Coarse graining flow of spin foam intertwiners

Bianca Dittrich,[1,*] Erik Schnetter,[1,2,3,†] Cameron J. Seth,[4,‡] and Sebastian Steinhaus[5,§]

[1] *Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, Ontario N2L 2Y5, Canada*
[2] *Department of Physics, University of Guelph, 50 Stone Road East, Guelph, Ontario N1G 2W1, Canada*
[3] *Center for Computation and Technology, Louisiana State University, Baton Rouge, Louisiana 70803, USA*
[4] *Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*
[5] *II. Institute for Theoretical Physics, University of Hamburg, Luruper Chaussee 149, D-22761 Hamburg, Germany*

(Received 23 September 2016; published 29 December 2016)

Simplicity constraints play a crucial role in the construction of spin foam models, yet their effective behavior on larger scales is scarcely explored. In this article we introduce intertwiner and spin net models for the quantum group $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$, which implement the simplicity constraints analogous to four-dimensional Euclidean spin foam models, namely the Barrett-Crane (BC) and the Engle-Pereira-Rovelli-Livine/Freidel-Krasnov (EPRL/FK) model. These models are numerically coarse grained via tensor network renormalization, allowing us to trace the flow of simplicity constraints to larger scales. In order to perform these simulations we have substantially adapted tensor network algorithms, which we discuss in detail as they can be of use in other contexts. The BC and the EPRL/FK model behave very differently under coarse graining: While the unique BC intertwiner model is a fixed point and therefore constitutes a two-dimensional topological phase, BC spin net models flow away from the initial simplicity constraints and converge to several different topological phases. Most of these phases correspond to decoupling spin foam vertices; however we find also a new phase in which this is not the case, and in which a nontrivial version of the simplicity constraints holds. The coarse graining flow of the BC spin net models indicates furthermore that the transitions between these phases are not of second order. The EPRL/FK model by contrast reveals a far more intricate and complex dynamics. We observe an immediate flow away from the original simplicity constraints; however, with the truncation employed here, the models generically do not converge to a fixed point. The results show that the imposition of simplicity constraints can indeed lead to interesting and also very complex dynamics. Thus we need to further develop coarse graining tools to efficiently study the large scale behavior of spin foam models, in particular for the EPRL/FK model.

DOI: 10.1103/PhysRevD.94.124050

I. INTRODUCTION

Spin foams provide a nonperturbative and background independent path integral quantization for general relativity [1–3]. The construction of spin foam models involves an auxiliary discretization as a regulator for the path integral. A key outstanding task is the removal of this regulator, a process we refer to as continuum limit. One also needs to establish whether spin foam models can reproduce in this limit familiar low energy physics, in particular a geometric phase in which the models resemble a smooth manifold. Related is the question of whether diffeomorphism symmetry, which is deeply rooted in the dynamics of general relativity, can be restored [4–6].

There are two main paths to remove dependence of the auxiliary discretizations: a refinement limit of the underlying discretization, see e.g. [7], or summing over the discretizations [8–10]. We here consider the first approach, based on the refinement limit for the following reasons: a number of works [11–15] have shown that discretization independent models, which at the same time restore a notion of diffeomorphism invariance, can be constructed via a refinement limit—implemented in practice via a coarse graining flow. (For a review, and an explanation of the interplay between refinement and coarse graining, see [7,16,17].) Secondly we are in particular interested in the fate of diffeomorphism symmetry, which in the canonical framework is implemented via constraints. The spin foam path integral is supposed to provide a projector onto wave functions satisfying the corresponding quantized constraints [18,19]. However, summing over discretizations does in general not lead to a projector [20,21]. In contrast one can show that with the restoration of diffeomorphism symmetry in the discrete path integral one also obtains a projector, implementing the constraints, including an anomaly free constraint algebra [11,12,22,23].

[*] bdittrich@perimeterinstitute.ca
[†] eschnetter@perimeterinstitute.ca
[‡] cjmpseth@uwaterloo.ca
[§] sebastian.steinhaus@desy.de

The main challenge for the investigation of spin foam models is their overwhelming algebraic complexity. This comes together with an incomplete understanding of possible infinities (possibly related to diffeomorphism symmetry), a question on which there has been recent progress however [24–28]. Furthermore a framework has been developed that clarifies a number of conceptual questions in the context of background independent renormalization [7,15–17]. To condense this framework to what is important for the current work, the initial models, which are constructed via an auxiliary discretization, are subjected to a coarse graining flow. The models typically flow to an attractive fixed point, defining a phase of the model. Such phases correspond to topological models (with local amplitudes), which are triangulation invariant (and also restore diffeomorphism symmetry), but do not feature propagating degrees of freedom. Fine-tuning of some parameters in the initial models, which correspond e.g. to ambiguities in choosing the path integral measure, might allow one to find phase transitions. In particular, second order phase transitions are characterized by unstable fixed points, which in the background dependent context describe conformal theories. In the spin foam context, we are interested in the fact that these are fixed points, i.e. that the fixed point model is invariant under at least some subset of discretization changes, defined by the coarse graining flow. On such a fixed point we can construct a meaningful refinement limit, e.g. via an inductive limit as outlined in [7,16,17]. The corresponding model features nonlocal amplitudes, but its dynamics is accessible via a system of so-called dynamical embedding maps that allow one to extract the large scale dynamics in terms of coarse grained observables, which also capture the "most relevant degrees of freedom" [16]. This framework is particularly adapted to so-called tensor network renormalization (TNR) schemes. Such schemes implement real space renormalization, based on (a) identifying the most relevant (or contributing) degrees of freedom in the path integral using the dynamics of the system and (b) explicitly integrating out these (relevant) degrees of freedom [29–32]. In particular, (b) is opposed to Monte Carlo simulations, in which the integral is accessed via a sampling process. As spin foams are real time path integrals, that is, expected to have complex and highly oscillating amplitudes, this latter method is however (in general) not applicable to spin foams. This yields another motivation for the use of tensor network renormalization.

Coming back to the overwhelming algebraic complexity of the models, different lines of attack have been taken. First of all, spin foams can be understood as generalized lattice gauge theory models [33], allowing also a notion of these models for finite groups [34]. This latter approach inspired also so-called spin net models, which are the main focus of this work. In short, spin net models share the same dynamical ingredients as lattice gauge theories, namely group elements and weights, which are associated to lower dimensional objects, namely vertices and edges respectively (instead of edges and faces). Instead of a local gauge symmetry, spin nets have a global symmetry. The Ising model is a typical example for a $\mathbb{Z}_2$ spin net. Remarkably, two-dimensional spin net models of the same gauge group share statistical properties with the four-dimensional lattice gauge theory [35]. Hence we study spin net models as dimensionally reduced analogues for spin foams, which capture a key ingredient of spin foam dynamics, namely the so-called simplicity constraints. Furthermore spin nets are equivalent to so-called melon spin foams,[1] which are spin foams defined on a discretization involving only two (spin foam) vertices but an arbitrary number of edges connecting these vertices. The coarse graining collects a number of spin foam edges to new "thicker" edges. The hope is that the coarse graining of spin net models will allow us to study and understand the behavior of the simplicity constraints under coarse graining, and that a similar coarse graining flow could hold in spin foams. In fact this avenue allowed the investigation of more and more complicated models [14,36,37] via tensor network renormalization, with [14] studying spin nets based on the quantum group $\mathrm{SU}(2)_k$, and revealing a rich phase diagram for these models. Furthermore, tensor network renormalization has been also applied to three-dimensional spin foam models, so far based on finite groups, where the results confirm the phase diagram obtained for the corresponding spin nets [31,38].

The main aim of this work is to study spin nets with the full algebraic complexity of the full (Euclidean) spin foam models, in particular the Barrett Crane (BC) and the so-called Engle-Pereira-Rovelli-Livine/Freidel-Krasnov (EPRL/FK) models [39–43]. Implementing a (positive) cosmological constant at the same time provides a convenient cutoff on the summation range for the variables; we therefore need to consider spin nets with a structure group $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$. We see that this requires a range of techniques to allow for the numerical implementation of the tensor network coarse graining. We detail these techniques as we believe that these are also helpful in more general contexts.

As mentioned the tensor network coarse graining is based on explicit summation of the models. Furthermore, the space of models, in which the coarse graining flow can take place, is very large, in general given by all possible tensors of a predefined rank and index range. The coarse graining process proceeds iteratively; the effective amplitudes of each coarse graining step are encoded in a tensor (of very high dimension) which is updated in each step

[1] A melon spin foam consists only of two vertices, which are connected by many dual edges. From this melon one obtains a spin net by cutting through the dual edges, by mapping the projectors on the edges of the foam to the vertices of the spin net. The spin net vertices then have the same valency as the spin foam edges.

based on the previous tensor. This allows in principle to keep track of many observables of the models, but is of course also a challenge for the numerical implementation.

Most importantly this algorithm allows us to track the coarse graining flow of the simplicity constraints, which are crucial for the spin foam dynamics. The simplicity constraints determine which spin values are allowed and the various models do differ in these sets. Under coarse graining one expects that the allowed set of spins changes: this is due to the coupling of "finer" spins to "coarser" spins, which does not need to respect the simplicity constraints. To understand the large scale behavior of spin foams it is crucial to study how the simplicity constraints change under coarse graining.

The recent work [44,45] takes in some sense an opposite approach to the one taken here: one works with the full models (more precisely EPRL/FK using coherent Livine-Speziale intertwiners [46]), but implements a drastic, geometrically motivated, truncation that for instance suppresses all curvature degrees of freedom, but keeps some torsion degrees of freedom. The authors of [44,45] employ furthermore a saddle point approximation for the spin foam amplitudes, such that the focus is on large spins. (In contrast, using quantum group models here, we rather concentrate on small spins.) As curvature is suppressed the associated Regge action vanishes, such that Monte Carlo methods can be readily employed. These allow the approximate computation of expectation values for observables arising in one coarse graining step. Such an expectation value is then also used as a criterion to truncate the amplitude for the coarse building block back to the initial one-parameter family of models. This one parameter encodes a certain freedom in the choice of path integral measure. From this procedure one can deduce a coarse graining flow which tracks only the parameter describing the path integral measure. Thus compared to the tensor network method, where the flow is computed in a very high dimensional parameter space, here one truncates the flow to a one-parameter space. Despite these drastic truncations very interesting results were found: this (truncated) flow shows indications for a phase transition, at which a notion of residual diffeomorphism invariance is recovered.

Another approach relies even more heavily on analytical techniques [47].[2] The works [27,47] consider Pachner moves in a general triangulation, which makes it however difficult to come up with an iterative (regular) coarse graining scheme. Thus one can compute the amplitudes for a coarser complex; the details of the truncation scheme and a full implementation of the flow still need to be explored. These methods do however allow for a general understanding of the divergence structure of the models [27].

Let us also mention the older works [48] which studied the BC model with Monte Carlo simulations. Here one uses a property specific to the BC model, namely that it admits a representation in which the amplitudes are positive [49]. This does not hold for the EPRL model, prohibiting so far Monte Carlo simulations for the action contribution to the path integral. The work [48] considered the $\mathrm{SU}(2) \times \mathrm{SU}(2)$ BC model on a very simple four-dimensional triangulation, given by the 5-simplex. It also implemented a cutoff in the spins ($j = 5/2$ and $j = 25/2$). Three different choices for measure factors were tested: one led to a fast "divergence" of the model, i.e. a phase were large spins dominated (with respect to the cutoff). One phase led to a fast convergence and a partition function dominated by $j = 0$ spins. This motivated the introduction of a third choice, on the border between these two behaviors. Also a quantum group BC model has been considered in [50]. The most interesting point here is that the expectation values in the quantum group values do not converge to the classical group case, indicating a discontinuity.

In these works [48,50] one has measured e.g. the relative frequency of spin values in the probability distribution defined by the BC model. Although such observables give some insight—in this case testing the suitability of measure factors—we believe that we need a more systematic development of order parameters admitting a diffeomorphism invariant meaning. A main point of concern in [48] is divergences and so-called bubbles that are actually a sign for a restoration of diffeomorphism invariance.

The tensor network employed here has the advantage of testing the model iteratively over a large range of scales (defined by the number of coarse graining steps). Divergences can in principle be dealt with (although these do not arise here due to using quantum groups) by normalizing the partition function in each coarse graining step. Also a key point is the ability to track how the simplicity constraints behave under coarse graining, as this understanding is crucial for the understanding of the (effective) dynamics in spin foams models.

In agreement with results in [45,48], we find that the measure is a relevant factor which can drive phase transitions. This is also intuitively understandable: choosing a measure factor that suppresses larger spins drives the system to the Ashtekar-Lewandowski phase, in which all spins $j > 0$ are not allowed. The similar question concerning the (dual) BF phase, which is characterized by allowing all representations weighted by their respective dimension, can be answered negatively: For both BC and EPRL analogue models we do not observe a flow to a BF phase for any choice of measure discussed here.

This article is organized into two main parts. One focuses on the computational methods used in this work, while the other one addresses the construction and results of the

[2] So far particular (simplifying) features of the model studied in [47] seem to be important in order to allow for analytical treatment; see also [28].

quantum gravity related models. We have designed these parts such that they can be read independently of one another:

The first part is aimed at researchers outside quantum gravity also using tensor network techniques. While avoiding technical details of the models under discussion we focus on the scope of the problem and the implemented improvements to the algorithm. In Sec. II A we discuss the scope of the models we intend to coarse grain with tensor network techniques, which has motivated the improvements to the algorithm we present in Sec. II B.

The second part is aimed at people familiar with quantum gravity and spin foam models and can be read without reference to the computational/numerical details. In Sec. III A we briefly recall and motivate the general class of models under discussion, including their relation to lattice gauge theories and spin foam models. In Sec. III B we construct models analogous to modern four-dimensional spin foam models and discuss their behavior under coarse graining.

We conclude with a discussion of the methods and results in Sec. III E.

Several of the methods used in this article have already been developed and used in previous articles. Thus we do not introduce them in full detail, but concisely introduce their main features in the appendixes.

II. PART I: IMPLEMENTATION

A. Coarse graining of spin net models: numerical challenges

1. A brief introduction to tensor network renormalization

Before introducing the models under discussion in this work let us briefly touch upon the numerical algorithms used to coarse grain the models, which are broadly summarized under the term tensor network renormalization.

Before applying tensor network renormalization [29,30,32] the partition function of the model is rewritten as a contraction of a tensor network. A tensor network is a collection of multidimensional arrays, i.e. tensors, at the vertices of a lattice, where each tensor has as many indices as the vertex has legs. Then the tensors are contracted according to the combinatorics of the network, that is, a shared leg implies that the respective indices get contracted.

There exist different ways to obtain such a tensor network, but they are usually straightforward. In the cases we are considering the partition function is already of tensor network form. Crucially even though the tensor network might coincide with the lattice the underlying model is defined on, the network is independent as it merely represents a rewriting of the model. This is particularly beneficial in the context of background independent approaches to quantum gravity as tensor networks do not refer to a background structure. In most cases one studies systems on regular lattices resulting also in a regular network with identical tensors at all vertices. Thus the coarse graining process can be straightforwardly iterated.

The fundamental idea of tensor network renormalization is to locally manipulate the network, given by tensors T, such that the same partition function is (approximately) described by a coarser network of effective tensors T',

$$Z = \mathrm{Ttr} T \ldots T \approx \mathrm{Ttr} T' \ldots T', \tag{1}$$

where Ttr denotes the tensor trace, i.e. the contraction of the tensors according to the network. Hence one studies a flow of tensors $T \to T' \to \ldots$ capturing the dynamics of the system, which leads to the original name of tensor renormalization group (TRG) [29]. In this article we use a method closely related to the TRG. Nevertheless we point out that a more advanced algorithm has been invented by Evenbly and Vidal [32] named TNR, which filters out short-range entanglement resulting in a proper renormalization group flow. This method is also closely related to the multiscale entanglement renormalization ansatz [51] used to construct ground states of condensed matter systems.

More concretely let us discuss the algorithm introduced in [29] as an example,[3] as we modify it in the rest of the article. Consider a two-dimensional square tensor network of identical tensors. Let the indices of the tensor run from 1 to χ. This index range χ is frequently referred to as the (initial) bond dimension.[4]

The general scheme of the algorithm is illustrated in Fig. 1. To coarse grain this network each four-valent tensor is split first into two three-valent ones, that is, the four-valent tensor is written as a contraction of two three-valent tensors along a new edge. This new edge is the effective edge of the coarse grained tensor network. *A priori* a four-valent tensor can be split in many different ways, but as approximations during the numerical algorithm are necessary this splitting should allow for error control. Thus the tensor is rewritten into a matrix by a pairwise grouping of its indices according to the intended splitting. This matrix is then split into two by a singular value decomposition (SVD) as follows:

[3] In general many schemes to coarse grain tensor networks exist, e.g. one which more closely resembles block spin transformations: By contracting the edges connecting four tensors on the corner of a square one obtains a new coarse tensor. This step is exact, yet the new tensor has "double" edges with a bond dimension χ^2. Due to this exponential growth of data and to relate the new tensor to the original one, approximations are necessary, which are usually implemented via variable transformations and truncations, such that the error is minimized. Graphically this is shown as a three-valent tensor mapping the two edges into an effective one. We usually refer to these maps as embedding maps.

[4] As the models under discussion here are already of tensor network form, the tensor actually inherits the variables of the original model as labels on its edges. We refer to these spaces on the edges also as edge Hilbert spaces $\mathcal{H}_e$.

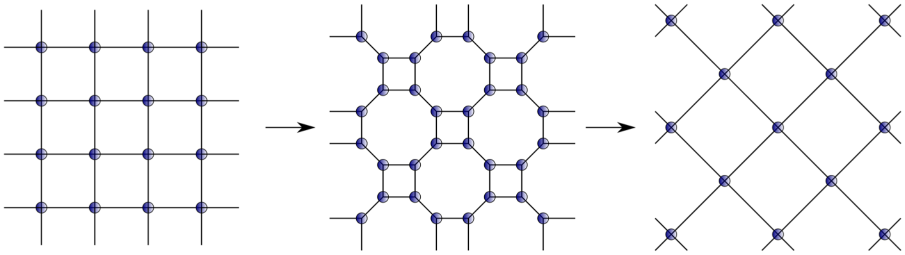


FIG. 1. Scheme of four-valent algorithm: Each four-valent tensor is split along a new edge into two three-valent tensors. Then four of these three-valent tensors are contracted along the original edges to give a new effective four-valent tensor, whose coarser edges are those introduced by the splitting of the original tensors.

$$T_{(ab);(cd)} =: M_{(ab);(cd)} = \sum_{i=1}^{\chi^2} U_{(ab);i}\lambda_i (V_{(cd);i})^\dagger, \quad (2)$$

where U and V are the unitary matrices of singular vectors; λ_i denote the singular values with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\chi^2} \geq 0$.

From U and V one then constructs three-valent tensors, e.g. $S_{ab;i} = U_{(ab);i}\sqrt{\lambda_i}$. Four of these tensors S are then contracted along the links of the original network to give the new effective tensors T',

$$T'_{ijkl} = \sum_{abcd} S_{ab;i} S_{bc;j} S_{cd;k} S_{da;l}. \quad (3)$$

For simplicity we have avoided enumerating the tensors S; in principle they can be different, but this is not important to illustrate the scheme.

The coarse edges of the new network of tensors T' are those obtained from splitting the initial tensors T. Thus the new tensors are actually labeled by the singular values. Note that the SVD (2) is exact such that the coarse network is an exact rewriting of the original partition function. From this we can conclude the physical interpretation underlying tensor network renormalization.

(i) The SVD serves as a variable transformation, reshuffling the original degrees of freedom into effective degrees of freedom on a coarser scale. Since (2) is exact no degrees of freedom are lost, while the relation to the original interpretation is encoded in the maps U and V. Moreover the SVD arranges the degrees of freedom according to their significance, which is indicated by the relative size of the associated singular values.

(ii) In general the tensor T' as obtained from (2) and (3) has a bond dimension of χ^2 compared to χ of the original tensor T. Without approximations this bond dimension grows exponentially with each iteration of the algorithm quickly rendering the scheme inefficient. Thus approximations must be implemented to keep the algorithm feasible. Due to the features of the SVD the quality of the approximations can be readily evaluated: The degrees of freedom are ordered in significance indicated by the size of the singular values. Hence it is straightforward to truncate less important degrees of freedom in (2) by dropping e.g. all λ_i with $i > \chi$. This approximation is actually the best one of $M_{(ab);(cd)}$ by a matrix of rank χ (with respect to least square error).

Of course the more singular values are taken over the better the approximation is, e.g. the position of a phase transition is more accurately determined.

Usually one iterates the algorithm for a fixed bond dimension until the system has converged to a fixed point tensor T^*. This tensor is then used to identify different phases of the model.

In the next section we introduce the models we coarse grain in this article via tensor network renormalization.

2. A brief introduction to quantum group spin nets

In this article we successfully apply tensor network renormalization to so-called spin net models [14,36,37] based on the quantum group $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$. The goal of this section is to give the reader an impression of why this is a remarkable achievement made possible by several improvements of tensor network algorithms. After giving a very short introduction to spin net models we explain why the main challenge is the size of the tensors, encoding the models, leading to a memory consumption that is too large to handle even with high performance computing resources. We then introduce several techniques which allow us to reduce memory usage enormously. We expect that these methods and ideas can also be facilitated by other researchers, in particular in high accuracy calculations.

Spin net models can be defined on lattices of arbitrary dimension, but we restrict ourselves to a two-dimensional square lattice in this article. The models are characterized by a global symmetry group, e.g. a non-Abelian finite or (compact) Lie group. The simplest nontrivial example is the $\mathbb{Z}_2$ Ising model, which is invariant under flipping all Ising spins. This model can be represented in either the group picture, e.g. with the group $\mathbb{Z}_2$ defining the fundamental variables (the Ising spins), or in the dual picture, where the variables are given by the irreducible representation labels of the group. This latter picture defines also a tensor network representation for the spin net models. For non-Abelian groups, the representations are higher than one dimensional and the representation labels are amended by vector space labels.

