntax [65]. For instance, consider the "correlation time" τ discussed in the previous section. As a matter of fact, this property implies that language, as a system, seems to have "little influence at long *syntactic* separations".

Indeed, for a given lexicon it should be possible to find these correlation times τ "experimentally" by analizing the different probability distributions for every type of sentence. An average over all plausible sentences should give a number characterizing each language. We conjecture that this number is universal, i.e., it is essentially the same for all languages. Moreover, we also conjecture it to be rather small, given the typical short-range syntactic influence of linguistic dependencies. This observation is clear in the case of language models, which is what we elaborated in detail in the previous section, but we expect it to be valid in general, since it is a property inherent to the renormalization picture and quasi-loop-free correlation structures.

We wish to remark that universal properties of language, and in particular short correlation distances, had already been observed by analyzing linguistic information with the tools of *complex networks* [38]. This is the field of physics and mathematics that analyzes complex systems and their structure from the network perspective (examples are ubiquitous: the internet, the power grid of a country, the synaptic network in the brain...). In this setting, the so-called *linguistic networks* allow for a study of the properties of syntax from a pure network-theory perspective. In order to avoid confusion, we stress that our approach here is radically different, since we start from a very different physical perspective: renormalization, and how this orchestrates correlations. This led to a TN picture of language models, and results on short correlation distances, which are different but complement those obtained using complex-network theory.

C. Optimality of language

Several of the properties from the previous section seem to be related to the conjectured "perfection and economy" of human language in the Minimalist Program, as well as to the conjectured efficient processing of linguistic information in the brain [7, 21]. Let us take for concreteness the language models that we analyzed before. The fact that the TN structures are mostly loop-free automatically implies that the retrieval of information can be done efficiently in all computational resources (a problem in the complexity class P). Such computational efficiency strongly depends on the quasi-loop-free renormalization structure of syntax trees, and is therefore generically valid, i.e., not just for the case of language models. In fact, loop-free structures are well-known to be the *cheapest non-trivial* class of correlation structures in terms of the manipulation of their information [23]. The surprising fact, is that human language is *even more efficient than this*, because of the properties of MERGE.

In particular, we saw that the uniqueness of the output of MERGE once the input is specified, implied a correlated factorization in the TN, which leads to a dramatic efficiency in the calculation of probabilities for TN language models. It looks, therefore, that the human language chose the cheapest possible option able to keep non-trivial correlations between information units. Our brain could have evolved to use a MERGE where the output is non-unique for a given input and still maintain a big part of the computational efficiency in the manipulation of information, but this just didn't happen. This observation makes precise the common-lore statement that language is, indeed, the cheapest non-trivial computational system. This may be one of the reasons why our brains choose to work with such correlation structures, instead of a different one. And we manage to externalize it through a physiological interface pretty well: we communicate (on average and most of the time) via sequential sounds in time produced with one mouth, instead of producing correlated sounds with, say, each one of our fingers, which would amount to 20 mutually correlated outputs, and thus a syntax full of correlation loops in turn implying computational inefficiency in the processing of its information.

D. Non-Markovian memory environment

A coarse-graining is a process that finds effective degrees of freedom to describe an emergent object, and inherently involves an *information loss* when moving from one scale to the next. It is well known in physics that renormalization is, usually, irreversible (the so-called "irreversibility of RG flows") [39]. In language, however, it is clear that even if syntax manipulates coarse-grained objects at some long-time scale, we still *know* about the information content of the short-time scales. This is, our brain seems to organize the information according to different time scales, but does not seem to fully *erase* the information when going from one scale to the next, at least for some period of time. For instance, when we say a sequence of the type $[_{NP}\ [_{A}\ X]\ [_{N}\ Y]]$ (an adjective X followed by a noun Y), we remember for a while what it actually refers to: "happy cat", "hot meal", "interesting paper", and so on. This seems to indicate that the "discarded" information seems not to be immediately erased, but just put apart for a while in some memory degree of freedom. To put it in physical jargon, one would say that the "memory environment" is *non-Markovian*, in the sense that there seems to be access for some period of time to the discarded information, shall this be needed. Understanding how and why this happens is indeed a relevant but different question to the one that we addressed in this paper.

E. Context-free grammars in other ambits

An interesting observation is that probabilistic context-free grammars (PCFG), though originally developed in linguistics, have proven recently very powerful in the probabilistic modelling of RNA and protein structures. In particular, PCFGs offer a way of determining the secondary structure of RNA, with a comparable accuracy to that obtained by energy minimization methods [40]. Concerning proteins, the situation is more complex but several achievements have already been reported using PCFG methods [41]. Many of the things that we mentioned previously in this work for the case of language, therefore, apply as well to the study of RNA and protein sequences. Even if being a very different scenario, the relevant correlation structures that appear in these biological problems happen to be similar to the ones that we described in this work, and therefore the same derivations could be applied to study those. The same is also true for the correlation structures present in *programming languages*, such as C++, Java, and so on. From a theoretical perspective, programming languages actually apply the rules of some grammar, i.e., rules by which words in a computer code are interpreted into meaningful machine instructions.

