SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area – PhD course in

Geometry and Mathematical Physics

Topological Invariants of Plumbed 3-Manifolds

Candidate:

Song Jin Ri

Advisor:

Prof. Pavel Putrov

Academic Year 2022-23

Abstract

In this thesis, we deal with topological invariants of plumbed 3-manifolds by means of not only mathematical analysis but also deep machine learning with Graph Neural Networks (GNN).

We first introduce a two-variable refinement $\hat{Z}_a(q,t)$ of plumbed 3-manifold invariants $\hat{Z}_a(q)$, which were previously defined for weakly negative definite plumbed 3-manifolds. For plumbed 3-manifolds with two high-valency vertices, we analytically compute the limit by using the explicit integer solutions of quadratic Diophantine equations in two variables. Based on numerical computations of the recovered $\hat{Z}_a(q)$ for plumbings with two high-valency vertices, we propose a conjecture that the recovered $\hat{Z}_a(q)$, if exists, is an invariant for all tree plumbed 3-manifolds. Then we suggest a formula of the $\hat{Z}_a(q,t)$ for the connected sum of plumbed 3-manifolds in terms of those for the components.

Next, we test the efficiency of applying Geometric Deep Learning to the problems in low-dimensional topology in a certain simple setting. Specifically, we consider the certain class of 3-manifolds described by plumbing graphs and apply GNNs to the problem of deciding whether a pair of graphs give homeomorphic 3-manifolds. We use supervised learning to train a GNN that provides the answer to such a question with high accuracy. Moreover, we also consider reinforcement learning by a GNN to find a sequence of Neumann moves that relates the pair of graphs if the answer is positive.

Preface

This thesis is based on the following publications:

- Song Jin Ri, "Refined and Generalized $\hat{Z}$ Invariants for Plumbed 3-Manifolds", *SIGMA* **19**(2023), 011, arXiv:2205.08197.
- Song Jin Ri and Pavel Putrov, "Graph Neural Networks and 3-Dimensional Topology", Mach. Learn.: Sci. Technol. vol.4 035026 (**2023**).

Acknowledgements

First of all, I am deeply thankful to my supervisor, Professor Pavel Putrov, who has willingly shared his precious time to help me with his useful comments and remarks through each stage of my entire process. I would like to acknowledge professors: Jacopo Stoppa, Alessandro Tanzini, Tamara Grava, and Rafael Torres. They gave me plenty of knowledge to lead me to the beautiful world of mathematics by their great efforts through wonderful lectures at the first year of my PhD. Many thanks to my PhD collegues who supported me greatly and were always willing to help me.

I would like to express the deepest appreciation to Professor Jin U Kang and Professor Hak Chol Pak in Kim Il Sung University, who continually and convincingly conveyed a spirit of adventure in regard to research, and an excitement in regard to teaching. I would also like to thank Kwang Il Ryom for always being with me in thicks and thins of 5-years-life in Trieste.

Last but not least, I would like to express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout 4 years of study and through the process of researching and writing this thesis.

Song Jin Ri
October 13, 2023
Trieste, Italy

Contents

To my family, for their constant love and support.

Chapter 1

Introduction

It is well-known that Dehn surgery provides a method to produce nontrivial 3-manifolds starting from a framed link in $\mathbb{R}^3$ and this method establishes a pivotal relation between links in $\mathbb{R}^3$ and closed oriented 3-manifolds [29, 44]. It also gives rise to an important relation between invariants of links and 3-manifolds. A typical example is the colored Jones polynomial for framed oriented links and the Witten–Reshetikhin–Turaev (WRT) invariant for closed oriented 3-manifolds [38, 45].

Khovanov homology [23] is another well-known invariant for knots and links. Furthermore, it is a categorification of the Jones polynomial, that is, the Euler characteristic of Khovanov homology is the Jones polynomial. Therefore, it is natural to ask if there is an invariant of 3-manifolds not only as the counterpart of Khovanov homology but also as a categorification of the WRT invariants. This is one of the major open questions in quantum topology.

Some progress in this direction has been made in [15, 14], where a physical definition of certain new invariants of 3-manifolds, often referred to as homological blocks, Gukov–Pei–Putrov–Vafa (GPPV) invariant, or as $\hat{Z}$ invariant, denoted by $\hat{Z}_a(q)$, was formulated by the study of $3d$ $\mathcal{N} = 2$ supersymmetric theory obtained by compactification of $6d$ $\mathcal{N} = (2,0)$ theory on a 3-manifold. The $3d$ $\mathcal{N} = 2$ theory depends on the choice of a Lie group G and the case $G = \mathrm{SU}(2)$ is mostly studied in [13, 15, 14] since it corresponds to the Jones polynomial. In this paper, we will also assume $G = \mathrm{SU}(2)$. The cases for other gauge groups, especially for $G = \mathrm{SU}(N)$ are studied in [36]. The invariant $\hat{Z}_a(q)$ is a power series in q with integer coefficients and a denotes a Spin^c structure of the 3-manifold. Moreover, a general relation between $\hat{Z}$ invariants and WRT invariants was conjectured. The conjecture says that the limit of a certain linear combination of $\hat{Z}_a(q)$ over all possible a as q goes to a root of unity would be equal to the WRT invariant. A rigorous mathematical definition of the invariants for general 3-manifolds is yet to be found. A concrete mathematical formula of $\hat{Z}_a(q)$ and its invariance are given in [13] and [15] only for weakly negative definite

plumbed 3-manifolds.

The main context of this thesis is divided into two parts. In the first part of this thesis we introduce a refined and generalized version of these q-series, which is defined for a much larger class of plumbed 3-manifolds. To do this, we first construct a refinement, the (q,t)-series, by introducing a new regulator variable t, which is also an invariant of *reduced* plumbed 3-manifolds. This (q,t)-series has a nice property that for weakly negative definite plumbings the evaluation at $t = 1$ is equal to the q-series $\hat{Z}_a(q)$. However, for weakly positive definite plumbings we can switch the (q,t)-series into the q-series in [13] by taking $t = 1$ and transforming the expansion in q^{-1} into an expansion in q as in [5]. Also, the (q,t)-series for negative definite plumbings is reminiscent of the 2-variable series introduced in [1], which is obtained by combining the lattice cohomology and the q-series.

Another property of the (q,t)-series is that the exponents of t are intimately related to non-negative integer solutions of quadratic Diophantine equations. Therefore, we can recover the q-series from the (q,t)-series by computing the limit $t \to 1$ even for a certain class of strongly indefinite plumbings by solving the corresponding quadratic Diophantine equations. In cases when the limit is finite, the recovered q-series is conjecturally also an invariant of plumed 3-manifolds.

In the second part of this thesis, we examine the Graph Neural Networks (GNN) approach to the problems in 3-dimensional topology, which ask whether two given plumbing graphs represent a same 3-manifold or not, and how to find out the sequence of Neumann moves that connects two plumbings if they are equivalent.

Geometric Deep Learning (GDL) [3] is an area of Machine Learning (ML) that has been under very active development during the last few years. It combines various approaches to ML problems involving data that has some underlying geometric structure. The neural networks used in GDL are designed to naturally take into account the symmetries and the locality of the data. It has been successfully applied to problems involving computer vision, molecule properties, social or citation networks, particle physics, etc (see [4] for a survey). It is natural to apply GDL techniques also to mathematical problems in topology. In general, ML has been already used in various problems in low-dimensional topology, knot theory in particular, [19, 20, 16, 9, 21, 8, 43, 22, 30, 17], as well as various physics-related problems in geometry (for a recent survey see [18]). However, the used neural network models were mostly not specific to GDL.

The goal of the second part of this thesis is to test the efficiency of GDL in a very simple setting in low-dimensional topology. Namely, we consider a

special class of 3-manifolds known as plumbed, or graph, 3-manifolds. Those are 3-manifolds that are specified by a choice of a graph with particular features assigned to edges and vertices. Such 3-manifolds are therefore very well suited for analysis by Graph Neural Networks (GNN). GNN is one of the most important and used types of neural networks used in GDL. In general, GNN are designed to process data represented by graphs.

In this thesis, we use GNNs for the following problems involving plumbed 3-manifolds. Different (meaning not isomorphic) graphs can correspond to equivalent, i.e. homeomorphic, 3-manifolds. Note that in 3 dimensions (or less) any topological manifold has a unique smooth structure and the notions of homeomorphism and diffeomorphism are equivalent. It is known that a pair of graphs that produce two equivalent 3-manifolds must be related by a sequence of certain *moves*, commonly known as Neumann moves [34]. These moves establish a certain equivalence relation on the graphs (in addition to the standard graph isomorphism). First, we consider a neural network that, as the input has a pair of plumbing graphs, and, as the output gives the decision whether the graphs correspond to homeomorphic 3-manifolds or not, i.e. whether the two graphs are equivalent, or not, in the sense described above. Supervised Learning (SL) is then used to train the network. The training dataset consists of randomly generated graph pairs, for which it is known whether the corresponding 3-manifolds are homeomorphic or not. The trained neural network, up until the very last layer, can be understood to produce an approximate topological invariant of plumbed 3-manifolds.

Second, we consider a neural network for which the input is a plumbing graph and the output is a sequence of Neumann moves that "simplifies" the graph according to a certain criterion. The aim is to build a neural network such that if it is applied to equivalent graphs it simplifies them to the same graph. If the result is successful this can be used to provide an explicit demonstration that a given pair of graphs give two homeomorphic 3-manifolds. Reinforcement Learning (RL) is used to train the network. For both cases, SL and RL, we consider different architectures of the neural networks and compare their performance.

Note that in principle there is an algorithm for determining whether two plumbing graphs give homeomorphic 3-manifolds or not, which was already presented in [34]. It involves bringing both graphs to a certain normal form (which is, in a sense, similar to the "simplification" process in the RL setup mentioned above) and then checking that normal forms are the same (i.e. isomorphic graphs). However, it is known that just checking isomorphism of graphs already goes beyond polynomial time. The plumbing graphs can be

considered as a particular class of more general Kirby diagrams that can be used to describe arbitrary closed oriented 3-manifolds, with Neumann moves being generalized to the so-called Kirby moves. Even in this case, in principle there exists an algorithm of checking whether two Kirby diagrams produce homeomorphic 3-manifolds or not [26]. There is a also of version of Kirby diagrams and moves for smooth 4-manifolds. Moreover, in this case, however, an algorithm for the recognition of diffeomorphic pairs does not exist. In 4 dimensions the notion of diffeomorphism and homeomorphism are not the same. In particular, there exist pairs of manifolds that are homeomorphic but not diffeomorphic. While the classification of 4-manifolds up to homeomorphisms (with certain assumptions on the fundamental group) is relatively not difficult, classification up to diffeomorphisms is an important open question. The setup with plumbed 3-manifolds that we consider in this thesis can be understood as a toy model for the problem of recognition of diffeomorphic pairs of general 3- and 4-manifolds, for which one can try to apply neural networks with similar architecture in the future.

The rest of the thesis is organized as follows. In Chapter 2, we review basic preliminaries about the plumbed 3-manifolds, the q-series $\hat{Z}_a(q)$, and some GNN architectures needed for the analysis that follows. In Chapter 3, we provide the formula for the $\hat{Z}_a(q,t)$ invariants of reduced plumbed 3-manifolds, and prove that they are independent of the plumbing representation. We give an analytical study on the recovering process for plumbings with two high-valency vertices by using the solutions of quadratic Diophantine equations in two variables. Also, we propose a conjecture that the recovered q-series is an invariant of arbitrary plumbed 3-manifolds. Then we provide a relation between the (q,t)-series of the disjoint union of two plumbing graphs and those of component plumbings. In Chapter 4, we first consider various GNN architectures for supervised learning of whether a pair of plumbing graphs provide homeomorphic 3-manifolds or not. We also consider reinforcement learning of the process of simplification of a plumbing graph representing a fixed (up to a homeomorphism) 3-manifold. Then we discuss the obtained results and mention possible further directions. In Appendices A and B, we provide a brief review on the solutions of quadratic Diophantine equations in two variables, the results of which are used in Chapter 3, and some basic algorithms that are specific to the problems considered in Chapter 4.