The second representation, involving representations of the symmetry group, is also called "spin representation." The name is due to the SU(2) representations j, which are referred to as spins. It is this representation which can be generalized also to quantum groups, in particular $\mathrm{SU}(2)_k$ [52,53], which is thoroughly explained in [14]. In this section we restrict the discussion to the most basic features of representation theory for $\mathrm{SU}(2)_k$ in order to discuss the index range of the initial tensor.

(i) $\mathrm{SU}(2)_k$ is a Hopf algebra, the q-deformation of the universal enveloping algebra $\mathcal{U}(\mathrm{SU}(2))$ for $q = \exp(\frac{i\pi}{k+2})$ at the root of unity [52,53]. $k \in \mathbb{N}$ is called the level of the quantum group. Very similar to SU (2) these quantum groups have irreducible representations labeled by spins $j \in \frac{1}{2}\mathbb{N}$. These range from 0 to $j_{\max} = \frac{k}{2}$, the maximal spin of the quantum group. As for SU(2) the representation vector spaces V_j are $(2j+1)$ dimensional. In this work we restrict ourselves to the integer representations of $\mathrm{SU}(2)_k$.[5]

(ii) Each edge of $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ spin nets carries the edge Hilbert space $\mathcal{H}_e = \bigoplus_{j^+,j^-=0}^{j_{\max}} V_{j^+} \otimes V_{j^{+*}} \otimes V_{j^-} \otimes V_{j^{-*}}$, where j^+ and j^- are $\mathrm{SU}(2)_k$ representations and j^* denotes their dual representation. Thus expressed naively as a tensor network each leg of a tensor carries the indices $\{j^+, j^-, m^+, n^+, m^-, n^-\}$, where the so-called magnetic indices m, n range from $-j$ to j, in integer steps.

(iii) The tensor itself encodes the "quantum group symmetries," i.e. it is a projector onto the invariant subspace in the product space of all representation spaces meeting at the vertex, $\mathrm{Inv}(\otimes_{e\supset v} V_{j_e})$. The projector on the full invariant subspace is called the Haar projector; see [14] or Appendix B for its definition.

Due to the symmetry of the model and the finite edge Hilbert spaces, it is in principle possible to directly turn the model into a tensor network. Thus one obtains the tensor

$$T(\{j^+\}, \{j^{+\prime}\}, \{j^-\}, \{j^{-\prime}\}, \{m^+\}, \{n^+\}, \{m^-\}, \{n^-\}). \quad (4)$$

To not overburden the notation, we suppress the indices i of the edges, which range from 1 to 4 in the case of a square network.

However this naive approach is not very feasible for either non-Abelian groups [37] or quantum groups: A quick estimate of the dimension of the edge Hilbert space for a small quantum group, e.g. $k = 4$ such that the spins range over $j = 0, 1, 2$, shows that the index range is roughly 6000 if $j^{\pm\prime} \neq j^{\pm *}$, due to the sheer amount of magnetic indices.

Fortunately due to the symmetries of the model, the dependence of the tensor T on the magnetic indices denoted is not arbitrary and given by the projector/intertwiner structure. This projector structure actually survives under tensor network renormalization and can be exploited to significantly reduce the index range of the initial tensor. To do so, two measures were introduced in [14,37].

(i) The initial tensor was rewritten into a so-called recoupling (or intertwiner) basis, in which the tensor is expanded into a sum over four-valent invariant tensors, which are labeled by (intermediate) spins $\{J^\pm, J^{\pm\prime}\}$. This basis is adapted to the intended splitting of the four-valent tensors into three-valent ones. As the spins $\{j^\pm j^{\pm\prime}\}$ associated to the original edges have to couple to the intermediate spins $\{J^\pm, J^{\pm\prime}\}$, the tensor T can be expressed in a block-diagonal form. Thus the crucial information on the tensor is encoded in an amplitude only depending on the spins $\{j^\pm\}$ on its edges and the spins $\{J^\pm\}$ labeling the basis. The projector structure, and with it the dependence of the tensor T on the magnetic indices, is explicitly preserved under coarse graining. This allows us to precontract the magnetic indices of the projective part into so-called recoupling symbols. Thus during the coarse graining cycle itself we only have to deal with the spin indices.

(ii) The intertwiner structure introduced above can be exploited further by considering the coupling rules of $\mathrm{SU}(2)_k$. In fact the intertwiner basis is written as the sum over two Clebsch-Gordan coefficients which are only nonvanishing if triangle inequalities are satisfied. Thus we introduce one superindex $K(J)$ for each intermediate label J counting the allowed possibilities of the pair (j_1, j_2) coupling to J, dismissing all vanishing entries due to coupling rules. Conversely one can translate $K(J)$ back into representations as $j_i(K(J))$ for $i = 1, 2$. Figure 2 illustrates the idea with reference to the splitting in the algorithm.

(iii) A similar super index $B(J, K(J), K'(J))$ is also defined for the $\mathrm{SU}(2)_k$ $\{6j\}$ recoupling symbol, which is a particular contraction of four Clebsch-Gordan coefficients and appears in the renormalization equations due to the treatment of the magnetic indices. Essentially one picks out one representation J, to which two pairs of two representations couple directly encoded in two superindices $K(J)$ and $K'(J)$. The last remaining representation has to satisfy several conditions and one stores the allowed choices in the index B, which depends on J, $K(J)$ and $K'(J)$. Thus one only stores (and sums over) the nonvanishing $\{6j\}$ symbols. As before we can

[5] This can be understood as the q-deformation of the algebra of SO(3). If k is odd we take $\frac{k-1}{2}$ as the maximal spin.

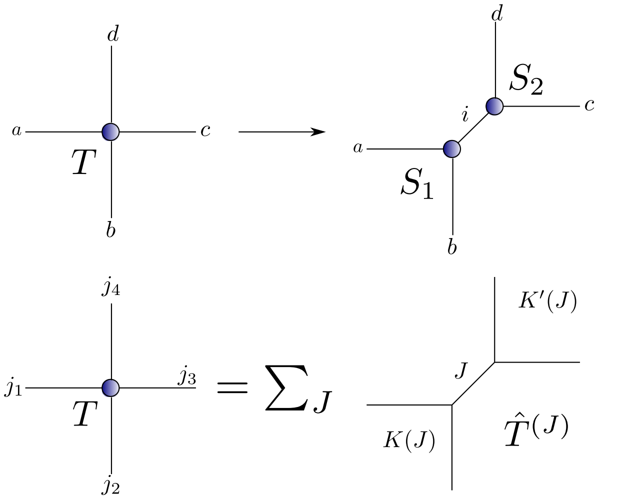


FIG. 2. Visualization of the recoupling basis: Above we show the intended splitting of the coarse graining algorithm. The recoupling basis is chosen such that the variables that are to be split couple to the same intermediate spin J. Instead of storing all boundary data $\{j_i\}$, $i = 1, \ldots, 4$ we only store the ones leading to nonvanishing tensors, stored in superindices $K(J)$ and $K'(J)$ respectively. We have suppressed additional representations $j^\pm$ and magnetic indices.

decode these indices back to the original spin values via functions $j_\alpha(J, K, K', B)$, $\alpha = 1, \ldots 6$. This index is explained in Fig. 3.

The first measure already drastically improves the initial index range of the tensors,

$$\begin{aligned} &T(\{j^\pm\}, \{j^{\pm\prime}\}, \{m^\pm\}, \{n^\pm\}) \\ &= \sum_{J^\pm, J^{\pm\prime}} \hat{T}^{(J^\pm, J^{\pm\prime})}(\{j_i^\pm\}, \{j_i^{\pm\prime}\}) \\ &\quad \times (P^{J^+}_{\{j_i^+\}} \otimes P^{J^{+\prime}}_{\{j_i^{+\prime}\}} \otimes P^{J^-}_{\{j_i^-\}} \otimes P^{J^{-\prime}}_{\{j_i^{-\prime}\}}), \end{aligned} \tag{5}$$

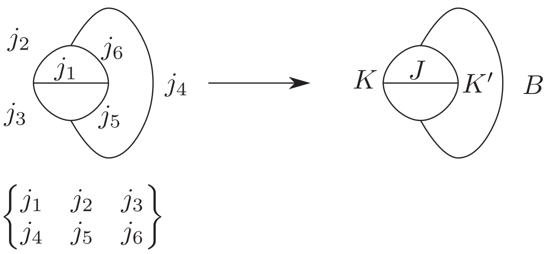

j_2
j_6
j_1
j_4
j_3
j_5
K
J
K'
B
$\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{Bmatrix}$

FIG. 3. Indices for the $\{6j\}$ symbol: We choose $J = j_1$ as the reference spin; then all nonvanishing choices for j_2, j_3 and j_5, j_6 are encoded in indices $K(J)$ and $K'(J)$ respectively. The nonvanishing choices for j_4 then depend on J, K and K' and are summarized in the superindex $B(J, K, K')$.

where the P represent a basis of four-valent intertwiners for one copy of irreducible representations. Their explicit form is not relevant here and can be found in Sec. III A and also [14]. Again the pivotal insight is that this structure is analytically dealt with and preserved under coarse graining, such that the information of the tensor is stored in $\hat{T}$ instead of T. Thus the tensor is specified by four $\mathrm{SU}(2)_k$ representations on each edge, which for $k = 4$ is an index range of $\chi = 3^4 = 81$. For the entire tensor we obtain a size of 81^5 for the four edges and four intermediate spins labeling the basis.

This expression can be further simplified by using superindices $K(J)$. The superindices always combine two representations coupling to the spin J. In the four-valent case we group two representations together according to the intended split, which we denote by sets $\{j_i\}_a$ and $\{j_i\}_b$, such that we also have superindices $K_a(J)$ and $K_b(J)$. Note that from here on we suppress the additional superscripts $^\pm$ and $'$ in order to simplify the notation. Unless specified otherwise $\{j\}$ is supposed to stand for $\{j^\pm, j^{\pm\prime}\}$, similar for J and $K(J)$,

$$\hat{T}^{\{J\}}(\{j_i\}_a, \{j_i\}_b) = \begin{cases} \hat{T}^{\{J\}}(\{j_i(K_a(J))\}_a, \{j_i(K_b(J))\}_b) & \text{if all pairs} \quad \{j_i\}_{a,b} = \{j_i(K_{a,b}(J))\}_{a,b}. \\ 0 & \text{else} \end{cases} \tag{6}$$

Since the projector structure is explicitly preserved it is sufficient to just consider the entries of $\hat{T}$ encoded in the superindices $K(J)$. Since these $K(J)$ combine the representations on two fine edges, one cannot talk about individual index ranges, but we can still give the size of the total tensor. Again for $k = 4$ this is given by 43^4, which is roughly 0.1% of the data without using superindices.

To sum up the measures briefly described in this section provide an interpretational and a computational advantage: By introducing the intertwiner basis we isolate the relevant data of T from the magnetic indices and encode them into a much smaller tensor $\hat{T}$. Moreover the tensor $\hat{T}^{\{J\}}$ is already in a block-diagonal form, where each block is labeled by four $\mathrm{SU}(2)_k$ representations $\{J\}$. Expressed in superindices $\{K_a(J)\}, \{K_b(J)\}$ it can be readily rearranged into a matrix M_{ab} to which SVD is applied in order to split $\hat{T}$. The new edges created in this SVD carry over the labels $\{J\}$ of the block, which are the same type of variables as in the original model. Thus one eventually obtains a new coarse tensor $\hat{T}'$ of the same form as the initial one. This procedure (with smaller index ranges) described here has already been used in [14].

However these significant improvements quickly turn out to be insufficient as soon as one goes to higher levels k

TABLE I. Characteristic numbers for the initial tensor $\hat{T}$ of $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ spin nets for different values of k. We assume 16 bytes per entry.

Level k	j_{max}	Maximal $K(J)$	Number of blocks	Size of largest block	Size of block in GB	Size of $\hat{T}$	Size of $\hat{T}$ in GB
4	2	$K(1) = 5$	81	25^4	∼0.0058	43^4	∼0.051
5	2	$K(1) = 6$	81	36^4	∼0.025	70^4	∼0.36
6	3	$K(1) = K(2) = 8$	256	64^4	∼0.25	160^4	∼9.77
7	3	$K(2) = 10$	256	100^4	∼1.5	246^4	∼54.6
8	4	$K(2) = 13$	625	169^4	∼12.2	461^4	∼673.1
9	4	$K(2) = 15$	625	225^4	∼38.2	671^4	∼3021
10	5	$K(2) = K(3) = 18$	1296	324^4	∼165	1112^4	∼23000

of the quantum group $\mathrm{SU}(2)_k$. In fact one easily reaches the limits of modern high performance computers, in particular in terms of memory usage. This issue is the subject of the next subsection.

3. Memory costs of $\mathbf{SU(2)}_k \times \mathbf{SU(2)}_k$ spin nets

In this section we discuss the main numerical obstruction that has to be overcome if one intends to coarse grain $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ spin nets, which is an immense cost to memory. In order to give an idea of the order of magnitude it is sufficient to just consider the size of the initial tensor $\hat{T}$ [using superindices $K(J)$].

As one increases the level k of the quantum group the size of the initial tensor increases due to two effects: j_{max} and thus the number of irreducible representations increases and the range of the superindex $K(J)$ increases as more pairs of representations (j_1, j_2) can couple to J. Because of the first effect, the number of intertwiners grows exponentially as they are labeled by four $\mathrm{SU}(2)_k$ representations $\{J\}$. Moreover each of these grows in size as well due to the larger range of the superindices. We summarize this in Table I.

From the data we can clearly observe an exponential increase in the size of the initial $\hat{T}$ for growing level k of the quantum group. The memory used to store it roughly increases by an order of magnitude for each increase of k. While the models $k = 4$ and $k = 5$ appear to be small enough to still run on modern notebooks (at least concerning the memory usage), one has to move to high performance machines for $k \geq 6$. However already for $k = 8$ the memory to only define the initial tensor exceeds the memory available on many modern machines. Note also that the memory cost for the initial $\hat{T}$ alone can only serve as a lower bound, in particular if one goes to higher bond dimension after several iterations of the algorithm.

It is apparent that the standard four-valent algorithm [14,29,30] is limited to smaller levels k due to the sheer size of the tensors, which encode the dynamics of the theory. Thus in order to go to larger quantum groups one has to find a way of encoding the same information in smaller building blocks carrying less data. In a way the tensor network algorithm [14,29,30] itself already holds the key to the solution of this problem. During the algorithm the four-valent tensor $\hat{T}$ is split via a SVD into two three-valent tensors $\hat{S}_1$ and $\hat{S}_2$. This transformation is exact as long as no singular values get truncated such that one can encode the same information into three-valent tensors. This allows us to redesign the algorithm [29,30] based on four-valent tensors to an algorithm (first introduced in [31]) which is equivalent in precision, but only involves three-valent tensors and requires moreover less computational time (scaling with χ^3 instead of χ^4). Fortunately in the intertwiner basis it is straightforward to define the initial three-valent tensor without a SVD, such that one can readily work with three-valent tensors $\hat{S}$ instead of $\hat{T}$. The sizes of three-valent and four-valent tensors are compared in Table II.

Unsurprisingly the three-valent tensors are far more economical in terms of memory usage compared to four-valent ones, essentially because they are parametrized by only one superindex $K(J)$ per intertwiner label J instead of two for four-valent ones. However as beneficial as this fact may be one still requires a tensor network algorithm suited to a three-valent tensor which avoids higher valent intermediate tensors as much as possible. In the next section we present such an algorithm, called a triangular algorithm, show that it uses significantly less memory than the original four-valent one and discuss further numerical optimizations.

B. Optimizing tensor network algorithms

In this section we discuss the tensor network algorithm used to coarse grain $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ spin net models. Particular attention is given to the so-called triangular algorithm which can be understood as a modification of the familiar algorithms [29,30] using three-valent tensors, denoted by $\hat{S}$, as its basic building block instead of four-valent ones, called $\hat{T}$. Originally it was invented in [31] and already applied in [54] yet it turns out to be indispensable for the models under discussion as motivated in the previous section. Furthermore we also discuss

TABLE II. Comparison of the size of three-valent tensors $\hat{S}$ and four-valent tensors $\hat{T}$ for various levels k of the quantum group.

Level k	$j_{\max}$	Maximal $K(j)$	Size of $\hat{S}$	Size of $\hat{S}$ in GB	Size of $\hat{T}$	Size of $\hat{T}$ in GB
4	2	$K(1) = 5$	11^4	~0.00022	43^4	~0.051
5	2	$K(1) = 6$	14^4	~0.0006	70^4	~0.36
6	3	$K(1) = K(2) = 8$	24^4	~0.005	160^4	~9.77
7	3	$K(2) = 10$	30^4	~0.013	246^4	~54.6
8	4	$K(2) = 13$	45^4	~0.062	461^4	~673.1
9	4	$K(2) = 15$	55^4	~0.14	671^4	~3021
10	5	$K(2) = K(3) = 18$	76^4	~0.5	1112^4	~23000

improvements to the code itself which are recommended if one is dealing with tensors and matrices of the size mentioned before.

1. Triangular algorithm

As already discussed the step from the four-valent algorithm [29,30] to a three-valent one is almost directly built into the four-valent algorithm. At an intermediate step each four-valent tensor is split into two three-valent ones by a SVD. Then four of the latter are glued together to form a new four-valent tensor, rotated by 45 degrees. However in the next iteration of the algorithm this new coarse tensor is again split into two coarse three-valent tensors that one in principle could directly construct from two fine three-valent ones. Thus the question arises whether one can instead work just with three-valent tensors as the basic building blocks. To make this idea more clear, let us consider the lattice dual to the tensor network.

The lattice dual to a four-valent (square) tensor network is again a square lattice. By splitting the four-valent tensor into three-valent ones each square is cut along its diagonal into two triangles, each dual to a three-valent tensor. As such one obtains a regular triangulation of the square lattice. In the next step of the algorithm four of these triangles are glued to form a coarser square. During the next iteration this square is again split into two triangles, where the (coarse) cut is precisely along the lines along which the finer triangles were glued. In fact it appears that the new coarse triangles are made up of two fine triangles plus a variable redefinition, mapping a "subdivided edge into a coarse edge." These mappings[6] are instrumental for defining the truncation scheme underlying the tensor network algorithm [16]. This motivates the triangular algorithm [31,54].

Before we discuss this algorithm in detail note that it is defined for a square lattice split into a triangulation as described above. This is necessary in order to straightforwardly iterate the algorithm.

The triangular algorithm is demonstrated for a larger network in Fig. 4. The first step is therefore to turn the four-valent tensor network into a three-valent one. In principle this can be done by performing the first step of the usual formalism, but it should rather be avoided if one has to deal with tensors of the size illustrated in Sec. II A. Fortunately it is possible to analytically define the three-valent tensors for many models including the spin nets under discussion; the Ising model is another example that is straightforward to split into three-valent tensors.

So assume the triangulation of the square lattice, or its dual tensor network of three-valent tensors $\hat{S}$, is given, where $\hat{S}$ depends on the following variables:

$$\hat{S}^{\{J\}}(\{j_a(K_a(J))\}). \tag{7}$$

The notation is purposely similar to the four-valent tensor $\hat{T}$. On the one hand the spins $\{J\}$ label again the projector basis and also serve as the label for the superindices $\{K_a(J)\}$ summarizing the (nonvanishing) configurations of fine spin $\{j_a\}$. On the other hand $\{J\}$ actually represent a variable in the model attached to one edge of the three-valent tensor/triangle. In fact this edge is distinguished both in the definition of $\hat{S}$ and in the lattice/network, as it is a coarser (that is by a factor of $\sqrt{2}$ longer) edge. This is also illustrated in Fig. 5.

Given this triangulation it is straightforward to identify a coarse triangle made up of two fine triangles glued along a fine edge. From the tensor network perspective one edge between two three-valent tensors gets contracted resulting in a new effective four-valent tensor describing a coarse triangle with a subdivided edge. Of course this immediately raises the question whether one runs into the same issue as the original algorithm of having to store an entire four-valent tensor. Fortunately this problem can be nicely circumvented.

Recall that both the three-valent $\hat{S}$ and the four-valent tensor $\hat{T}$ were expressed in a so-called intertwiner basis, such that they are labeled by spins $\{J\}$ [denoting the basis and superindices $\{K(J)\}$]. The same expansion can be applied to the new four-valent tensor $\hat{\mathcal{T}}$ obtained from two three-valent ones, see also Fig. 6,

[6] The maps can be seen as coarse graining maps or if one considers the inverse maps, as embedding maps, which are crucial for the continuum limit [16].