Intriguingly, one can also make a turnaround in the derivation that we presented here, and consider some TN structures as the natural correlation output of grammars. To be more precise, one could argue that TTNs and MPS can, in general, always be regarded as the output of some set of "generalized" context-free grammar rules where one allows for several possible outputs of a MERGE operation for a given input, being the outputs associated to complex "weigths". As such, this then implies that ground states of gapped $1d$ local quantum many-body Hamiltonians are, roughly speaking, nothing but generalized grammatical structures. Whether this simple observation has consequences in the (analytical and numerical) study of quantum many-body systems remains as a provocative open question.

F. On typical human abilities

Intriguingly, similar structures to the ones presented here for the case of language and grammar have also been found in different but somehow related scenarios. For instance, it was recently noticed that the correlation structure of neural network algorithms (which mimic in part the behavior of neurons in the brain) is, in fact, that of a Tree Tensor Network [42]. Renormalization-like algorithms are also common in the study of image compression, such as those based on wavelets [43], and even on Matrix Product States [44], where information of a picture gets organized according to different $2d$ length scales. Matrix Product States have also been used in the context of machine learning [45]. Moreover, it has been argued that the harmonic structure of tonal music may be, in fact, also a result of the MERGE syntactic operation [46]. As a matter of fact, it is believed that the faculty of language appeared in evolution almost simultaneously to the faculties of mathematics and music, with some people arguing in favour of the three faculties being actually three different manifestations of the same basic ability, which became available to our ancestors due to some genetic mutation throughout evolution [47]. A very subtle, somehow missed point, but key in this regard, is that the mathematical faculty looks itself *also* as a coarse-graining of (mathematical) information. This is in fact a consequence of MERGE being the successor function in mathematics [48]. In order to make this point more explicit, let us directly cite a rather popular paragraph (at least in the linguistics' community) in one of the recent works of N. Chomsky [49]:

> *"Suppose that a language has the simplest possible lexicon: just one lexical item, call it "one". Application of MERGE to the lexical item yields {one}, call it "two". Application of MERGE to {one} yields {one, {one}}, call it "three". And so on. In efect, MERGE applied in this manner yields the successor function. It is straightforward to define addition in terms of MERGE(X, Y), and in familiar ways, the rest of arithmetic. The emergence of the arithmetical capacity has been puzzling ever since Alfred Russell Wallace, the co-founder of modern evolutionary theory, observed that the "gigantic development of the mathematical capacity is wholly unexplained by the theory of natural selection, and must be due to some altogether distinct cause", if only because it remained unused. It may, then, have been a side product of some other evolved capacity (not Wallace's conclusion), and it has often been speculated that it may be abstracted from the faculty of language by reducing the latter to its bare minimum. Reduction to a single-membered lexicon is a simple way to yield this consequence."*

Moreover, and at an experimental level, neuroscientists have recently discovered what could be the signature of the MERGE operation in neural activity, by analizing the neural activation of epileptic patients performing several language tasks [50].

Given all this, we take the liberty to take off and hypothesize, somehow phylosophically and because everything seems to point in this direction, that the human abilities of language, mathematics, and probably others, may actually be different manifestations of a fundamental single ability of the human brain, namely, *the ability to organize and process information according to different physical scales*. To put it simple: one could say that the human brain looks like some kind of biological information-renormalization machine. When it comes to human language, this allows the brain to build a *language system of discrete infinity*, i.e., a discrete and recursive system able to produce infinitely-many outputs.

