Decorated tensor network renormalization for lattice gauge theories and spin foam models

Bianca Dittrich,1 Sebastian Mizera,1,2 and Sebastian Steinhaus1

1Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, Ontario, Canada N2L 2Y5
2Girton College, University of Cambridge, Huntingdon Road, Cambridge CB3 0JG, United Kingdom

Tensor network techniques have proved to be powerful tools that can be employed to explore the large scale dynamics of lattice systems. Nonetheless, the redundancy of degrees of freedom in lattice gauge theories (and related models) poses a challenge for standard tensor network algorithms. We accommodate for such systems by introducing an additional structure decorating the tensor network. This allows to explicitly preserve the gauge symmetry of the system under coarse graining and straightforwardly interpret the fixed point tensors. Using this novel information encoded in the decoration might eventually lead to new methods incorporating both analytical and numerical techniques.

INTRODUCTION

Tensor network algorithms [1–5] have become a well used method for the coarse graining of classical partition functions. In particular for models in which Monte Carlo methods are not available (due to sign problems for fermionic systems or due to a priori complex amplitudes as in spin foams) [6] tensor network algorithms may be the only way to explore the phase structure of a given system. We are in particular interested in the phase structure of spin foams, candidate models for quantum gravity. For these models the key open question is the recovery of a dynamics describing continuum manifolds in the limit involving many building blocks.

This work will address the question of efficient algorithms for lattice gauge theories, including generalized models such as spin foams. Here the problem is to deal with the gauge symmetry, that is a large redundancy in the description. This issue has been addressed for a framework where tensor networks are used as a variational ansatz for the low energy wave function (for quantum lattice models) [11, 12], however, to our knowledge not for algorithms in which the tensor network represents the partition function.

A tensor network representation for lattice gauge theories and spin foam models appeared in [13, 14] and later also in [15]. As we will explain later, these representations are not economic in terms of bond dimension of the initial tensor. As the aim is to go to rather ‘large’ structure group, that is SU(2)k with k large (or as in [14] even SU(2) which would involve infinite bond dimension), this issue is rather important.

The main aim of this work is therefore to develop methods for coarse graining (generalized) lattice gauge theories, that are sufficiently efficient to be applied to higher dimensions (here we will consider 3D, eventually the methods have to be applied to 4D models) and for ‘large’ structure groups.

To this end we will go beyond tensor networks, allowing more general indexed objects, which we will term “decorated tensor networks”. This allows a more efficient representation of lattice gauge theories and spin foam models and furthermore a comparably simple algorithm that emulates the 2D algorithms of [3, 4]. Although this generalization of tensor networks and the related algorithm were designed for (generalized) lattice gauge theories the framework is also applicable to e.g. Ising like models in two and higher dimension. Here the decorated tensor networks lead to an interesting mixture of keeping part of the variables of the original model, i.e. the Ising spins, and integrating out and transforming the other part of the variables (by field redefinitions), which is the essence of tensor network models. The field redefinitions inherent in tensor network methods turn however the interpretation of the phases and fixed points into a difficult task, see also the discussion in [16]. For this reason the algorithm presented here might be also useful for other models than lattice gauge theories.

Spin foams: In the following we will outline the relevance of our aim for the coarse graining of spin foams. Spin foams [6–8] are generalized versions of lattice gauge theories, but feature a more complicated structure. Roughly speaking, the models are parametrized by weights for intertwiners. This means that the weights depend on a combination of several representation labels, instead of being parametrized by face weights, which are functions of one representation label only and describe standard lattice gauge models. The phase structure of spin foams, beyond those phases that also appear in lattice gauge theory, is basically unknown, except for the recent developments in [10, 17, 18]. Tensor network algorithms are indeed so far the only tool that is available for the real space coarse graining of these systems (as...
The full spin foam models are defined in four dimensions and involve a structure group $SU(2) \times SU(2)$ (this is for models describing Euclidean signature metrics, which face less technical challenges than those describing Lorentzian metrics, based on $SO(3,1)$). However, these models are barely understood beyond a very few building blocks, additionally the issue of divergences arises [23–25]. [9, 13, 14] therefore proposed a program in which one considers a reduction of the structure group to a finite group, and a dimensional reduction to two dimensional spin net models. Nevertheless a key input of spin models is barely understood beyond a very few building blocks. We also shortly comment on extensions to 4D and non–Abelian gauge theories. Afterwards we present a similar algorithm applicable to Ising–like systems which analogously preserve the original spin variables of the system, such that correlation functions can be straightforwardly calculated. We close with a discussion of the algorithms and results.

**REPRESENTATIONS FOR LATTICE GAUGE THEORIES**

Lattice gauge models feature a partition function

$$Z = \sum_{g_\mathbf{f}} \prod_{h_\mathbf{f}} \omega \left( \prod_{e \subseteq \mathbf{f}} g_e \right),$$

(1)

where $g_e$ denotes group elements attached to the (oriented) edges $e$ of a lattice, $f$ denotes an (oriented) face or plaquette of the lattice and $h_f = \prod_{e \subseteq f} g_e$ the face holonomy, that is the oriented product of group elements associated to the edges belonging to this face. The face weights $\omega$ have to be class functions, i.e. $\omega(g h g^{-1}) = \omega(h)$, for the partition function to be invariant under gauge transformations $g_e \rightarrow h_e^{-1} g_e h_t(e)$. Here $h_v$ denote group elements, acting as gauge parameter, associated to the vertices of the lattice, $s(e)$ and $t(e)$ the source and target vertex of the edge $e$ respectively.

We assumed a finite structure group in [1], for application to Lie groups we have to replace the sum in [1] by an integral. Later we will perform a duality transformation, in which the partition function appears as a finite/infinite sum for finite/compact Lie groups. Moreover in this representation we can also easily deal with the quantum group case.

Although the representation [1] features variables based on edges, as in tensor networks, the weights are actually associated to the faces and not to the vertices. Additionally the gauge symmetry leads to an over-parametrization of the system, which should be avoided. A duality transformation (or strong coupling / high temperature expansion) rewrites the partition function using a group Fourier transform for the weights:

$$\omega(h) = \sum_\rho \tilde{\omega}(\rho) \chi_\rho(h),$$

(2)

with $\chi_\rho$ denoting the character of the representation, see for instance [9, 31]. Here the sum is over the irreducible

---

2 This theory is dual to the (non–gauge) Ising model [31], which
unitary representations of the structure group. The partition function \( Z \) turns into

\[
Z = \sum \prod_{\rho_f} \tilde{\omega}(\rho_f) \prod_{e} \langle (\rho_e)_{f \supset e} \vert (\rho_f)_{f \supset e} \rangle . \tag{3}
\]

Here \( \rho_e \) are the Haar projectors, that is representation theoretical objects associated to the edges of the lattice. These Haar projectors map onto the subspace associated to the trivial representation in the tensor product of representations \( \rho_f \) associated to the faces adjacent to the edge \( e \), that is the invariant subspace in the representation space \( \otimes_{f \supset e} V_{\rho_f} \). Thus we have magnetic indices \( m_e, n_e \) that are contracted in a particular pattern, as shown in figure 1.