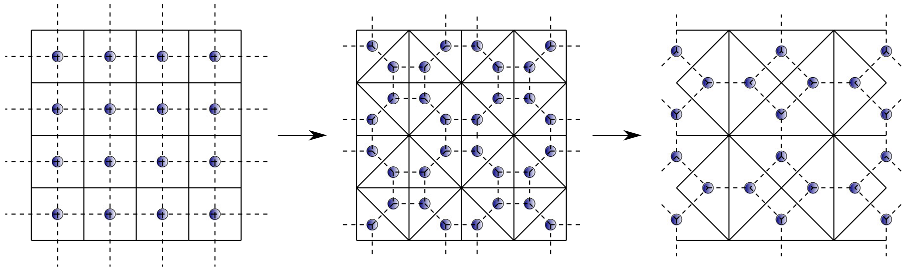


FIG. 4. Triangular algorithm for an extended network: First the square lattice is split into a regular triangulation as in the original algorithm [29,30]. Instead of defining new coarse squares/four-valent tensors one combines two triangles into a coarse triangle, thus working with three-valent tensors instead.

$$\begin{aligned}&\hat{\mathcal{T}}^{\{\bar J\}}(\{J_a(K_a(\bar J))\},\{j_b(K_b(\bar J))\})\\ &\quad=\sum_{\{B(\bar J,K_a,K_b)\}} S^{\{J_a(K_a(\bar J))\}_1}(\{j_b(K_b(\bar J)\}_1,\\ &\quad\{j(B)\})S^{\{J_a(K_a(\bar J))\}_2}(\{j(B)\},\\ &\quad\{j_b(K_b(\bar J)\}_2)\times f(J,K_a,K_b,B). \qquad (8)\end{aligned}$$

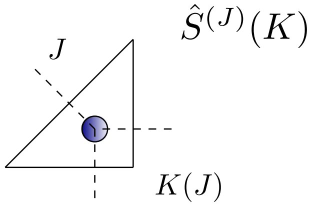


FIG. 5. Labels of triangular tensor: The triangular tensor essentially inherits its block-diagonal form from the four-valent one. The new coarse edge carries the reference spin J; the fine spins are combined into a superindex $K(J)$.

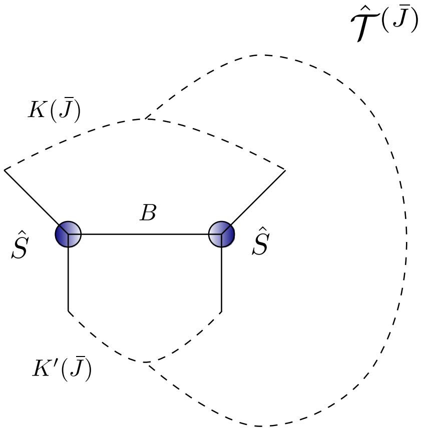


FIG. 6. Intermediate four-valent tensor $\hat{\mathcal{T}}$: Two three-valent tensor are contracted to form an intermediate four-valent tensor. It is directly defined in block-diagonal form labeled by $\bar J$ and indicated by the dashed recoupling lines. The combinatorics of this diagram exactly match the $\{6j\}$ symbol, such that one only sums over nonvanishing configurations labeled by the superindex B.

For the sake of simplicity we combine all $\mathrm{SU}(2)_k$ specific expressions, that is, recoupling symbols, quantum dimension factors, etc., in the function f, which is also expressed entirely as a function of $\{\bar J, K_a, K_b, B\}$. See Appendix C for the complete formula. Crucially the tensor $\hat{\mathcal{T}}$ is directly expressed in its new intertwiner basis labeled by new indices $\{\bar J\}$. Consequently the old indices J of $\hat S$ get expressed in terms of the superindex $K_a(\bar J)$, the fine (uncontracted) j in terms of $K_b(\bar J)$ and the fine contracted j in terms of $B(\bar J, K_a, K_b)$. Again this insures that one only sums over representations allowed by the coupling rules.

As we have discussed in Sec. II A, storing the entire tensor $\hat{\mathcal{T}}$ is a costly endeavor. Fortunately this is not necessary in this case: Recall that our goal is to construct a new three-valent tensor $\hat S'$ from $\hat{\mathcal{T}}$, which then again serves as the starting point for the next iteration of the algorithm. To do so we have to combine the two indices of the subdivided coarse edge, which are labeled by the superindices $\{K_b\}$, into one effective index via a variable transformation, which is computed with a SVD. The form of $\hat{\mathcal{T}}^{\{\bar J\}}$ is precisely chosen with this goal in mind, as we intend to preserve the coarse edges labeled by $J_a(K_a)$ and (variable) transform the fine ones labeled by $j_b(K_b)$.

Thus we apply a SVD to

$$\hat{\mathcal{T}}^{\{\bar J\}}_{\{K_a\};\{K_b\}} =: M^{\{\bar J\}}_{ab} = \sum_i U^{\{\bar J\}}_{ai}\lambda_i (V^{\{\bar J\}}_{bi})^\dagger, \qquad (9)$$

where U_{ai} and V_{bi} denote the ith left and right singular vectors of M_{ab} respectively, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are the respective singular values ordered in size. The sum over the index i counting the singular values runs over the full range of superindices $\{K(J)\}$.

The last step to obtain the new coarse tensor $\hat S'$ is the contraction of the indices $\{j_b(K_b(\bar J))\}$ with the singular vectors $V^{\{\bar J\}}_{\{j_b\},i}$,

$$\begin{aligned}(\hat S')^{\{\bar J,i\}}(\{J_a(K_a(\bar J))\}) &= \sum_{\{j_b(K_b)\}} \hat{\mathcal{T}}^{\{\bar J\}}_{\{K_a\};\{K_b\}} V^{\{\bar J\}}_{\{j_b\},i}\\ &= U^{\{\bar J\}}_{\{J_a(K_a)\},i}\lambda_i. \qquad (10)\end{aligned}$$

The last identity follows from the fact that the matrices of singular vectors U, V are unitary. For Eq. (10) to be valid, one has to insert a resolution of the identity $VV^{\dagger}$ into the tensor network in order not to change the partition function. As a consequence the tensors opposite of $\hat{\mathcal{T}}$ get contracted with $V^{\dagger}$; however there the second identity in (10) does not apply in general.[7]

Having described the first iteration of the triangular algorithm, it is time to address the elephant in the room: the size of the intermediate tensor $\hat{\mathcal{T}}$ and how one can avoid saving all of it in the triangular algorithm. Therefore we draw the reader's attention to Eqs. (8)–(10): the intertwiner basis $\{\bar{J}\}$ puts both the tensors $\hat{\mathcal{T}}$ and $\hat{S}$ into the same block-diagonal form, as the new coarsest edge of $\hat{S}$ inherits the labels from $\hat{\mathcal{T}}$. Therefore, in order to compute one block $\{\bar{J}\}$ of $\hat{S}$ one only has to know the block $\{\bar{J}\}$ of $\hat{\mathcal{T}}$. As a result one can compute $\hat{S}'$ block by block, for which one only has to compute and store the respective block of $\hat{\mathcal{T}}$. Thus, as we can see from Tables I and II, the main limiting factor in the triangular algorithm is actually the size of the largest block $\hat{\mathcal{T}}^{\{\bar{J}\}}$, which is roughly 2 orders of magnitude smaller than the full four-valent tensor for any level k of the quantum group.

This concludes the principle discussion of the three-valent algorithm, in which we paid particular attention to the (avoidance of the) memory problem. In order to iterate the code one necessarily has to implement a truncation scheme after the SVD, as in any other tensor network algorithm. This is the subject of the next section. Furthermore we explain and justify the simplifications made in the algorithm.

2. Truncation scheme and simplifications

Any numerical tensor network algorithm requires a truncation scheme after the SVD has been performed. If we consider Eq. (10) again, we realize that the coarse three-valent tensor $\hat{S}'$ comes with an additional label i attached to $\{\bar{J}\}$ enumerating the singular values from the previous iteration in this block. Interpretationwise it tells us that $\{\bar{J}\}$ appears with a certain multiplicity, which has to be stored again. In the following iterations these index ranges keep growing exponentially such that one eventually is forced to truncate. However this should be done such that the error is as small as possible.

The decomposition of a tensor (rearranged as a matrix) by a SVD is optimal for that, as truncating the singular values to the largest χ values gives the best approximation (in terms of the least square error) of this matrix by another matrix of (lower) rank χ. For the full tensor, that is, all blocks $\{J\}$, one should compute all of the singular values for each block, compare them and take the largest χ of those. The approximation is improved if χ is increased.

As straightforward as this idea is it is rather cumbersome to realize in the context of this work. Due to the size of four-valent tensors in this model, one cannot simply compute the SVD of all blocks one after another and store them, as the full U and V each take up as much memory as the original $\hat{\mathcal{T}}$. Moreover computing the full SVD, i.e. all singular values and vectors, of matrices of the size shown in Table I (the size of largest block column) is very costly. Instead one would have to compute one block $\{J\}$ of $\hat{\mathcal{T}}$, compute only its singular values and store them. Then one deletes the current block and continues with the next one. Once all the singular values have been computed they are compared and the largest χ singular values are taken over. Afterwards one computes $\hat{\mathcal{T}}$ (block by block) again and computes the SVD only for the amount of singular values taken over.

Furthermore note that increasing the number of singular values taken over directly affects the sizes of the three-valent and four-valent tensors (and matrices) in the next and next-to-next iteration due to the asymmetry of the three-valent tensors. Thus one may yet again run into memory issues. It has been noted in [36] that it is best advised not to lower the number of singular values in one block in the following iterations.

Instead of this elaborate truncation scheme we use the same simple one as in [14], namely we take over only one singular value per block $\{J\}$. Even though it may appear to be very low at first sight, it actually does take many singular values into account: If we consult Table I again we realize that for $k = 8$, which is the largest model we have studied, we already have 625 blocks. Experience from previous models [14] showed that for many models this scheme is already suitable for exploring the phase structure of the model. Nevertheless for one model discussed in this article this approximation scheme clearly breaks down.

Concerning the triangular algorithm a further comment is in order: In the original tensor network algorithm [29,30] one usually gets four different three-valent tensors from splitting the four-valent tensor in two ways. Therefore in full generality the triangular algorithm is formulated for four three-valent tensors, where each of these four gets renormalized. Fortunately in the model under discussion, as well as the Ising model [31], one finds out that all four three-valent tensors are identical. This also extends to the fact that Eq. (10) applies to all three-valent tensors $\hat{S}$. Thus it is sufficient to perform the three-valent algorithm with just one tensor $\hat{S}$, which slightly reduces memory usage but more importantly saves a lot of computational time, as less SVDs and summations have to be performed.

This concludes the discussion of the triangular algorithm. In the next section we briefly discuss several improvements of the code and parallelization.

[7] To improve the algorithm one should actually perform the SVD for both pairs of tensors and compare which map minimizes the error.

3. Code improvements and parallelization

The triangular algorithm describes the current state as a block-structured matrix, i.e. as a matrix consisting of dense blocks of varying sizes, where some blocks are known to be identically 0 due to symmetry, and certain blocks are mere transpositions of others. It goes without saying that one needs to make use of this structure to improve performance and reduce memory requirements.

The operations described above implementing the renormalization flow define a new block-structured matrix in terms of an existing one via tensor contractions. This creates intermediate objects that can be significantly larger than both the initial and the final state. It is thus crucial to perform the tensor contractions in an optimal order to reduce the amount of memory required for intermediate states.

As modern workstations have multiple cores, it is necessary to find parallelism for good performance. The triangular algorithm can be parallelized in two ways. First, each tensor operation (i.e. matrix multiplication or singular value decomposition) can be executed in a parallel operation. This is worthwhile only for sufficiently large blocks, with more than about 20^2 elements on a modern workstation. Second, the blocks making up the resulting tensor can all be evaluated simultaneously. We found the latter to lead to the best performance, because while there are some large blocks, most blocks are too small to be parallelized by themselves. We implemented this both via a shared memory OpenMP [55] parallelization with dynamically scheduled loop in a C++ code, as well as via a distributed memory parallelization in a Julia code [56].[8]

C. Summary of triangular algorithm

Before we continue to discuss the models studied in this article in more detail, in particular from the perspective of spin foam quantum gravity, we summarize and conclude this part of the article about the employed tensor network algorithm. In Sec. II A we have illustrated in detail that for models equipped with a large symmetry group, like $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ in our case, the four-valent algorithm [29,30] is quickly limited by memory on modern machines. This even holds when one exploits the symmetries of the model as in [14,37]. By a shift of perspective from four-valent to three-valent tensors we have remedied this issue and invented the triangular algorithm [31,54], which is only limited by the size of the largest block (in terms of the symmetries) instead of the size of the entire tensor. Indeed in our context the triangular algorithm allowed us to go to much larger levels k of the quantum group.

In addition to that the shift to smaller fundamental building blocks should have more potential for other practitioners of tensor network algorithms, and also for smaller symmetry groups. Since the triangular algorithm is more economical than its four-valent counterpart it should be possible to further increase the bond dimension of the model and thus improve the approximation or lower the computational costs at the same level of accuracy. Moreover it might be worthwhile to modify the entanglement filtering algorithm [32] to the triangular case.

This concludes the part of this article focusing on the technical and numerical aspects of this work. In the next sections we go into more detail about the models under discussion, in particular establishing the relation to modern spin foam models and presenting the results of applying the above-mentioned triangular algorithm to these models.

III. PART II: RESULTS

A. Intertwiner and spin net models

After focusing on the technical and numerical challenges one faces when renormalizing quantum group spin net models, the rest of this article focuses more on the technical details of the models and their relation to spin foam models. This entails a brief introduction to and motivation of the models, including a discussion of intertwiner models [57], a detailed construction of spin net models mimicking modern spin foam models and a presentation of the results from coarse graining these models.

Spin net models are defined on a lattice of arbitrary dimension.[9] To each vertex one assigns a group element $g_v \in G$, e.g. a finite or a (compact) Lie group, and weight functions $\omega_e : G \to \mathbb{C}$. For concreteness and similarity to the quantum group case we assume that G is finite. The edges of the lattice come with an orientation and the associated weights are evaluated on the product of group elements at the "source" and "target" of the edge, i.e. $\omega_e(g_{s(e)} g_{t(e)}^{-1})$, where $s(e)/t(e)$ denote the source/target vertex of e respectively. Crucially the edge weights are invariant under conjugation, i.e. $\omega_e(hgh^{-1}) = \omega_e(g) \forall h \in G$. Thus the system possesses a global symmetry, as it is invariant under (left and right) multiplying the same group element to all group elements on the vertices. To conclude, the partition function of this system is given by

$$Z = \frac{1}{|G|^E} \sum_{\{g_v\}} \prod_e \omega_e(g_{s(e)} g_{t(e)}^{-1}). \tag{11}$$

[8] Readers interested in the codes can request them by contacting the authors.

[9] This lattice need not be regular for tensor network methods to work. However regular lattices result in regular tensor networks, whose numerical coarse graining can be straightforwardly iterated.

By $|G|$ we denote that the number of group elements of G and E is the number of edges in the lattice. The simplest nontrivial model is the $\mathbb{Z}_2$ Ising model (for vanishing external magnetic field).

A dual description of spin net models can be derived from (11) by a group Fourier transform, as can be found in [34,58]. To do so one expands the class functions $\omega_e(g)$ into characters $\chi_\rho(g)$ of irreducible representations ρ of the group G. As the characters factorize over group elements (into representations) so does (11), such that the group summation can be performed for each vertex individually. Eventually one obtains a group theoretic object, the Haar projector $\mathcal{P}_v$,[10]at each vertex of the lattice, such that (11) is rewritten as

$$Z = \frac{1}{|G|^E} \sum_{\{\rho_e, m_e, n_e\}} \prod_e \tilde{\omega}_{\rho_e} \prod_v \mathcal{P}_v(\{\rho_e\}_{e\supset v})^{\{m_e\}}_{\{n_e\}}$$
$$= \sum_{\{\rho_e, m_e, n_e\}} \prod_e \tilde{\omega}_{\rho_e} \prod_v \sum_{|\iota_d\rangle} {}^{\{m_e\}}|\iota_d\rangle\langle\iota_d|_{\{n_e\}}. \tag{12}$$

$\tilde{\omega}_{\rho_e}$ denotes the Fourier transformed edge weight. The indices of $\mathcal{P}_v$ are contracted with the indices of the other projectors according to the connectivity of the graph, such that each edge essentially carries the Hilbert space $\mathcal{H}_e = \bigoplus_\rho V_\rho \otimes V_{\rho^*}$, where ρ^* denotes the dual representation to $\rho.\mathcal{P}$ itself is a projector satisfying $\mathcal{P}\cdot\mathcal{P} = \mathcal{P}$. More information and details on the derivation of these expressions can be found in [34].

In order to apply tensor network algorithms, the partition function (12) needs to be rewritten as a contraction of tensors, i.e. as a sum over tensor indices. This implies that we must assign all amplitudes to the vertices and variables to the edges.[11] As the projectors are already assigned to the vertices, it remains to split the edge weights ω_e by assigning $\sqrt{\omega_e}$ to the source and target vertex of each edge e. Hence we define the following tensor T:

$$T(\{\rho\},\{m\},\{n\}) \coloneqq \frac{1}{|G|^2} \prod_{e\supset v} \sqrt{\tilde{\omega}_{\rho_e}}\, \mathcal{P}_v(\{\rho_e\}_{e\supset v})^{\{m_e\}}_{\{n_e\}}$$
$$= \prod_{e\supset v} \sqrt{\tilde{\omega}_{\rho_e}} \sum_{|\iota_d\rangle} {}^{\{m_e\}}|\iota_d\rangle\langle\iota_d|_{\{n_e\}}. \tag{13}$$

Given this T, the sums over irreducible representations and magnetic indices in (12) are expressed as the contraction of tensor indices according to the combinatorics of the network, called the tensor trace Ttr,

$$Z = \mathrm{Ttr} T\ldots T. \tag{14}$$

As briefly discussed in Sec. II A the representation (11) is not available for quantum groups, as these are not groups but Hopf algebras. Here we understand the quantum group $\mathrm{SU}(2)_k$ as the q-deformation of the universal enveloping algebra $\mathcal{U}(\mathrm{SU}(2))$ with $q = \exp(\frac{i\pi}{k+2})$ being a root of unity [52,53]. k denotes the level of the quantum group. Since the representation theory of $\mathrm{SU}(2)_k$ is well understood and actually very similar to the one of SU(2), we take representation (12) as our starting point. The most notable difference to the undeformed case is the cutoff in the spins $j_{\max} = \frac{k}{2}$, which depends on the level k. Details on the derivation of the Haar projector for $\mathrm{SU}(2)_k$ can be found in [14].

At this stage we briefly comment on the relation of spin net models and spin foams: the dynamical ingredients of spin foams are very similar to spin nets; however there are two major differences. First, group elements are assigned to edges instead of vertices and weights are assigned to faces. In representation (12), spin foams carry representations on the faces and intertwiners on the edges of the foam. The second difference is that spin foams possess a local gauge symmetry instead of a global one. As a result the partition functions of spin foams and spin nets are very similar in form. Another difference is the chosen dimension of the systems. Spin foam models are usually describing four-dimensional spacetimes, whereas we study here two-dimensional spin net models. There are several reasons for this choice. It is a known feature that four-dimensional lattice gauge theories and two-dimensional spin systems share certain statistical properties and have similar phase structures [35]. In addition to that two-dimensional spin net models on a square lattice feature four-valent projectors on their vertices as do four-dimensional spin foam models defined on (the dual of a) triangulation.

Among these reasons, the similarity of dynamical ingredients was the main motivation for the construction of spin net models [14,36,37] as it allows us to capture a key dynamical ingredient of spin foams, the so-called simplicity constraints. These simplicity constraints appear in the Plebansky formulation of general relativity [59] and break the symmetries of topological BF theory to obtain a theory with propagating degrees of freedom. Spin foam models take this Plebanski action as their starting point, but first discretize and quantize the topological theory. Discretized versions of the simplicity constraints are then imposed at the quantum level and are expected to result in propagating degrees of freedom. However this construction is not unique and is the cause of different spin foam models; see again [3] for recent review. Due to their (dynamical) similarities we hope that spin nets can serve as analogue models for spin foams and that the phases and possibly some key features of the coarse graining flow agree in these

[10] $\mathcal{P}_v : V_{\rho_1} \otimes \ldots \otimes V_{\rho_n} \to \mathrm{Inv}(V_{\rho_1} \otimes \ldots \otimes V_{\rho_n})$ is the projector onto the invariant subspace of the tensor product of representation vector spaces V_ρ. It can also be seen as a sum over an orthonormal basis of intertwiners $|\iota_d\rangle \in \mathrm{Inv}(V_{\rho_1} \otimes \ldots \otimes V_{\rho_n})$.

[11] In general there exist many choices on how to rewrite a partition function as a tensor network. In fact the tensor network can be different from the underlying discretization.

models. Due to their simpler structure spin nets allow the tracking of the simplicity constraints during coarse graining. Besides spin nets are also useful in developing coarse graining techniques that might be also applicable to spin foams.

In fact spin nets can even be identified with spin foams defined on a very special underlying discretization, known as melon spin foam [14]: Such a melon consists of two spin foam vertices glued together via many spin foam edges. The spin net coarse graining corresponds to a bundling of a number of spin foam edges into thicker edges. The simplicity constraints determine also how the two spin foam vertices are glued to each other. For instance later on we encounter so-called factorizing spin net models, which translate to factorizing, and hence unglued, spin foam vertices.

As mentioned simplicity constraints are of particular interest for the dynamics of spin foams. At the core of the construction of modern spin foam models is the insight of Plebanski that the Palatini action (a first order formulation of the Einstein-Hilbert action) can be written as a constrained topological field theory [3,59]. Since it is understood how to discretize and quantize the unconstrained topological theory, known as BF theory, many modern spin foam models use it as the starting point [3,39,41,42]. In order to obtain propagating degrees of freedom (a version of) the simplicity constraints are implemented at the discrete quantum level. As it is generically the case for discretizations this procedure is not unique and the root of differences among modern spin foam models. In more detail the simplicity constraints affect the projectors $\mathcal{P}$ on the invariant subspace, e.g. in (12), by forbidding certain intertwiners $|\iota\rangle$ such that $\mathcal{P}$ projects only onto a subspace. In this regard spin foam models can be seen as extensions of lattice gauge theories [33].