Chapter 2

Preliminaries

2.1 Plumbed 3-manifolds and q-series

In this section we review some known facts about plumbed 3-manifolds and their invariants, and we set up notational conventions.

2.1.1 Plumbing graphs

A *plumbing graph* is a finite weighted graph Γ, that is, a graph consisting of a finite number of vertices and edges together with the data of integer weights associated to vertices. In Section 2.1–3.2, we will assume that Γ is a tree, and we will consider disconnected plumbings in Section 3.3.

Let V be the set of vertices of Γ. For each vertex $v \in V$, m_v denotes the weight of the vertex v, and the degree $\deg(v)$ describes the number of edges connected to the vertex. Let $s = |V|$ be the cardinality of the set V. Then we define the symmetric $s \times s$ matrix $M = M(\Gamma)$, called *linking matrix* of Γ, by

$$M_{v_1,v_2} = \begin{cases} m_v, & \text{if } v_1 = v_2 = v, \\ 1, & \text{if } v_1,\ v_2 \text{ are connected by an edge,} \\ 0, & \text{otherwise.} \end{cases} \qquad v_i \in V.$$

From Γ, we can construct plumbed 3-manifolds in the following way: we first obtain the framed link $L(\Gamma)$ in $S^3 = \partial B^4$ by taking an unknot with framing m_v for each vertex v and making these unknots forming Hopf links whenever corresponding vertices are connected by an edge, e.g., see Figure 2.1. By attaching two-handles to B^4 along $L(\Gamma)$, we get the 4-manifold, denoted by $W(\Gamma)$. It can be also obtained by plumbing disk bundles over S^2 with Euler numbers m_v. Then its boundary $Y = Y(\Gamma) = \partial W(\Gamma)$ is the closed and oriented 3-manifold obtained

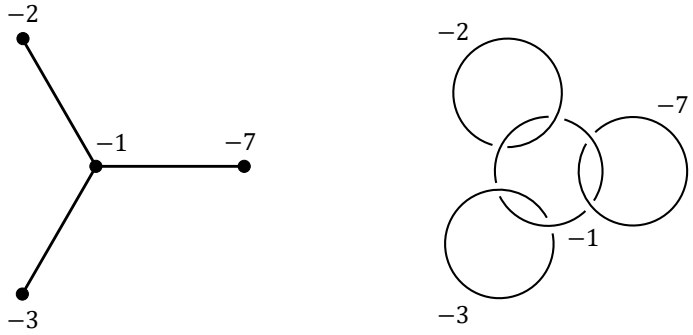


FIGURE 2.1: A plumbing graph Γ on the left and its associated framed link $L(\Gamma)$ on the right.

by Dehn surgery from $L(\Gamma)$ whose first homology is

$$H_1(Y) \cong \mathbb{Z}^s / M\mathbb{Z}^s.$$

Two different plumbing graphs can represent the same homeomorphism class of $Y(\Gamma)$ as following theorem [34, Theorem 3.2].

Theorem 2.1. *If Γ_1 and Γ_2 are connected plumbings with no cycles, then Γ_1 and Γ_2 are related by Neumann moves depicted in Figure 2.2.*

Since Neumann moves of type (d) is relevant to disconnected plumbings, we will restrict ourselves to a certain class of plumbing graphs (defined explicitly later) such that any two plumbings that represent the homeomorphic 3-manifolds are related by only three types of Neumann moves, i.e., type (a), (b), and (c).

For our purpose, we establish some terminology of plumbings that can be helpful for understanding the structures of plumbing graphs. Given a graph Γ, we divide the set V of vertices into three disjoint subsets V_1, V_2 and V_h with respect to the degree of vertices as following:

$$V_1 = \{v \in V \mid \deg(v) = 1\}, \qquad V_2 = \{v \in V \mid \deg(v) = 2\},$$
$$V_h = \{v \in V \mid \deg(v) = 2 + p_v \geq 3,\ p_v \in \mathbb{Z}_{>0}\}.$$

Then we name the elements of V_1, V_2 and V_h by *valency one*, *valency two* and *high-valency vertices*, respectively.

Definition 2.2. Suppose that Γ has at least one high-valency vertex.

(i) An *arm* is a connected component of $\Gamma \setminus V_h$.

(ii) An arm is called a *bridge* if its two endpoints are in V_h.

(iii) An arm is called *branch* if one endpoint is in V_h and the other is in V_1.

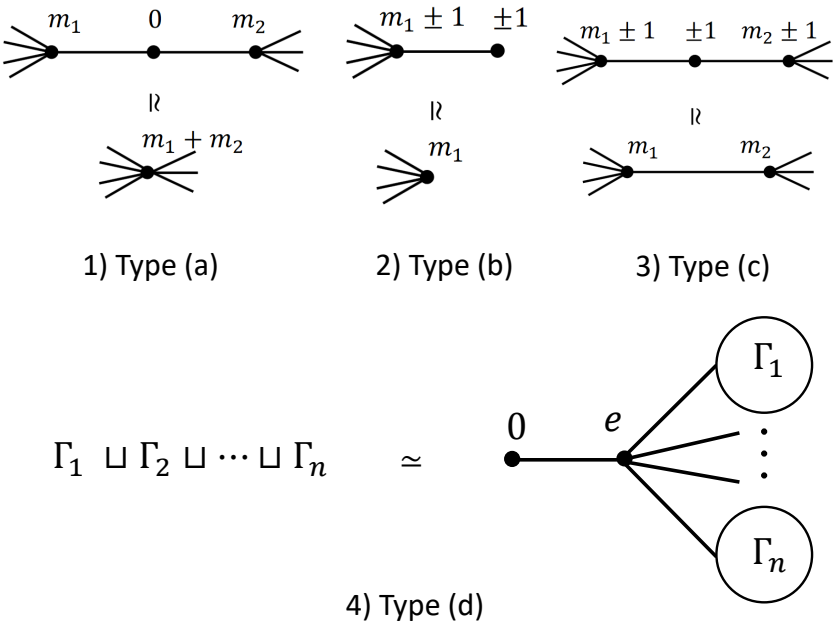


FIGURE 2.2: There are 4 different types of Neumann moves, which preserve the resulting 3-manifold up to homeomorphism. For Neumann moves of type (d), e can be any finite integer.

For example, two graphs in Figure 2.2 (b) have bridges between m_1 and m_2 for the bottom one or from $m_1 \pm 1$ to $m_2 \pm 1$ for the top, respectively, and the top graph in Figure 2.2 (c) has a branch from $m_1 \pm 1$ to ± 1.

Since a branch Γ_{wv} from w to v can be thought as a linear plumbing, the following continued fraction

$$m'_w = m_w - \cfrac{1}{u_1 - \cfrac{1}{u_2 - \cfrac{1}{\ddots - \cfrac{1}{m_v}}}} \tag{2.1}$$

can give us the information of the branch. It is actually a complete invariant under the Neumann moves creating or annihilating valency two vertices along the branch. In (2.1), m_w is the weight of the starting high-valency vertex of the branch Γ_{wv}, u_is represent the weights of valency two vertices on the linear plumbing between w and v, and m_v is the weight of the ending valency one vertex. If the continued fraction m'_w of Γ_{wv} is an integer, the branch is called *pseudo* branch because it can be annihilated or removed by a sequence of Neumann moves. For example, the top graph in Figure 2.2 (b) has a pseudo-branch

$\Gamma_{m_1\pm1,\pm1}$.[1] A high-valency vertex is $w \in V_h$ called a *pseudo* high-valency vertex if it satisfies the following condition

$$\deg(w) - n_w = 1, 2,$$

where n_w denotes a number of pseudo branches connected to the vertex w. This means that a pseudo high-valency vertex can be turned into a valency one or two vertex once we annihilate or collapse all of its pseudo branches by using Neumann moves.

For a given bridge Γ_{w_1,w_2} between two high-valency vertices $w_1, w_2 \in V_h$, we also examine the continued fraction of valency two vertices laid on the bridge if they exist. The bridge is called a *pseudo* bridge if it contains at least one valency two vertex and the continued fraction is zero. A pseudo bridge can also be annihilated by a sequence of Neumann moves of type (a) and (c).

In order to exclude Neumann moves of type (d), we also define a *bad branch* to be a branch which has a segment, including valency one vertex, whose continued fraction is equal to 0.

We define a plumbing graph to be *reduced* if there is at least one vertex with degree greater than two and there is no pseudo high-valency vertices and bad branches. One can change any plumbing tree into a reduced plumbing by removing pseudo high-valency vertices by using Neumann moves. Note that removing pseudo high-valency vertex could create another pseudo high-valency vertex, hence it is needed to keep removing all pseudo high-valency vertices as they appear until there is no pseudo high-valency vertex anymore. We also notice that a plumbing without any high-valency vertex represents a lens space.

2.1.2 Identification of Spin^c structures

Let us review briefly the identification of Spin^c structures on plumbed 3-manifolds in terms of plumbing data. The affine space Spin$^c(Y)$ in the case of $Y = Y(\Gamma)$ for a plumbing tree Γ has already been studied in the Heegard–Floer homology literature [35], and a slightly different description has been used in order to construct an invariant $\hat{Z}_a(q)$ of the plumbed 3-manifold equipped with the Spin^c structure a in [13]. Here we recall Spin^c structures on plumbed 3-manifolds $Y(\Gamma)$ by following [13].

The Spin^c structures on Y can be inherited by those on the 4-manifold $W = W(\Gamma)$ with boundary Y. More explicitly, a natural identification for Spin^c

[1] By abuse of notation, we often use the weights to denote the vertices where the meaning is clear in the context.

structures on W

$$\mathrm{Spin}^c(W) \cong 2\mathbb{Z}^s + \vec{m}$$

translates into a natural identification for the plumbed 3-manifold Y

$$\mathrm{Spin}^c(Y) \cong \left(2\mathbb{Z}^s + \vec{m}\right) / \left(2M\mathbb{Z}^s\right),$$

where $\vec{m}$ is the vector made of the weights m_v for $v \in V$. Note that both identifications take the conjugation of Spin^c structures to the involution $a \leftrightarrow -a$ on the right-hand side.

Another natural identification

$$\mathrm{Spin}^c(Y) \cong \left(2\mathbb{Z}^s + \vec{\delta}\right) / \left(2M\mathbb{Z}^s\right) \cong 2\,\mathrm{Coker}\, M + \vec{\delta}, \tag{2.2}$$

can be obtained by using the map

$$\begin{aligned} \phi\colon\ \left(2\mathbb{Z}^s + \vec{m}\right) / \left(2M\mathbb{Z}^s\right) &\xrightarrow{\cong} \left(\left(2\mathbb{Z}^s + \vec{\delta}\right) / \left(2M\mathbb{Z}^s\right)\right), \\ \left[\vec{\ell}\ \right] &\to \left[\vec{\ell} - M\vec{u}\right], \end{aligned}$$

where $\vec{\delta}$ is the vector of the degrees of the vertices in Γ, and $\vec{u} = (1, 1, \ldots, 1)$.