Spin foam models implement simplicity constraints by replacing this tensor \( \rho_e \) by a projector \( \rho'_e \) onto a smaller subspace inside the trivial representation space, thus \( \tilde{\omega}(\rho'_f) \) can serve also as the partition function for spin foam models (for further explanation see for instance \([9, 13, 14]\)). The particular choice of simplicity constraints is then encoded into a particular choice of projector \( \rho'_e \).

For Abelian structure groups, for instance \( Z_K \), the partition function \( Z \) simplifies to (we will denote the representations for \( Z_K \) by \( k \in 0, \ldots, K - 1 \))

\[
Z = \sum \prod_{f} \tilde{\omega}(k_f) \prod_{e} \delta \left( \sum_{f \supset e} o(e, f) k_f \right) , \tag{4}
\]

where \( o(e, f) = -1 \) if face and edge orientation disagree and \( o(e, f) = +1 \) if these agree. Thus the Haar projectors reduce to Gauss constraints, imposing that the representations around the faces meeting in one edge sum up to zero.

Although we have now variables associated to faces, this form of the partition function (3) can be brought into the form of a tensor network. To this end one associates tensors \( \rho_e \) to the midpoints of each edge and introduces auxiliary tensors to the midpoints of the faces, that make sure, that each tensor \( \rho_e \) around a given face \( f \) see the same representation label \( \rho_f \), see also figure 2. This representation appeared in \([13, 14]\) and also later in \([15]\).

However, this representation (3) is rather inefficient if it comes to coarse graining algorithms. The dependence of the tensor in the magnetic indices is prescribed by the gauge invariance of the model, and thus one would rather wish to pre–contract these indices, as in fact happens for the symmetry protecting algorithms developed in \([17, 18]\) (for models with global symmetry). Thus one would only have to deal with representation and possibly intertwiner labels (which for multiplicity free \([\ast]\) groups are again representation labels), as in \([17, 18]\). This would also allow to understand the coarse graining in terms of weights on intertwiners.

Indeed, a standard technique in spin foams, is to split the projector \( \rho_e \) by using an orthonormal basis \( \{ e \} \) of the invariant subspace of \( \otimes_{f \supset e} V_{\rho_f} \) into two parts (referred

\[\text{FIG. 1. Projector } P_e \text{ is associated to each edge of the lattice. The magnetic indices are contracted with neighbouring projectors sharing the same face } f. \text{ In case the orientations of the faces } f \supset e \text{ and the edge } e \text{ match, i.e. } o(e, f) = 1, \forall f \supset e, \text{ then } P_e : \text{Inv} \left( \otimes_{f \supset e} V_{\rho_f} \right) \rightarrow \text{Inv} \left( \otimes_{f \supset e} V_{\rho_f} \right), \text{ the contragradient representation has to be used for opposing orientations.}
\]

\[\text{FIG. 2. Tensor network representation of a spin foam defined on a cubical lattice. The tensors } T_e \text{ on the edges capture the intertwiner degrees of freedom, while the tensors } T_f \text{ on the faces are an auxiliary structure. For a given face } f \text{ they ensure that all edge tensors } T_e \text{ with } e \subset f \text{ carry the same representation label } \rho_f \text{ assigned to that face.}
\]

\[\text{3 I.e. groups in which the tensor product of two irreducible unitary representations reduces to a sum in which every irreducible unitary representation appears at most once.}\]
FIG. 3. Construction of the vertex amplitude: The projectors \( P_e \) on the edges are expanded in the intertwiner basis \( \{ \iota_e \} \) according to (5), the opposing triangles pictorially representing \( |\iota_e\rangle \langle \iota_e| \). Together with the contraction of magnetic indices shown in figure 1, the projectors can be split and associated to the vertex of the lattice.

\[
(P_e)_{\{n_f\}_{f \supset e}} = \sum_{\iota_e} \{n_f\}_{f \supset e} |\iota_e\rangle \langle \iota_e| \{m_f\}_{f \supset e}, \tag{5}
\]

see figure 3. These two parts can be associated to the source vertex and target vertex of the edge \( e \) respectively. The sum is over intertwiner labels \( \iota_e \) (which are again representation labels assuming multiplicity free groups) appearing in the tensor product \( \otimes_{f \supset e} V_{\rho_f} \).

Now every half edge, labelled by an edge–vertex pair \((e, v)\), carries an object \(|\iota_e\rangle\). The structure of the magnetic index contraction is such, that the sum over the magnetic indices factorizes over the vertices, i.e. the set of objects \( \iota_e \) associated to a given vertex \( v \) can be contracted to a so called vertex amplitude \( A_v (\rho_f)_{f \supset v}, (\iota_e)_{e \supset v} \) that only depends on representation labels associated to the faces and edges adjacent to the vertex \( v \). This is shown in figure 4. Thus we have now a form of the partition function

\[
Z = \sum_{\rho_f, \iota_{e \supset v}} \prod_f \hat{\omega}_f \prod_v A_v (\rho_f)_{f \supset v}, (\iota_e)_{e \supset v}, \tag{6}
\]

which almost is a tensor network, if it would not be for the representation labels on the faces. (The face weights can be absorbed into the vertex weights for a regular lattice.)

This could again be dealt with by introducing auxiliary tensors put into the middle of the faces. One can then attempt to group tensors of this inhomogeneous network into a homogeneous one, resulting in tensors with more edges and also higher initial bond dimension (see [15] where this has been applied to the representation \( (3) \)). For a cubical 3D lattice a possible grouping is depicted in figure 5 and would result in 12–valent tensors, which are also connected via diagonal edges.

Alternatively\(^5\), one can actually copy a given face index \( \rho_f \) to all edges adjacent to this face. To enforce equality between the copied indices we have to multiply the vertex amplitudes (i.e. the tensors) with Kronecker deltas. For a cubical lattice each edge would then carry four rep-

\(^4\) We assume here for simplicity that all faces adjacent to the edge \( e \) have an orientation that agrees with the one of \( e \).