Despite these difference in the construction of spin foam models they are remarkably in agreement with the semiclassical limit for one vertex amplitude, i.e. the amplitude assigned to a 4-simplex, which is given by the cosine of the Regge action [60], a discretization of general relativity on a triangulation, of this simplex [61–64]. While this is an encouraging result examinations for larger 2-complexes are scarce and the dynamics of spin foams is not well understood as well. Concerning the simplicity constraints this is a crucial issue to tackle as they play the crucial role of implementing the dynamics. Whether this dynamics is nontrivial, in the sense of describing propagating degrees of freedom, and furthermore compatible with general relativity in an appropriate limit is on open question. In turn a better understanding of the dynamics can lead to an improved construction of the models.

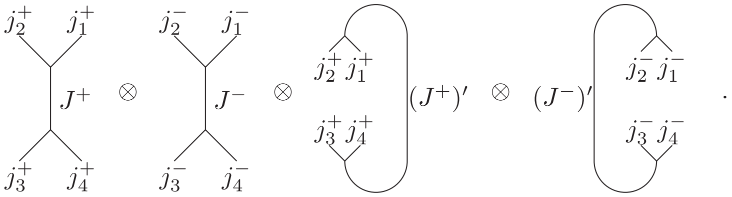
Thus progress towards this goal would be achieved by examining the effect of the simplicity constraints if the foam consists of more than one building block. This is the question of how the subspaces the constraints project on change under coarse graining, i.e. how the constraints act effectively on a coarser scale. Due to their dynamical similarity to spin foams, yet for a simpler dynamics, we can address these questions in spin net models and subject them to a real space renormalization procedure. The purpose of this article is therefore to construct spin net models mimicking the properties of two four-dimensional spin foam models, namely the BC [39] and the EPRL/FK [41,42] model, and coarse grain them via tensor network renormalization.

To this end the choice of the proper symmetry group is crucial, where studying spin nets for the quantum group $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ actually kills two birds with one stone. On the one hand one requires a cutoff on the (irreducible) representations in order to apply tensor network renormalization. On the other hand quantum groups also provide us with a physical motivation from quantum gravity as it is conjectured that spin foam models for quantum groups model gravity with a cosmological constant $\Lambda \neq 0$ [65–72]. This insight stems from the Turaev-Viro model [73], a spin foam model for discretized Euclidean quantum gravity in three dimensions with a positive cosmological constant. Its basic amplitude is precisely the $\mathrm{SU}(2)_k$ $6j$-symbol describing a constantly (positively) curved quantum tetrahedron describing the case $\Lambda > 0$. In this setting the maximum spin $j_{\max}$ can be interpreted as an infrared cutoff via the cosmological constant with $\Lambda \sim 1/j_{\max}$. The small value of Λ in current observations would thus suggest a large level k of the quantum group. Extensions of these ideas to four-dimensional spin foam models are a topical field of research [50,69,70] and have recently uncovered an interesting connection to Chern-Simons theory [74]. Moreover, canonical loop quantum gravity frameworks, implementing a cosmological constant, can also be constructed, thus providing the boundary Hilbert spaces for these models [72,75–77].

After this motivation of quantum group spin nets let us return to the discussion of the model itself. As discussed above, the partition function is of the general form (12) and can be written as a tensor network with (13) and (14). With $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ as the underlying quantum group irreducible representations are labeled by two irreducible $\mathrm{SU}(2)_k$ representations j^+ and j^-; consequently the edge Hilbert space is $\mathcal{H}_e = \bigoplus_{j^+,j^-} V_{j^+} \otimes V_{j^-} \otimes V_{j^{+*}} \otimes V_{j^{-*}}$. Due to the quantum group symmetry encoded in the projectors the tensors T are of a very specific form derived in [14] and alluded to in Secs. II A and II B,

$$T(\{j^+\},\{j^-\},\{m^+\},\{m^-\},\{n^+\},\{n^-\}) =$$

$$= \sum_{J^\pm,(J^\pm)'} \hat{T}^{(J^\pm,(J^\pm)'}(\{j^+\},\{j^-\})\, d_{J^+} d_{J^-} d_{(J^+)'} d_{(J^-)'}$$



(15)

This equation expresses the change of basis, namely to the recoupling basis, at the heart of the tensor network algorithm. d_J denotes the quantum dimension $[2J+1]$ of the representation J, with $[n]$ being the quantum number of n (see Appendix A). $\hat{T}$, which is a function solely of representation labels, is the tensor expressed in the new, block-diagonal basis. The graphical expressions denote the Haar projector of $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ (modulo dimension factors and signs) and encode the dependence on the magnetic indices. Essentially each trivalent vertex is dual to a Clebsch-Gordan coefficient ${}_q\mathcal{C}^{\{j\}}_{\{m\}}$, where the first two diagrams encode the indices $\{m^\pm\}$ and the latter two encode $\{n^\pm\}$. A short explanation on this graphical calculus can be found in Appendix B; see also [14] for more details.

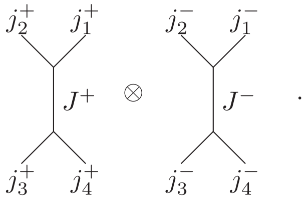
The crucial point about identity (15) is that any spin net model equipped with the quantum group symmetry can be written in this form. Thus the different methods and concepts of implementing simplicity constraints result in different tensors $\hat{T}$. Moreover the symmetry structure of the model is precisely preserved under coarse graining, such that we can directly examine the renormalization group flow of $\hat{T}$, which one could then interpret as the flow of the simplicity constraints. Again we recall that the basis (15) is crucial for developing and optimizing the algorithm such that it can be applied to larger levels k of the quantum group, which is the subject of Secs. II A and II B.

Before we go on and discuss in more detail the construction of BC and EPRL spin nets, we also introduce $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ intertwiner models and give the reader a brief context to other models examined before.

1. Intertwiner models

Seen from spin net models, intertwiner models resemble simpler version with fewer degrees of freedom. Looking at the spin net for an arbitrary group (12), intertwiner models only have an edge Hilbert space $\mathcal{H}_e = \otimes_\rho V_\rho$, so the dual representations are missing. Thus the model is considerably simpler, in particular compared to numerically coarse grain, but of a similar form as (15),

$$t(\{j^+\},\{j^-\},\{m^+\},\{m^-\},) = \sum_{J^\pm,(J^\pm)'} \hat{t}^{(J^\pm)}(\{j^+\},\{j^-\})\, d_{J^+} d_{J^-} \tag{16}$$



Indeed these models can prove useful in better understanding the full spin net models, as some of the fixed points obtained via coarse graining actually factorize, that is, the sets of representations decouple.

In previous work a similar behavior has already been encountered: $\mathrm{SU}(2)_k$ intertwiner models have been introduced in [57], where several (families of) topological fixed points were derived by requiring triangulation independence. These topological fixed points were then used in [14] as initial data for $\mathrm{SU}(2)_k$ spin nets and a rich phase structure with possibly second order phase transitions were found. Some of the fixed points describing the phases turned out to be factorizing as the representation j and its dual j^* completely decoupled. Indeed taking the tensor product of fixed points of intertwiner models is also a fixed point of spin nets, as long as the representations j and $j' \neq j^*$ are uncoupled. Note however, e.g. in (15), that in the initial spin net models $j' = j^*$. So one can interpret a spin net as two entangled, or rather interacting, intertwiner models.[12] Pushing these analogies even further, one can interpret $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ spin nets as a tensor product of two interacting $\mathrm{SU}(2)_k$ spin nets or as a tensor product of four interacting $\mathrm{SU}(2)_k$ intertwiner models.

In the next section we discuss the construction of two spin net models in detail, an analogue BC and an analogue EPRL model.

B. BC and EPRL models

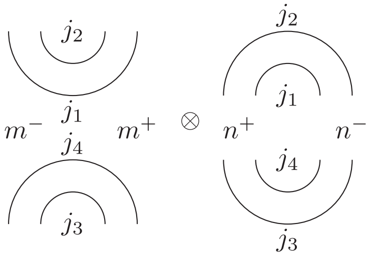
As we have discussed before most four-dimensional spin foam models are built by imposing a version of the simplicity constraints on a discretized and quantized BF theory, with the goal to break the (too many) symmetries of the latter to obtain a theory with propagating degrees of freedom. In this article, we intend to construct analogue models by mimicking the four-dimensional procedures. In order to keep this concise, we focus on the two models which have been studied most thoroughly in the literature, namely the BC and EPRL models, which differ significantly in their construction.

1. The BC construction

The BC model is one of the first four-dimensional spin foam models that have been constructed, both for Euclidean [39] and Lorentzian signatures [78]. As discussed before we focus on the Euclidean version in this article. A BC spin foam model implementing a cosmological constant has been constructed in [50]. This work also includes the Monte Carlo simulation of some observables, like relative frequencies of spin values.

Nowadays the BC model is disfavored, for mainly two reasons [3,79,80]. On the one hand it suffers from metric discontinuities in the semiclassical limit due to nonmatching shapes of tetrahedra along which 4-simplices are glued. On the other hand the BC model (intentionally) does not make contact with the (kinematical) Hilbert space of loop quantum gravity (LQG) (as a boundary Hilbert space), such that it cannot be used to define a physical inner product in LQG. In particular the latter point is overcome in the EPRL model. Indeed, a motivation of the EPRL model was the failure of the BC model to reproduce all properties of the continuum graviton propagator [81].

Despite these disadvantages the BC model is an interesting model to study due to its geometric construction and remarkable simplicity. Essentially it is constructed by describing a triangulated four-dimensional Riemannian manifold by assigning bivectors to all triangles of the 4-simplices plus constraints. These are identified with Lie algebra elements and quantized by expressing them as group theoretic objects, namely by assigning $\mathrm{SU}(2) \times \mathrm{SU}(2)$ representations (j^+, j^-) to the triangles and intertwining maps to the tetrahedra. Simplicity constraints are implemented by requiring the bivectors to be simple, which is translated to the representation labels $j^+ = j^-$[13] Representations (j, j) are thus also called simple representations. Four-valent intertwiners assigned to tetrahedra are shown to be unique. They can be expanded into three-valent ones, where one requires that the intermediate representations are simple again. This splitting is not unique; however all possible recoupling schemes are related to one another by fusion, where the fusion coefficients are given by $\{6j\}$ symbols. Following this general idea, we define the following spin net tensor implementing the BC conditions on the representations:

$$T_{\mathrm{BC}}(\{j\}, \{m^{\pm}\}, \{n^{\pm}\}) = c_{\{j_i\}}$$


. (17)

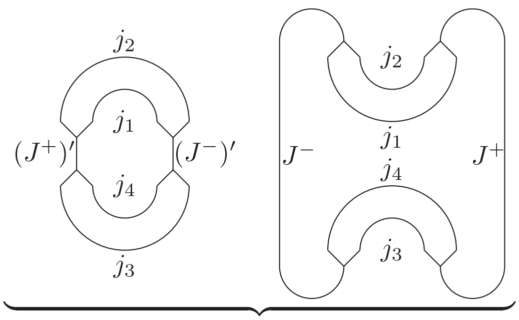
The first diagram represents the representation j_i, the second one their duals j_i^*. The normalization constant $c_{\{j_i\}}$ turns out to only depend on the quantum dimension of the representations $\{j_i\}$. In the graphical calculus the half-circle represents a Clebsch-Gordan coefficient for representations (j_i^+, j_i^-) coupling to $j = 0$, i.e. ${}_q C^{j_i^+ j_i^- 0}_{m_i^+ m_i^- 0}$. This is only nonvanishing if $j_i^+ = j_i^- =: j_i$.

Note that the tensor T_{BC} generically is not a projector onto the invariant subspace given by the basis in (15). In order to define the spin net one thus has to project it down onto the invariant subspace. To this end we contract both the sets of magnetic indices of T_{BC} with the Haar projector $\mathcal{P}$ for $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$,

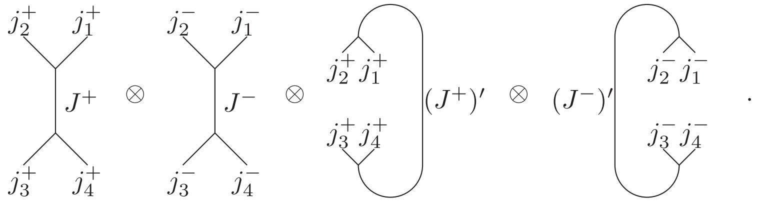
[12] This is also the reason that taking topological fixed points of intertwiner models results in a flow of the spin net model under coarse graining.

[13] This condition is inferred from the classical condition on the bivectors stating that self-dual and antiself-dual parts have the same norm.

$$\mathcal{P} \circ T_{\mathrm{BC}} \circ \mathcal{P} = \sum_{\{J\}} c_{\{j_i\}}\, d_{J^+} d_{J^-} d_{(J^+)'} d_{(J^-)'} \tag{18}$$



$$= \frac{\delta_{J^+,J^-}\delta_{(J^+)',(J^-)'}}{d_J d_{J'}} \prod_{i=1}^4 \delta_{j_i^+,j_i^-}$$



The diagrams are straightforwardly calculated by using several identities and orthogonality relations of the Clebsch-Gordan coefficients [52]. As can be read off from the result T_{BC} precisely implements the Barrett-Crane conditions on the representations.

The only component left to define $\hat{T}_{\mathrm{BC}}$ is the normalization constant $c_{\{j_i\}}$. This can be fixed by requiring the projector condition [82]; see Appendix D for a derivation,

$$T_{\mathrm{BC}} \circ T_{\mathrm{BC}} \overset{!}{=} T_{\mathrm{BC}} \Rightarrow c_{j_1,j_2,j_3,j_4} = (d_{j_1} d_{j_2} d_{j_3} d_{j_4})^{-1}. \tag{19}$$

We however introduce a free parameter in this normalization, allowing a power α, instead of just (-1). The reason is that this normalization determines the (path integral) measure in spin foams. Different principles have been suggested to fix this measure [26,82–84], leading to different proposals. There is however one very strong requirement which is expected to give a unique answer, namely to ensure a restoration of diffeomorphism invariance and triangulation independence in the continuum limit [11,12,85]. This principle does in fact fix the measure in three-dimensional Regge calculus uniquely [13]. For four-dimensional Regge calculus one can show that there is no local measure satisfying this requirement [86], which again emphasizes the need to study which measures could lead to a diffeomorphism invariant model via coarse graining. Thus it is important to allow for some freedom of choice in the initial measure, as this can also determine the phase the models are flowing to. (In [45] the measure is the only free parameter and its tuning does indeed indicate a phase transition.)

To conclude the construction, we obtain the following initial tensor $\hat{T}_{\mathrm{BC}}$ in block-diagonal form:

$$\hat{T}^{\{J\}}_{\mathrm{BC}}(\{j_i\}) = (d_{j_1} d_{j_2} d_{j_3} d_{j_4})^{\alpha} (d_J d_{J'})^{-1} \times \delta_{J^+,J^-} \delta_{(J^+)',(J^-)'} \prod_{i=1}^4 \delta_{j_i^+,j_i^-}. \tag{20}$$

For the triangular algorithm, which we are using in this work, one rather has to define a three-valent tensor from the four-valent one. This is given by

$$\hat{S}^{\{J\}}_{\mathrm{BC}}(\{j_i\}) = (d_{j_1} d_{j_2})^{\alpha} (d_J d_{J'})^{-1} \times \delta_{J^+,J^-} \delta_{(J^+)',(J^-)'} \prod_{i=1}^4 \delta_{j_i^+,j_i^-}. \tag{21}$$

Note that in the triangular algorithm, the indices $\{J\}$ do not have the interpretation of an intermediate label, but rather are the irreducible representations assigned to a coarser edge of the tensor (obtained from splitting a square along its diagonal).

This concludes the construction of the BC model. In the next section we present the construction of the EPRL model, which is based on very different concepts and is more elaborate.

2. The EPRL construction

Originally the EPRL model was motivated as a modification of the BC model, in particular in the imposition of the simplicity constraints onto discretized and quantized BF theory [40,41,46]. These constraints do not form a closed algebra, more precisely the off-diagonal constraints are second class. Imposing them strongly, as it is done in the BC model, might therefore restrict the degrees of freedom of the model more than in the classical theory. It is frequently argued that the uniqueness of the BC intertwiner supports this reasoning, as the intertwiner degrees of freedom are completely constrained.

The EPRL model lifts this issue by imposing the constraints weakly, that is, not as an operator equation but at the level of expectation values, e.g. by a Gupta-Bleuler criterion. In this article, we do not use the original derivation, but rather follow the more recent and straightforward method which was motivated by the closely related FK model [42]. Instead of imposing the quadratic simplicity constraints, which can be shown to reduce BF theory to general relativity, one imposes so-called linear simplicity constraints. In the classical and discrete setting imposing these linear constraints on each face of a triangulation is actually equivalent to imposing the quadratic simplicity constraints. Also the linear constraints are second class; thus they are also imposed weakly. One possible solution, in the Gupta-Bleuler condition, results in a condition on the $\mathrm{SU}(2)\times\mathrm{SU}(2)$ representations (j^+,j^-). In the rest of this article we assume the Barbero-Immirzi parameter $\gamma<1$,

$$(j^+,j^-) \coloneqq \left(\frac{1+\gamma}{2}l, \frac{1-\gamma}{2}l\right). \tag{22}$$

l denotes another SU(2) representation. Given this relation of SU(2) representations to $\mathrm{SU}(2)\times\mathrm{SU}(2)$ representations labeled by γ one defines a map $Y_\gamma : \mathcal{H}_{(1+\gamma)l/2,(1-\gamma)l/2} \to \mathcal{H}_l$ relating the respective Hilbert spaces. Essentially this map restricts the representations (j^+,j^-) to those compatible with the simplicity constraints.

Similarly one also constructs $\mathrm{SU}(2)\times\mathrm{SU}(2)$ intertwiners from SU(2) ones. First one maps all SU(2) representations to $\mathrm{SU}(2)\times\mathrm{SU}(2)$; however the resulting vector is not necessarily an intertwiner, i.e. it does not lie in the invariant subspace. Thus one has to contract this object again with the Haar projectors from both sides in order to obtain an $\mathrm{SU}(2)\times\mathrm{SU}(2)$ intertwiner. Note that this map between invariant subspaces is not an isometry [87], that is, the norm of the intertwiners is not preserved under this map. Nevertheless, due to this construction of the EPRL model its boundary Hilbert space is actually isomorphic to the (kinematical) Hilbert space of loop quantum gravity (for a fixed graph). Thus the EPRL model lifts the second shortcoming of the BC model as it can be used to define transition amplitudes for states of loop quantum gravity.

In the construction of the EPRL spin nets we essentially follow the same route outlined in the previous two paragraphs. First we consider a map from $\mathrm{SU}(2)_k$ representations to $\mathrm{SU}(2)_k\times\mathrm{SU}(2)_k$ ones implementing simplicity constraints, where the maximum spin j_{max} of the quantum group requires particular care. Then we lift the Haar projector of $\mathrm{SU}(2)_k$ to a $\mathrm{SU}(2)_k\times\mathrm{SU}(2)_k$ representation theoretic object, which we denote as the EPRL tensor T_{EPRL}. As this generically is not a projector onto the $\mathrm{SU}(2)_k\times\mathrm{SU}(2)_k$ invariant subspace, it is then contracted by Haar projectors of $\mathrm{SU}(2)_k\times\mathrm{SU}(2)_k$. To put it in a nutshell we essentially restrict the model to projectors that can arise from the $\mathrm{SU}(2)_k$ Haar projector given the map implementing the simplicity constraints.

The spirit of the construction is very similar to the corresponding spin foam model as defined by Meusburger and Fairbairn [69]. We work here with the Euclidean version; Lorentzian spin foam vertex amplitudes have been also constructed by [69] and [70].

As already mentioned above, the map (22) from $l\to(j^+(l),j^-(l))$ requires some attention in the Euclidean theory: Both the representations l and j^+, j^- must be $\frac{1}{2}\mathbb{N}$. If this is not the case, the particular mapping is forbidden, i.e. will be assigned a vanishing weight. As we usually start from a representation l, this gives restrictions on the Barbero-Immirzi parameter γ in order to obtain a nontrivial map, i.e. beyond just mapping the trivial representations to one another. Thus one quickly realizes that $\gamma\in\mathbb{Q}$ is the necessary condition to do so. Again this is a particular condition on the Euclidean theory; a similar restriction does not exist for the Lorentzian one. In the case we are considering here, there is a further restriction, as we only consider integer representations. On a more technical level, one can understand this identification as a map from $V_l\to V_{j^+}\otimes V_{j^-}$, so essentially a Clebsch-Gordan coefficient. Thus the coupling rules of SU(2) also influence whether a nontrivial map exists.

In the case of quantum groups further restrictions occur, as it has been already studied in [69] (see also [70] for an independent derivation of the Lorentzian model) in the case $\gamma<1$. As discussed above $\mathrm{SU}(2)_k$ (at root of unity) has a natural cutoff on the spins, $j_{\mathrm{max}}=\frac{k}{2}$. Representations labeled by larger $\frac{1}{2}\mathbb{N}$ exist, but are referred to as having vanishing quantum dimension.[14] The $\mathrm{SU}(2)_k$ spin nets in [14] have been explicitly constructed to avoid these representations; thus we have to ensure that no allowed spin l gets mapped to such a representation. Similar to [69], we achieve this by requiring

$$(j^+(j_{\mathrm{max}}),j^-(j_{\mathrm{max}})) = \left(\frac{1+\gamma}{2}j_{\mathrm{max}}, \frac{1-\gamma}{2}j_{\mathrm{max}}\right) \in \{(j,j')\in(\mathbb{N},\mathbb{N}) : j,j'\le j_{\mathrm{max}}\}. \tag{23}$$

Again this puts many restrictions on the possible choices of γ, in particular for small levels k of the quantum group. In many cases only the trivial map exists. The following nontrivial cases are possible (we omit the trivial identification):

(i) $k=6$ $(j_{\mathrm{max}}=3)$ for $\gamma=\frac{1}{3}$: $l=3\mapsto(j^+=2,j^-=1)$.