V. CONCLUSIONS AND PERSPECTIVES

The observations and results in this paper are highly interdisciplinary. Let us briefly summarize here the main points. We have argued that the linguistic MERGE operation entails renormalization in physics: the information content in, e.g., sequences of words (short time scale) gets renormalized by MERGEs up to sentences (long time scale). We have made this observation concrete for language models, and have found that probabilities of meaningful sentences are naturally given by quasi-loop-free TNs, which in turn organize correlations according to different renormalization time scales. Such language models are naturally related to probabilistic context-free grammars, though not restricted only to them. We have discussed some of the properties of these TN language models: locally-built syntactic correlations at every scale, very high efficiency of information processing because of correlated factorization of the TN, short-range syntactic correlations, and practical refinement levels. We also proposed how to promote probabilistic language models to probability distributions of quantum states, argued that such quantum states may be useful when it comes to sampling the distribution, showed how they can be built efficiently in a quantum computer, and used their entanglement properties to provide a classical lower bound on the statistical perplexity of finding a set of words in a sentence. We discussed also how this useful formalism may be generalized to other types of grammars, and discussed a number of implications of our observations in several ambits. These concern the legitimacy of language models, universality and optimality of language, some required properties of the memory environment, the potential application of our formalism for RNA and protein sequencing as well as programming languages and quantum many-body systems via context-free grammars, and the overall picture of several human faculties all somehow boiling down to MERGE. In the end, we have taken the liberty to hypothesize that the human brain seems to have a natural fundamental ability to organize information according to different physical scales, from which other faculties may materialize.

Our work opens the possibility to use all the mathematical and physical knowledge about TN states, both classical and quantum, in the theoretical and computational study of language and grammar. This includes a wide variety of applications not just in linguistics, but also in RNA and protein sequencing [40, 41] and the design of computer languages, just to name some well-known examples. In particular, the different ways to quantify correlations and the information content in the network, as well as associated numerical algorithms [18], should find useful applications in these scenarios. Moreover, the efficient descriptions of probability distributions of relevant grammars by means of quantum states, opens the exciting possibility to use possible quantum computers and quantum simulators to deal with problems in all these ambits. A prominent example is AI, where our results show that quantum information tools can be used to to validate, simulate, assess, and improve state-of-the-art language models, as well as that quantum computers can be used to implement perfect random sampling of language, which is impossible with classical technology. This is particularly relevant given the recent big advances in the development of experimental quantum processors.

By digging deeper into linguistic concepts it is indeed possible to take our equivalences further. We do this in Appendix A. All in all, our conjecture that MERGE in linguistics is connected to RG in physics turns our to be extremely fruitful, since many of the key linguistic ideas from the last century fit perfectly with know physical concepts linked to renormalization. We have also seen that, as a consequence, many concepts in computational linguistics also match perfectly with well-known physical conceptions. The main equivalences discussed in this paper, including those in the appendix, are summarized in Table I.

Linguistics	Physics
MERGE	Coarse-graining
Relabelling	Rescaling
Derivation	RG flow
Phase	RG scale
Phase impenetrability	RG irreversibility
Optimality and efficiency	Loop-free structures
Prob. language model	1d tensor network
N-gram models	Mean-field theory
Prob. context-free grammar	3-index tensor & MPS/TTN
Dependency grammar	$(k > 3)$-index tensor & 1d MERA
$\sqrt{\text{Prob. language model}}$	Quantum circuit
Perplexity	Quantum entanglement

TABLE I: Main equivalences and connections between linguistics and physics proposed in this paper. The upper part corresponds to concepts usually discussed in theoretical linguistics, and the lower part to concepts in computational linguistics. 1d means that the "physical" degrees of freedom span along one dimension, which in the case of language is time. The "square-root" symbol in the lower-left panel is a way of saying that the corresponding quantum circuit produces probability amplitudes that are the square root of the actual probabilities given by the language model.

Only good things can happen by studying language from the perspective of physics [51]. The fields of physics and linguistics have been traditionally very far away from each other. But indeed, linguistics focuses on the study of the laws of language, and physics on the study of the laws of Nature. For a linguist, the human language is the universe, and it has deep connections with how our brain processes and manipulates information, as well as other situations whose correlations are orchestrated by grammar-like rules. From the perspective of physics, it feels just natural to think that classical and quantum information theories should be somehow useful for this purpose. Being able to formalize mathematically some of the most relevant aspects of language and grammar in terms of physical ideas is already an important achievement. We strongly believe that the cross-fertilization of physics and linguitics will become increasingly relevant in the future. Phylosophical questions, such as those encountered sometimes in linguistics, usually lead to deep, profound scientific problems, and our work here is no exception to this rule.

Acknowledgments

We acknowledge the Universitat Autònoma de Barcelona, the Max Planck Instutute of Quantum Optics, and the University of Mainz, where the ideas in this paper materialized from interdisciplinary discussions between the two authors over the past seven years. Discussions with Ondiz Aizpuru, Gemma de las Cuevas, Laura García-Álvarez, José I. Latorre, Enrique Solano, Miles Stoudenmire and Juan Uriagereka are acknowledged. A special acknowledgement to Sergi Quintana, for explaining us the basics of Chomsky's generative grammar 24 years ago.