2.1.3 WRT invariants

The physical definition of WRT invariants was originally inspired by a more physical approach by Witten [45]. For a closed, connected, oriented 3-manifold M^3, WRT invariant is defined by the following functional integral in a QFT with Chern-Simons action:

$$Z_{\mathrm{CS}}(M^3; k) = \int_{\mathcal{A}_{\mathrm{SU}(2)}(M^3)} D\mathcal{A}\ \exp\left[2\pi i(k-2)\mathrm{CS}(\mathcal{A})\right],$$

where we set

$$q = e^{\hbar} = e^{2\pi i/k},$$

and $\mathcal{A} \in \Omega^1(M^3, \mathfrak{su}_2)$ is a connection 1-form of a $\mathrm{SU}(2)$ principal bundle on M^3. Also, $\mathrm{CS}(\mathcal{A})$ is a Chern-Simons action given by

$$\mathrm{CS}(\mathcal{A}) = \frac{1}{8\pi^2} \int_{M^3} \mathrm{Tr}\left(\mathcal{A}d\mathcal{A} + \frac{2}{3}\mathcal{A}^3\right) \in \mathbb{C} \mod \mathbb{Z}.$$

The functional integral is performed over $\mathcal{A}_{\mathrm{SU}(2)}(M^3)$, the space of all connections modulo gauge transformation

$$\mathcal{A} \to g\mathcal{A}g^{-1} + gdg^{-1}, \qquad g : M^3 \to \mathrm{SU}(2).$$

Then we get

$$Z_{\mathrm{CS}}(S^2 \times S^1; k) = 1, \qquad Z_{\mathrm{CS}}(S^3; k) = \frac{q^{1/2} - q^{-1/2}}{i\sqrt{2k}}.$$

In the math literature, the other choice of normalization

$$\tau_k(M^3) = \frac{i\sqrt{2k}}{q^{1/2} - q^{-1/2}} Z_{\mathrm{CS}}$$

is used such that

$$\tau_k(S^3) = 1$$

and the invariant behaves nicely under connected sum operation as follows:

$$\tau_k(M^3 \sharp N^3) = \tau_k(M^3)\tau_k(N^3).$$

2.1.4 The q-series

The q-series has been firstly proposed in [15] aimed to categorify the WRT invariant. It has the integrality property and it is ordinary power series in q. For rational homology spheres, its relation to WRT invariants is conjectured as follows [15, 13].

Conjecture 2.3. *Let M be a closed 3-manifold and $Spin^c(M)$ be the set of $Spin^c$ structures of M, with the action of $\mathbb{Z}_2$ by conjugation. Set*

$$T := Spin^c(M)/\mathbb{Z}_2.$$

Then, for every $a \in T$, there exists

$$\Delta_a \in \mathbb{Q}, \qquad c \in \mathbb{Z}_+, \qquad \hat{Z}_a(q) \in 2^{-c} q^{\Delta_a} \mathbb{Z}[[q]],$$

with $\hat{Z}_a(q)$ converging in the unit disk $|q| < 1$ such that, for infinitely many k, the radial limits $\lim_{q \to e^{2\pi i/k}} \hat{Z}_a(q)$ exist and can be used to recover the Chern-Simons path integral

in the following way:

$$Z_{CS}(M;k) = (i\sqrt{2k})^{-1} \sum_{a,b\in T} e^{2\pi i k\cdot lk(a,a)} |W_b|^{-1} S_{ab} \hat{Z}_b(q)|_{q\to e^{2\pi i/k}}.$$

Here the coefficients S_{ab} are given by

$$S_{ab} = \frac{e^{2\pi i\cdot lk(a,b)} + e^{-2\pi i\cdot lk(a,b)}}{|W_a|\cdot\sqrt{|H_1(M;\mathbb{Z})|}},$$

where the group $W_x = Stab_{\mathbb{Z}_2}(x)$ is $\mathbb{Z}_2$ if $x = \bar{x}$ and is 1 otherwise. The linking numbers $lk(a,b)$ are the usual linking numbers on $H_1(M;\mathbb{Z})$ and a non-canonical isomorphism $Spin^c(M) \cong H_1(M;\mathbb{Z})$ is used.

The simplest version of Conjecture 2.3 happens when M is an integral homology 3-sphere. Then we have $T = 0$, $S_{00} = 1$, and the conjecture predicts the existence of a single series $\hat{Z}_0(q)$ that converges to

$$2(q^{1/2} - q^{-1/2})\tau_k(M) = 2i\sqrt{2k}\cdot Z_{\mathrm{CS}}(M;k)$$

as $q \to e^{2\pi i/k}$.

The q-series was originally defined for plumbed 3-manifolds coming from negative definite plumbing graphs. For a later convenience, we recall the formula of the q-series $\hat{Z}_a(q)$ for weakly negative definite plumbed 3-manifolds following [13] and [15].

Let Γ be a plumbing graph satisfying the weakly negative definite condition, that is, the linking matrix $M = M(\Gamma)$ is invertible and M^{-1} is negative definite on the subspace of $\mathbb{Z}^s$ spanned by the high-valency vertices. Using identifications (2.2), fix a representative $\vec{a} \in 2\mathbb{Z}^s + \vec{\delta}$ of a class

$$a \in \left(2\mathbb{Z}^s + \vec{\delta}\right)/\left(2M\mathbb{Z}^s\right).$$

Then the q-series $\hat{Z}_a(q)$ is defined by

$$\hat{Z}_a(q) = (-1)^{\pi} q^{\frac{3\sigma - \sum_v m_v}{4}} \cdot \mathrm{CT}_{\vec{z}}\left\{\mathcal{D}(\vec{z})\cdot\Theta_a^M(\vec{z})\right\}, \tag{2.3}$$

where

$$\mathcal{D}(\vec{z}) = \prod_{v\in V}\left(z_v - \frac{1}{z_v}\right)^{2-\deg(v)} \tag{2.4}$$

and

$$\Theta_a^M(\vec{z}) = \sum_{\vec{\ell}\in 2M\mathbb{Z}^s + \vec{a}} q^{-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}} \prod_{v\in V} z_v^{\ell_v}. \tag{2.5}$$

In (2.3), $\mathrm{CT}_{\vec{z}}$ is the operation of taking the constant terms of the formal power series in z_v. Also, for the factors in front of the operation $\mathrm{CT}_{\vec{z}}$, $\pi = \pi(M)$ denotes the number of positive eigenvalues of the linking matrix M and $\sigma = \sigma(M)$ is the signature of M, i.e., $\sigma = 2\pi - s$.

Note that all rational functions in (2.4) should be understood as the symmetric expansion, that is, the average of the expansion as $z_v \to 0$ and $z_v \to \infty$. For example, given a high-valency vertex $w \in V_h$ with $\deg(w) = 2 + p_w$, we have

$$\begin{aligned}\left(z_w - \frac{1}{z_w}\right)^{-p_w} &= \frac{1}{2}\left(z_w - \frac{1}{z_w}\right)^{-p_w}\Bigg|_{|z_w|<1} + \frac{1}{2}\left(z_w - \frac{1}{z_w}\right)^{-p_w}\Bigg|_{|z_w|>1} \\ &= \frac{1}{2}\left[\left(-\sum_{i=0}^{\infty} z_w^{2i+1}\right)^{p_w} + \left(\sum_{i=0}^{\infty} z_w^{-(2i+1)}\right)^{p_w}\right] \\ &= \frac{1}{2}\sum_{r_w=0}^{\infty} A(r_w, p_w)\left[(-1)^{p_w} z_w^{2r_w+p_w} + z_w^{-2r_w-p_w)}\right], \end{aligned} \tag{2.6}$$

where $A(r,p)$ is a binomial coefficient

$$A(r,p) = \binom{r+p-1}{r} = \frac{(r+1)(r+2)\cdots(r+p-1)}{(p-1)!}.$$

Remark. The weakly negative definite condition should be imposed to ensure that $\hat{Z}_a(q)$ is well-defined. In fact, together with this condition, (2.5) implies that the exponents of q in $\hat{Z}_a(q)$ has a lower bound, and that only finitely many terms can contribute to the same exponent of q.

Remark. The condition that M is invertible means the corresponding plumbed 3-manifold is a rational homology sphere. We will mostly be interested in the case where M is invertible. The q-series for plumbings satisfying $\det M = 0$ was dealt in [7, 6].

Remark. Recently, Conjecture 2.3 is proved for plumbed 3-manifolds in [33] under the assumption of negative definite condition. However, the convergence of $\hat{Z}_a(q)$ for general 3-manifolds to the WRT invariants, as q approaches a root of unity, has not yet proved rigorously in the literature.

2.2 Graph neural networks

Here we provide a brief review on some of GNNs for a later purpose. In general, GNNs are deep machine learning-based models that operate on graphs. In the literature of GNNs, graphs are a data structure which models a set of objects

(called *nodes*) and their relationships (called *edges*). Since graph data can be easily found in our everyday life, for example, world wide web, social networks, scientific citations, brain and protein networks, traffic networks and so on, GNNs have recently become a very hot topic in the community of machine learning. One of the motivations of GNNs roots in the long-standing history of neural networks for graphs.

There are 3 main computational modules to build a typical GNN architecture: propagation modules, sampling modules and pooling modules. The propagation module is designed to capture the aggregated information of a specific node or edge, or the entire graph by propagating the local information between nodes connected by edges. Among various propagation modules, the convolution operators and recurrent operators are commonly used modules. Sampling modules are usually used to conduct propagation on large graphs and they are often combined with the propagation module. When we need the representations of high-level subgraphs or graphs, pooling modules are needed to extract information from nodes. Since in this paper we will use only convolution operators, which are one of the most frequently used propagation modules, we focus on some of convolution operators. For a broad review on various modules, we refer the reader to [48].

A general pipeline for designing a GNN model usually consists of 4 steps:

- Find graph structure.
  In this step, we have to find out the graph structure for the specific application. These are usually two scenarios: structural scenarios and non-structural scenarios.

- Specify graph type and scale.
  Here we have to investigate the graph type and scale. Graphs are usually categorized as directed or undirected graphs, homogeneous or heterogeneous graphs, and static or dinamic graphs. Note that those types of graphs can be combined, for example, plumbing graphs are undirected, homogeneous, and static graphs, and that there are also several other graph types designed for different tasks. There is no universal criterion to distinguish "small" and "large" graphs. The criterion often depends on the computation devices and its performance.

- Design loss function.
  The loss function is based on the task types and the training settings. For GNNs, there are usually three types of tasks: node-level, edge-level and graph-level tasks. For example, our task in Chapter 4 is graph-level

classification problem. We can also categorize graph learning tasks into three different training settings: supervised, unsupervised and reinforcement learning setting. In Section 4.1, we deal with the supervised learning setting and in Section 4.2, we work in the reinforcement learning setting. Once we determine the task type and training setting, we can design a specific loss function for the task.

- Build model using computational modules.
  Based on the above three steps, we can start building the model by combining various computational modules to obtain better representations.

Let us now look at the convolution operators, which are the mostly used operators for the GNN models. The convolution operators are motivated by convolutional neural networks (CNN), which have achieved a notable progress in various areas dealing with Euclidean structured data such as images, videos, and texts. In general, the role of convolution operators can be described as

$$\mathbf{x}_i^{(k)} = \gamma^{(k)} \left(\mathbf{x}_i^{(k-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(k)} \left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{j,i} \right) \right),$$

where $\mathbf{x}_i^{(k)} \in \mathbb{R}^F$ denotes node features of node i in the k-th layer and $\mathbf{e}_{j,i} \in \mathbb{R}^D$ denotes edge features of the edge connecting from node j to node i. We also note that $\bigoplus$ over a neighborhood $\mathcal{N}(i)$ of node i is a differentiable, permutation invariant function such as sum, mean and max, and γ and ϕ denote differentiable functions such as Multi Layer Perceptrons (MLPs).

Convolutional operators are often categorized as spectral operators and spatial operators. Spectral methods are theoretically based on graph signal processing and define the convolution operator in the spectral domain by using the graph Fourier transform, while spatial methods define convolution operations with differently sized neighborhoods on the graph based on the graph topology by maintaining the local invariance of CNNs.

Among various convolution operators existing in the literature, the following will appear in the next sections.