(ii) $k=10$ $(j_{\mathrm{max}}=5)$ for $\gamma=\frac{3}{5}$: $l=5\mapsto(j^+=4,j^-=1)$.

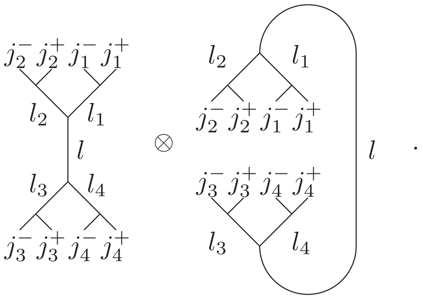
[14] For $j>j_{\mathrm{max}}$ the quantum dimension $d_j=[2j+1]_q$ is no longer positive definite.

(iii) $k=12$ ($j_{\max}=6$) for $\gamma=\frac{1}{3}$: $l=3 \mapsto (j^+=2, j^-=1)$ and $l=6 \to (j^+=4, j^-=2)$.

As we argue below, due to freedom in the normalization as in the BC case, the model for $k=12$ is the most interesting one, as it will actually be a whole one-parameter family of models. However, at least for spin nets, $k=12$ is currently beyond efficient simulation, despite the optimization efforts described in Secs. II A and II B. Nevertheless, the associated intertwiner model can be studied without problems.

Concretely for spin nets, we first construct the EPRL tensor T_{EPRL} from the $\mathrm{SU}(2)_k$ Haar projector,

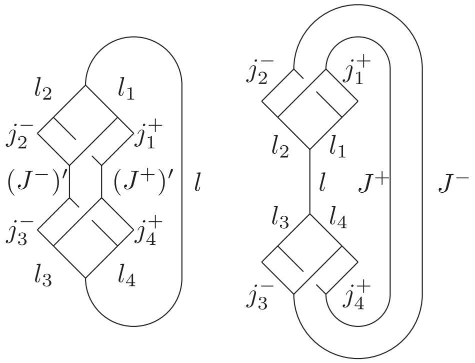
$$T_{\mathrm{EPRL}}(\{j\},\{m\},\{n\}) := \sum_l c_{\{l\}} \qquad (24)$$

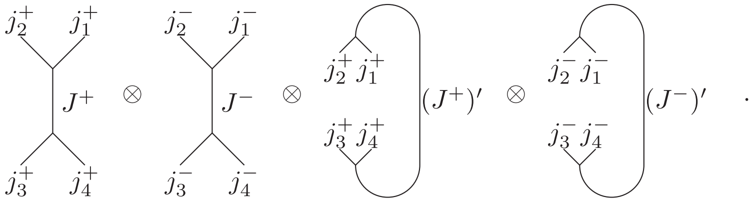

j_2^- j_2^+ j_1^- j_1^+
l_2
l_1
l
l_3
l_4
j_3^- j_3^+ j_4^- j_4^+
⊗
l_2
l_1
j_2^- j_2^+ j_1^- j_1^+
l
j_3^- j_3^+ j_4^- j_4^+
l_3
l_4
.

Note again that $j_i^\pm = (1\pm\gamma)l_i/2$. $c_{\{l\}}$ is the normalization constant. Also it is important that both copies of $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ representations are generated from the same $\mathrm{SU}(2)_k$ intertwiner [and thus identical, i.e. $(j_i^\pm)' = (j_i^\pm)^*$]. If the latter were independent, i.e. replace the second l by l', this would result in a different, factorizing model.

As for the BC case, T_{EPRL} is not a projector onto the invariant subspace; therefore it has to be contracted with the Haar projector from both sides,

$$\mathcal{P} \circ T_{\mathrm{EPRL}} \circ \mathcal{P} = \sum_{\{J\}} c_{\{j_i\}}\, d_{J^+} d_{J^-} d_{(J^+)'} d_{(J^-)'} \qquad (25)$$



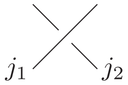


A comment on the choice of Haar projector is in order. In contrast to the BC model, we have slightly changed the Haar projector,[15] which allows us to simplify the diagrams.

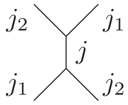
Another peculiarity of quantum groups is the over- and undercrossings of representations, which one has to keep track of since they do not commute. They can be transferred into one another employing the so-called $\mathcal{R}$ matrix [52,57],

[15] The $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ Haar projector used for the BC model is a tensor product $\mathcal{P}_q \otimes \mathcal{P}_{\bar{q}}$, whereas the one for the EPRL is $\mathcal{P}_q \otimes \mathcal{P}_q$.

$$\mathcal{R} =$$

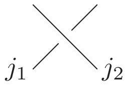


$$= \sum_j d_j \; q^{-\frac{1}{2}(j_1(j_1+1)+j_2(j_2+1)-j(j+1))}$$

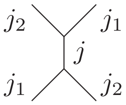


$$, \qquad (26)$$

$$\mathcal{R}^{-1} =$$



$$= \sum_j d_j \; q^{\frac{1}{2}(j_1(j_1+1)+j_2(j_2+1)-j(j+1))}$$

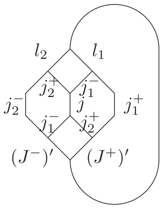


$$.$$

Using these identities we can replace the crossings in the diagrams. Furthermore the diagrams can be manipulated further using identities derived in [14] (see also Appendix B). Eventually we can define the block-diagonal form of T_{EPRL}, namely $\hat{T}_{\mathrm{EPRL}}$. Here we restrict ourselves to the triangular version $\hat{S}_{\mathrm{EPRL}}$. See Appendixes E and F for derivations of the normalization and the diagrams respectively,

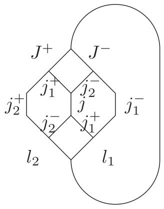
$$\hat{S}^{\{J\}}_{\mathrm{EPRL}}(\{j_i\}) = (-1)^{j_1^+ + j_1^- + j_2^+ + j_2^- - l}\,(d_{l_1} d_{l_2})^{\alpha}\, d_l$$

$$\times \left[\sum_j d_j\, q^{-\frac{1}{2}(j_2^+(j_2^+ +1) + j_1^-(j_1^- +1) - j(j+1))} \right.$$



$$\left. \vphantom{\sum_j} \right]$$

$$\times \left[\sum_j d_j\, q^{-\frac{1}{2}(j_1^+(j_1^+ +1) + j_2^-(j_2^- +1) - j(j+1))} \right.$$



$$\left. \vphantom{\sum_j} \right]. \qquad (27)$$

The normalization constant $c_{\{l_i\}}$ is again computed by contracting T_{EPRL} with itself, and as in the BC case we introduce more freedom by allowing it to appear with a power α. Note that the normalization only depends on the quantum dimensions of the $\mathrm{SU}(2)_k$ representations l_i. Therefore α only plays a role if there exists a nontrivial map $l_i \mapsto (j^+, j^-)$ for $l_i \neq 0$, $j_{\max}$, as $d_0 = d_{j_{\max}} = 1$. The smallest k for which this is possible (in the model discussed here) is $k = 12$. Even though the diagrams turn out to be nicely symmetric, we have not found a simpler expression for them. Interestingly both diagrams turn out to give the same expression.

Before we continue with the discussion of the related intertwiner models, we comment on the simplicity constraints. As discussed above, these are implemented at the level of representations, here explicitly in the maps $l_i \mapsto (j_i^+, j_i^-)$. In the diagrams we then observe that $j_1^\pm$ and $j_2^\pm$ couple to $J^\pm$, which are then again coupled to l. Note that the simplicity constraints are not explicitly implemented in the latter coupling, such that we expect a flow of the simplicity constraints. As one would interpret the theory at a coarser scale as an effective theory of the finer one, we *a priori* do not see a reason to enforce the constraints there, too.

In the next section we discuss the respective intertwiner models and their behavior under coarse graining.

C. Intertwiner models

As already discussed above intertwiner models can be motivated from spin nets as simpler versions of those. In fact one can interpret the latter as the tensor product of two interacting intertwiner models; thus one could also denote them as entangled.[16] This insight in itself is already helpful in interpreting the fixed point of spin net models, as they often turn out to be factorizing.

Nevertheless the study of intertwiner models themselves is already interesting in itself as their construction is analogous to spin nets, such that coarse graining them gives us first hints and insights into the behavior of the simplicity constraints under coarse graining with much lower computational costs. The latter is crucial for studying the EPRL model, as the interesting case, i.e. $k = 12$, for spin net models is currently out of reach.

[16] This is in full analogy to quantum information where two subsystems are entangled when they cannot be written as a product state, e.g. the Bell states.

Therefore in the next two subsections we very briefly introduce the respective BC and EPRL intertwiner models and discuss the results under coarse graining.

1. BC intertwiner model

For completeness, let us give the initial three-valent tensor (in block-diagonal form) for BC intertwiner models, which can be straightforwardly defined by omitting the dual representations [compare also with (21)],

$$\hat{S}^{\{J\}}_{\mathrm{BC}}(\{j_i\}) = (d_{j_1} d_{j_2})^{\alpha} (d_J)^{-1} \delta_{J^+,J^-} \prod_{i=1}^{4} \delta_{j_i^+,j_i^-}. \quad (28)$$

As before we keep the factor α as a means to study different models. It reflects the fact that the normalization is not uniquely defined, as it is also the case for edge and face amplitudes in spin foam models.

Under coarse graining we find a very simple pattern valid for all levels k of quantum groups: the first important fact is that the BC intertwiner is a fixed point of the renormalization group flow for $\alpha_c = \frac{1}{2}$. This is not surprising as the original four-valent BC intertwiner [39] is unique, i.e. it does not depend on the recoupling scheme chosen. On that specific fixed point the model thus is discretization independent.

If we consider $\alpha \neq \alpha_c$ we do observe the following behavior: As the model is not on the fixed point, we observe that channels other than the BC ones, i.e. $J^+ \neq J^-$, get excited, that is come associated with a nonvanishing singular value. This signifies a weakening of the simplicity constraints under the coarse graining flow. As we see below the BC simplicity conditions are however restored at the fixed points.

If one orders the singular values into a matrix $J^+ = J^-$ give the diagonal elements; $J^+ \neq J^-$ are off-diagonal elements. Crucially this matrix is symmetric, that is, the model is invariant under exchanging J^+ and J^-. Eventually the models flow back to only BC channels and converge to one of two different fixed points. One of them is again the usual BC fixed point (for $\alpha = \frac{1}{2}$), that is, all channels $J^+ = J^-$ are equally excited with $J^\pm \in \{0, 1, \ldots, j_{\max}\}$. The model flows back to this fixed point for roughly all $\alpha > 0$.

For $\alpha < 0$ we observe a flow to a different fixed point. Again only channels $J^+ = J^-$ are allowed; however only $J^\pm \in \{0, j_{\max}\}$ for k even and $J^\pm = 0$ for k odd. (For odd k the maximal representation is half integer, which we have excluded.) This new fixed point, which is similar to the Ashtekar-Lewandowski vacuum in LQG, also exists in the initial model if $\alpha \to -\infty$. Then all representations $j \neq 0, j_{\max}$ in (28) get suppressed as they possess a quantum dimension $d_j > 1$. Due to the coupling rules of $\mathrm{SU}(2)_k$ any combinations of the trivial and the maximal representation can only couple to either the trivial or the maximal representation.

Thus, to sum up, we find that the BC intertwiner model has a simple phase structure valid for all levels k of the quantum group. There exist two (attractive) fixed points of the renormalization group flow, both compatible with the BC condition $J^+ = J^-$. The BC intertwiner, given for $\alpha = \frac{1}{2}$, is the fixed point allowing all representations, whereas the other fixed point (for $\alpha \to -\infty$) is similar to the Ashtekar-Lewandowski vacuum of LQG, where only the trivial and the maximal representations are allowed. That the maximal representation remains is a peculiarity of the quantum group as it also has a quantum dimension of 1 as the trivial one. Thus we expect that only the trivial one remains in the limit towards SU(2). For any other value of α the system initially flows away from the BC condition before eventually converging to one of the two fixed points. It appears that this condition is very strictly implemented and deviations from it are possible, but are dynamically disfavored. Therefore it is interesting to study the BC spin net model under similar aspects.

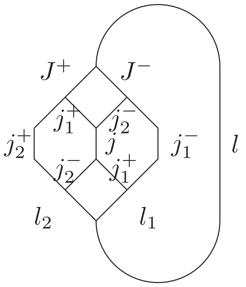
In the next subsection we study the EPRL intertwiner model in detail.

2. EPRL intertwiner model

Before we discuss the behavior of the EPRL model under coarse graining, let us present the initial three-valent tensor [again compare with (27)],

$$\hat{S}^{\{J\}}_{\mathrm{EPRL}}(\{j_i\}) = (-1)^{j_1^+ + j_1^- + j_2^+ + j_2^- - l} \, (d_{l_1} d_{l_2})^{\alpha} \, \sqrt{d_l}$$

$$\times \Bigg[\sum_j d_j \, q^{-\frac{1}{2}(j_1^+(j_1^+ + 1) + j_2^-(j_2^- + 1) - j(j+1))}$$



$$\Bigg]. \quad (29)$$

Compared to the BC model this expression is clearly more complicated and involved, which is reflected in the flow under coarse graining. Again we keep the parameter α free in order to keep a freedom in the normalization choice. As discussed above this is actually only relevant when the map from $\mathrm{SU}(2)_k$ representations to the $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ ones is nontrivial, i.e. there exists an l different from $j = 0$ or $j = j_{\max}$ that gets mapped to an $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ representation. This is the case for $k = 12$; thus we focus on this one in the following.

Before we start with the discussion of the flow under coarse graining it is instructive to focus first on the diagram in the definition of the model. We see that the two representations J^+ and J^- couple to l; however this coupling does not need to fulfil the conditions of the simplicity constraints. Thus, if one considers the coupling rules of $\mathrm{SU}(2)_k$, one quickly realizes that channels (J^+, J^-) are allowed that are initially not part of the conditions given by the simplicity constraints. These include for example several diagonal channels (with $J^+ = J^-$). Therefore it is clear that this EPRL intertwiner model immediately flows away from the original one for any value of α. Note that this is the first clear difference to the BC model, where the BC simplicity condition $J^+ = J^-$ automatically followed from recoupling theory and thus remained valid at least in the first iteration. In this regard the EPRL model appears to restrict the intertwiner degrees of freedom less. Moreover, the symmetry of the BC model under exchange of J^+ and J^- does not exist in the EPRL one by construction.

Indeed under coarse graining we observe an almost orthogonal behavior of the EPRL model compared to the BC model. The truncation scheme we have used in this article, see also Sec. II B, is taking over the most relevant (effective) degree of freedom per block $\{J^\pm\}$. While this scheme works very well for the BC model, as there usually one degree of freedom is clearly the most important one (given by the size of its singular value), this does not hold any more for the EPRL model. Already from the first iteration, one finds that the several effective degrees of freedom per block are too relevant to be truncated, that is, are not negligible in reference to the most important one. In subsequent iterations this gets even more pronounced for more of the channels. Therefore in order to study the system in more detail one should employ a more elaborate truncation scheme. However we refrain from doing so for three reasons: first our main goal is to study the respective spin net models, for which simply increasing the number of degrees of freedom is out of reach (for the interesting models). The second reason concerns the increasing number of degrees of freedom that are too relevant to be truncated: previously we have observed this only close to (arguably) second order phase transitions, which are indicated by an (almost) scale invariance. We do not find such an indication here. Although this might be due to the truncation scheme, it hints at the possibility of a very intricate behavior of the simplicity constraints. This brings us to the third reason, which is the need to have a geometric interpretation of the additional degrees of freedom appearing in each block. With the truncation of one singular value per block the models also preserved part of their initial form, thus allowing us to retain the original geometrical interpretation. This would change with a more complicated truncation scheme, which also should be accompanied with a geometric understanding of the emerging effective degrees of freedom. Nevertheless it would be more promising to study this model with an algorithm more suited for cases where the number of relevant degrees of freedom increases [32], in order to better understand the origin of these degrees of freedom.

Despite these shortcomings we still report on some of the observations that we have made under renormalization. Of course these results should be taken with a grain of salt as they might change under a more accurate algorithm. One of the first observations is that phases are very difficult to identify as in many cases the model does not converge to a fixed point, but rather oscillates. This behavior is most likely related to the low cutoff per block, such that it can happen that two degrees of freedom in the same block $\{J\}$ change their significance, leading to a different new tensor and subsequent flow. Nevertheless, for $\alpha < 0.3$ we observe a flow of the model towards a phase similar to the Ashtekar-Lewandowski vacuum of LQG, where only $(J^+, J^-) = (0, 0)$ is allowed.

For (roughly) $\alpha > 0.3$ we observe another peculiarity that we have never encountered in related models so far. Usually the trivial representation, here with the channel $(J^+, J^-) = (0, 0)$ always appears as the most important one, i.e. with the largest singular value, such that it is suitable to normalize the tensor with respect to it. However for $\alpha > 0.3$ other representations become more relevant than the trivial one, i.e. their associated singular value is larger. That is unexpected as inevitably all degrees of freedom couple to the trivial representation again. Moreover, if we increase α further to roughly $\alpha > 1$ we also observe that some of the singular values associated to nontrivial representations increase over several iterations and appear to diverge, e.g. are several orders of magnitude larger than the one associated to the trivial representation. The analysis is made difficult also by the oscillation of the singular values as no clear pattern can be uncovered under coarse graining, such that changing the normalization is not straightforward as well. One example for this is again the behavior for $\alpha > 1$, where we observe some singular values increasing rapidly iteration after iteration before suddenly dropping to values smaller than the trivial representation before converging to the Ashtekar-Lewandowski vacuum. We again attribute these features to the cutoff scheme. In Fig. 7 we plot the singular values for three different values of α to illustrate the behavior of the model in the three different regimes described above.

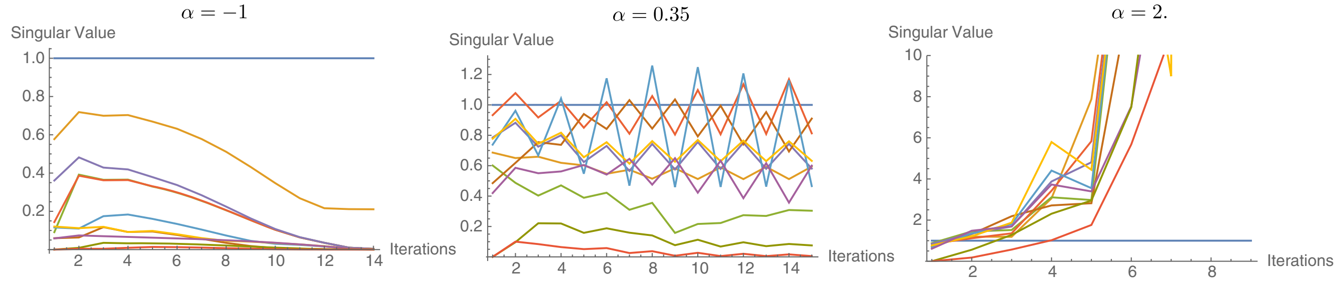


FIG. 7. Plots of singular values for the EPRL-intertwiner model for $k = 12$: We plot the flow of singular values for three different initial values of α from the regions mentioned in the main text. For $\alpha = -1$, we observe a convergence to the Ashtekar-Lewandowski phase, where only $(J^+, J^-) = (0, 0)$ is allowed. For the other values the behavior is ambiguous, as we observe oscillations (for $\alpha = 0.35$ and $\alpha = 2$) and diverging singular values for $\alpha = 2$. The singular values are normalized with respect to channel $(J^+, J^-) = (0, 0)$. We plot only a selection of values, namely all diagonal ones with $J^+ = J^-$, as well as $(J^+, J^-) = (1, 2), (2, 1), (2, 4), (4, 2)$, in which the EPRL model is explicitly asymmetric [(2,1) and (4,2) are initially allowed]. As we cannot identify interesting phases, we refrain from labeling the singular values.

One reason for the rather irregular behavior of the EPRL model might be due to the few spin values that are allowed in the initial model, which are moreover asymmetric under exchange of j^+ and j^-. For the case $k = 12$ with maximal spin $j^{\pm}_{\max} = 6$ (and the Barbero-Immirzi parameter chosen as $\gamma = \frac{1}{3}$) these include besides the trivial spin assignment only the two cases $(j^+, j^-) = (2, 1)$ and $(j^+, j^-) = (4, 2)$. Compare this with the BC model (for finite α) in which all configurations satisfying $j^+ = j^-$ appear. We expect that this might actually hinder the Euclidean EPRL-Λ models, as defined in their current form [69], displaying a suitable continuum limit, at least for large cosmological constant (correspondingly small k). For larger k this issue should be attenuated as more configurations become allowed and more choices for the Immirzi parameter lead to nontrivial configurations. In the Lorentzian case one has *a priori* infinitely many representations appearing; however the imposition of the (EPRL) simplicity constraints does lead to a cutoff in the spins. That is, also in this case the combination of using a quantum group and implementing a Barbero-Immirzi parameter suppresses (infinitely) many spin combinations.

Clearly one should not interpret too much of results obtained from coarse graining the EPRL intertwiner model, due to the reasons mentioned above. Nevertheless we have reported on several qualitative features which are strikingly different from the BC model or other models studied so far, namely the quickly growing number of relevant degrees of freedom for any parameter of the model, that is, without a sign of a nearby phase transition. Thus it is apparent that the EPRL construction allows for a far more intricate dynamics than the BC one, which is difficult to characterize yet. The dynamics of the model has to be studied in more detail, which requires new tools, possibly combining both analytical and numerical methods.