Appendix A: More equivalences by digging deeper

Our paper is written having in mind a reader with background on physics and mathematics. However, the topic itself is strongly interdisciplinary. Because of this, in this appendix we would like to add some extra information useful for the reader with knowledge about theoretical linguistics. In particular, we would like to define a few concepts more precisely in linguistic jargon. Thanks to this, we will see that by digging deeper into the linguistic jargon, more equivalences with physics will show up, in turn strengthening our thesis that MERGE in linguistics and RG in physics are deeply linked to each other.

To begin with, the term *Universal Grammar* (UG) is nothing but a label for the striking difference in cognitive capacity between "us and them", i.e., humans versus the rest of animal species. UG is thus the research topic of generative grammar in its attempt to understand what it is and how it evolved in our species. Finding a satisfying answer to the latter question may be impossible with the tools we have right now, but any theory of UG seeking to address the former must meet a criterion of evolvability: any properties, mechanisms, etc. attributed to UG should have emerged in what appears to have been a unique and relatively sudden event on the evolutionary timescale [52]. This line of thought presupposes that UG (the genetic encoding of the the human linguistic capacity) manifests *bona fide* traits of perfect design, in the sense that contains operations and mechanisms that follow from conceptual necessities, efficiency principles or interface demands. In this respect, linguistic expressions (sentences, phrases, words, etc.) are built up by adhering to these principles, therefore in an optimal fashion. While these notions are intuitively clear, their precise formulations remain vague and controversial.

One of the most important mathematical achievements of generative grammar is the so-called "Chomsky Hierarchy" [53], a classification of formal grammars according to their complexity. As Chomsky showed sixty years ago, human languages manifests both context-free and context-sensitive properties, needed to construct PHRASES and CHAINS respectively, shown in the Sentences A1(a,b):

a. John killed John. (A1)

b. John was killed $<$ John $>$

In Sen.A1(a) (a PHRASE) we have two tokens of the lexical item "John" that participate in phrasal dependencies to yield a compositional interpretation whereby the first John is the agent of a killing event, and the second John is the patient of such event. What we have in Sen.A1(b) (a CHAIN) is more complex. This time, we don't have two tokens of "John", but two occurrences of the same lexical item – as if they were one and the same object in two positions at the same time, where the notation $<$ John $>$ means that the word itself is not pronounced at that position, but it is also interpreted there from the logical point of view. This is what is called CHAIN in linguistics. In languages of the English type, the first (leftmost) occurence is spelled-out, whereas the second (rightmost) is necessary to keep a syntax-semantics homomorphism (that is, to capture the desideratum that a specific interpretation is tied to a specific position). Notice that the same type of object (a CHAIN) is necessary in Sen.A2, where "John" is pronounced to the left of *seem*, although it is interpreted as the patient of *killed.*

John seems to have been killed $<$ John $>$ (A2)

In order to account for these properties, generative grammar has resorted to phrase structure rules (PSR) and transformations. The most articulated version of PSR is known as X-bar Theory, which resorted to different devices that have been subject to a revision within minimalism. In particular, Chomsky [10] argued that the basic properties of PSR could be understood by means of a computational operation, dubbed MERGE [10], which captures two empirical properties of human language that are non-negotiable: *discrete infinity* and *displacement.* To be able to account for those properties, one must assume an operation that constructs hierarchically structured expressions with displacement. And that is what MERGE does. MERGE applies to two objects X and Y (be these words or bigger units), yielding a new one, K, which is the set containing X and Y, i.e., $\{X, Y\}$. If X, Y are distinct (taken directly from the *lexicon* or independently assembled), K is constructed by what is called EXTERNAL MERGE (EM); if Y is part of X (if Y is contained in X), then we have what is called INTERNAL MERGE (IM). The latter scenario is that of Sentences A1(b) and A2 above, where MERGE turns "John" into a discontinuous object (a CHAIN). For completeness, if the operation is at the beginning of a derivation (e.g., with bare lexical items from a lexicon), it is called FIRST MERGE, and if it operates with partially-derived items (phrases), it is called ELSEWHERE MERGE.

Chomsky [10] takes MERGE to be strictly binary, as it is what is minimally necessary to create hierarchical structure. Generation by MERGE thus entails a restrictive class of recursively defined, binary-branching and discrete-hierarchical structures.

It is also worth mentioning that in X-bar Theory, the label identifies the properties of the entire phrase, at the cost of this being a theory-internal symbol that departs from inclusiveness demands. An alternative to this is a label-free representation (see Fig.1), where endocentricity (the assumption that all phrases must be headed) is not preserved. This entails that syntactic objects can be exocentric, as seems to be necessary for objects formed by the combination of two phrases, $\{XP, YP\}$. Syntactic objects are "endocentric" if they contain an element that can be determined by Minimal Search – typically, a head. Given this logic, $\{X, YP\}$ is endocentric and $\{XP, YP\}$ exocentric. Consequently, such a system freely generates objects of different kinds, without stipulating their endocentric nature.