- Graph Embedding Network (GEN) [28]
  GEN is designed for deep graph similarity learning and embeds each graph into a vector, called a graph embedding. More explicitly, it first computes initial node embeddings $\mathbf{x}_i^{(1)}$ from the node features $\mathbf{x}_i^{(0)}$ through MLP

$$\mathbf{x}_i^{(1)} = \mathrm{MLP}\left(\mathbf{x}_i^{(0)} \right),$$

then it executes the single message propagation to compute node embeddings $\mathbf{x}_i^{(2)}$ by the information in its local neighbourhood $\mathcal{N}(i)$ [2]

$$\mathbf{x}_i^{(2)} = \mathrm{MLP}\left(\mathbf{x}_i^{(1)}, \sum_{j \in \mathcal{N}(i)} \mathrm{MLP}\left(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(1)}\right)\right).$$

Once the node embeddings $\mathbf{x}_i^{(2)}$ are computed, an aggregator computes a graph embedding by aggregating the set of node embeddings. In Section 4.1.1, we describe the details of the aggregator which we will apply not only to GEN but also the other models GCN and GAT.

- Graph Convolutional Network (GCN)
  GCN is introduced in [24] as a variant of convolutional neural networks for graphs. It operates as the following formula:

$$\mathbf{z}_i = \mathbf{\Theta}^\mathsf{T} \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j,$$

  where $\mathbf{z}_i$ is the output for the i-th node, $\mathbf{\Theta}$ is a matrix of filter parameters, and $\hat{d}_i$ is the degree of i-th node.

- Graph Attention Network (GAT)
  GAT is proposed in [42], which incorporates the attention mechanism into the message propagation. The mechanism of GAT can be formulated as

$$\mathbf{x}_i' = \alpha_{i,i} \mathbf{\Theta} \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \mathbf{\Theta} \mathbf{x}_j.$$

  Here the attention coefficients α are given by

$$\alpha_{i,j} = \frac{\exp\left(\mathrm{LeakyReLU}(\mathbf{a}^\mathsf{T}[\mathbf{\Theta}\mathbf{x}_i \| \mathbf{\Theta}\mathbf{x}_j])\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(\mathrm{LeakyReLU}(\mathbf{a}^\mathsf{T}[\mathbf{\Theta}\mathbf{x}_i \| \mathbf{\Theta}\mathbf{x}_k])\right)},$$

  where the attention mechanism $\mathbf{a}$ is implemented by a single-layer feedfoward neural network, and $\|$ is the concatenation operator.

All the neural networks including GNNs are implemented based on PyTorch [37] and PyTorch Geometric [11]. [3]

[2] It is also possible to apply a finite number of propagation process iteratively, but we will only consider single propagation here.

[3] Python code is available on Github.

Chapter 3

Refined and generalized $\hat{Z}$-invariants

This chapter deals with a two-variable refinement $\hat{Z}_a(q,t)$ of plumbed 3-manifold invariants $\hat{Z}_a(q)$, which were previously defined for weakly negative definite plumbed 3-manifolds. We provide a number of explicit examples in which we argue the recovering process to obtain $\hat{Z}_a(q)$ from $\hat{Z}_a(q,t)$ by taking a limit $t \to 1$. For plumbed 3-manifolds with two high-valency vertices, we analytically compute the limit by using the explicit integer solutions of quadratic Diophantine equations in two variables. Based on numerical computations of the recovered $\hat{Z}_a(q)$ for plumbings with two high-valency vertices, we propose a conjecture that the recovered $\hat{Z}_a(q)$, if exists, is an invariant for all tree plumbed 3-manifolds. Finally, we provide a formula of the $\hat{Z}_a(q,t)$ for the connected sum of plumbed 3-manifolds in terms of those for the components.

3.1 The (q,t)-series invariants

In this section we construct the formula of the (q,t)-series $\hat{Z}_a(q,t)$ for a certain class of plumbed 3-manifolds by introducing a regulator t into the q-series $\hat{Z}_a(q)$, previously defined only for weakly negative definite plumbings.

3.1.1 Definition

Let Γ be a reduced plumbing graph, that is, a tree plumbing that has at least one high-valency vertex, but does not have any pseudo high-valency vertex. We assume that its linking matrix $M = M(\Gamma)$ is invertible. We keep our notations as in Section 2.1.

Definition 3.1. The (q,t)-series $\hat{Z}_a(q,t)$ for a reduced plumbing, equipped with some Spin^c structure $\vec{a} \in 2\mathbb{Z}^s + \vec{\delta}$, is defined by

$$\hat{Z}_a(q,t) = (-1)^{\pi} q^{\frac{3\sigma - \sum_v m_v}{4}} \cdot \mathrm{CT}_{\vec{z}} \left\{ \frac{1}{|\mathcal{S}|} \sum_{\xi \in \mathcal{S}} \prod_{w \in V_h} \mathcal{D}_w(\vec{z},t)|_{\xi_w} \cdot \Theta_a^M(\vec{z}) \right\}. \tag{3.1}$$

Let us elaborate on the various elements of this formula.

For the factors in front of the $\mathrm{CT}_{\vec{z}}$ operation, the $\pi = \pi(M)$ and σ denote the number of positive eigenvalues and the signature of the linking matrix M, respectively. The theta function is the same as the one in (2.5) for the q-series, and $\mathrm{CT}_{\vec{z}}$ denotes the operation of taking the constant term of the Laurent series in $z_v \in V$.

The $\mathcal{S}$ is a set of vectors $\xi \in \{\pm 1\}^{V_h}$ that satisfy the following property

$$\xi_{w_1}\xi_{w_2} = (-1)^{\pi(\Gamma_{w_1,w_2})+1} \quad \text{for each pseudo bridge } \Gamma_{w_1,w_2} \text{ in } \Gamma,$$

where ξ_w denotes the element in a vector ξ corresponding to the high-valency vertex $w \in V_h$, and $\pi(\Gamma_{w_1,w_2})$ is the number of positive eigenvalues of the linking matrix associated to Γ_{w_1,w_2}. One can easily see that the set $\mathcal{S}$ for a given Γ is well-defined as long as Γ is a tree. Moreover, the cardinality $|\mathcal{S}|$ of the set $\mathcal{S}$ is equal to 2 to the power of the number of high-valency vertices minus the number of pseudo bridges in Γ. A vector $\xi \in \mathcal{S}$ is called a *chamber* for Γ.

For a given chamber ξ, the discriminant function at this chamber is defined by

$$\mathcal{D}_w(\vec{z},t)|_{\xi_w} := \begin{cases} (-1)^{p_w} \sum_{r_w=0}^{\infty} A(r_w,p_w) t^{2r_w+p_w} z_w^{2r_w+p_w} \prod_{v\in I_w} \mathcal{D}_v^{\varphi(v)}(z_v,t) & \text{for } \xi_w = +1, \\ \sum_{r_w=0}^{\infty} A(r_w,p_w) t^{2r_w+p_w} z_w^{-2r_w-p_w} \prod_{v\in I_w} \mathcal{D}_v^{-\varphi(v)}(z_v,t) & \text{for } \xi_w = -1. \end{cases} \tag{3.2}$$

Here I_w, associated to each $w \in V_h$, is the set of valency one vertices that are end-points of all branches starting from w. And the discriminant $\mathcal{D}_v^{\varphi(v)}$ of the valency one vertex $v \in I_w$ depends on $\varphi(v)$ by

$$\mathcal{D}_v^{\varphi(v)} := \begin{cases} z_v t - 1/z_v t & \text{for } \varphi(v) = +1, \\ z_v/t - t/z_v & \text{for } \varphi(v) = -1, \\ z_v - 1/z_v & \text{for } \varphi(v) = 0. \end{cases}$$

The function $\varphi(v)$ returns a sign, determined by the branch Γ_{wv}, as follows: if the branch Γ_{wv} from w to v is not a pseudo-branch, then $\varphi(v) = 0$. If the branch Γ_{wv} is a pseudo-branch, by definition, the weights of the vertices laid on the branch

should satisfy

$$m'_w = m_w - \cfrac{1}{u_1 - \cfrac{1}{u_2 - \cfrac{1}{\ddots - \cfrac{1}{m_v}}}},$$

for some finite integer m'_w. Then we define

$$\varphi(v) = (-1)^{\pi(\Gamma_{wv}) - \pi(\Gamma_{m'_w})},$$

where $\Gamma_{m'_w}$ is the graph that consists of the single vertex weighted by m'_w. Recall that the notation $\pi(\Gamma_{wv})$ denotes the number of positive eigenvalues of the linking matrix corresponding to Γ_{wv}, so it is immediate to see $(-1)^{\pi(\Gamma_{m'_w})} = \mathrm{sgn}(m'_w)$.

For completeness, we define $\hat{Z}_a(q,t) = \hat{Z}_a(q)$ if there is no high-valency vertex in Γ. The (q,t)-series is well-defined even when Γ does not satisfy the weakly negative (or positive) definite because there are only finitely many contributions to any monomial $t^{n_t} q^{n_q}$.

Remark. Comparing (3.2) with (2.6), one can see that the new variable t is introduced as the regulator that appears in the standard ζ-regularization process. Moreover, if Γ is weakly negative definite, then we can recover $\hat{Z}_a(q)$ from $\hat{Z}_a(q,t)$ by taking $t = 1$.

Remark. In [13], two-variable series $F_K(x,q)$ for plumbed knot complements is introduced as the analogue of the invariants $\hat{Z}_a(q)$. It is also possible to construct the t-deformation of F_K as the analogue of $\hat{Z}_a(q,t)$ for knot complements in the same way. We note that another version of t-deformation and a-deformation has been discussed in [10].

Remark. Another two-variable series $\hat{\hat{Z}}_a(q;t)$ is introduced in [1] based on the lattice cohomology. This two-variable series is also considered as the refinement of the q-series, but it is only defined for negative definite plumbings. Even for negative definite plumbings, any direct relation between the two-variable series and our (q,t)-series has not been found.

3.1.2 Invariance

For the (q,t)-series $\hat{Z}_a(q,t)$ to be an invariant of the plumbed 3-manifold Y equipped with the Spin^c structure a, it should be independent on the presentation

of Y as a plumbing, that is, it has to be unchanged under the Neumann moves in Figure 2.2.

Proposition 3.2. *If two reduced plumbings represent a same 3-manifold, then there exists a sequence of Neumann moves such that none of those Neumann moves creates any pseudo high-valency vertices.*

Proof. Since two reduced plumbings Γ and Γ' realize the same 3-manifold, there exists a sequence of plumbings $G = \{\Gamma_0 = \Gamma, \Gamma_1, \Gamma_2, \dots, \Gamma_{n-1}, \Gamma_n = \Gamma'\}$ such that any adjacent two plumbings Γ_{i-1} and Γ_i are related by a Neumann move depicted in Figure 2.2 for $i = 1, \dots, n$. It is important to note that in this proof we consider a Neumann move as an action applied to a certain vertex on a branch or on a bridge, instead of regarding it as a transformation from a plumbing graph to another.

As a first step, we pick up a subsequence of Neumann moves such that the first move in the subsequence creates a pseudo branch or a pseudo bridge, the last move deletes the pseudo branch or bridge, and all the moves in between are actions on the branch or bridge. It is clear that the subsequence is redundant since the relevant continued fraction should be preserved, therefore, we are going to remove all such subsequences. Notice that another subsequence with this property can be appeared after removing a subsequence, so we keep removing subsequences until they will not appear anymore. We also note that the remaining sequences of Neumann moves are still well-defined since all the moves in this subsequence are limited on a redundant branch or bridge.