Let us further remark that the fact that the BC intertwiner model allows for a fixed point (for $\alpha = 1/2$), that is, defines a triangulation invariant two-dimensional model, is crucial for the arguments that were invoked to show uniqueness of the model in [88]; see also the discussion in [14]. In contrast we have not found a corresponding topological model that would be triangulation invariant and originates from the EPRL intertwiner models by coarse graining. Again one reason seems to be the restrictions on the allowed spins imposed by the Barbero-Immirzi parameter. Also our coarse graining method is not sufficient to capture the phase structure, that is, find *local*, triangulation invariant models, to which the initial models flow under coarse graining. It might however be that the fixed points for the EPRL model feature nonlocal amplitudes, which in particular applies to second order phase transitions [86,89].

This concludes the section on intertwiner models. In the next section we move on towards coarse graining spin net models.

D. BC spin net models

In this section we present the results obtained by applying the tensor network renormalization algorithm introduced and thoroughly discussed in Secs. II A and II B to BC spin nets constructed in Sec. III B. Note that the details of the algorithm are not vital for understanding the results.

Before doing this however we mention again why we do not consider the EPRL spin net model here as well. On the technical side this is due to the size of the tensors for $k = 12$, the first nontrivial model we can study. With all the optimizations developed and implemented, the largest obstacle remaining is the size of the largest block $\{J\}$ of the tensor. For $k = 12$ this block consists of 25^8 entries, which equals roughly 2.3 TByte of memory usage. Moreover as we have already observed for the intertwiner models, a more accurate cutoff scheme, which further increases the memory usage, as well as an algorithm better suited to dealing with an increasing number of relevant degrees of freedom [32] are necessary. Therefore we do not expect reliable results and thus leave this question for future research.

Returning to the BC model, the initial tensor is given in (21) and is of a very simple form. The thoughtful reader might wonder why we expect an interesting flow of this tensor under coarse graining when it essentially consists of two BC intertwiners, which have been previously identified (individually) as fixed points of the renormalization procedure. As we have already commented before the two intertwiners making up the projector sitting at the vertices of the spin net are not independent, but depend on the same labels. In particular the second copy of representations satisfies $(j_i^\pm)' = (j_i^\pm)^*$. Due to the particular conditions of the BC model this actually also translates to the new effective labels $J^\pm$. The two intertwiners are only completely independent if all $(j_i^\pm)'$ are completely unrelated to the $j_i^\pm$, such that the model could be written exactly as a tensor product of two intertwiners. As the renormalization algorithm does not mix the two copies of the representations, i.e. $\{j\}$ and $\{j'\}$, they stay independent during the entire coarse graining process. Thus tensor products of fixed points are indeed fixed points of the renormalization group flow. Using the same terminology, the intertwiner degrees of freedom in the BC spin net are interacting; one could say the intertwiners are entangled (as they do not factorize). As a result one observes a nontrivial renormalization group flow (at least initially) beyond factorizing models.

In this article we do not consider only one BC spin net model, but a one-parameter family described by the parameter α, which appears as the power of the normalization constant. Again this relates to the ambiguities in the choice of face and edge amplitudes in spin foam models. From the intertwiner models in Sec. III C we have already observed that α influences the model quite significantly: if $\alpha < 0$ it suppresses representations j in the model with quantum dimension $d_j > 1$, which is all other than the trivial and maximal one. For $\alpha > 0$ it conversely emphasizes the representations. We see that this also affects the BC spin net model.

A further comment on the simplicity constraints is in order: as discussed thoroughly above, the simplicity constraints in the BC model are essentially imposed by requiring that all $j_i^+ = j_i^-$. This holds also for the new edge labels $J^+ = J^-$ introduced in the first coarse graining iteration; however, as we have already seen for the intertwiner models away from one of its two fixed points, this condition gets violated dynamically under coarse graining. Thus we also expect this for spin nets, as long as the initial model is not a fixed point.

This immediately leads us to one of the first observations: the BC spin net model is not a fixed point of the renormalization group flow for any finite value of α; it is only a fixed point if one sends $\alpha \to -\infty$, such that only the trivial and maximal representation are allowed. Similar to the intertwiner model we find at least two (extended) phases, i.e. fixed points the model converges to, for each level k of the quantum group. Additionally we find up to two more phases for particular levels k.

The phases/fixed points of the model are characterized by the singular values assigned to the intertwiner channels $\{J\}$ and are best organized in a matrix, with (J^+, J^-) and $((J^+)', (J^-)')$ denoting rows and columns respectively.[17] Actually this matrix is (and stays) symmetric with respect to both diagonals, i.e. under exchanging both sets of representations and also under exchanging J^+ and J^-.

Let us begin with the two phases found for any level k of the quantum group, which are quite similar to the ones of the intertwiner model. Actually one of the fixed points is precisely a tensor product of two BC intertwiners.

(i) The first phase one finds for very small α is the phase in which only the trivial $(j^+, j^-) = (0,0)$ and the maximal representation $(j^+, j^-) = (j_{\max}, j_{\max})$ (for even k) are allowed. As already discussed before, this is very similar to the Ashtekar-Lewandowski vacuum of LQG. This phase itself is not factorizing as it actually requires that $J = J'$. The fixed point is summarized in the following matrix (for even k):

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & \dots & & 0 & 1 \end{pmatrix}. \tag{30}$$

(ii) The second phase we find appears for larger α and we call it the factorizing BC phase. It is characterized by allowing all representations $(J^+, J^-) = (J, J)$, where both copies of representations, i.e. $J^\pm$ and $(J^\pm)'$, are independent of one another. Summarized in a matrix of singular values this looks as follows (for $k = 4$, $j_{\max} = 2$):

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \vdots & & \vdots & & \vdots & & \vdots & 0 \\ 0 & & & & 0 & & & & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & & & & 0 & & & & 0 \\ 0 & & & & 0 & & & & 0 \\ 0 & & & & 0 & & & & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{31}$$

[17] The ordering of (J^+, J^-) in the (rows and columns of the) matrix is as follows: $(J^+, J^-) = (0,0)$, $(0,1)$, $\dots(0, j_{\max})$, (1,0), $(1,1)$, $\dots(j_{\max}, j_{\max} - 1)$, $(j_{\max}, j_{\max})$.

As this model is factorizing, this matrix can be written as a tensor product of two vectors, one for (J^+, J^-) and one for $((J^+)', (J^-)')$, which have entries 1 if $J^+ = J^-$ and 0 otherwise. These exactly characterize the original BC intertwiner fixed point.

For most levels k of the quantum group we have studied we only find these two extended phases, that is, both fixed points are attractive. In detail these were the levels $k = 5$, $k = 6$ and $k = 7$. The parameter α at which the transition occurs varies for these three models; we summarize these values in Fig. 8. Moreover if we tune the system towards the phase transition we do not observe an (almost) scale invariance, that is, an increase in the number of iterations that the models need to flow to a fixed point. Instead the system rather quickly flows to one or the other fixed point—the nonvanishing singular values specifying that the phase has converged after roughly ten to fifteen iterations. Thus it is unlikely that these phase transitions are of second order.

There exist two models that possess a more interesting phase structure, namely for $k = 4$ and $k = 8$. There we find the following two phases.

(i) The first phase only occurs for the model $k = 4$ and appears in between the Ashtekar-Lewandowski and the factorizing BC phase. It is summarized best in a matrix,

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{32}$$

As one can see only channels on the main diagonals are allowed, together with the condition that the sum of spins [individually for (J^+, J^-) and $((J^+)', (J^-)')$] must be even. Thus the BC condition is broken on this fixed point. Clearly both copies of representations are not independent and the model is thus not factorizing, without requiring that $J' = J^*$. Interestingly we have not found a similar phase in any other model. Due to the similarities of $\mathrm{SU}(2)_k$ for $k = 4$ restricted to integer representations to the finite group S_3, a similar phase/fixed point might exist for $S_3 \times S_3$ as well.

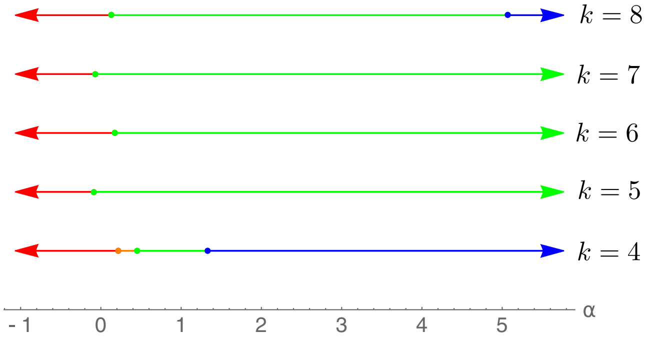


FIG. 8. Phases of BC spin nets: Red indicates the Ashtekar-Lewandowski phase, green the factorizing BC phase and blue the factorizing phase with singular values alternating between 1 and 0. Only for $k = 4$ do we find a nonfactorizing phase, here orange, in between the Ashtekar-Lewandowski and the factorizing BC phase. The arrows indicate that this phase continues up to $\pm\infty$. Across the different levels k, the transition between the Ashtekar-Lewandowski and factorizing BC phase is close to $\alpha = 0$. It appears that the transition from the factorizing BC phase to the factorizing and alternating phase moves to larger α as k is increased; however one would need to study even larger k to confirm that.

(ii) Another phase can be found for $k = 4$ and $k = 8$ for large α, which is also violating the BC condition $J^+ = J^-$. Essentially all channels are allowed as long as the sum of spins for each copy of representations is even. Thus the matrix of singular values is alternating between 0 and 1 in each column and row (see $k = 4$),

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{33}$$

This phase is factorizing yet again, as we can write the matrix as a tensor product of two vectors, whose values alternate between 1 and 0. Thus we have found another fixed point of intertwiner models, to which the BC intertwiner model does not flow.

Due to these additional fixed points the phase structure for $k = 8$ and in particular $k = 4$ is more interesting than for the other models. Again these fixed points are attractive and come with extended phases. We have summarized the values of α at which the transitions occur in Fig. 8. Furthermore in Fig. 9 we plot the flow of singular values for $k = 4$ for four different values of α, one for each phase.

Of course, more phases imply more phase transitions, which might possibly be of second order. Let us first focus on the case $k = 4$, more precisely the transitions between

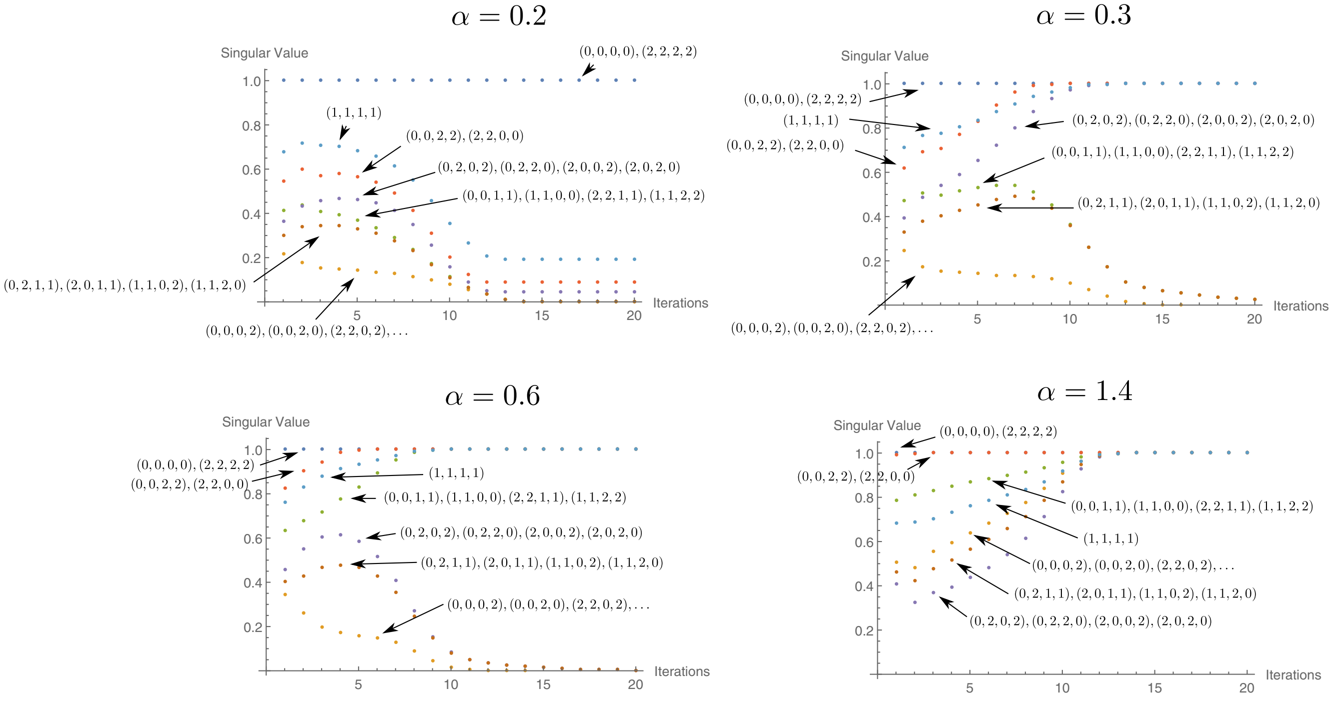


FIG. 9. Plots of singular values [per channel $(j^+, j^-, (j^+)', (j^-)')$] for the $k = 4$ BC model: The plots are for four different choices of α each flowing to a different fixed point characterizing one of the four phases. The color coding of the plots is identical in each figure. Depending on the phase, the singular values of different channels converge to either 0 or 1. (The fact that some singular values do converge to a nonzero value smaller than 1 for $\alpha = 0.2$ is a known, yet unphysical, feature of tensor network algorithms, which is overcome in the recent algorithm by Evenbly and Vidal [32].) Note that we do not plot those singular values that always converge to 0 and thus do not help in differentiating the phases. Note that the convergence happens rather quickly after 10 to 15 iterations, even though some values of α are quite close to phase transitions. This already indicates that the transitions are not of second order.

the only nonfactorizing phase (orange in Fig. 8) we have found.

As we have discussed above in most models there exists a direct transition between the Ashtekar-Lewandowski and the factorizing BC phase, which is not of second order. In the $k = 4$ model, the nonfactorizing phase sits right in between the two above-mentioned phases, thus splitting one phase transition into two. At the transition at lower α, i.e. from the Ashtekar-Lewandowski phase to the nonfactorizing phase, we observe only that the system requires a few more iterations (roughly 20) before converging to its fixed point. However no matter how close we tune α towards the transition, we do not find signs of an almost scale invariance. The situation is also very similar at the new transition to the factorizing BC phase. Thus these two transitions are likely not of second order either.

The last remaining transition, from factorizing BC to the factorizing and alternating model, is present in the models $k = 4$ and $k = 8$. However, as for the other phase transitions we do not find signs that the transition is of second order; in particular we find no (almost) scale invariance the further we tune the model towards the transition.

To sum up the initial BC spin net model, in contrast to the BC intertwiner, is not a fixed point of the renormalization group flow. Thus the model flows under coarse graining, where we find two to four different (extended) phases/fixed points depending on the level k of the quantum group. It appears that we find more phases in the case in which the level k is a multiple of 4, which might be related to the following fact: If it exists, the representation $\frac{j_{\max}}{2} = \frac{k}{4}$ has the maximal quantum dimension (of the quantum group) and allows for the largest number of couplings to other representations. As we restrict our models to integer representations this particular representation only exists for $k = 4l$, $l \in \mathbb{N}$. If we increase α the respective weight of configurations containing $j = \frac{k}{4}$ grows faster than others while this representation furthermore couples to most other representations. We expect this to be the origin of the last factorizing phase the model flows to for large α, yet it seems that α has to be significantly larger for larger levels k of the quantum group.

In fact, if one is looking for fixed points for the intertwiner models, leading to two-dimensional topological models, the models with even levels k and especially levels $k = 4l$, $l \in \mathbb{N}$ play a special role in featuring more fixed points or equivalently phases compared to the cases where k is odd [57]. For these additional phases the condition of even spins also appears. We therefore believe that the additional fixed points we found might appear for larger values of $k = 4l$ as well.

A nontrivial phase structure with multiple phases also implies transitions between these phases. Second order phase transitions are particularly interesting for discrete models as they provide a nontrivial way of taking the continuum limit and obtaining propagating degrees of freedom. A typical sign for such a transition is an almost scale invariance close to the transition. In tensor networks this manifests itself as the tensors remain unchanged for a growing number of iterations the closer the system is tuned towards the transition.[18] Unfortunately we observe no such behavior for BC spin nets at any of the transitions we have found, no matter how close we tune towards the transition. Thus we conclude that the transitions of the BC spin net models are very likely not of second order and therefore do not allow for propagating degrees of freedom.

These results allow us to draw a few tentative conclusions for the BC spin foam model. Of course, the nonexistence of second order phase transitions in BC spin nets does not prove that the same holds for BC spin foams, but it can be taken as an indication. (As mentioned spin nets can be interpreted as spin foams based on a particular 2-complex, which includes only two vertices, but a large number of edges.) The BC model has been criticized as implementing the simplicity constraints too strongly, which could suppress propagating degrees of freedom. The absence of a second order phase transition in the corresponding spin net model can be taken as an indicator that this is indeed the case. Of course this has to be confirmed by studying the BC spin foam model itself.

E. Summary and discussion

In this article we have employed tensor network renormalization to thoroughly study $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ intertwiner and spin net models, which are constructed analogously to four-dimensional BC and EPRL/FK spin foam models. We have illustrated the numerical challenges in examining these models for larger levels k of the quantum group and described how we have overcome these by a three-valent version of familiar tensor network algorithms and further steps that reduced drastically the computational resources required. Let us briefly summarize and discuss the results.

The analogue BC models show an interesting structure. In the intertwiner case we find two fixed points; one is similar to the Asthekar-Lewandowski vacuum in LQG, and the other is the initial BC intertwiner itself. Thus in a certain sense the BC model already implements a version of triangulation independence. For most levels k of the quantum group BC spin nets similarly show two phases, again one of Ashtekar-Lewandowski type. The other is best described as a factorizing implementation of the BC condition on the representations, that is $j^+ = j^-$. In the interpretation of spin nets as a melon spin foam [14], i.e. two spin foam vertices connected by many edges, this implies a decoupling of the two spin foam vertices. Since the simplicity constraints weaken the gluing or coupling of spin foam vertices, this can be interpreted as models in which the simplicity constraints are implemented too strongly.

Notably the initial BC model is not a fixed point of spin net models. For particular levels k we have found up to two *new* phases, one of which is not factorizing. It thus could represent an interesting new phase, where spin foam vertices are not decoupled, but nevertheless implement a version of the simplicity constraints.

None of these fixed points corresponds to BF theory, which implies that the simplicity constraints are implemented strong enough such that the BF symmetry is not recovered under coarse graining. However we have not found any signs indicating that the phase transitions are of second order, which can be taken as an indication that the simplicity constraints are implemented too strongly. This latter conclusion does however depend on how accurately the two-dimensional spin net models mimic the four-dimensional spin foam models: a close relationship holds between two-dimensional spin systems and four-dimensional lattice gauge systems. Spin foams can be understood as generalized gauge systems [33]; the question is therefore whether this relationship survives the generalization. We hope that these questions can be resolved in the near feature by studying the coarse graining flow of four-dimensional spin foam models, e.g. the BC model can be readily studied via Monte Carlo simulations [48–50].

For the EPRL/FK model we have only been able to study intertwiner models, as the restrictions to imposing the simplicity constraints require large levels k for nontrivial models. Generically we do not observe a convergence to a fixed point, only for parameters (in the one-parameter family of models parametrizing the measure) which disfavor representations with quantum dimension $d_j > 1$, i.e. all but $j^\pm = 0$, j_{max}; the model appears to flow to the Ashtekar-Lewandowski fixed point. This is rooted in the construction of the (Euclidean) model, where only very few entries in the initial tensor are actually nonzero. We expect this to be less severe for larger levels k and in the Lie group case; however it could be a general flaw of the Euclidean theory. However these results must be taken with a grain of salt as the applied truncation scheme cannot account for all degrees of freedom relevant under coarse graining.

We have however seen that the coarse graining flow, in particular in the EPRL/FK case, shows a surprising complexity. Of course to encounter such a complexity one needs a sufficiently large parameter space, in which the coarse graining flow takes place. Indeed tensor network algorithms provide such a large parameter space, in

[18] Observing such a behavior does not prove that the transition is of second order, but it is a strong indication, e.g. this is observed at the phase transition of the two-dimensional Ising model.

addition to a truncation scheme, adjusted to the dynamics of the system. As we have seen this leads of course to numerical challenges. We addressed these challenges with a range of techniques in this work, but a further improvement is needed.

In general one can always expect that the set of spin representations, allowed to appear in the initial models as prescribed by the simplicity constraints, is enlarged. This happens because under the coarse graining flow spins associated to finer building blocks are coupled to each other to give spins associated to coarse grained building blocks. The question is then whether the flow leads to a complete washing out of simplicity constraints (if one flows to the BF model in which all spins are allowed) or whether the flow leads to some subset of allowed spins that can be interpreted as a stable form of the simplicity constraints. Another possibility is, as we have seen, a flow to the trivial phase, where only spin $j = 0$ (or in the quantum group case also a $j_{\max}$ with quantum dimension 1) is allowed to appear. The tensor network algorithm employed here allows us to track which spins are allowed and how relevant the spin configurations actually are.