\(^5\) We thank J. Hnybida for pointing this out.
representation labels coming from the adjacent faces and an additional (intertwiner) label \( v_e \), giving a priori \( 6 \times 5 \) representation indices for a tensor describing a cubical lattice in three dimensions. These labels would be restricted by the condition that the four representations per edge actually couple to the intertwiner \( v_e \). For Abelian groups the intertwiners are trivial, and this condition turns into the Gauss constraint that the four representation labels sum to the trivial representation. This would then still give for a structure group \( \mathbb{Z}_K \) a bond dimension \( K^3 \). This again is a rather inefficient way to proceed.

**DECORATED TENSOR NETWORK
RENORMALIZATION**

Here we propose to generalize the concept of tensor networks in order to keep the gauge constraints satisfied explicitly, and hence allow for more efficient and accurate study of the phase space of a given lattice gauge model.

For the following discussion we found it simpler to transform the representation to the dual lattice (which for spin foams is actually the triangulation). Thus ‘tensors based on vertices’ are now replaced by ‘amplitudes associated to building blocks’. To some degree this is just a change of perspective, see also [32], but it allows for more flexibility in designing (and imagining) algorithms. For a cubical 3D lattice this means that we now associate amplitudes to cubes. Indices, which have been associated to building blocks’, to some degree this is just an implied coupling to the intertwiner labels are associated to the faces \( \iota_e \). The coarse-graining / blocking algorithms now proceed by gluing cubes to larger cubes, by summing over the labels associated to bulk edges and bulk faces.

**Abelian gauge models in 3D**

Let us specialize to Abelian gauge models in 3D. These models are dual to Abelian ‘edge’ models with global symmetry \[ [31, [33, [34], as we will see shortly. However, as we want to test algorithms also applicable to non-Abelian gauge models, where such a duality is not available, we will keep the lattice gauge (spin foam) representation of the model.

For Abelian models, for instance with structure group \( \mathbb{Z}_K \), we denote the representations associated to the (oriented) edges by \( k_e \) with \( k_e = 0, \ldots, K - 1 \). We have Gauss constraints on the faces, prescribing that the representations of the edges associated to a given face, have to couple to the trivial representation, that is all face labels are \( k_f = 0 \). This gives the restrictions

\[
\sum_{e \in f} o(e, f) k_e = 0 ,
\]

where \( o(e, f) = -1 \) if face and edge orientation disagree and \( o(e, f) = +1 \) if these agree. These Gauss constraints can be solved (but as mentioned we are not going to do so) by introducing variables (gauge potentials) \( k_e \) on the vertices. Setting \( k_e = k_{\bar{e}} - k_{(e)} \) gives a solution to the Gauss constraints [7]. As mentioned, this leads back to an edge model in which the gauge group serves now as global symmetry group (e.g. Ising model if we started with Ising gauge theory). Such an edge model could be turned into a tensor network by standard methods. (The easiest way is to use a dual representation - which again will feature Gauss constraints now based on the vertices of the lattice.)

Due to the Gauss constraints the edge labels \( k_e \) are not independent of each other. In fact, for a cube there are five independent constraints of the form [7]. An independent set of variables \( k_e \) can be easily found by choosing a maximal tree (a connected subgraph without loops) among the edges of a cube. Thus by solving the Gauss constraints for the 5 edges which are not part of the tree, the cube amplitude does depend only on 7 variables instead of 12 variables. The initial cube amplitudes are given as

\[
A_c = \prod_{e \in c} (\bar{\omega}(k_e))^{1/4} ,
\]
where $\tilde{\omega}$ are the (Fourier transformed) face weights in $A$. Later we will extend this to also contain additional variables, see figure $\text{[2]}$. Note that the amplitude can describe not only the cubic models: It is for example possible to embed a tessellation of the cube into 6 tetrahedra into $\text{[3]}$, since the total number of independent variables is the same as that of the cube.

Algorithm in the leading order approximation

Let us first convey the spirit of the coarse graining algorithm for the leading order approximation. In this case, throughout the algorithm we will keep the same index structure for the cube amplitudes $A_c$. This amplitude will however flow and loose its initial factorizing form.

The basic algorithm is a (decorated) 3D generalization of the Gu–Wen algorithm $\text{[4]}$. We divide each cube into two prisms using a singular value decomposition (SVD). Then four of these prisms are glued to a new cube, see figure $\text{[7]}$. (The new building blocks are actually parallelepipeds or rhomboids, but we will refer to them as cubes.) We changed the lattice length, say in the plane $(xy)$, by a factor of $\sqrt{2}$, see figure $\text{[7]}$. We then rotate the lattice, so that in the next iteration we coarse grain in an orthogonal plane.

To divide each cube into two prisms, we have to introduce diagonals for two faces (opposite to each other) of the cube. For Abelian models the edges of the subdivided faces determine the representation label for the new edge, as we also impose the Gauss constraints for the new triangular faces. As we introduce two new variables and two additional constraints, the counting of independent variables does not change.

To split the cube we apply singular value decomposition. In general this provides a splitting of a matrix $M_{AB}$,

$$M_{AB} = \sum_I \left(U_{AI} \sqrt{\lambda_I} \right) \left(\sqrt{\lambda_I} V_{BI}^\dagger\right) =: \sum_I S_{AI} S_{BI}^2$$

into two matrices. Cutting off the singular values at the highest one, $\lambda_1$, we obtain a factorizing amplitude $M_{AB} \sim S_{AI}^2 S_{BI}^2$. (9)

We thus would summarize the edge labels of one prism to the index $A$ and the edge labels of the second prism to $B$. But it seems that we need also to split four of the edges, which are shared by the two prism halves. Again because of the Gauss constraints (now for the new cut faces), these are actually just three independent variables.

We could copy each of the independent labels $k_e$ to a pair $k_e, k_e'$, multiply the cube amplitude with Kronecker delta, $\delta(k_e, k_e')$ and then proceed with a singular value decomposition into two matrices, where $k_e$ and $k_e'$ will be associated to the two different prisms.

However, the Kronecker delta $\delta(k_e, k_e')$ leads to matrix $M_{AB}$ in block form, where the blocks are labelled by the three independent shared edge variables $k_e = k_e'$. Thus instead of proceeding with an SVD for the full matrix at once (and just keeping one of the SV’s) we perform a singular value decomposition for each of the blocks and keep one SV per block (thus $K^3$ SV’s in total).