If the relevant plumbings are all reduced after removing subsequences, then the proof is done. Therefore, without loss of generality, let us assume that there exists at least one non-reduced plumbing in the sequence G. For simplicity, suppose that all non-reduced plumbings in the sequence G contain only one pseudo high-valency vertex, because the cases with more than one pseudo high-valency vertex can be extended in a similar way. Among non-reduced plumbings, choose the non-reduced plumbings Γ_j and Γ_k with the smallest index j and the largest one k. This means that Γ_{j-1} and Γ_{k+1} are reduced plumbings. Furthermore, since we have removed all redundant subsequences of Neumann moves, there are only two possible ways to create the pseudo-high valency vertex in Γ_j. The first is that the pseudo high-valency vertex in Γ_j is created by adding a pseudo branch to a valency two vertex on a pseudo bridge in Γ_{j-1}. The other is that a high-valency vertex with degree more than 3 in Γ_{j-1} is split into two high-valency vertex connected by a pseudo bridge, one of which is the pseudo high-valency vertex.

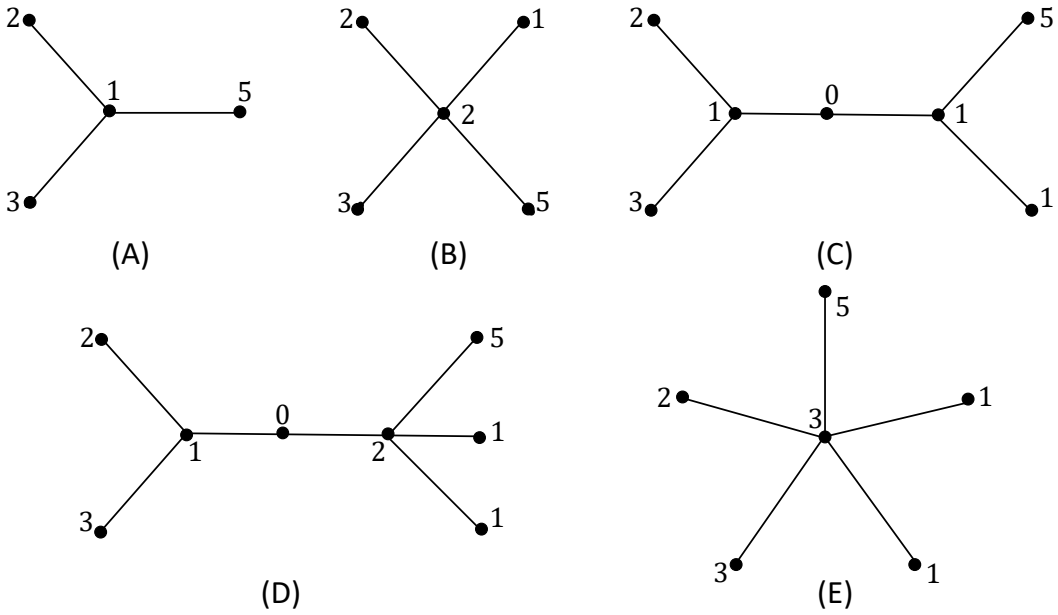


FIGURE 3.1: A sequence of plumbings in which the first and last plumbings are reduced, but there are two non-reduced plumbings in between.

Since two cases are related to a pseudo bridge, the key is to collapse the pseudo bridge by using Neumann moves. More explicitly, we insert a plumbing Γ'_{j-1} between Γ_{j-1} and Γ_j where Γ'_{j-1} is obtained by collapsing the pseudo bridge on which the pseudo high-valency vertex is laid in Γ_j. Then we get Γ'_i from Γ_i for $i = j, j+1, \ldots, k$ by collapsing the relevant pseudo bridge. We also insert a plumbing Γ'_{k+1} between Γ'_k and Γ_{k+1} by rebuilding a pseudo bridge as same as the one in Γ_{k+1} if it exists in Γ_{k+1}.

Then the sequence $\{\Gamma_0, \ldots, \Gamma_{j-1}, \Gamma'_{j-1}, \Gamma'_j, \ldots, \Gamma'_k, \Gamma'_{k+1}, \Gamma_{k+1}, \ldots, \Gamma_n\}$ satisfies the statement of the proposition. Notice that Γ'_{j-1} and Γ'_{k+1} might not appear in the sequence case by case. □

An example of Proposition 3.2 is depicted in Figure 3.1 and Figure 3.2. The plumbings (A), (B), and (E) shown in Figure 3.1 are reduced, while the plumbings (C) and (D) are clearly not reduced. From this sequence, one can construct the sequence of reduced plumbings depicted in Figure 3.2 by using method mentioned in the proof of Proposition 3.2.

Theorem 3.3. *The series $\hat{Z}_a(q,t)$ defined in (3.1) is an invariant for the reduced plumbed 3-manifolds.*

Proof. Since $\hat{Z}_a(q,t)$ is equal to $\hat{Z}_a(q)$ for the plumbings without high-valency vertices by definition, we can assume that $V_h(\Gamma)$ is not an empty set. We use a prime to distinguish the quantities associated to the top graphs in Figure 2.2. For

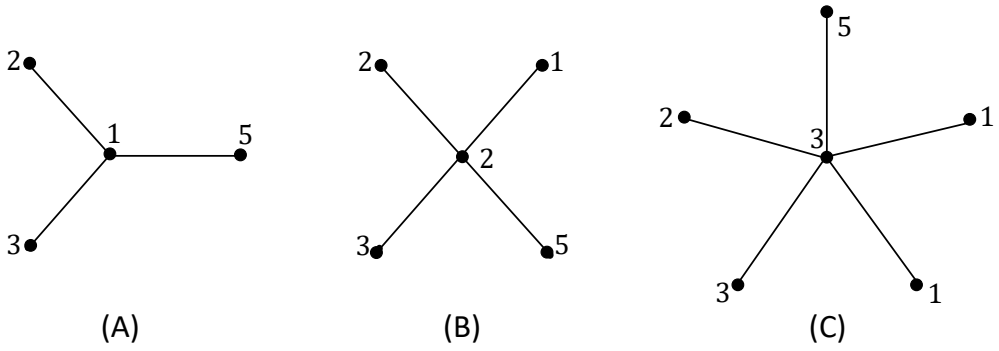


FIGURE 3.2: A sequence of reduced plumbings obtained from the sequence shown in Figure 3.1 by Proposition 3.2.

example, M is the linking matrix for the bottom graph Γ, and M' denotes the one for the top graph Γ'. According to the result of Proposition 3.2, it is enough to consider Neumann moves that does not create a pseudo high-valency vertex. Let us consider by each type of Neumann moves depicted in Figure 2.2.

Before we dive into the main consideration on each type of Neumann moves, we recall the defining formulae of $\hat{Z}_a(q,t)$ for a convenience. We defined the (q,t)-series in Section 3.1.1 by

$$\hat{Z}_a(q,t) = (-1)^{\pi} q^{\frac{3\sigma-\sum_v m_v}{4}} \cdot \mathrm{CT}_{\vec{z}} \left\{ \frac{1}{|\mathcal{S}|} \sum_{\xi\in\mathcal{S}} \prod_{w\in V_h} \mathcal{D}_w(\vec{z},t)|_{\xi_w} \cdot \Theta_a^M(\vec{z}) \right\},$$

with the following discriminant function

$$\mathcal{D}_w(\vec{z},t)\,|_{\xi_w} := \begin{cases} (-1)^{p_w} \sum_{r_w=0}^{\infty} A(r_w,p_w) t^{2r_w+p_w} z_w^{2r_w+p_w} \prod_{v\in I_w} \mathcal{D}_v^{\varphi(v)}(z_v,t) & \text{for } \xi_w = +1, \\ \sum_{r_w=0}^{\infty} A(r_w,p_w) t^{2r_w+p_w} z_w^{-2r_w-p_w} \prod_{v\in I_w} \mathcal{D}_v^{-\varphi(v)}(z_v,t) & \text{for } \xi_w = -1. \end{cases}$$

For Neumann moves of type (b)

Consider first the move (b) in Figure 2.2, with the signs on top being -1. By linear algebra, it is immediate to see that we have $\sigma' = \sigma - 1$, $\pi' = \pi$, hence the quantity $3\sigma - \sum_v m_v$ does not change. Furthermore, the factors in front of the $\mathrm{CT}_{\vec{z}}$ operation in (3.1) does not change either.

For the bottom graph Γ, let us write a vector $\vec{\ell} \in \mathbb{Z}^s$ as a concatenation

$$\vec{\ell} = (\vec{\ell}_1, \vec{\ell}_2),$$

such that $\vec{\ell}_1$ describes the left part of the graph including the vertex m_1 and $\vec{\ell}_2$ describes the right part. Then we can construct a vector $\vec{\ell}'$ for the top graph Γ' by

$$\vec{\ell}' = (\vec{\ell}_1, 0, \vec{\ell}_2) \in \mathbb{Z}^{s+1},$$

that satisfies

$$(\vec{\ell}, M^{-1}\vec{\ell}) = (\vec{\ell}', (M')^{-1}\vec{\ell}'). \tag{3.3}$$

Moreover, it is shown in [13] that $\vec{\ell}'$ has the property $\vec{\ell}' \in 2M'\mathbb{Z}^{s+1} + \vec{a}'$ whenever $\vec{\ell} \in 2M\mathbb{Z}^s + \vec{a}$, where $\vec{a}'$ is a representative of the Spin^c structure a' for $Y(\Gamma')$ that is the counterpart of a through the isomorphism $Y(\Gamma) \cong Y(\Gamma')$.

Taking these into account, when the move happens inside a branch Γ_{wv} from a high-valency vertex $w \in V_h$ to a valency one vertex $v \in I_w$, observe that there is no change for $\varphi(v)$ whether the branch is pseudo or not. This implies that the discriminant function does not change, hence we obtain the same result.

For the case when the move happens inside a bridge between two high-valency vertices, two discriminants for Γ' and Γ have the same structure, therefore they yield the same $\hat{Z}_a(q,t)$.

Let us now consider the case of the move (b) with the sign $+1$ for the top graph in a similar way. As before, the quantity $3\sigma - \sum_v m_v$ is still unchanged, but we have $\pi' = \pi + 1$, so $(-1)^\pi$ switches the sign. Given $\vec{\ell} = (\vec{\ell}_1, \vec{\ell}_2)$ for the bottom graph, a vector for the top graph

$$\vec{\ell}' = (\vec{\ell}_1, 0, -\vec{\ell}_2)$$

satisfies (3.3) [13].

For the move on a bridge that is not pseudo, the discriminant does not change, but due to the change of sign in $\vec{\ell}_2$, the theta function changes as if we do the substitutions $z_v \to z_v^{-1}$ for all the vertices v corresponding to $\vec{\ell}_2$. From this, we obtain extra -1 sign, that cancels out the disagreement of signs by $(-1)^\pi$. Note that the powers of t remain the same by the substitutions $z_v \to z_v^{-1}$. If the move happens on a pseudo-bridge, the proof is similar, but the only difference is that due to the minus sign in $-\vec{\ell}_2$ for $\vec{\ell}'$ the choice of the chamber for the top graph should be opposite than that for the bottom, because the move increases by 1 the number of the positive eigenvalues of the linking matrix associated to the bridge.

If the move appears on a non-pseudo branch Γ_{wv}, then the mechanism is same as one for the move on a bridge, so we get the same result. However, if the move appears on a pseudo-branch Γ_{wv}, then we need to care about the change of $\varphi(v)$ for the discriminant function because the move (b) with the sign $+1$ increases

the number $\pi(\Gamma_{wv})$ of positive eigenvalues of the linking matrix associated to the branch by 1. In this case, v is the only vertex corresponding to $\vec{\ell}_2$, hence the change of the sign for ℓ_v produces the extra sign, but it gives the same powers of t because of the change of $\varphi(v)$. Also, the extra sign remedies the change of $(-1)^{\pi}$.

For Neumann moves of type (c)

Next, we consider the move (c) when the sign for the blow-up vertex is -1. We have $\pi' = \pi$, $\sigma' = \sigma - 1$, and the quantity $3\sigma - \sum_v m_v$ for the top graph is 1 lower than that for the bottom one. This yields an extra factor $q^{-1/4}$ for the top graph.