Moreover, MERGE is subject to efficiency and economy conditions. One such condition is inclusiveness, which precludes the introduction of extraneous objects, like the ones that X-bar Theory deployed: traces, bar-levels, projections, etc. Inclusiveness also bars introduction of features that are not present in lexical items.

To further clarify MERGE, we stress that the combination of two objects, X and Y, yields a new one, K, which is the set $\{X, Y\}$. Once we have $\{X, Y\}$, we may want to merge K and some object W, which can be either internal to K or external to it (see above). In any event, the merger of W cannot change or tamper with $\{X, Y\}$, which behaves as a unit. More precisely, subsequent applications of MERGE must yield Eq.A3(a), not A3(b):

$$a.\ \mathrm{MERGE}(K, W) = \{\{X, Y\}, W\} \qquad \text{(A3)}$$

$$b.\ \mathrm{MERGE}(K, W) = \{\{X, W\}, Y\}$$

The driving force of this work is the fact that MERGE and renormalization seem to play a similar role on various respects. As noted above, MERGE takes two objects, X and Y, to yield a new one, K, thus removing X and Y from the computational workspace (WS). In the simplest scenario, MERGE maps WS $= [X, Y]$ onto WS$' = [\{X, Y\}]$, reducing the complexity of WS. Notice that MERGE never extends the WS, at least in terms of cardinality; thus WS $= [\{X, Y\}]$ and WS$' = [\{W, \{X, Y\}\}]$ are equally bigger, since they only contain one set. A new element can be added to WS (or WS$'$) in only one way: by taking two items W, Z from the lexicon and introducing $\{W, Z\}$ into WS as a new element, yielding WS$'' = [\{W, Z\}, \{X, Y\}]$. Of course, cardinality can be reduced if we apply EM (EXTERNAL MERGE) and neither of the elements are taken from the lexicon, as if we map WS$'' = [\{W, Z\}, \{X, Y\}]$ onto WS$''' = [\{\{W, Z\}, \{X, Y\}\}]$. This idea is indeed very similar to that of a coarse-graining in physics, in the sense made precise throughout the paper.

Additionally, the possibility that computational load is reduced by MERGE is perhaps somewhat new, as this typically follows from a principle in linguistics that is called STRICT CYCLICITY. The notion of cycle (and thus cyclicity) goes back to the fifties, where work in phonology [55] showed that the application of stress-assigning rules apply from innermost to outermost units of a word, putting aside linear order information. More generally, an object is build under cyclic principles if it is COMPOSITIONAL, which means that its interpretation is fixed by the elements it contains and the way in which they are combined. Consider this with Sentences A4, where the interpretation is crucially different (Brutus is an agent in (a), and a patient in (b)), although both examples contain the same three words:

a. Brutus stabbed Caesar. (A4)
b. Caesar stabbed Brutus.

The concept of STRICT CYCLICITY is a stronger version of cyclicity. The key intuition behind it is that for certain linguistic object constructed in a derivation (say, a VP), further computation should not modify it. Let us see this with the example in Eq.A5, where the verb "leave" is merged with the NP "the room" to yield the complex VP "leave the room", which we can call K for ease of reference.

$$\text{MERGE}(\text{leave}, \{\text{the}, \text{room}\}) = \{\text{leave}, \{\text{the}, \text{room}\}\} \tag{A5}$$

What is of interest here is that the interpretation of K (that is, of "leave the room") is determined at that stage of the derivation (at that "cycle"), and cannot be changed at subsequent stages ("cycles"). Therefore, if we add "Mary" to obtain "Mary leaves the room" (call it K'), as in Eq.A6, the interpretation of K will be the same in Eq.A5 and in Eq.A6.

$$\text{MERGE}(\text{Mary}, K) = \{\text{Mary}, \{\text{leaves}, \{\text{the}, \text{room}\}\}\} \tag{A6}$$

In a nutshell, the interpretation of complex objects is constructed stepwise, in a step-by-step fashion, and whatever has been done at a stage s cannot be undone at statge $s + 1$ (Eqs.A5 and A6 above). This, in turn, is quite analogue to the idea of irreversibility of RG flows in physics, which matches perfectly with our interpretation of MERGE as a coarse-graining of information.

Such stages at a derivation, where a "computation" is done and cannot be altered afterwards, correspond with the so-called linguistic PHASES, and the device responsible for ensuring that the interior of a PHASE is no longer accessible is the PHASE IMPENETRABILITY CONDITION (PIC for short). What has been called phase roughly corresponds with the notion of cycle described above. Using the physical interpretation that we introduce in this paper, one would say that a PHASE in linguistics is the analogous of an RG scale in physics.