That is, we actually perform SVD’s of matrices $M_{A'B'}^{\{k_e\}}$ which are labelled by three independent shared edge labels $\{k_e\}$. The index $A'$ summarizes all the other (initially two independent) edge indices of one prism, and $B'$ the other (initially two independent) edge indices of the other prism.

This will result in amplitudes associated to the prisms $A_p$, with four possible prisms resulting from cutting the cube along the two possible face diagonals. The prisms...
are glued to cubes as depicted in figure 7. These cubes come with a further decoration, namely four half diagonal edges on two opposite sides. Due to Gauss constraints these (eight) additional edges just amount to two further independent variables for the new cubes. These variables can be associated to the faces.

We now rotate the cubes and start the splitting into prisms again. It turns out that we only have to subdivide undecorated faces. During the gluing of the prism the only summation is actually over the decorated faces - thus for gluing a cube we just have to sum over two variables, as the edge shared by all four prisms is Gauss constrained by outer edges. This summation can be done in steps, by gluing first two prisms to half cubes and then just multiplying these to find the new cubes amplitude. Hence the computational complexity scales with $K^9$ for the last and the most expensive step. It takes negligible time on a laptop for the Ising gauge model in this lowest approximation (less than a second for 50 iterations).

The Ising gauge models flows either to the high temperature / strong coupling phase with $\lambda^{k_1,k_2,k_3} = \delta_{k_1,1}\delta_{k_2,1}\delta_{k_3,1}$ or the low temperature / weak coupling phase with $\lambda^{k_1,k_2,k_3} = 1 \ \forall k_1, k_2, k_3$. In the leading order approximation we observe the phase transition at $\beta_c = 0.703$, within 8% of the critical temperature $\beta_c = 0.761$ found by Monte Carlo methods [35–37]. By comparison, the simplest isotropic Migdal Kadanoff recursion relation [16, 19, 20] leads to a phase transition $\beta_c = 0.683$.

Apart from the simplicity of this algorithm compared to the other options, let us point out another advantage: this is in performing the SVD in block form, where the blocks are labelled by variables, which are straightforward to interpret. In fact, for spin foams the representation labels carry geometric information and represent in 3D indeed the length of the edges. The same strategy applied to the spin net models provided invaluable information about the phase space structure [18] and in general offers an easy way to monitor the outcome of the coarse graining algorithm. For non-Abelian models, the block form SVD is in parallel to the block form in spin net models and the block labels can be interpreted as intertwiner channels. This enforces the hope that spin nets and spin foams do indeed share statistical properties [13, 14, 17].

**Improving the truncation**

In order to improve the truncation one would take into account more than one singular value per block in the process of splitting of cubes into prisms. This requires the introduction of additional indices on the faces, here denoted by $I_f \ldots$, that describe the correlations between the two split parts of the cube.

In the following iterations, faces are split into triangles and hence face indices have also have to be ‘divided’: We do so by putting the face index $I_f$ onto the newly introduced diagonal edge $e$ and denote it by $I_e (= I_f)$, the diagonal $e$ now carries an index pair $(k_e, I_e)$. Thus if we glue the entire complex, the index on the diagonal edge will ensure that the four triangular faces that are glued in pairs onto each other, carry the same index $I_e$, as is the case before cutting. Eventually, all the faces will carry variables $I_f$, and all the edges will have variable pairs $(k_e, I_e)$. We define the initial amplitude as

$$A_e = \prod_{e \in c} (\hat{\omega}(k_e))^{1/4} \prod_{I_e,1} \delta_{I_e,1} \prod_{I_f,1} \delta_{I_f,1},$$

(10)

where the introduced indices can be understood as (initially empty) higher-order corrections decorating the cube amplitude, cf. figure 6.

Let us return to the process of splitting the cube into prisms, which is implemented by a SVD. There are now three independent indices $k_e$ and four indices $I_e$ that are shared by both prisms. These indices could therefore be used as parameters for the SVD, as described before in the case of just having the three $k_e$ indices. One would now have $K^3 \times \chi_e^5$ blocks where $\chi_e$ is the lengths of the $I_e$ indices. Assuming we choose also $\chi_e$ singular values per block we would take into account $K^3 \times \chi_e^5$ singular values in this splitting and thus have an ‘effective’ bond dimension of $\chi_{\text{eff}} = K^3 \times \chi_e^5$.

However we have now a situation where one expects that the singular values from a block labelled by $I_e = 2$ for all four edges are much smaller than from the block labelled by $I_e = 1$. This can be accommodated by comparing the singular values for all block labels among each other and selecting the $\chi$ largest ones, where $\chi$ is now a choice for the effective bond dimension. The $\chi$ largest singular values have to be distributed among the various blocks, resulting into blocks of varying size and also into vanishing blocks if these do not get any singular values assigned. The size of an index $I_e$ can thus get reduced a posteriori. This makes the algorithm much more complicated to implement. A possibility, to be explored in future work, is to fill entries up with zeros to reach a constant block size and then dealing with sparse matrices or tensors.

---

6 This can be done by fixing the number of singular values per block, and thus working with tensors or pre–determined size. A more complicated version [13, 14] is to compare singular values across blocks and introduce a global cut–off for the number of singular values. Thus the size of the blocks might actually vary.

Experience with the models in [13, 14] has shown that reducing the size of a block might lead to instabilities, which can be addressed by introducing an algorithm in which the block size can increase, but not decrease.

7 See [13, 14, 38] for an implementation of an algorithm with varying block size in 2D.
An alternative procedure is to keep just the block form with the $k_e$ indices as parameters and to accommodate the $I_e$ indices by first multiplying the amplitude with Kronecker delta’s $\delta_{I_e I'_e}$ and then to split by an SVD with the $I_e$ indices assigned to one prism and the $I'_e$ indices assigned to another prism. This automatically compares the singular values across the $I_e$ block structure, with a constant block size $\chi_c$ (i.e. $K^3 \times \chi_c$ singular values are taken into account with each splitting), however with the disadvantage that one has to split factors of Kronecker deltas with an SVD.

After the splitting of the cubes into prisms one proceeds as before, gluing four prisms to a cube in two steps. We sum over only the bulk variables, i.e. $I_e$ and $k_e$ labels on the half diagonals, as well as face indices $I_f$ for the faces without half diagonals. The most expensive step is the gluing of the two pairs of prisms into the cube, the cost of which scales with $K^9 \times \chi_c^2$.