For the bottom graph Γ, we write vectors as

$$\vec{\ell} = (\vec{\ell}_2, \ell_1),$$

where ℓ_1 denotes the vertex with weight m_1. Then we define corresponding vectors for the top graph Γ'

$$\vec{\ell}'_{\pm} = (\vec{\ell}_2, \ell_1 \pm 1, \mp 1).$$

By simple linear algebra, one can observe that

$$(\vec{\ell}, M^{-1}\vec{\ell}\,) = (\vec{\ell}'_{\pm}, (M')^{-1}\vec{\ell}'_{\pm}) + 1. \tag{3.4}$$

The extra factor $+1$ in (3.4) gives rise to $q^{1/4}$ for the top graph that cancels with $q^{-1/4}$ coming from the factor in front of the operation $\mathrm{CT}_{\vec{z}}$.

Let us consider the case where the vertex w decorated with m_1 is a high-valency vertex in the bottom graph. We assume that the high-valency vertex does not have a pseudo bridge. One can follow the similar argument for the case when it has a pseudo bridge.

The corresponding part of the discriminant function for Γ is given by

$$\frac{1}{2}\sum_{r=0}^{\infty} A(r,p)t^{2r+p}\left[(-1)^p z_1^{2r+p}\mathcal{D}^+ + z_1^{-2r-p}\mathcal{D}^-\right],$$

where $\mathcal{D}^{\pm}$ are the contributions from vertices in I_w corresponding to $\vec{\ell}_2$, and p is determined by $\deg(w) = 2 + p$. From the role of the operation $\mathrm{CT}_{\vec{z}}$ in (3.1), it follows that ℓ_1 has values in the form of $\ell_1 = \pm(2r + p)$ for non-negative integers r. Without loss of generality, suppose that $\ell_1 = 2r + p$. Then the vectors $\vec{\ell} = (\vec{\ell}_2, 2r + p)$ pick up monomials in t given by

$$\frac{1}{2}A(r,p)t^{2r+p}.$$

For the top graph, the discriminant function has the following portion

$$\frac{1}{2}\sum_{s=0} A(s,p+1)t^{2s+p+1}$$
$$\times\left[(-1)^{p+1}z_1^{2s+p+1}\left(z_0t-\frac{1}{z_0t}\right)\mathcal{D}^+ + z_1^{-2s-p-1}\left(\frac{z_0}{t}-\frac{t}{z_0}\right)\mathcal{D}^-\right],$$

where z_0 is the variable for the newly introduced blow-up vertex in Γ'. Corresponding to $\vec{\ell}=(\vec{\ell}_2,2r+p)$, we have vectors of the form

$$\vec{\ell}'_\pm=(\vec{\ell}_2,2r+p\pm1,\mp1).$$

Therefore, the monomials chosen by those vectors are

$$-\frac{1}{2}A(r-1,p+1)t^{2r+p-1}\cdot t+\frac{1}{2}A(r,p+1)t^{2r+p+1}\cdot\frac{1}{t}=\frac{1}{2}A(r,p)t^{2r+p}.$$

This means that we get the same answer for $\hat{Z}_a(q,t)$.

For the case when the move applies to a valency one vertex $v_1\in I_w$ for some $w\in V_h$, it is essential to check that the discriminant for Γ' can be obtained from the one for Γ by the substitution $z_1\to z_0$ since we have $\varphi(v_1)=\varphi(v_0)$. Then, vectors for Γ are described by

$$\vec{\ell}=(\vec{\ell}_2,\pm1)$$

and the corresponding vectors for Γ' are given by

$$\vec{\ell}'=(\vec{\ell}_2,0,\pm1).$$

Since we have the same part of discriminant function corresponding to the high-valency vertex for top and bottom plumbings, the proof can be done in the same way as in [13, Proposition 4.6].

Move (c) with the sign $+1$ is similar, but we use the vectors of the form

$$\vec{\ell}'=(\vec{\ell}_2,\ell_1\pm1,\pm1).$$

For Neumann moves of type (a)

At last, let us move on to the move (a). Here we have $\pi'=\pi+1$ that yields an extra sign, and the factor $3\sigma-\sum_v m_v$ is unchanged. For the top graph Γ', we write vectors as

$$\vec{\ell}'=(\vec{\ell}_1,\ell_1,0,\ell_2,\vec{\ell}_2),$$

where $\vec{\ell}_1$ is corresponding to the left side of the graph (not including v_1) and $\vec{\ell}_2$ is

corresponding to the right side of the graph (not including v_2). From $\vec{\ell}'$ we define a vector for the bottom graph Γ as

$$\vec{\ell} = (\vec{\ell}_1, \ell_1 - \ell_2, -\vec{\ell}_2),$$

that satisfies the following equality

$$(\vec{\ell}, M^{-1}\vec{\ell}\,) = (\vec{\ell}', (M')^{-1}\vec{\ell}').$$

Let v_1, v_0, v_2 be the vertices shown in the top graph and v_b the vertex shown in the bottom one. There are four different cases depending on where those vertices are placed in the graph as follows:

(i) Three vertices v_1, v_0, v_2 form a segment of a non-pseudo bridge. In this case, it is obvious that v_0 should be a valency two vertex. Then both of v_1 and v_2 cannot be high-valency vertices because they are laid on a non-pseudo bridge. Either of v_1 and v_2 can be a valency one vertex.

(ii) Three vertices form a pseudo bridge. Thus, v_1 and v_2 are high-valency vertices at the boundary of the pseudo bridge and v_0 is a valency two vertex.

(iii) Three vertices form a pseudo branch. In this case, v_1 is a high-valency vertex and v_2 is a valency one vertex. As always, v_0 is a valency two vertex.

(iv) Three vertices form a part of a bridge. In this case, v_1 can be either a high-valency vertex or a valency two vertex. Then v_2 is a valency two vertex if v_1 is a high-valency vertex, or valency one vertex if v_1 is a valency two vertex.

Let us deal with all possible cases one by one. First, we consider the case when v_1, v_0, v_2 are laid on a non-pseudo bridge, but at least one of v_1 and v_2 is not a high-valency vertex. Then the discriminant functions for Γ and Γ' have the same structure, and the minus sign in front of $\vec{\ell}_2$ in $\vec{\ell}' = (\vec{\ell}_1, \ell_1, 0, \ell_2, -\vec{\ell}_2)$ produces the opposite sign for monomials in t to the sign from $\vec{\ell} = (\vec{\ell}_1, \ell_1 - \ell_2, \vec{\ell}_2)$. This sign difference is recovered by the extra sign from $(-1)^{\pi}$. Thus we get the same result for $\hat{Z}_a(q,t)$.

If the vertices v_1, v_0, v_2 are laid on a pseudo bridge, we should put into our consideration the change of choice of chambers for the top graph since the move creates a positive eigenvalue of the linking matrix.

The second case is when v_1, v_0, v_2 form a bridge between v_1 and v_2, that is, v_1 and v_2 are both high-valency vertices, and v_0 is a valency two vertex. This case

is just the one depicted in Figure 2.2 (A). For the bottom graph Γ, we can assume that the vertex v_b does not have a pseudo bridge, then the corresponding portion of the discriminant function is given by

$$\frac{1}{2}\sum_{r=0}^{\infty} A(r,p)t^{2r+p}\left[(-1)^p z_b^{2r+p}\mathcal{D}_1^+\mathcal{D}_2^+ + z_b^{-2r-p}\mathcal{D}_1^-\mathcal{D}_2^-\right],$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the discriminant functions of valency one vertices in I_{v_b} corresponding to $\vec{\ell}_1$ and $\vec{\ell}_2$, respectively. Then the vectors of the form

$$\vec{\ell} = (\vec{\ell}_1, 2r+p, \vec{\ell}_2)$$

choose the following monomials by the role of the operator $\mathrm{CT}_{\vec{z}}$

$$\frac{1}{2}A(r,p)t^{2r+p}.$$

For the top graph, since the elements ξ_{v_1} and ξ_{v_2} of a chamber ξ should be opposite, we have

$$\frac{1}{2}\left[\sum_{r_1=0}^{\infty}(-1)^{p_1}A(r_1,p_1)t^{2r_1+p_1}z_1^{2r_1+p_1}\mathcal{D}_1^+\sum_{r_2=0}^{\infty}A(r_2,p_2)t^{2r_2+p_2}z_2^{-2r_2-p_2}\mathcal{D}_2^-\right.$$
$$\left.+\sum_{r_1=0}^{\infty}A(r_1,p_1)t^{2r_1+p_1}z_1^{-2r_1-p_1}\mathcal{D}_1^-\sum_{r_2=0}^{\infty}(-1)^{p_2}A(r_2,p_2)t^{2r_2+p_2}(-1)^{p_2}z_2^{2r_2+p_2}\mathcal{D}_2^+\right],$$

where $p_1 = \deg(v_1) - 2$ and $p_2 = \deg(v_2) - 2$ such that $p = p_1 + p_2$. Associated to $\vec{\ell} = (\vec{\ell}_1, 2r+p, \vec{\ell}_2)$, we have the following vectors

$$\vec{\ell}' = (\vec{\ell}_1, 2r_1+p_1, 0, -2r_2-p_2, -\vec{\ell}_2).$$

Here the sum of r_1 and r_2 should be equal to r. From such vectors, we obtain

$$-\frac{1}{2}\sum_{r_1+r_2=r}A(r_1,p_1)A(r_2,p_2)t^{2r_1+2r_2+p_1+p_2} = -\frac{1}{2}A(r,p)t^{2r+p},$$

where we have used $A(r,p) = \sum_{r_1+r_2=r}A(r_1,p_1)A(r_2,p_2)$.

Thirdly, let us consider when v_1, v_0, v_2 are vertices on a branch from v_1 to v_2 through v_0. Then, the discriminant function for the bottom graph Γ has the following part

$$\frac{1}{2}\sum_{r=0}^{\infty}A(r,p)t^{2r+p}\left[(-1)^p z_b^{2r+p}\mathcal{D}^+ + z_b^{-2r-p}\mathcal{D}^-\right].$$

The vectors $\vec{\ell} = (\vec{\ell}_1, \ell_b) = (\vec{\ell}_1, 2r+p)$ pick up the factors of the form

$$\frac{1}{2}A(r,p)t^{2r+p}.$$

On the other hand, for the top graph Γ' we have

$$\frac{1}{2}\sum_{s=0}^{\infty} A(s,p+1)t^{2s+p+1} \times \left[(-1)^{p+1}z_1^{2s+p+1}\left(\frac{z_2}{t}-\frac{t}{z_2}\right)\mathcal{D}^+ + z_1^{-2s-p-1}\left(z_2 t - \frac{1}{z_2 t}\right)\mathcal{D}^-\right],$$

where we have used $\varphi(v_2) = -1$ since the branch $\Gamma'_{v_1v_2}$ is a pseudo-branch and the move increases the number of positive eigenvalues by 1. Corresponding to $\vec{\ell} = (\vec{\ell}_1, 2r+p)$, we have two vectors $\vec{\ell}' = (\vec{\ell}_1, 2r+p+1, 1)$ and $\vec{\ell}' = (\vec{\ell}_1, 2r+p-1, -1)$, which produce the following factors:

$$\frac{1}{2}A(r,p+1)t^{2r+p+1}\cdot\frac{1}{t} + \frac{1}{2}A(r-1,p+1)t^{2r+p-1}\cdot t = -\frac{1}{2}A(r,p)t^{2r+p}.$$

The minus sign in the right side of the above equation cancel with the extra sign from $(-1)^{\pi}$. Therefore, we obtain the same result.

At last, if v_1, v_0, v_2 are a part of a branch, then the degree of the relevant high-valency vertex remains the same by the move and this implies the same structure of the discriminant function for bottom and top plumbings. Therefore the proof gets much easier by the similar way to the corresponding case in [13, Proposition 4.6]. □

Now we present a simple example of the (q,t)-series for a 3-manifold realized by the plumbing shown in Figure 3.3. It is known as Poincaré homology sphere. It is also an example of a Brieskorn 3-sphere, denoted by $\overline{\Sigma(2,3,5)}$. The overline denotes the reverse of the orientation compared to the standard one.