To be more precise, a phase is defined in linguistics as a domain D where uninterpretable features (number and person features of verbs) are valued. When a phase is closed off, the complement domain Ω (which can itself be complex, in the sense of having some inner structure) of the phase head P cannot be modified from the outside; this means, for instance, that the case of an NP within Ω (e.g., "the book" in the VP "read the book") cannot be changed once the phase headed by P is complete [56]. Among other things, this entails that "the book", which is the Direct Object of "read" Sentence A7 (it receives accusative case from "read"), cannot also be the Direct Object of the matrix verb "believe":

I believe that John read the book. (A7)

That "the book" is the Direct Object of "read" and not of "believe" is shown in Sentences A8, where we see that this NP can be passiviced in the embedded clause, but not in the matrix clause ($*$ signals ungrammaticality):

a. I believe that the book was read. (A8)
b. $*$The book was believed that John read.

This "shielding" effect that makes the VP impenetrable is captured by the phase impenetrability condition mentioned above. Physically, this is the irreversibility of the RG flow when moving from one RG scale to the next. There are various approaches to Phase Theory [54], but all of them share the key intuition that PHASES are domains where complexity is reduced by somehow allowing the system to "forget" about an amount of structure that has been created and which will be no longer accessible. This process of "forgetting" is, in fact, analogous to the process of "discarding irrelevant degrees of freedom" in an RG-step in physics.

Moreover, the "rescaling" step in RG has not been discussed in this paper, but also appears naturally when particularizing to specific models of language. For instance, in the Matrix Syntax model [37] this rescaling appears naturally in order to recover the correct linguistic labels after a MERGE operation (see Ref.[37] and the discussions therein for more information). We believe that this is a general feature: the "rescaling" in physics is nothing but the "mathematical relabelling" that one needs in order to recover the correct labels (NP, VP, etc) after a MERGE operation when dealing in practice with models of language.

[1] See, e.g., https://en.wikipedia.org/wiki/Linguistics

[2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (3rd ed.)*, Upper Saddle River, New Jersey: Prentice Hall (2009).

[3] N. Chomsky, Á. J. Gallego, and D. Ott, *Generative Grammar and the Faculty of Language: Insights, Questions, and Challenges*. Ms., MIT / UAB / UOttawa (2017). Available at http://ling.auf.net/lingbuzz/003507

[4] R. Descartes, *Discours de la mthode*, 1662.

[5] N. Chomsky, *The Language Capacity: Architecture and Evolution*. Psychonomic Bulletin and Review **24**: 200-203 (2017).

[6] M. D. Hauser, N. Chomsky and W. T. Fitch, *The Faculty of Language: What is It, Who Has It, and How Did It Evolve?*, Science **298**: 1569-1579 (2002); S. R. Anderson, *Doctor Dolittle's Delusion. Animals and the Uniqueness of Human Language*. New Haven, CT: Yale University Press (2004); N. Chomsky, *Some Simple Evo-devo Theses: How True Might They Be for Language?*, in R. K. Larson, V. Déprez, and H. Yamakido (eds.), *The Evolution of Human Language: Biolinguistic Perspectives*, 45-62. Cambridge: Cambridge University Press (2012).

[7] N. Chomsky, *A minimalist program for linguistic theory*, MIT occasional papers in linguistics no. 1. Cambridge, MA: Distributed by MIT Working Papers in Linguistics, 1993.

[8] N. Chomsky, *Three factors in language design*. Linguistic Inquiry **36**: 1-22 (2005).

[9] D. W. Thompson, *On Growth and Form*, Cambridge University Press (1917); A. M. Turing, Phylosophical Transactions of the Royal Society B, **237** (642): 37 - 42 (1952).

[10] N. Chomsky, *Bare Phrase Structure*, Evolution and Revolution in Linguistic Theory, Essays in honor of Carlos Otero., eds. Hector Campos and Paula Kempchinsky, 51109, 1995.

[11] See, e.g., https://en.wikipedia.org/wiki/Emergentism

[12] P. W. Anderson, Science, New Series, Vol. **177**, No. 4047, 393-396 (1972).

[13] There are plenty of books and introductory articles on renormalization in physics. Some good original sources, though, are L. Kadanoff, Physics **2**, 263 (1966); K. G. Wilson, Rev. Mod. Phys. **47**, 4, 773 (1975); K. G. Wilson, Sci. Am. **241**, 140-157 (1979); Also K. G. Wilson's nobel prize lecture from 1982, available at www.nobelprize.org.

[14] See, e.g., R. Shankar, Rev. Mod. Phys. **66**, 129 (1994); S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[15] See, e.g., S. Weinberg, *The Quantum Theory of Fields* (3 volumes), Cambridge University Press (1995).