A reduction in costs can be obtained by summarising the variables on the top and bottom faces (with total bond dimension $\chi_c^4$ each), $T$ and $B$ respectively, into single face indices, $T'$ and $B'$ (treating the one independent $k_e$ index of one half diagonal as parameter). Such a truncation map, introducing a cut-off $\chi_c^4 \rightarrow \chi_c$, can be implemented using the SVD:

$$A_{\{BRT\}} = \sum_{T'=1}^{\chi} U_{\{BR\}, T' \lambda T'} V^*_{T'T},$$

(11)

where $R$ denotes all the remaining indices. After applying the truncation map on both sides (here $(V^\dagger)_T T'$ from the right and $V_{B'B}$ from the left) the new truncated amplitude becomes

$$A_{\{B'RT'\}} = \sum_{B=1}^{\chi^4} V_{B'B} U_{\{BR\}, T' \lambda T'} .$$

(12)

In fact we could have also chosen $U$ as a truncation map. We can apply the technique from [3] to determine which of the maps, $U$ or $V$, gives a better truncation by comparing the root mean square values of the rejected singular values in both cases.

The simplification reduces the memory usage from $O(K^9 \times \chi_c^{24})$ to $O(K^9 \times \chi_c^{18})$, which allows for numerical testing of this higher-order algorithm. For the 3D Ising gauge model we obtain an improved critical temperature, $\beta_c = 0.710$ for $\chi_c = 2$.

The memory and computational costs of this improvement scheme are quite high, also because additional indices are associated to edges and faces. An alternative scheme only introduces additional face indices. Imagine we put a vertex in the middle of the cube and attach a tensor with six edges carrying the $I_f$-type indices. Then this splitting would correspond to splitting of a tensor network edge. To do so we proceed as discussed before: We formally replace the face index $I_f$ with a pair $(I_f, I'_f)$ (for the two triangular faces) and multiply the amplitude of the cube with two Kronecker deltas $\delta(I, I')$ for the two pairs of triangular faces. We then proceed with an SVD, treating the $k_e$ indices as parameters but not the two $I_f$ indices that are being split. Thus, the number of singular values taken into account at each splitting is $K^3 \times \chi_f$, that is $\chi_f$ singular values per block labelled by the three independent indices $k_e$ of the newly introduced face. This defines the effective bond dimension, i.e. the number of indices per tensor leg, as $K^3 \times \chi_f$.

Proceding as before with gluing the prisms to cubes, the cost of the algorithm goes as $K^9 \times \chi_f^{14}$. We can reduce it further by applying the embedding maps on the top and bottom faces in two steps: $\chi_f^2 \rightarrow \chi_f$ after gluing two prisms, and once again after two pairs of them are put together. The overall complexity of the algorithm scales with $K^9 \times \chi_f^{10}$. By comparison the (higher order singular value) 3D algorithm of [5] has scaling of $\chi_f^{11}$ for the computational costs (for models that can be brought into tensor form with bond dimension $\chi$). For the Ising model we found a steady improvement of the critical temperature with increasing $\chi_f$, reaching $\beta_c = 0.729$ for $\chi_f = 8$.

**Further simplifications**

The 3D algorithm still requires large computational resources, especially related to memory consumption of the decorated tensor, which largely pertains to dealing with the dimension higher than two. However, another reason is that the algorithm is based on cubes, which are not the most elementary building blocks. A 3D cube has 12 edges and 6 faces, whereas a tetrahedron has only 6 edges and 4 faces. The difficulty is to come up with regular coarse graining schemes involving tetrahedra. One straightforward scheme of coarse graining tetrahedra into tetrahedra right away is based on cubes subdivided into 6 tetrahedra each. In this case 8 smaller tetrahedra are coarse grained into a larger tetrahedron with all edges doubled in length.

Here we propose an algorithm based on prisms, which can be obtained from the cube algorithm described above by cutting the cubes. Instead of starting with cubes and subdividing these into prisms, we start with prisms right away, see figure [8]. We divide two of these into pairs of tetrahedra and pyramids, before re-glueing into larger prisms back again.

The advantage is, that in the leading order approximation, the basic amplitude carries fewer indices, decreasing the computation cost to $O(K^8)$, as well as the memory consumption from $O(K^9)$ to $O(K^7)$. While we find no qualitative differences to the cubic algorithm, the obtained critical temperature is near $\beta_c = 0.630$ (within 18% of the Monte Carlo result [33][37]). Although less accurate, the technique might find applications in the models with large structure groups, where the cubic al-
FIG. 8. Simplified algorithm based on prisms, tetrahedra and pyramids. Basic building blocks are now prisms, which after splitting and re-glueing form the coarse-grained polyhedra.

algorithm would be too expensive.

**OUTLOOK: NON-ABELIAN MODELS AND FOUR DIMENSIONAL MODELS**

Here the initial amplitude of the cube depends not only on the representation labels $\rho_e$ attached to the edges but also on intertwiner labels $\iota_f$ for the faces. Often, these face intertwiner labels can be chosen to be equal to representation labels $\rho_f$, describing the coupling of a pair of edges via $\rho_f$ to the other pair of edges. If this pair of edges is chosen to be adjacent, then $\rho_f$ can be attached to a corresponding diagonal as in the Abelian case. Details for this algorithm for Non-Abelian groups will appear elsewhere, together with an application to spin foams with quantum group symmetry.

Let us also point out that the algorithms generalize to four dimensions. We have now hypercubes, the (in 3D) edge indices $\rho_e$ are replaced by plaquette indices (i.e. attached to the two-dimensional objects) $\rho_p$ and the (in 3D) face indices $\rho_f$ are now attached to the 3D cubes which make up the boundary of the hypercube. Thus the splitting and coarse graining can as before be considered to take place in a plane.

**DECORATED TENSOR NETWORKS FOR OTHER MODELS**

In this section we lay out that the main idea of the decorated tensor network algorithm can also be applied to other models, besides (generalized) lattice gauge theories.

Let us describe the general structure of decorated tensor models for lattice gauge theories: A tensor carrying the indices $I, J, M, \ldots$ sits in the centre of each cube with its edges piercing the faces of the cube, equipping them with indices. The tensors in neighbouring cubes are glued together by contracting the index of the edge connecting them, which is equivalent to summing over the index on the common face. In addition, this tensor network comes with supplemental decorations, i.e. the labels attached to the edges of the cube. In fact these are the variables which keep most of the physical information. Indeed one can imagine to contract the ‘internal’ tensor network by summing over all indices $I, J, M, \ldots$ and be left with a model with the original type of variables.