Interestingly we have not found that the simplicity constraints are completely washed out, that is, a flow to the BF models. This shows that the spin foam construction principle—building models by imposing simplicity constraints on the BF model—has the potential to lead to interesting models. For the BC spin net models we have found various fixed points, including a new fixed point, which does describe a nonfactorizing spin net model. On the other hand the EPRL/FK intertwiner model showed a very intricate flow. We attribute that in part to the fact that in the Euclidean quantum group model only very few spin values are allowed initially. However, also in the more general EPRL/FK models the set of allowed spins is indeed much more intricate than in the BC model. We thus expect a much more complex flow that also depends on the Immirzi parameter. To study such aspects it is of course necessary to allow for a sufficiently large parameter space in which the flow can take space.

The results in this article are a first hint of the complex dynamics of spin foam models beyond a few building blocks. In order to explore this regime of many degrees of freedom, numerical algorithms and suitable truncations are required to efficiently identify and study the relevant dynamics. Once achieved this will allow us to contrast the models with observations and measurements, such that they can be eventually verified or falsified. To advance towards this goal works such as the one presented here are necessary as they shed light on promising directions and technical issues that have to be overcome.

ACKNOWLEDGMENTS

The authors thank Florian Girelli for discussions at the start of this project. S. S. thanks Benjamin Bahr for discussions on the construction of the EPRL/FK model. C. S. has been supported by an NSERC USRA fellowship and an NSERC Discovery grant awarded to Florian Girelli. S. S. has been supported by Grant No. BA 4966/1-1 of the German Research Foundation (DFG). This research was supported in part by the Perimeter Institute for Theoretical Physics. Research at the Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Research, Innovation and Science.

APPENDIX A: $\mathrm{SU}(2)_k$ AND GRAPHICAL CALCULUS

In the following appendixes we include several technical details necessary to thoroughly understand the calculations in this article, which are however not necessary to understand the main ideas. Moreover we keep this brief as most of these topics have been more extensively addressed in [14,54].

In this appendix we briefly include several basic facts about the quantum group $\mathrm{SU}(2)_k$. The notation and conventions used are taken from [52], where one can also find a more detailed discussion of quantum groups. Interested readers should also consult [53].

By $\mathrm{SU}(2)_k$ we actually mean the q-deformation $\mathcal{U}_q(\mathfrak{su}(2))$ of the universal enveloping algebra $\mathcal{U}(\mathfrak{su}(2))$ of the Lie algebra $\mathfrak{su}(2)$ as in [52]. This algebra is generated by three operators $J_\pm$, J_z with commutation relations

$$\begin{aligned} [J_z, J_\pm] &= \pm J_\pm, \\ [J_+, J_-] &= \frac{q^{J_z} - q^{-J_z}}{q^{1/2} - q^{-1/2}}. \end{aligned} \tag{A1}$$

Given the deformation parameter q one defines quantum numbers of the quantum group,

$$[n] = \frac{q^{\frac{n}{2}} - q^{-\frac{n}{2}}}{q^{\frac{1}{2}} - q^{-\frac{1}{2}}}. \tag{A2}$$

For $\mathrm{SU}(2)_k$, q is a root of unity, $q = \exp(\frac{2\pi}{(k+2)} i)$, where $k \in \mathbb{N}$ is called the level of the quantum group. In this case quantum numbers are periodic,

$$[n] = \frac{\sin(\frac{2\pi n}{2k+4})}{\sin(\frac{2\pi}{2k+4})}, \tag{A3}$$

with 0's at $n = 0$ and $n = k + 2$.

As for SU(2), the finite dimensional representations of $\mathrm{SU}(2)_k$ are labeled by $j \in \frac{\mathbb{N}}{2}$ and can be defined on $2(j+1)$-dimensional representation spaces V_j. The quantum dimension d_j of representation j is defined as the quantum number of the classical dimension,

$$d_j := [2j+1]. \tag{A4}$$

Given the periodicity of quantum numbers only representations $j \leq \frac{k}{2}$ have a strictly positive quantum dimension. Representations $j = 0, \frac{1}{2}, \ldots, \frac{k}{2}$ are called admissible; representations $j > \frac{k}{2}$ are of so-called quantum trace 0.

The tensor product of two representations V_{j_1}, V_{j_2} is defined via the coproduct Δ. The action of the $\mathrm{SU}(2)_k$ algebra on $V_{j_1} \otimes V_{j_2}$ is defined as

$$\Delta(J_\pm) = q^{-J_z/2} \otimes J_\pm + J_\pm \otimes q^{J_z/2}, \\ \Delta(J_z) = \mathbb{I} \otimes J_z + J_z \otimes \mathbb{I}. \tag{A5}$$

The tensor product $V_{j_1} \otimes V_{j_2}$ can be decomposed into a direct sum of irreducible representations plus a part consisting of trace 0 representations (which are modded out). With an orthogonal basis $|j, m\rangle$ in the representation spaces, the decomposition is given by Clebsch-Gordan coefficients

$$|j, m\rangle = \sum_{m_1, m_2} {}_q C^{j_1 j_2 j}_{m_1 m_2 m} |j_1 m_1\rangle \otimes |j_2, m_2\rangle. \tag{A6}$$

If three admissible representations j_I, j_K and j_L are coupled in this way, the Clebsch-Gordan coefficients are nonvanishing if several conditions are satisfied,

$$j_I + j_K \geq j_L \quad \text{for permutations } \{J, K, L\} \text{ of } \{1,2,3\}, \\ j_1 + j_2 + j_3 = 0 \mod 1, \\ j_1 + j_2 + j_3 \leq k. \tag{A7}$$

The last condition in (A7) is special to the quantum deformed case at the root of unity and indicates that $V_{j_1} \otimes V_{j_2}$ can include trace 0 parts, which can be modded out [53]. However, some equations (for instance the definition of the $[6j]$ symbol) are only valid up to trace 0 parts [53].

In particular we have the completeness relation

$$\sum_{m_3, j_3 \text{admiss}} {}_q C^{j_1 j_2 j_3}_{m_1 m_2 m_3} {}_q C^{j_1 j_2 j_3}_{m_1' m_2' m_3} = \Pi^{j_1 j_2}_{m_1 m_2, m_1' m_2'}, \tag{A8}$$

where $\Pi^{j_1 j_2}_{m_1 m_2, m_1' m_2'}$ projects out the trace 0 part in $V_{j_1} \otimes V_{j_2}$. The orthogonality relation for the Clebsch-Gordan coefficients is given as

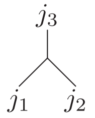
$$\sum_{m_1, m_2} {}_q C^{j_1 j_2 j}_{m_1 m_2 m} {}_q C^{j_1 j_2 j'}_{m_1 m_2 m'} = \delta_{jj'} \delta_{mm'} \theta_{j_1 j_2 j}, \tag{A9}$$

where $\theta_{j_1 j_2 j} = 1$ if the coupling conditions (A7) are satisfied and vanishing otherwise.

APPENDIX B: DIAGRAMMATIC CALCULUS

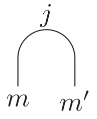
When studying spin net models the notion of the dual representation is necessary. For quantum groups this is more complicated to define than in the classical case, but can be conveniently overcome with the graphical calculus invented in [14] (and also used in [54]).

A special direction must be specified for the quantum group, which we take to be the vertical direction. Then the drawings are interpreted as maps from a tensor product of representation spaces of $\mathrm{SU}(2)_k$ (incoming lines from below) to a tensor product of representation spaces (outgoing lines on top). Each line carries a representation label j and a magnetic index m. Clebsch-Gordan coefficients ${}_q \mathcal{C}^{j_1 j_2 j_3}_{m_1 m_2 m_3}$[19] are a basic example: They are interpreted as a map $V_{j_1} \otimes V_{j_2} \to V_{j_3}$, symbolizing how the spins j_1 and j_2 (with their respective magnetic indices) couple to j_3,



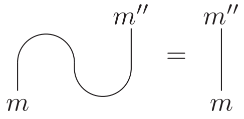
$$:= {}_q \mathcal{C}^{j_1\, j_2\, j_3}_{m_1\, m_2\, m_3} . \tag{B1}$$

A special case of this Clebsch-Gordan coefficient is given by $j_1 = j_2 = j$ and $j_3 = 0$, which we call "cap." It represents a map $V_j \otimes V_j \to \mathbb{C}$,



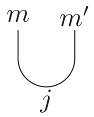
$$:= {}_q \mathcal{C}^{j\, j\, 0}_{m\, m'\, 0} \sqrt{d_j} = (-1)^{j-m} q^{\frac{m}{2}} \delta_{m,-m'} . \tag{B2}$$

From this cap we can similarly define a "cup" by requiring that they give the identity if we concatenate them,



$$= \delta^{m''}_{m} , \tag{B3}$$

which gives

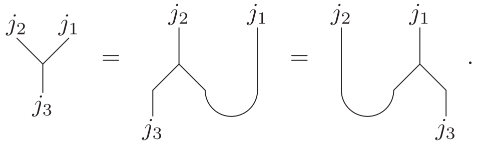


$$= (-1)^{j+m} q^{\frac{m}{2}} \delta_{m,-m'} . \tag{B4}$$

Using cups and caps we obtain Clebsch-Gordan coefficients for the quantum group with inverse (here complex conjugate) deformation parameter $\overline{q}$ as follows:

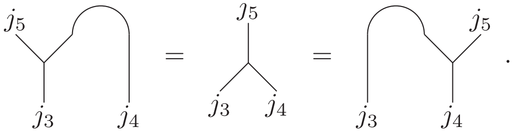
[19] This is not the standard Clebsch-Gordan coefficient defined in [52], but it is modified by the quantum dimension: ${}_q \mathcal{C}^{j_1 j_2 j_3}_{m_1 m_2 m_3} = {}_q C^{j_1 j_2 j_3}_{m_1 m_2 m_3} (\sqrt{d_{j_3}})^{-1}$.

$$_{\bar q}\mathcal{C}^{j_1\, j_2\, j_3}_{m_1\, m_2\, m_3} =$$



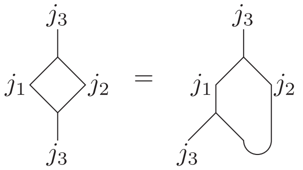
(B5)

This map is hence interpreted as mapping $V_{j_3} \to V_{j_1} \otimes V_{j_2}$; thus it is dual to (B1). Of course one can analogously obtain (B1) again,



(B6)

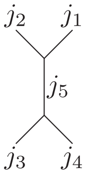
Concatenating these two maps gives a map $V_{j_3} \to V_{j_3}$ proportional to the identity.



$$= (-1)^{j_1+j_2-j_3} d^{-1}_{j_3} \delta_{m_3\, m_3'} \,. \tag{B7}$$

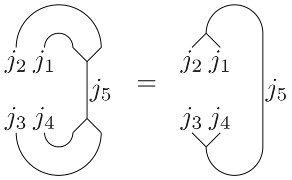
With this graphical calculus already seen in the main body of the article, several important identities can be compactly written.

An important ingredient is the Haar projector $\mathcal{P}$, where we restrict ourselves here to the one for $\mathrm{SU}(2)_k$. The version for $\mathrm{SU}(2)_k \times \mathrm{SU}(2)_k$ is obtained by tensoring the expression as seen in the main part of this article. A four-valent intertwiner is given as follows:



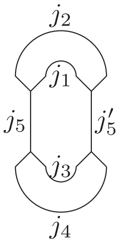
$$= \sum_{m_5} {}_{\bar q}\mathcal{C}^{j_1\, j_2\, j_5}_{m_1\, m_2\, m_5}\; {}_{q}\mathcal{C}^{j_3\, j_4\, j_5}_{m_3\, m_4\, m_5} \,. \tag{B8}$$

Its dual is defined by placing cups on its bottom legs and caps on its top ones avoiding crossing of legs. Graphically this is nicely expressed as



$$= (-1)^{2j_5} \sum_{m_5} q^{m_5}\; {}_{\bar q}\mathcal{C}^{j_1\, j_2\, j_5}_{m_1\, m_2\, m_5}\; {}_{q}\mathcal{C}^{j_3\, j_4\, j_5}_{m_3\, m_4\, m_5} \,. \tag{B9}$$

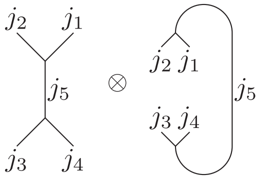
Diagrams (B8) and (B9) determine $\mathcal{P}$ up to normalization. To compute $\mathcal{P} \cdot \mathcal{P}$ we have to evaluate the following diagram:



$$= (-1)^{j_1+j_2+j_3+j_4}\, (d_{j_5})^{-1}\, \delta_{j_5 j_5'} \,. \tag{B10}$$

We obtain

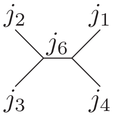
$$\mathcal{P}_{(\{m\},\{n\})}(j_1,j_2,j_3,j_4) := \sum_{j_5} (-1)^{j_1+j_2+j_3+j_4}\, d_{j_5}$$



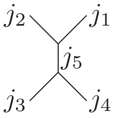
, (B11)

where the magnetic indices m are encoded in the first diagram, n in the second.

The change of the recoupling scheme

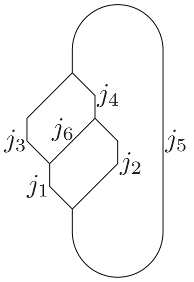


$$= \sum_{j_5} \sqrt{\frac{d_{j_5}}{d_{j_6}}} \begin{bmatrix} j_1 & j_2 & j_5 \\ j_3 & j_4 & j_6 \end{bmatrix}$$

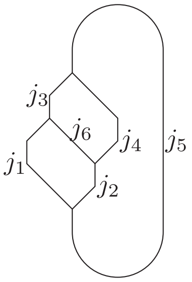


(B12)

is given by the $[6j]$ symbol, which is also defined by a graphical identity,



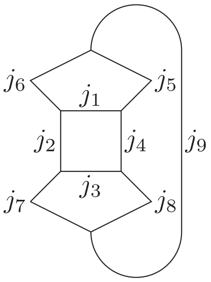
$$=$$



$$= \begin{Bmatrix} j_1 & j_2 & j_5 \\ j_4 & j_3 & j_6 \end{Bmatrix} =: \frac{(-1)^{j_1+j_2+j_3+j_4}}{\sqrt{d_{j_5} d_{j_6}}} \begin{bmatrix} j_1 & j_2 & j_5 \\ j_4 & j_3 & j_6 \end{bmatrix}. \tag{B13}$$

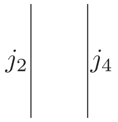
See [14] for a derivation.

A diagram worth mentioning is the following, as it appears in the four-valent tensor network algorithm (see again [14] for a derivation):

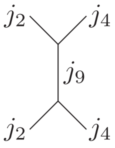


$$. \tag{B14}$$

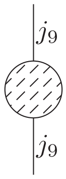
With the following two identities, it is possible to split this diagram into two $[6j]$ symbols. Fortunately this splitting is precisely the splitting necessary for the three-valent algorithm such that the equation for the coarse grained tensor can be readily read off from the four-valent algorithm (see [54] for a more thorough derivation),



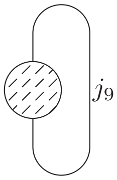
$$=$$



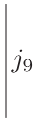
$$(-1)^{j_2+j_4-j_9} d_{j_9} \quad , $$



$$=$$

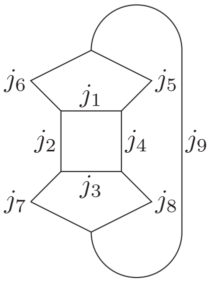


$$\frac{(-1)^{2j_9}}{d_{j_9}}$$

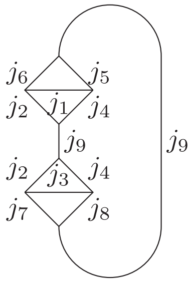


$$. \tag{B15}$$

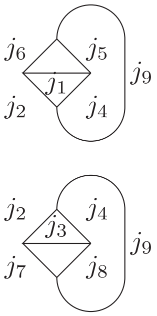
Thus we obtain for (B14)



$$= (-1)^{j_2+j_4-j_9} d_{j_9}$$



$$= (-1)^{j_2+j_4-j_9}$$



$$= \frac{(-1)^{j_2+j_4+j_9}(-1)^{j_5+j_6+j_7+j_8}}{d_{j_9}\sqrt{d_{j_1}d_{j_3}}} \begin{bmatrix} j_2 & j_4 & j_9 \\ j_5 & j_6 & j_1 \end{bmatrix} \begin{bmatrix} j_2 & j_4 & j_9 \\ j_8 & j_7 & j_3 \end{bmatrix}. \tag{B16}$$

APPENDIX C: RENORMALIZATION EQUATION

For the sake of completeness we provide the equations to compute the renormalized three-valent tensor $\hat{S}$.

In the three-valent algorithm, two three-valent tensors $\hat{S}$ are contracted among a common edge[20] to an intermediate four-valent tensor. For efficiency we directly compute the block-diagonal form of $\hat{\mathcal{T}}$,

$$\begin{aligned}\hat{\mathcal{T}}^{(\{\bar{J}\})}(\{j_1\},\{j_2\};\{j_c\},\{j_a\}) = \sum_{\{j_b\}} & \frac{\sqrt{(-1)^{j_c^+ + j_a^+ + \bar{J}^+}}}{\sqrt{d_{\bar{J}^+}}\sqrt{d_{j_b^+}}} \frac{\sqrt{(-1)^{j_c^- + j_a^- + \bar{J}^-}}}{\sqrt{d_{\bar{J}^-}}\sqrt{d_{j_b^-}}} \frac{\sqrt{(-1)^{(j_c^+)' + (j_a^+)' + (\bar{J}^+)'}}}{\sqrt{d_{(\bar{J}^+)'}}\sqrt{d_{(j_b^+)'}}} \frac{\sqrt{(-1)^{(j_c^-)' + (j_a^-)' + (\bar{J}^-)'}}}{\sqrt{d_{(\bar{J}^-)'}}\sqrt{d_{(j_b^-)'}}} \\ & \times \sqrt{d_{j_1^+}d_{j_2^+}}\sqrt{d_{j_1^-}d_{j_2^-}}\sqrt{d_{(j_1^+)'}d_{(j_2^+)'}}\sqrt{d_{(j_1^-)'}d_{(j_2^-)'}} \\ & \times \begin{bmatrix} j_c^+ & j_a^+ & \bar{J}^+ \\ j_1^+ & j_2^+ & j_b^+ \end{bmatrix} \begin{bmatrix} j_c^- & j_a^- & \bar{J}^- \\ j_1^- & j_2^- & j_b^- \end{bmatrix} \begin{bmatrix} (j_c^+)' & (j_a^+)' & (\bar{J}^+)' \\ (j_1^+)' & (j_2^+)' & (j_b^+)' \end{bmatrix} \begin{bmatrix} (j_c^-)' & (j_a^-)' & (\bar{J}^-)' \\ (j_1^-)' & (j_2^-)' & (j_b^-)' \end{bmatrix} \\ & \times (\hat{S})^{(\{j_1\})}(\{j_b\},\{j_a\})(\hat{S})^{(\{j_2\})}(\{j_c\},\{j_b\}). \end{aligned} \tag{C1}$$

The $[6j]$ symbols stem from the treatment of magnetic indices to arrive at the block-diagonal form. The notation is explained in Fig. 10. Note that in the actual algorithm we work with superindices to only sum and store nonvanishing contributions. As before $\{\bar{J}\}$ denotes four $\mathrm{SU}(2)_k$ representations $\bar{J}^+$, $\bar{J}^-$, $(\bar{J}^+)'$, $(\bar{J}^-)'$.

To define a new effective three-valent tensor $\hat{S}'$, we need to define a map mapping $\{j_a\}$, $\{j_c\}$ into a new coarser edge labeled by $\{\bar{J}\}$. As usual this is done by a singular value decomposition: From $\hat{\mathcal{T}}^{(\{\bar{J}\})}$ we define a matrix by grouping together the coarse edges $\{j_1\}$, $\{j_2\}$ and the finer edges $\{j_a\}$, $\{j_c\}$,

$$\tilde{S}^{(\{\bar{J}\})}_{(\{j_1\},\{j_2\});(\{j_a\},\{j_c\})} = \sum_i U^{(\{\bar{J}\})}_{(\{j_1\},\{j_2\});i} \lambda_i^{(\{\bar{J}\})} (V^{(\{\bar{J}\})}_{(\{j_a\},\{j_c\});i})^\dagger. \tag{C2}$$

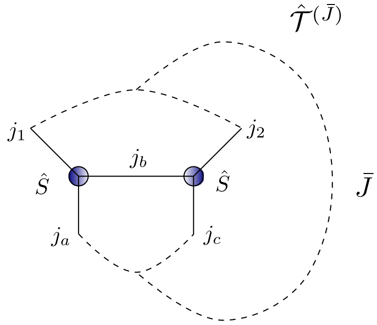


FIG. 10. Notation for intermediate tensor $\hat{\mathcal{T}}$.

In our truncation scheme, we take over one singular value per block $\{\bar{J}\}$. Thus we obtain $(\hat{S}')^{(\{\bar{J}\})}$ by contracting the legs $\{j_a\}$, $\{j_c\}$ with the map $V^{(\{\bar{J}\})}_{(\{j_a\},\{j_c\});1}$.[21] However as V is a unitary matrix it immediately follows that $(\hat{S}')^{(\{\bar{J}\})}$ is given by

[20] In principle, one has four different tensors S_i, $i = 1, \ldots, 4$, but for the models under discussion here, they turn out to be all identical. Thus the algorithms are significantly simplified.