[16] F. Verstraete *et al*, Phys. Rev. Lett. **94**, 140601 (2005); G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007);

[17] See, e.g., https://en.wikipedia.org/wiki/Language_model

[18] F. Verstraete, J. I. Cirac, and V. Murg, Adv. Phys. **57**,143 (2008); J. I. Cirac and F. Verstraete, J. Phys. A: Math. Theor. **42**, 504004 (2009); J. Eisert, Modeling and Simulation **3**, 520 (2013); N. Schuch, QIP, Lecture Notes of the 44th IFF Spring School (2013); R. Orús, Eur. Phys. J. B **87**, 280 (2014); R. Orús, Ann. Phys.-New York **349** 117158 (2014).

[19] N. Chomsky, *Problems of Projection*. Lingua **130**: 33-49 (2013).

[20] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, New York (2000).

[21] B. Sengupta and M. N. Stemmler, Proceedings of the IEEE, Vol. **102**, No. 5 (2014).

[22] N. A. Smith, M. Johnson, Computational Linguistics. **33** (4): 477 (2007).

[23] Y. Shi, L. Duan and G. Vidal, Phys. Rev. A **74**, 022320 (2006); L. Tagliacozzo, G. Evenbly and G. Vidal, Phys. Rev. B **80**, 235127 (2009); V. Murg *et al*, Phys. Rev. B **82**, 205105 (2010); M. Gerster *et al*, Phys. Rev. B **90**, 125154 (2014).

[24] M. Fannes, B. Nachtergaele, R. F. Werner, Commun. Math. Phys. **144**, 443-490 (1992); A. Klümper, A. Schadschneider, J. Zittartz, J. Phys. A **24**, L955 (1991); A. Klümper, A. Schadschneider, J. Zittartz, Europhys. Lett. **24**, 293 (1993); U. Schollwöck, Ann. Phys. **326**, 96 (2011).

[25] I. V. Oseledets, SIAM J. Sci. Comput., **33**(5), 22952317 (2011).

[26] N. Chomsky, *Syntactic structures*. The Hague/Paris: Mouton (1957).

[27] See, e.g., H. Liu, *Dependency Grammar: from Theory to Practice*. Beijing: Science Press (2009).

[28] See https://en.wikipedia.org/wiki/N-gram♯cite_note-1

[29] C. H. Papadimitrou, *Computational Complexity*, (Addison Wesley, 1994).

[30] N. Schuch *et al*, Phys. Rev. Lett. **98**, 140506 (2007).

[31] K. Temme, F. Verstraete, Phys. Rev. Lett. **104**, 210502 (2010); G. De las Cuevas *et al*, New J. Phys. **15**, 123021 (2013).

[32] See, e.g., https://en.wikipedia.org/wiki/QR_decomposition

[33] C. Holzhey, F. Larsen, and F. Wilczek, Nucl. Phys. B **424**, 443 (1994); G. Vidal *et al.*, Phys. Rev. Lett. **90**, 227902 (2003); J. I. Latorre, E. Rico, and G. Vidal, Quantum Inf. Comput. **4**, 48 (2004); J. Eisert and M. Cramer, Phys. Rev. A **72**, 042112 (2005); R. Orús *et al.*, Phys. Rev. A **73**, 060303(R) (2006).

[34] See, e.g., R. Bathia, *Matrix Analysis*, Springer-Verlag, New York, (1997); M. Nielsen and G. Vidal, QIC Vol. **1** No. 1, 76-93 (2001).

[35] L. P. Kadanoff, J. Stat. Phys. **137**: 777 (2009).

[36] G. Sidorov *et al*, *Syntactic Dependency-based n-grams as Classification Features*, LNAI 7630: 111 (2012).

[37] R. Orús, R. Martin and J. Uriagereka, arXiv:1710.00372; R. Martin, R. Orús and J. Uriagereka, to appear in the conference proceedings of *Generative Syntax: Questions, Corssroads and Challenges*, edited by UAB.

[38] R. Ferrer i Cancho and R. V. Solé, Proc. R. Soc. Lond. B **268**, 2261-2265 (2001).

[39] A. B. Zamolodchikov, JETP Lett. **43**: 730732 (1996); J. I. Latorre *et al*, Phys. Rev. A **71**, 034301 (2005); R. Orús, Phys. Rev. A **71**, 052327 (2005).

[40] S. R. Eddy and R. Durbin, Nucleic Acids Research. **22**

(11): 20792088 (1994); Y. Sakakibara *et al.*, Nucleic Acids Research. **22** (23): 51125120 (1994); R. Durbin, S. Eddy, A. Krogh and G. Mitchinson, eds, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge University Press (1998).