This model would in general be non-local, if the $I, J, M, \ldots$ indices have non-trivial range. Thus we can directly understand how non-local couplings, which appear in other real space renormalization schemes, are turned into additional indices and how the cut-off on these indices can also be understood as a truncation of non-local interactions that can occur, see also the discussion in [32, 39].

Let us proceed with the application of this idea to Ising-like models in 2D. The algorithm will amount to a sophisticated decimation procedure, i.e. keep a subset of the original Ising spins, separated at larger and larger distances, while the decimated spins are absorbed into a tensor network. These Ising spins can serve as signature observables, for instance one can straightforwardly compute a correlation function from the final tensor (this will be however at e.g. a distance half the system size in the contraction of six tensors to a torus topology).

Again instead of starting with a tensor or a vertex model, we rather think of squares as the basic building blocks. To these squares we assign an initial amplitude, which is a function of the four Ising spins sitting on the four corners of the square. Additionally, we will use a dual lattice for the tensor network, i.e. each square is decorated with a vertex in its centre with four emanating edges that cut the edges of the squares at the midpoints. The indices on these tensor edges will allow to go to a higher order approximation. Thus we again obtain the structure of a decorated tensor network.

The coarse graining algorithm is a straightforward adaptation of the Gu–Wen algorithm [4] taking the dec-

\[ A \text{ similar algorithm as just presented for the 3D gauge models can also be applied to 3D Ising-like models, see also figure 10.} \]
oration by the Ising spins into account, see figure 9. The idea is to cut the squares along their diagonals into triangles by a SVD. As in the Gu–Wen algorithm, this requires to split the 4-valent tensor in the center of the square into two 3-valent ones, but additionally one has to ‘split’ the Ising spins connected by the diagonal. Naively, one can double these spins and identify them by introducing Kronecker Deltas, which allows us to straightforwardly perform the SVD, yet with a significant amount of irrelevant data in case the doubled spins are not equal. Instead it is more efficient to use the pair of split spins as block labels: the matrix that we intend to split decomposes into smaller block matrices, labelled by the pair of spins, on which we apply a SVD separately. Consequently the new index of the 3-valent tensor piercing the edge connecting the pair of split spins is precisely labelled by these spins.

After splitting the square along both diagonals, we obtain four triangular amplitudes which are connected to form the new (rotated) square as in figure 9. This incorporates the summation over four tensor indices, as in the Gu–Wen algorithm, and one Ising spin in the center of the new square. In the lowest possible approximation, i.e. tensor index range one, one only sums over the Ising spin.

**Simplification – Triangular algorithm**

In fact, the code depicted in figure 9 and also the related Gu–Wen algorithm, can be simplified by shifting the perspective from square building blocks to triangular ones.

If we consider the second part of the iteration, i.e. the glueing of the triangles and contraction of the respective indices, we realize that this can be performed in two steps: First we glue two small triangles into a larger triangle, then we glue the two larger triangles together to form the new square. However, in the next iteration of the algorithm we split this square again into the larger triangles we previously glued.

Hence, instead of performing the summations and subsequent splitting of the square, we propose an algorithm purely based on triangular amplitudes, see figure 11. Therefore we keep track of four different triangles, denoted as $S_1, \ldots, S_4$: these amplitudes are pairwise glued together, once vertically, once horizontally, forming new larger triangles, where the new long edge carries two tensor indices and one Ising spin. To arrive back at an amplitude comparable to the initial one, we have to define a truncation map (three-valent tensors) blocking these indices into a new common tensor index, whose index range determines the quality of the approximation. Again these maps are computed via a (Higher Order) SVD as in [5]; note that the truncation maps applied to opposing triangles should be related by hermitian conjugation, such that they form a partition of unity (before implementing the truncation on the number of singular values).

This modification of the algorithm is not restricted to decorated tensor networks and can be readily applied also to the original Gu–Wen algorithm, for which it leads to a reduction of computational costs from $\chi^6$ to $\chi^5$.

The interesting feature of this decorated tensor network algorithm is the mixture of keeping part of the original variables and allowing for the additional tensor indices field redefinitions. An essential ingredient is a SVD per block as extensively used in the algorithms developed in [13, 14, 17, 18]. Especially for the non-Abelian models the knowledge of the singular values per block was key to be able to interpret the fixed point tensors. This might help to address some criticism to tensor network methods in [16], which points out the difficulty in extracting physical information from the fixed point tensors and separating physical from unphysical (due to the field redefinitions there is a kind of weak gauge symmetry for the tensor models) information. In terms of computation...
efficiency the algorithm is rather the same as the original Gu–Wen algorithm (apart from the reduction due to basing the algorithm on triangles instead of squares).

Results: We have applied decorated tensor network methods to the 2D Ising model defined on a square lattice. To lowest approximation, i.e. $\chi = 1$ per configuration of spins on the new edge corresponding to an effective bond dimension $\chi_{\text{eff}} = 4$, we observe a phase transition at the inverse temperature $\beta \approx 0.49664$. In comparison to the exact solution, $\beta_c = \frac{\ln(1+\sqrt{2})}{2} \approx 0.44068$, this is an error of roughly 13 %, yet still an improvement over the Migdal–Kadanoff methods, where one finds $\beta^{(\text{MK})} \approx 0.3047$, an error of roughly 31 %. Additionally, we can compute critical exponents, which essentially confirm the results presented in [16] on the Gu–Wen algorithm on the square lattice for the lowest cut-off. Exploiting the fact that decorated tensor networks preserve (some) of the original Ising spins, we can compute correlation functions. In figure 12 we show the correlation functions (and its first derivative) of two ‘neighbouring’ spins on a 2D lattice with periodic boundary conditions after different numbers of iterations, i.e. different distances between the spins on the original lattice. This correlation function clearly indicates the phase transition between the ordered and disordered phase, where the correlation functions intersect right on the phase transition because of scale invariance of the fixed point tensor.

Increasing the cut-off the phase transition temperature approaches the exact result; the highest cut-off tested is $\chi_{\text{eff}} = 60$, for which we observe a phase transition around $\beta \approx 0.440706$ (an error of roughly 0.06 %).

For the 3D Ising model our method gives in the lowest order approximation a phase transition between
Sor networks: Instead of encoding all variables into the gravity, adds to their desirability. Tpered in fermionic systems and spin foams in quantum deal with negative and complex amplitudes, as encoun-

ted at larger and larger scales. Yet their capability to also
deal with negative and complex amplitudes, as encoun-
tered in fermionic systems and spin foams in quantum gravity, adds to their desirability.