[21] To not alter the partition function one actually inserts $VV^\dagger$. $V^\dagger$ gets contracted with the opposite tensor.

$$(\hat{S}')^{(\{\bar{J}\})}(\{j_1\},\{j_2\}) = U^{(\{\bar{J}\})}_{(\{j_1\},\{j_2\});1}\lambda_1^{(\{\bar{J}\})}. \tag{C3}$$

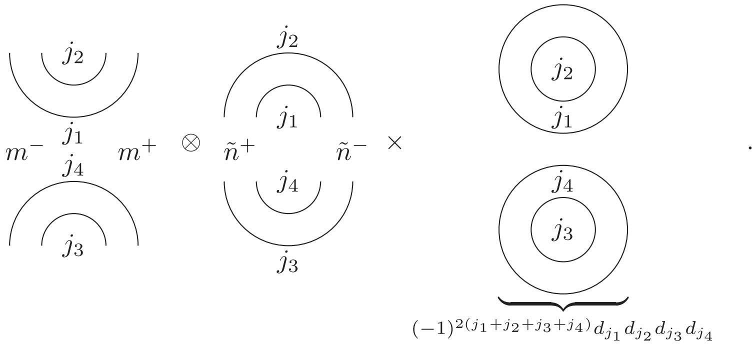
APPENDIX D: NORMALIZATION BC MODEL

In this section we briefly derive the normalization of the BC model. We fix it by requiring that the BC tensor T_{BC} (17) contracted with itself gives T_{BC} again,

$$T_{\mathrm{BC}} \circ T_{\mathrm{BC}} \overset{!}{=} T_{\mathrm{BC}}. \tag{D1}$$

$T_{\mathrm{BC}} \circ T_{\mathrm{BC}}$ is of the following form:

$$(T_{\mathrm{BC}} \circ T_{\mathrm{BC}})(\{j\},\{m^{\pm}\},\{\tilde{n}^{\pm}\}) = c^2_{\{j_i\}}$$



(D2)

For the normalization constant $c_{\{j_i\}}$ we thus obtain the following condition:

$$c^2_{\{j_i\}}\underbrace{(-1)^{2(j_1+j_2+j_3+j_4)}}_{=1}d_{j_1}d_{j_2}d_{j_3}d_{j_4} \overset{!}{=} c_{\{j_i\}} \Rightarrow c_{\{j_i\}} = (d_{j_1}d_{j_2}d_{j_3}d_{j_4})^{-1}. \tag{D3}$$

As explained in the main body of the article the normalization in spin foam models is not uniquely fixed. Thus we choose $c_{\{j_i\}} = (d_{j_1}d_{j_2}d_{j_3}d_{j_4})^{\alpha}$ to accommodate for this uncertainty.

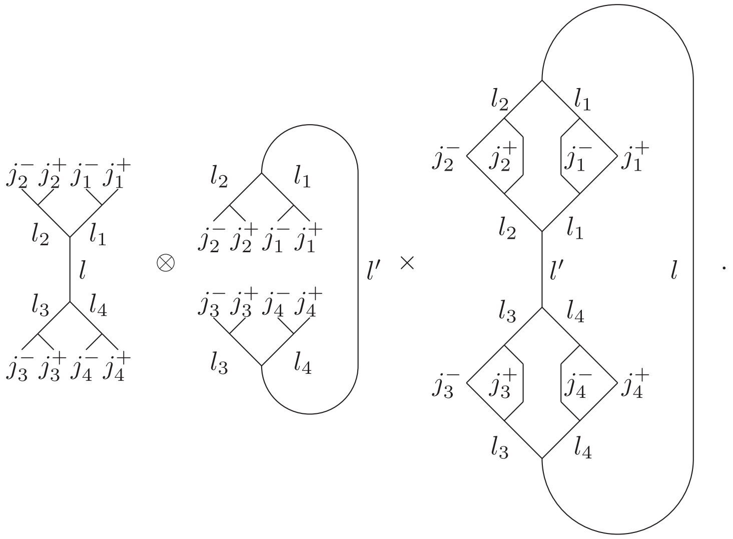
APPENDIX E: NORMALIZATION OF THE EPRL MODEL

The derivation for the normalization of the EPRL spin net model is analogous to the BC case. Again we study T_{EPRL} (24) contracted with itself, which should give T_{EPRL} again,

$$T_{\mathrm{EPRL}} \circ T_{\mathrm{EPRL}} \underset{=}{!} T_{\mathrm{EPRL}}. \tag{E1}$$

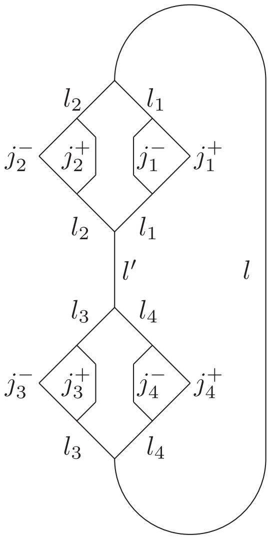
For $T_{\mathrm{EPRL}} \circ T_{\mathrm{EPRL}}$ we obtain

$$(T_{\mathrm{EPRL}} \circ T_{\mathrm{EPRL}})(\{j\},\{m^{\pm}\},\{\tilde{n}^{\pm}\}) := \sum_{l,l'} c_{\{l\}}c_{\{l'\}}$$

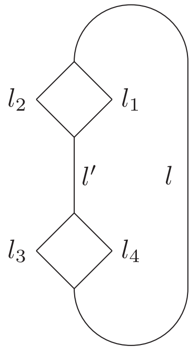


(E2)

The last diagram is straightforward to compute given the graphical identities in Appendix B,



$$= (-1)^{\sum_{i=1}^4 j_i^+ + j_i^- - l_i} \, (d_{l_1} d_{l_2} d_{l_3} d_{l_4})^{-1} \times$$



$$= (-1)^{\sum_{i=1}^4 j_i^+ + j_i^- - 2l} \, (d_{l_1} d_{l_2} d_{l_3} d_{l_4})^{-1} \, d_l^{-2} \delta_{ll'} \quad . \tag{E3}$$

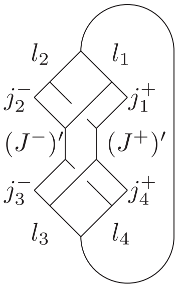
This implies for the normalization constant $c_{\{l\}}$,

$$c_{\{l\}}^2 (-1)^{\sum_{i=1}^4 j_i^+ + j_i^- - 2l} (d_{l_1} d_{l_2} d_{l_3} d_{l_4})^{-1} (d_l)^{-2} \overset{!}{=} c_{\{l\}} \Rightarrow c_{\{l\}} = (-1)^{\sum_{i=1}^4 j_i^+ + j_i^- - 2l} d_{l_1} d_{l_2} d_{l_3} d_{l_4} d_l^2. \tag{E4}$$

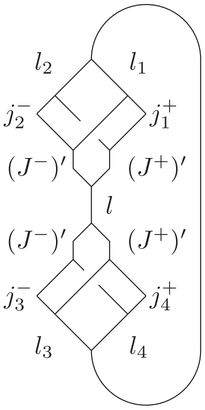
Again as the normalization is not uniquely determined in spin foams we keep this arbitrary; we choose $c_{\{l\}} = (-1)^{\sum_{i=1}^4 j_i^+ + j_i^- - 2l} (d_{l_1} d_{l_2} d_{l_3} d_{l_4})^{\alpha} d_l^2$.

APPENDIX F: DERIVATION OF THE EPRL AMPLITUDE

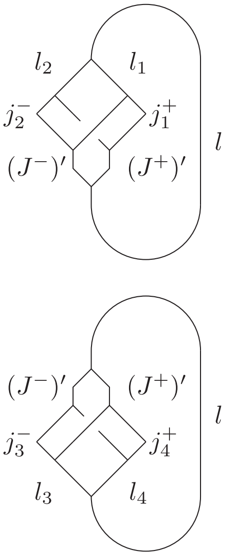
In this appendix we quickly outline how to derive the diagrams in the three-valent EPRL model from the four-valent one. Essentially one applies the identities (B15). We demonstrate this only for one diagram in (25), as it follows for the other diagram analogously,



$$= (-1)^{(J^-)' + (J^+)' - l} d_l$$



$$= \underbrace{(-1)^{(J^-)' + (J^+)' - l} d_l \times \frac{(-1)^{4l}}{d_l}}_{(-1)^{(J^-)' + (J^+)' - l}}$$



$$. \tag{F1}$$

To arrive at the final expression it remains to include the identity for the $\mathcal{R}$ matrices, which is explained in the main part of the paper.

[1] M. P. Reisenberger and C. Rovelli, Sum over surfaces, form of loop quantum gravity, Phys. Rev. D **56**, 3490 (1997).

[2] C. Rovelli, *Quantum Gravity* (Cambridge University Press, Cambridge, 2004).

[3] A. Perez, The spin foam approach to quantum gravity, Living Rev. Relativ. **16**, 3 (2013).

[4] B. Dittrich, Diffeomorphism symmetry in quantum gravity models, Adv. Sci. Lett. **2**, 151 (2009).

[5] B. Bahr and B. Dittrich, (Broken) gauge symmetries and constraints in Regge calculus, Classical Quantum Gravity **26**, 225011 (2009).

[6] B. Dittrich, How to construct diffeomorphism symmetry on the lattice, *Proc. Sci.*, QGQGS20112011 (2011) 012 [arXiv:1201.3840].

[7] B. Dittrich, The continuum limit of loop quantum gravity—a framework for solving the theory, arXiv:1409.1450.

[8] D. Oriti, *Approaches to quantum gravity*, edited by D. Oriti (Cambridge University Press, Cambridge, 2009), pp. 310–331.

[9] S. Carrozza, Flowing in group field theory space: a review, SIGMA **12**, 070 (2016).

[10] M. Kisielowski, J. Lewandowski, and J. Puchta, Feynman diagrammatic approach to spin foams, Classical Quantum Gravity **29**, 015009 (2012).

[11] B. Bahr and B. Dittrich, Improved and perfect actions in discrete gravity, Phys. Rev. D **80**, 124030 (2009).

[12] B. Bahr, B. Dittrich, and S. Steinhaus, Perfect discretization of reparametrization invariant path integrals, Phys. Rev. D **83**, 105026 (2011).

[13] B. Dittrich and S. Steinhaus, Path integral measure and triangulation independence in discrete gravity, Phys. Rev. D **85**, 044032 (2012).

[14] B. Dittrich, M. Martín-Benito, and S. Steinhaus, Quantum group spin nets: refinement limit and relation to spin foams, Phys. Rev. D **90**, 024058 (2014).

[15] B. Bahr, On background-independent renormalization of spin foam models, arXiv:1407.7746.

[16] B. Dittrich, From the discrete to the continuous: towards a cylindrically consistent dynamics, New J. Phys. **14**, 123004 (2012).

[17] B. Dittrich and S. Steinhaus, Time evolution as refining, coarse graining and entangling, New J. Phys. **16**, 123041 (2014).

[18] J. J. Halliwell and J. B. Hartle, Wave functions constructed from an invariant sum over histories satisfy constraints, Phys. Rev. D **43**, 1170 (1991).

[19] C. Rovelli, The projector on physical states in loop quantum gravity, Phys. Rev. D **59**, 104015 (1999).

[20] L. Freidel, Group field theory: an overview, Int. J. Theor. Phys. **44**, 1769 (2005).

[21] T. Thiemann and A. Zipfel, Linking covariant and canonical LQG II: spin foam projector, Classical Quantum Gravity **31**, 125008 (2014).

[22] B. Dittrich and P. A. Höhn, From covariant to canonical formulations of discrete gravity, Classical Quantum Gravity **27**, 155001 (2010).

[23] V. Bonzom and B. Dittrich, Dirac's discrete hypersurface deformation algebras, Classical Quantum Gravity **30**, 205013 (2013).

[24] C. Perini, C. Rovelli, and S. Speziale, Self-energy and vertex radiative corrections in LQG, Phys. Lett. B **682**, 78 (2009).

[25] A. Riello, Self-energy of the Lorentzian Engle-Pereira-Rovelli-Livine and Freidel-Krasnov model of quantum gravity, Phys. Rev. D **88**, 024011 (2013).

[26] V. Bonzom and B. Dittrich, Bubble divergences and gauge symmetries in spin foams, Phys. Rev. D **88**, 124021 (2013).

[27] L.-Q. Chen, Bulk amplitude and degree of divergence in four-dimensional spin foams, Phys. Rev. D **94**, 104025 (2016).

[28] A. Banburski and L.-Q. Chen, A simpler way of imposing simplicity constraints, Phys. Rev. D **94**, 104003 (2016).

[29] M. Levin and C. P. Nave, Tensor Renormalization Group Approach to Two-Dimensional Classical Lattice Models, Phys. Rev. Lett. **99**, 120601 (2007).

[30] Z.-C. Gu and X.-G. Wen, Tensor-entanglement-filtering renormalization approach and symmetry protected topological order, Phys. Rev. B **80**, 155131 (2009).

[31] B. Dittrich, S. Mizera, and S. Steinhaus, Decorated tensor network renormalization for lattice gauge theories and spin foam models, New J. Phys. **18**, 053009 (2016).

[32] G. Evenbly and G. Vidal, Tensor Network Renormalization, Phys. Rev. Lett. **115**, 180405 (2015).

[33] B. Bahr, B. Dittrich, F. Hellmann, and W. Kaminski, Holonomy spin foam models: definition and coarse graining, Phys. Rev. D **87**, 044048 (2013).

[34] B. Bahr, B. Dittrich, and J. P. Ryan, Spin foam models with finite groups, J. Grav. **2013**, 549824 (2013).

[35] J. B. Kogut, An introduction to lattice gauge theory and spin systems, Rev. Mod. Phys. **51**, 659 (1979).

[36] B. Dittrich, F. C. Eckert, and M. Martín-Benito, Coarse graining methods for spin net and spin foam models, New J. Phys. **14**, 035008 (2012).

[37] B. Dittrich, M. Martín-Benito, and E. Schnetter, Coarse graining of spin net models: dynamics of intertwiners, New J. Phys. **15**, 103004 (2013).

[38] C. Delcamp and B. Dittrich (to be published).

[39] J. W. Barrett and L. Crane, Relativistic spin networks and quantum gravity, J. Math. Phys. (N.Y.) **39**, 3296 (1998).

[40] J. Engle, R. Pereira, and C. Rovelli, Flipped spin foam vertex and loop gravity, Nucl. Phys. **B798**, 251 (2008).

[41] J. Engle, E. Livine, R. Pereira, and C. Rovelli, LQG vertex with finite Immirzi parameter, Nucl. Phys. **B799**, 136 (2008).

[42] L. Freidel and K. Krasnov, A new spin foam model for four-dimensional gravity, Classical Quantum Gravity **25**, 125018 (2008).

[43] W. Kaminski, M. Kisielowski, and J. Lewandowski, Spin foams for all loop quantum gravity, Classical Quantum Gravity **27**, 095006 (2010); Erratum, Classical Quantum Gravity **29**, 049502(E) (2012).

[44] B. Bahr and S. Steinhaus, Investigation of the spin foam path integral with quantum cuboid intertwiners, Phys. Rev. D **93**, 104029 (2016).

[45] B. Bahr and S. Steinhaus, Numerical Evidence for a Phase Transition in Four-Dimensional Spin Foam Quantum Gravity, Phys. Rev. Lett. **117**, 141302 (2016).

[46] E. R. Livine and S. Speziale, A new spin foam vertex for quantum gravity, Phys. Rev. D **76**, 084028 (2007).

[47] A. Banburski, L.-Q. Chen, L. Freidel, and J. Hnybida, Pachner moves in a four-dimensional Riemannian holomorphic spin foam model, Phys. Rev. D **92**, 124014 (2015).

[48] J. C. Baez, J. D. Christensen, T. R. Halford, and D. C. Tsang, Spin foam models of Riemannian quantum gravity, Classical Quantum Gravity **19**, 4627 (2002).

[49] J. C. Baez and J. D. Christensen, Positivity of spin foam amplitudes, Classical Quantum Gravity **19**, 2291 (2002).

[50] I. Khavkine and J. D. Christensen, q-deformed spin foam models of quantum gravity, Classical Quantum Gravity **24**, 3271 (2007).

[51] G. Vidal, Class of Quantum Many-Body States that can be Efficiently Simulated, Phys. Rev. Lett. **101**, 110501 (2008).

[52] L. C. Biedenharn and M. A. Lohe, *Quantum Group Symmetries and q-Tensor Algebras* (World Scientific, Singapore, 1995).

[53] J. S. Carter, D. E. Flath, and M. Saito, *The Classical and Quantum 6j-Symbols* (Princeton University Press, Princeton, 1995).

[54] S. Steinhaus, Coupled intertwiner dynamics: a toy model for coupling matter to spin foam models, Phys. Rev. D **92**, 064007 (2015).

[55] OpenMP is an application programming interface (API) for parallel computing in C/C++/Fortran, http://openmp.org/wp/.

[56] Julia is a high-level, high-performance dynamic programming language for technical computing, http://julialang.org/.

[57] B. Dittrich and W. Kamiński, Topological lattice field theories from intertwiner dynamics, arXiv:1311.1798.

[58] R. Savit, Duality in field theory and statistical systems, Rev. Mod. Phys. **52**, 453 (1980).

[59] J. F. Plebanski, On the separation of Einsteinian substructures, J. Math. Phys. (N.Y.) **18**, 2511 (1977).

[60] T. Regge, General relativity without coordinates, Nuovo Cimento **19**, 558 (1961).

[61] J. C. Baez, J. D. Christensen, and G. Egan, Asymptotics of 10j symbols, Classical Quantum Gravity **19**, 6489 (2002).

[62] L. Freidel and D. Louapre, Asymptotics of 6j and 10j symbols, Classical Quantum Gravity **20**, 1267 (2003).

[63] F. Conrady and L. Freidel, On the semiclassical limit of four-dimensional spin foam models, Phys. Rev. D **78**, 104023 (2008).

[64] J. W. Barrett, R. Dowdall, W. J. Fairbairn, H. Gomes, and F. Hellmann, Asymptotic analysis of the EPRL four-simplex amplitude, J. Math. Phys. (N.Y.) **50**, 112504 (2009).

[65] L. Smolin, Linking topological quantum field theory and nonperturbative quantum gravity, J. Math. Phys. (N.Y.) **36**, 6417 (1995).

[66] S. Major and L. Smolin, Quantum deformation of quantum gravity, Nucl. Phys. **B473**, 267 (1996).

[67] R. Borissov, S. Major, and L. Smolin, The geometry of quantum spin networks, Classical Quantum Gravity **13**, 3183 (1996).

[68] K. Noui and P. Roche, Cosmological deformation of Lorentzian spin foam models, Classical Quantum Gravity **20**, 3175 (2003).

[69] W. J. Fairbairn and C. Meusburger, Quantum deformation of two four-dimensional spin foam models, J. Math. Phys. (N.Y.) **53**, 022501 (2012).

[70] M. Han, Four-dimensional spin-foam model with quantum Lorentz group, J. Math. Phys. (N.Y.) **52**, 072501 (2011).

[71] M. Dupuis and F. Girelli, Quantum hyperbolic geometry in loop quantum gravity with cosmological constant, Phys. Rev. D **87**, 121502 (2013).

[72] M. Dupuis and F. Girelli, Observables in loop quantum gravity with a cosmological constant, Phys. Rev. D **90**, 104037 (2014).

[73] V. G. Turaev and O. Y. Viro, State sum invariants of three manifolds and quantum 6j symbols, Topology **31**, 865 (1992).

[74] H. M. Haggard, M. Han, W. Kamiński, and A. Riello, SL(2,C) Chern-Simons theory, a nonplanar graph operator, and four-dimensional loop quantum gravity with a cosmological constant: semiclassical geometry, Nucl. Phys. **B900**, 1 (2015).

[75] M. Dupuis and F. Girelli, Quantum hyperbolic geometry in loop quantum gravity with cosmological constant, Phys. Rev. D **87**, 121502 (2013).

[76] B. Dittrich and M. Geiller, Quantum gravity kinematics from extended TQFTs, arXiv:1604.05195.

[77] C. Delcamp and B. Dittrich, From three-dimensional TQFTs to four-dimensional models with defects, arXiv:1606.02384.

[78] J. W. Barrett and L. Crane, A Lorentzian signature model for quantum general relativity, Classical Quantum Gravity **17**, 3101 (2000).

[79] J. Engle, E. Livine, R. Pereira, and C. Rovelli, LQG vertex with finite Immirzi parameter, Nucl. Phys. **B799**, 136 (2008).

[80] Y. Neiman, A look at area Regge calculus, arXiv:1308.1012.

[81] E. Alesci and C. Rovelli, The complete LQG propagator. I. Difficulties with the Barrett-Crane vertex, Phys. Rev. D **76**, 104012 (2007).

[82] B. Bahr, F. Hellmann, W. Kaminski, M. Kisielowski, and J. Lewandowski, Operator spin foam models, Classical Quantum Gravity **28**, 105003 (2011).

[83] B. Bahr, On knottings in the physical Hilbert space of LQG as given by the EPRL model, Classical Quantum Gravity **28**, 045002 (2011).

[84] E. Bianchi, D. Regoli, and C. Rovelli, Face amplitude of spin foam quantum gravity, Classical Quantum Gravity **27**, 185009 (2010).

[85] M. Bojowald and A. Perez, Spin foam quantization and anomalies, Gen. Relativ. Gravit. **42**, 877 (2010).

[86] B. Dittrich, W. Kamiński, and S. Steinhaus, Discretization independence implies nonlocality in four-dimensional discrete quantum gravity, Classical Quantum Gravity **31**, 245009 (2014).

[87] W. Kaminski, M. Kisielowski, and J. Lewandowski, The EPRL intertwiners and corrected partition function, Classical Quantum Gravity **27**, 165020 (2010); Erratum, Classical Quantum Gravity **29**, 049501(E) (2012).

[88] M. P. Reisenberger, On relativistic spin network vertices, J. Math. Phys. (N.Y.) **40**, 2046 (1999).

[89] B. Bahr, B. Dittrich, and S. He, Coarse graining free theories with gauge symmetries: the linearized case, New J. Phys. **13**, 045009 (2011).