The linking matrix for the plumbing is given by

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 1 & 0 & 0 & 5 \end{pmatrix},$$

and we have

$$\pi = 3, \qquad \frac{3\sigma - \sum_v m_v}{4} = -\frac{5}{4}.$$

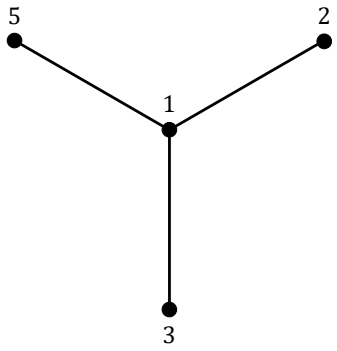


FIGURE 3.3: A plumbing graph that realizes Poincaré homology sphere $\overline{\Sigma(2,3,5)}$.

As $H_1\big(\overline{\Sigma(2,3,5)},\mathbb{Z}\big) \cong 0$, there is a unique value $a = 0$ to consider. The formula (3.1) for the manifold reads as

$$\hat{Z}_0(q,t) = -\frac{1}{2}q^{-\frac{5}{4}}\mathrm{CT}_{\vec{z}}\left\{\sum_{r=0}^{\infty} t^{2r+1}\left(z_1^{-2r-1}-z_1^{2r+1}\right)\right.$$
$$\left.\times\left(z_2-\frac{1}{z_2}\right)\left(z_3-\frac{1}{z_3}\right)\left(z_4-\frac{1}{z_4}\right)\times\sum_{\vec{\ell}\in 2M\mathbb{Z}^4+\vec{\delta}} q^{-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}}\prod_{v=1}^{4} z_v^{\ell_v}\right\}.$$

Due to the operation $\mathrm{CT}_{\vec{z}}$, the values of ℓ_v should be

$$\ell_1 = \pm(2r+1), \qquad \text{and} \qquad \ell_v = \pm 1 \quad \text{for } v = 2,3,4.$$

Observe that such vectors $\vec{\ell}$ also satisfy the condition $\vec{\ell} \in 2M\mathbb{Z}^4 + \vec{\delta}$. Then, we obtain the (q,t)-series for $\overline{\Sigma(2,3,5)}$ as

$$\begin{aligned}\hat{Z}_0(q,t) = tq^{-\frac{3}{2}}\big(1 - q - q^3 - q^7 + tq^8 + q^{14} \\ + q^{20} + t^2q^{29} - q^{31} - t^2q^{42} - t^2q^{57} + \cdots\big).\end{aligned} \tag{3.5}$$

Since the plumbing graph in Figure 3.3 has the weakly negative definite property, $\hat{Z}_0(q,t)$ has a lower bound on the exponents of q, and $\hat{Z}_0(q,t=1)$ recovers the q-series for $\overline{\Sigma(2,3,5)}$.

3.2 Recovering the q-series

Let Γ be a reduced plumbing graph that realizes a 3-manifold $Y = Y(\Gamma)$. By applying the formula (3.1), we have the invariants $\hat{Z}_a(q,t)$ for Y, which can be

expressed by

$$\hat{Z}_a(q,t) = \frac{1}{|\mathcal{S}|} q^{\Delta_a} \sum_{\xi \in \mathcal{S}} \sum_{n \in \mathbb{Z}} C_{\xi,n}(t) q^n,$$

where $\Delta_a \in \mathbb{Q}$, $\mathcal{S}$ is the set of all possible chambers for Γ, and $C_{\xi,n}(t) \in \mathbb{Z}[[t]]$ is in general the formal power series in t as the coefficient of q^n inside a chamber $\xi \in \mathcal{S}$.

As we have seen in the example of Section 3.1, $C_{\xi,n}(t)$ turns out to be a finite polynomial in t if Γ has the weakly negative (or positive) definite property. Furthermore, by simply setting $t = 1$, we can recover the q-series from the (q,t)-series. Being inspired by this, it would be nice if we can recover the q-series by taking a limit $t \to 1^-$. However, without weakly definite property, $C_{\xi,n}(t)$ is in general infinite series and $\lim_{t\to 1^-} C_{\xi,n}(t)$ might be ill-defined. This means that it would be not easy to recover the q-series for such plumbings. By abuse of notation, we will denote the limit from below $t \to 1^-$ by $t \to 1$. We notice that $C_{\xi,n}(t)$ is convergent for $|t| < 1$ due to the boundedness of its coefficient polynomial.

Fortunately, there are some examples of plumbings, for which it is possible to recover the q-series by computing the limit $t \to 1$. Since the (q,t)-series $\hat{Z}_a(q,t)$ for reduced plumbings are invariant under Neumann moves, its recovered version, $\hat{Z}^R_a(q) = \lim_{t\to 1} \hat{Z}_a(q,t)$, are also invariant, therefore indeed define a topological invariant of plumbed 3-manifolds.

Remark. The recovered q-series $\hat{Z}^R_a(q)$ is exactly same as previously defined q-series $\hat{Z}_a(q)$ when plumbings are weakly negative definite. However, there are some examples where the recovered q-series are not equal to previously defined q-series $\hat{Z}_a(q)$, for example, weakly positive definite plumbings [5, 13]. Furthermore, when plumbings have strongly indefinite property, q-series $\hat{Z}_a(q)$ for them have not been defined, while $\hat{Z}^R_a(q)$ still exists in certain cases.

Example 3.4. The first example we consider is to compute the (q,t)-series for two reduced plumbings, shown in Figure 3.4, that are equivalent by a Neumann move of type (a) in Figure 2.2. The interesting point here is that the plumbing depicted Figure 3.4 (a) has the weakly negative definite property but another plumbing in Figure 3.4 (b) which is obtained by applying one Neumann move to the plumbing (a) does not have the weakly negative definite property. This means the plumbing (a) has the q-series by applying the formula (2.3) while the plumbing (b) does not have it. We are going to overcome such drawback by the recovering process.

By the formula (2.3), the q-series for the plumbing (a) is given by

$$\hat{Z}_0(q) = q^{\frac{417}{2}} \left(1 - q^{29} - q^{211} + q^{252} - q^{393} + q^{442} - q^{667} + q^{726} + \cdots\right), \tag{3.6}$$

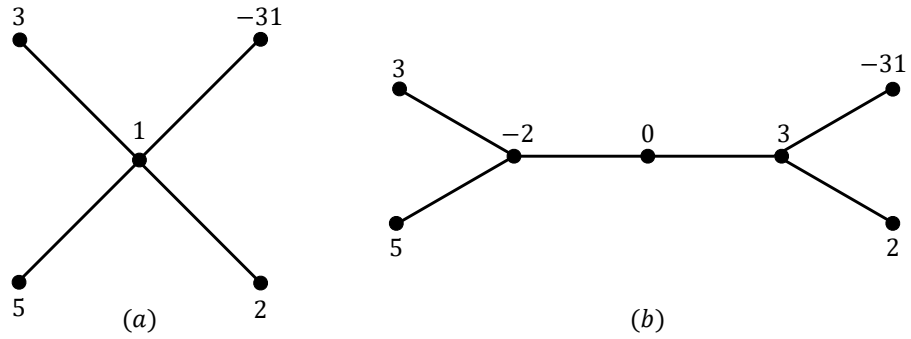


FIGURE 3.4: A Neumann move from a plumbing (a) to (b) does not preserve the weakly negative definite property.

where we notice that the determinant of the linking matrix for the plumbing is 1 and it has the unique q-series.

The formula (3.1) for the plumbing (a) reads as

$$\begin{aligned}\hat{Z}_0(q,t) = -\frac{1}{2}q^{\frac{23}{4}}\mathrm{CT}_{\vec{z}}\Bigg\{ &\sum_{r=0}^{\infty}(r+1)t^{2r+2}\left(z_1^{2r+2}+z_1^{-2r-2}\right)\\ &\times\left(z_2-\frac{1}{z_2}\right)\left(z_3-\frac{1}{z_3}\right)\left(z_4-\frac{1}{z_4}\right)\left(z_5-\frac{1}{z_5}\right)\\ &\times\sum_{\vec{\ell}\in 2M\mathbb{Z}^5+\vec{\delta}} q^{-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}}\prod_{v=1}^{5}z_v^{\ell_v}\Bigg\},\end{aligned}$$

which gives us the following result

$$\hat{Z}_0(q,t) = q^{\frac{417}{2}}t^2\left(1-q^{29}-q^{211}+q^{252}-q^{393}+q^{442}-q^{667}+q^{726}+\cdots\right). \tag{3.7}$$

It is clear that the limit $t\to 1$ of (3.7) is the same as q-series in (3.6).

Moreover, one can also check that (q,t)-series for the plumbing (b) is equal to (3.7), which is obvious by Theorem 3.3. We note that q-series for the plumbing (b) does not exist because of non-weakly negative definite property, but we can associate the recovered q-series to it.

Example 3.5. Let us consider a plumbing depicted in Figure 3.5. This is not a reduced plumbing since it has a bad branch. As we will see in Section 3.3 by using Neumann move of type (d), this plumbing represents a 3-manifold which is diffeomorphic to the disjoint union of two 3-manifolds. In order to compute

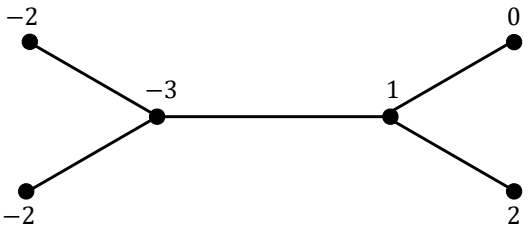


FIGURE 3.5: A plumbing graph that does not satisfy the weakly definite property.

the (q,t)-series, the first step is as usual to get the linking matrix

$$M = \begin{pmatrix} -3 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & -2 & 0 & 0 & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 \end{pmatrix},$$

and its inverse

$$M^{-1} = \frac{1}{8}\begin{pmatrix} -4 & 0 & -2 & -2 & 4 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 \\ -2 & 0 & -5 & -1 & 2 & 0 \\ -2 & 0 & -1 & -5 & 2 & 0 \\ 4 & 8 & 2 & 2 & -8 & -4 \\ 0 & 0 & 0 & 0 & -4 & 4 \end{pmatrix}.$$

Then, we have the following quantities

$$\det M = 16, \qquad \pi = \pi(M) = 2, \qquad \frac{3\sigma - \sum_v m_v}{4} = -\frac{1}{2}$$

and the set of chambers is given by

$$\mathcal{S} = \{(1,1),(1,-1),(-1,1),(-1,-1)\},$$

because no pseudo bridge is contained in the plumbing. Notice that the submatrix of M corresponding to the high-valency vertices

$$M_h = \begin{pmatrix} -1/2 & 0 \\ 0 & 0 \end{pmatrix}$$

is obviously not negative or positive definite, which means that the q-series $\hat{Z}(q)$ is ill-defined.