Therefore we have introduced and tested a novel tensor network renormalization algorithm, called decorated tensor networks: Instead of encoding all variables into the tensor network and thus potentially blurring the gauge symmetry (e.g. represented in the form of Gauss constraints for the representation labels), in particular after several coarse graining steps and variable redefinitions, we base our algorithms on building blocks of the discretisation, here cubes in the 3D cubical lattice, which are coloured by the original variables of the system and carry a tensor in their center; hence the tensor gets ‘decor-

ated’ by the original variables of the system.

We tested the method with the Ising gauge model in 3D, which led to acceptable results. (We are not aware of other results with tensor network algorithms applied to gauge theories.) The method seems to be in particular effective for large structure groups, which will appear in spin foams and lattice gauge theories (in fact with infi-
nite initial bond dimension). For spin foams decorated tensor networks provide in fact the first systematic coarse graining scheme.

We presented several possibilities to improve the lowest order approximations. We observed a systematic improvement of the phase transition temperature with increasing cut-off. Again it is difficult to compare with other tensor network methods – general discussions suggest that implementation of entanglement filtering [22, 40] might improve very much the effectiveness of 3D algorithms.

The decorated tensor network scheme allows for a clear identification and preservation of (part of) the original variables of the system. As an additional advantage the fixed points of this renormalization procedure are more straightforward to interpret. Note that the tensor network itself plays a distinctly different role in this algo-
rithm: Instead of wholisticly encoding the dynamics of the system, it rather appears as way of locally encoding higher order corrections. To lowest cut–off, i.e. if all ten-
sor indices are trivial, the systems keeps its original form. Higher order corrections can be understood as a higher multiplicity of particular colouring / configurations of the building blocks.

In fact, a slightly different perspective can be taken as well: the network of tensors, located in the center of the building blocks, can be contracted at any point. After this contraction one arrives back at a system depending on the same type of variables as the original ones, yet with generically non–locally interacting building blocks. Hence one can understand the decorated tensor network as encoding non–local interactions between the building blocks, where the cut–off on the tensor indices can be di-
rectly translated into a truncation of the non–local interactions, see the discussions in [32, 41]. This perspective will help to compare tensor network results with other, for instance continuum methods, based on truncations of (non–local) coupling terms in the effective action.

In this vain we hope that decorated tensor networks might eventually facilitate new methods mixing both analytialc (for the decoration indices) and numerical tech-
niques (for the tensor indices). This can possibly provide new approaches to go beyond finite groups towards Lie groups with infinite initial bond dimension and to allow to deal with (gauge) divergences occurring in spin foam models [23][25].

Regarding the application to spin foams, the decorated tensor networks allow to keep the geometric interpretation of the spin foam variables. The coarse graining process might thus allow to conclude whether the blocking of variables (as encoded in so-called embedding or truncation maps [32][41]), determined by the SVD truncation, also proceeds in a geometric manner. As models of quantum gravity spin foams pose additional challenges. One is that the notion of scale, in terms of complexity of boundary data, as well as diffeomorphism symmetry have to be emergent [11][13]. To this end non–local amplitudes are unavoidable [41][45], which indeed can be taken care off with the decorated tensor networks. As described in [41] tensor networks can be understood as a truncation scheme to compute physical states and a physical vacuum. We hope that the decorated tensor networks allow in particular to retain the geometric interpretation of the initial amplitudes, with the actual tensor network providing an improvement [46][47] over the bare (initial) amplitudes.

ACKNOWLEDGEMENTS

The authors thank Leo Kadanoff for an enlightening discussion. S.M. would like to thank Perimeter Institute for hospitality during this work and Girton College, Cambridge for support. S.St. would like to thank Perimeter Institute for an Isaac Newton Chair Graduate Research Scholarship. This research was supported by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through Industry Canada and by the Province of Ontario through the Ministry of Research and Innovation.

[1] T. Nishino and K. Okunishi, “Corner Transfer Matrix Renormalization Group Method,” J. Phys. Soc. Jpn. 65, 891 (1996), arXiv:cond-mat/9507087 [cond-mat].

[2] N. Maeshima, Y. Hieida, Y. Akutsu, T. Nishino, and K. Okunishi, “Vertical density matrix algorithm: A Higher dimensional numerical renormalization scheme based on the tensor product state ansatz,” Phys.Rev. E64, 016705 (2001), arXiv:cond-mat/0101360 [cond-mat].

[3] M. Levin and C. P. Nave, “Tensor renormalization group approach to 2d classical lattice models,” Phys. Rev. Lett. 99, 120601 (2007), arXiv:cond-mat/0611687 [cond-mat].

[4] Z.-C. Gu and X.-G. Wen, “Tensor-Entanglement-Filtering Renormalization Approach and Symmetry Protected Topological Order,” Phys. Rev. B 80, 155131 (2009), arXiv:0903.1069 [cond-mat.str-el].

[5] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, “Coarse-graining renormalization by higher-order singular value decomposition,” Phys. Rev. B 86, 045139 (2012), arXiv:1201.1144 [cond-mat.stat-mech].

[6] C. Rovelli, Quantum gravity (Cambridge University Press, Cambridge, 2004).

[7] C. Rovelli, “Zakopane lectures on loop gravity,” PoS QGQS2011, 003 (2011), arXiv:1102.3660 [gr-qc].

[8] A. Perez, “The Spin Foam Approach to Quantum Gravity,” Living Rev.Rel. 16, 3 (2013) arXiv:1205.2019 [gr-qc].

[9] B. Bahr, B. Dittrich, and J. P. Ryan, “Spin foam models with finite groups,” J.Geom. Phys. 2013, 549824 (2013), arXiv:1103.6264 [gr-qc].

[10] B. Bahr, B. Dittrich, F. Hellmann, and W. Kaminiski, “Holonomy Spin Foam Models: Definition and Coarse Graining,” Phys.Rev. D87, 044048 (2013), arXiv:1208.3388 [gr-qc].

[11] L. Tagliacozzo and G. Vidal, “Entanglement renormalization and gauge symmetry,” Phys.Rev. B83, 115127 (2011), arXiv:1007.1415 [cond-mat-str-el].

[12] L. Tagliacozzo, A. Celli, and M. Lewenstein, “Tensor Networks for Lattice Gauge Theories with continuous groups,” (2014), arXiv:1405.4811 [cond-mat.str-el].