[41] D. Searls, Biopolymers **99** (3): 203217. (2013); A. Krogh *et al.*, J. Mol. Biol. **235**: 15011531 (1994); C. Sigrist *et al.*, Brief Bioinform. **3** (3): 265274 (2002); W. Dyrka and J.-C. Nebel, BMC Bioinformatics. **10**: 323 (2009); W. Dyrka, J.-C. Nebel and M. Kotulska, Algorithms for Molecular Biology. **8**: 31 (2013).

[42] Y. Levine *et al*, arXiv:1704.01552.

[43] A. Graps, IEEE Computational Science and Engineering, Volume 2, Issue 2, 50-61 (1995).

[44] J. I. Latorre, arxiv:quant-ph/0510031

[45] E. M. Stoudenmire and D. J. Schwab, Advances in Neural Information Processing Systems **29**, 4799 (2016).

[46] J. Katz, D. Pesetsky, http://ling.auf.net/lingBuzz/000959

[47] J. K. Alcock *et al*, Brain Lang. **75**, 3446 (2000); C. S. L. Lai *et al*, Nature **413**, 519523 (2001); I. Peretz, Psychol. Belg. **49**, 157175 (2009); K. Rimfeld *et al*, Scientific Reports **5**, 11713 (2015)

[48] See, e.g., https://en.wikipedia.org/wiki/Successor_function, and also Paul R. Halmos *Naive Set Theory*, Nostrand (1968).

[49] N. Chomsky, *On Phases*, MIT Press (2008).

[50] M. J. Nelson *et al.*, PNAS Vol. **114**, No. 18 (2017).

[51] For other examples in different contexts see, e.g., M. Piattelli-Palmarini, G. Vitiello, arXiv:1506.08663; M. Piattelli-Palmarini, G. Vitiello, Journal of Physics: Conf. Series **880** 012016 (2017); R. Solé, Phil. Trans. R. Soc. B **371**: 20150438 (2016); R. Orús, R. Martin, J. Uriagereka, *to appear soon.*

[52] J. Bolhuis, I. Tattersall, N. Chomsky, and R.C. Berwick, *How Could Language Have Evolved?* PLoS Biology **12**: e1001934 (2014); R. C. Berwick and N. Chomsky *Why Only Us*, Cambridge, MA: MIT Press (2016).

[53] N. Chomsky, *Three models for the description of language*, -IRE Transactions on Information Theory **2**: 113-124 (1956).

[54] Á. J. Gallego (ed.), *Phases. Developing the Framework.* Berlin: De Gruyter (2012).

[55] N. Chomsky, M. Halle, F. Lukoff, *On Accent and Juncture in English.* In *For Roman Jakobson: Essays on the occasion of his sixtieth birthday*, M. Halle et al. (eds.), 65-80. The Hague: Mouton and Co. (1956); N. Chomsky, M. Halle, *The Sound Pattern of English.* New York: Harper Row (1968).

[56] N. Chomsky, *Problems of Projection.* Lingua 130: 33-49 (2013); N. Chomsky, *Problems of Projection. Extensions.* In E. di Domenico et al. (eds.), In *Structures, Strategies and Beyond*, 1-16. Amsterdam: John Benjamins. (2015).

[57] A clarification is in order: here we understand "renormalization" as the tool that allows the mathematical description of the information in a system at different physical scales, accounted for by relevant degrees of freedom at every scale. Of course, the implementation of this idea in several contexts leads to different consequences. Well-known examples in physics are the existence of critical systems, critical exponents, universality classes, phase transitions, the c-theorem, the reshuffling of Hilbert spaces, the β-function, fixed points, RG flows, scaling laws, relevant / irrelevant / marginal perturbations... the list is unending. In our case, however, we do not assume necessarily the existence of any of these in the case of language (though some of them may also be there), and adopt instead the most fundamental and general perspective of what "renormalization" means at its very basic core at the level of information.

[58] To be defined in Eq.(17).

[59] Other operations can be accounted for by introducing extra links in the graphical representation, as we shall explain, but the renormalization picture still holds.

[60] Notice that the converse is not true.

[61] Unlike some TNs with loops, which are $\sharp$P-hard and therefore need an exponential time to be contracted [30].

[62] To be precise, this is correct in average, since depending on the tree it is possible to choose specific pairs of points with longer separation [23].

[63] This is in fact a very efficient procedure to compute the overall probability of a given tree in a language model.

[64] A recent attempt in this direction, also related to quantum physics and linear algebra, is the Matrix Syntax model in Ref.[37].

[65] In words of N. Chomsky, "there is only one human language" (private communication).