[13] B. Dittrich, F. C. Eckert, and M. Martin-Benito, “Coarse graining methods for spin net and spin foam models,” New J. Phys. 14, 035008 (2012), arXiv:1109.4927 [gr-qc].

[14] B. Dittrich and F. C. Eckert, “Towards computational insights into the large-scale structure of spin foams,” J.Phys.Conf.Ser. 360, 012004 (2012), arXiv:1111.0967 [gr-qc].

[15] Y. Liu, Y. Meurice, M. P. Qin, J. Unnuth-Yockey, T. Xi, Xiang, Z. Y. Xie, J. F. Yu, and H. Zou, “Exact blocking formulas for spin and gauge models,” Phys.Rev. D88, 056005 (2013), arXiv:1307.6543 [hep-lat].

[16] E. Efrati, Z. Wang, A. Kolan, and L. P. Kadanoff, “Real Space Renormalization in Statistical Mechanics,” Rev.Mod. Phys. 86, 647–667 (2014) arXiv:1301.6323 [cond-mat.stat-mech].

[17] B. Dittrich, M. Martin-Benito, and E. Schnetter, “Coarse graining of spin net models: dynamics of intertwiners,” New J. Phys. 15, 103004 (2013), arXiv:1306.2987 [gr-qc].

[18] B. Dittrich, M. Martin-Benito, and S. Steinhaus, “Quantum group spin nets: refinement limit and relation to spin foams,” Phys.Rev. D 90, 024058 (2014) arXiv:1312.0905 [gr-qc].

[19] A. A. Migdal, “Recursion Equations in Gauge Theories,” Sov.Phys.JETP 42, 413 (1975).

[20] L.P. Kadanoff, “Notes on Migdal’s Recursion Formulas,” Annals Phys. 100, 359–394 (1976).

[21] M. Levin, “Real space renormalization group and the emergence of topological order,” (2007), talk at workshop: “Topological Quantum Computing” at IPAM, UCLA, Los Angeles.

[22] A. J. Ferris, “The area law and real-space renormalization,” Phys. Rev. B 87, 125139 (2013) arXiv:1301.2608 [cond-mat.str-el].

[23] C. Perini, C. Rovelli, and S. Speziale, “Self-energy and vertex radiative corrections in LQG,” Phys. Lett. B682, 78–84 (2009), arXiv:0810.1714 [gr-qc].
A. Riello, “Self-energy of the Lorentzian Engle-Pereira-Rovelli-Livine and Freidel-Krasnov model of quantum gravity,” Phys.Rev. D88, 024011 (2013), arXiv:1302.1781 [gr-qc].

V. Bonzom and B. Dittrich, “Bubble divergences and gauge symmetries in spin foams,” Phys.Rev. D88, 124021 (2013) arXiv:1304.6632 [gr-qc].

A. Feiguin, S. Trebst, A. W. W. Ludwig, M. Troyer, A. Kitaev, Z. Wang, and M. H. Freedman, “Interacting anyons in topological quantum liquids: The golden chain,” Phys. Rev. Lett. 98, 160409 (2007) arXiv:cond-mat/0612341 [cond-mat.str-el].

C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, “Non-Abelian anyons and topological quantum computation,” Rev.Mod.Phys. 80, 1083–1159 (2008) arXiv:0707.1889 [cond-mat.str-el].

B. Dittrich and W. Kamiński, “Topological lattice field theories from intertwiner dynamics,” arXiv:1311.1798 [gr-qc].

J. B. Kogut, “An Introduction to Lattice Gauge Theory and Spin Systems,” Rev.Mod.Phys. 51, 659 (1979).

S. Singh and G. Vidal, “Tensor network states and algorithms in the presence of a global SU(2) symmetry,” Phys.Rev. B86, 195114 (2012), arXiv:1208.3919 [cond-mat.str-el].

R. Savit, “Duality in Field Theory and Statistical Systems,” Rev.Mod.Phys. 52, 453 (1980).

B. Dittrich, “From the discrete to the continuous: Towards a cylindrically consistent dynamics,” New J.Phys. 14, 123004 (2012) arXiv:1205.6127 [gr-qc].

H. A. Kramers and G. H. Wannier, “Statistics of the two-dimensional ferromagnet. Part I,” Phys.Rev. 60, 252–262 (1941).

H. A. Kramers and G. H. Wannier, “Statistics of the Two-Dimensional Ferromagnet. Part II,” Phys.Rev. 60, 263–276 (1941).

K. Binder and E. Luijten, “Monte Carlo tests of renormalization group predictions for critical phenomena in Ising models,” Phys.Rept. 344, 179–253 (2001).

H.-O. Heuer, “Critical crossover phenomena in disordered ising systems,” Journal of Physics A: Mathematical and General 26, L333 (1993).

M. E. Fisher, “The theory of equilibrium critical phenomena,” Reports on Progress in Physics 30, 615 (1967).

F. C. Eckert, Coarse graining simplified spin foam models, Master’s thesis, University of Heidelberg (2012), diploma thesis.

Bianca Dittrich and Sebastian Steinhaus, “Time evolution as refining, coarse graining and entangling,” arXiv:1311.7565 [gr-qc].

G. Vidal, “Entanglement Renormalization,” Phys.Rev.Lett. 99, 220405 (2007) arXiv:cond-mat/0512165 [cond-mat].

B. Dittrich, “The continuum limit of loop quantum gravity - a framework for solving the theory,” (2014), arXiv:1409.1450 [gr-qc].

B. Dittrich, “Diffeomorphism symmetry in quantum gravity models,” Adv. Sci. Lett. 2, 151 (2009), arXiv:0902.3594 [gr-qc].

B. Dittrich, “How to construct diffeomorphism symmetry on the lattice,” PoS QGQGS2011, 012 (2011), arXiv:1201.3840 [gr-qc].

B. Dittrich and S. Steinhaus, “Path integral measure and triangulation independence in discrete gravity,” Phys. Rev. D 85, 044032 (2012) arXiv:1110.6866 [gr-qc].

B. Dittrich, W. Kamiński, and S. Steinhaus, “Discretization independence implies non-locality in 4D discrete quantum gravity,” (2014), arXiv:1404.5288 [gr-qc].

B. Bahr and B. Dittrich, “Improved and Perfect Actions in Discrete Gravity,” Phys. Rev. D 80, 124030 (2009), arXiv:0907.4323 [gr-qc].

B. Bahr, B. Dittrich, and S. Steinhaus, “Perfect discretization of reparametrization invariant path integrals,” Phys. Rev. D 83, 105026 (2011) arXiv:1101.4775 [gr-qc].