Among 16 Spin^c structures, in this example for simplicity we consider only one of them, that is coming from $(0,0,\dots,0) \in \operatorname{Coker} M$ by the following identification

$$\mathrm{Spin}^c(Y) \cong 2\operatorname{Coker} M + \vec{\delta}.$$

Then the application of the formula (3.1) to the plumbing leads to the following result:

$$\begin{aligned}\hat{Z}_0(q,t) = \frac{1}{4}q^{\frac{5}{8}}\big[\cdots &+ \big(-t^6+t^8-t^{10}+t^{12}-t^{14}+\cdots\big)q^{-3}\\ &+\big(-t^4+t^6-t^8+t^{10}-t^{12}+\cdots\big)q^{-2}\\ &+\big(-t^2+t^4-t^6+t^8-t^{10}+\cdots\big)q^{-1}\\ &+\big(t^2-2t^4+t^6-t^8+t^{10}+\cdots\big)q^{0}\\ &+\big(2t^4-3t^6+t^8-t^{14}+t^{16}+\cdots\big)q\\ &+\big(-t^4+3t^6-2t^8-t^{12}+t^{14}+\cdots\big)q^2\\ &+\big(t^4-t^6+2t^8-3t^{10}+t^{12}+\cdots\big)q^3\\ &+\big(t^6-2t^8+3t^{10}-2t^{12}-t^{20}+\cdots\big)q^4+\cdots\big].\end{aligned} \tag{3.8}$$

Now let us compute $\lim_{t\to 1}\hat{Z}_0(q,t)$ to recover the q-series. To do this, we need to find out the analytical property of the series $C_{\xi,n}(t)$, the coefficients of the term q^n in (3.8) from the contribution of the chamber $\xi \in \mathcal{S}$. By the role of the action $\mathrm{CT}_{\vec{z}}$ in (3.1) it follows that the vectors $\vec{\ell} = (\ell_1,\ell_2,\ell_3,\ell_4,\ell_5,\ell_6)$ should be of the form

$$\ell_1 = \xi_1(2r+1), \qquad \ell_2 = \xi_2(2s+1), \qquad \ell_i = \pm 1 \quad \text{for } i = 3,4,5,6,$$

where ξ_1 and ξ_2 are the coordinates of the chamber vector $\xi = (\xi_1,\xi_2) \in \{\pm 1\}^2$. Therefore, $C_{\xi,n}(t)$ is given by

$$C_{\xi,n}(t) = \sum_{r,s=0}^{\infty}\sum_{\varepsilon_i\in\{\pm1\}} \xi_1\xi_2\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6\, t^{2r+2s+2}\Big|_{-\frac{1}{2}-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}=\frac{5}{8}+n}, \tag{3.9}$$

where $\vec{\ell}$ is the vector of the form

$$\vec{\ell} = \big(\xi_1(2r+1),\xi_2(2s+1),\varepsilon_3,\varepsilon_4,\varepsilon_5,\varepsilon_6\big)$$

and the convention

$$\sum_{r,s}\sum_{\varepsilon_i}\Big|_{Q(\vec{\ell})=0}$$

denotes that the only r, s and ε_i satisfying the quadratic equation $Q(\vec{\ell}) = 0$ contribute to the summations.

By swapping the order of sums in (3.9), we get

$$C_{\xi,n}(t) = \sum_{\varepsilon_i\in\{\pm1\}} \xi_1\xi_2\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6 \sum_{r,s=0}^{\infty} t^{2r+2s+2}\Big|_{-\frac{1}{2}-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}=\frac{5}{8}+n}, \tag{3.10}$$

which means that given a chamber $\xi \in \mathcal{S}$ and chosen ε_i's, we have the sum of monomials in t with powers of $2r+2s+2$ from all non-negative integer solutions of the quadratic equation in two variables r and s

$$-\frac{1}{2} - \frac{(\vec{\ell}, M^{-1}\vec{\ell})}{4} = \frac{5}{8} + n,$$

moreover, the product $\xi_1\xi_2\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6$ determines the sign of the sum over r and s. Therefore, we naturally move on to the solutions of the quadratic Diophantine equations (QDEs). In Appendix A, we briefly review the algorithm to solve the QDEs in two variables for convenience.

In general, the quadratic form from a plumbing with two high-valency vertices

$$\frac{3\sigma - \sum_v m_v}{4} - \frac{(\vec{\ell}, M^{-1}\vec{\ell})}{4} = \Delta_a + n$$

is a quadratic equation with rational coefficients, but multiplying it by $4\det M$ we can obtain the one with integer coefficients. For a fixed chamber $\xi \in \mathcal{S}$ and chosen ε_i's, we denote the quadratic equation with integer coefficients by

$$Q_{\xi,\vec{\varepsilon}} = 4\det M\left(\frac{3\sigma - \sum_v m_v}{4} - \frac{(\vec{\ell}, M^{-1}\vec{\ell})}{4} - \Delta_a - n\right) = 0.$$

By putting the solutions of quadratic equations into (3.10), we obtain the explicit formula for $C_{\xi,n}(t)$, for example, $n = 2$ as follows:

$$\begin{aligned}
C_{\xi_{++},2}(t) &= -t^2 + t^4 + 2t^6 \\
&\quad + \sum_{k=1}^{\infty}\left(t^{4k^2+6k-4} + t^{4k^2+10k} + t^{4k^2+6k-2} + t^{4k^2+2k-4}\right), \\
C_{\xi_{+-},2}(t) &= -t^4 - 3t^6 - 2t^8 \\
&\quad + \sum_{k=1}^{\infty}\left(-t^{4k^2+14k+4} - t^{4k^2+10k-2} - t^{4k^2+6k-4} - t^{4k^2+10k}\right),
\end{aligned} \tag{3.11}$$

where $\xi_{++} = \{+1,+1\}$ and $\xi_{+-} = \{+1,-1\}$. We notice that $C_{\xi_{++},n}(t) = C_{\xi_{--},n}(t)$ and $C_{\xi_{+-},n}(t) = C_{\xi_{-+},n}(t)$ due to the quadratic form. Therefore, it is

enough to evaluate the limit of the form

$$\lim_{t\to 1}\sum_{k=0}^{\infty} t^{a(k+b)^2+c}, \qquad \text{for } a>0.$$

First, we change the variable by using $t=\mathrm{e}^{-\epsilon}$, then we have

$$\lim_{t\to 1}\sum_{k=0}^{\infty} t^{a(k+b)^2+c}=\lim_{\epsilon\to 0^+}\sum_{k=0}^{\infty}\mathrm{e}^{-\epsilon(a(k+b)^2+c)}.$$

We recall the Euler–Maclaurin formula

$$\sum_{k=N_i}^{N_f} f(k)=\int_{N_i}^{N_f} f(x)\mathrm{d}x+\frac{1}{2}(f(N_f)+f(N_i)) \\ +\sum_{i=2}^{j}\frac{b_i}{i!}\left[f^{(i-1)}(N_f)-f^{(i-1)}(N_i)\right]-\int_{N_i}^{N_f}\frac{B_j(\{1-x\})}{j!}f^{(j)}(x)\mathrm{d}x,$$

where N_i and N_f are real numbers such that N_f-N_i is a positive integer number, B_j and b_j denote Bernoulli polynomials and numbers, respectively, j is any positive integer and $\{x\}$ denotes the fractional part of a real number x. Applying the Euler–Maclaurin formula, we have

$$\sum_{k=0}^{\infty}\mathrm{e}^{-\epsilon(a(k+b)^2+c)}=\int_0^{\infty}\mathrm{e}^{-\epsilon a(x+b)^2-\epsilon c}\mathrm{d}x+1+O(\epsilon) \qquad \text{as } \epsilon\to 0,$$

then the evaluation of Gaussian integral returns

$$\sum_{k=0}^{\infty}\mathrm{e}^{-\epsilon(a(k+b)^2+c)}=1-b+\frac{1}{2}\sqrt{\frac{\pi}{\epsilon a}}+O(\sqrt{\epsilon}), \tag{3.12}$$

where we have used the error function's Maclaurin series

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x\mathrm{e}^{-y^2}\mathrm{d}y=\frac{2}{\sqrt{\pi}}\sum_{j=0}^{\infty}\frac{(-1)^j x^{2j+1}}{j!(2j+1)}.$$

By substituting (3.12) into (3.11), we obtain

$$C_{\xi_{++},2}(\epsilon)=3+\sqrt{\frac{\pi}{\epsilon}}+O(\sqrt{\epsilon}), \qquad C_{\xi_{+-},2}(\epsilon)=-5-\sqrt{\frac{\pi}{\epsilon}}+O(\sqrt{\epsilon}).$$

Even though the term $\sqrt{\frac{\pi}{\epsilon}}$ is singular at $\epsilon\to 0$, this singularity will disappear when we take a sum over all possible chambers. Thus, we see that the coefficient of $q^{\frac{21}{8}}$ in the recovered q-series is finite and equal to -5.

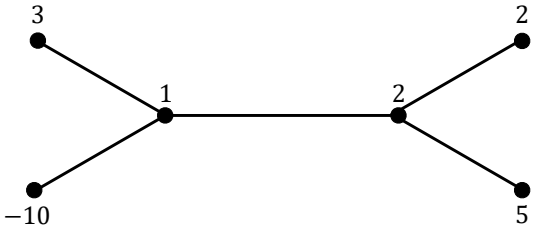


FIGURE 3.6: A plumbing graph that has the strongly indefinite property and whose linking matrix has determinant equal to 1.

By using a similar approach, we obtain the following recovered q-series

$$\hat{Z}_0^R(q) = \lim_{t\to 1} \hat{Z}_0^R(q,t) = q^{\frac{5}{8}}\big(\cdots + q^{-3} + q^{-2} + q^{-1} - 1 - q - q^2 - q^3 - q^4 + \cdots\big),$$

where the superscript denotes the recovered q-series.

In the previous example, we have seen that for some plumbings without weakly definiteness the $(q,\ t)$-series can be recovered into the q-series by computing the limit $t \to 1$. However, this does not hold for all strongly indefinite plumbings. The next example shows a such case.

Example 3.6. We consider a reduced plumbing shown in Figure 3.6. We note that just for simplicity we choose a plumbing whose linking matrix has determinant 1 such that there is a unique Spin^c structure $a = 0$.

For this plumbing, the linking matrix and its inverse are given by

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 1 \\ 1 & 0 & 3 & 0 & 0 & 0 \\ 1 & 0 & 0 & -10 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 5 \end{pmatrix},$$

$$M^{-1} = \begin{pmatrix} -390 & 300 & 130 & -39 & -150 & -60 \\ 300 & -230 & -100 & 30 & 115 & 46 \\ 130 & -100 & -43 & 13 & 50 & 20 \\ -39 & 30 & 13 & -4 & -15 & -6 \\ -150 & 115 & 50 & -15 & -57 & -23 \\ -60 & 46 & 20 & -6 & -23 & -9 \end{pmatrix}.$$

From the linking matrix, we obtain the following quantities

$$\pi = \pi(M) = 4, \qquad \frac{3\sigma - \sum_v m_v}{4} = \frac{3}{4}.$$

Since there is no pseudo bridge in the plumbing, the set $\mathcal{S}$ of all possible chambers is given by

$$\mathcal{S} = \{(1,1),(1,-1),(-1,1),(-1,-1)\}.$$

Therefore, the formula (3.1) reads

$$\begin{aligned}\hat{Z}_0(q,t) = \frac{1}{4}q^{\frac{3}{4}}\mathrm{CT}_{\vec{z}}\Bigg\{&\sum_{r=0}^{\infty} t^{2r+1}\left(z_1^{2r+1}-z_1^{-2r-1}\right)\sum_{s=0}^{\infty} t^{2s+1}\left(z_2^{2s+1}-z_2^{-2s-1}\right)\\ &\times\left(z_3-\frac{1}{z_3}\right)\left(z_4-\frac{1}{z_4}\right)\left(z_5-\frac{1}{z_5}\right)\left(z_6-\frac{1}{z_6}\right)\\ &\times\sum_{\vec{\ell}\in 2M\mathbb{Z}^6+\vec{\delta}} q^{-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}}\prod_{v=1}^{6} z_v^{\ell_v}\Bigg\}.\end{aligned} \tag{3.13}$$

The application of the formula (3.13) gives the following result:

$$\begin{aligned}\hat{Z}_0(q,t) = \frac{1}{2}q^{\frac{1}{2}}\big[&\cdots+\left(2t^{30}+t^{46}-t^{108}+\cdots\right)q^{-4}+\left(t^8-t^{10}-t^{22}+\cdots\right)q^{-3}\\ &+\left(2t^6-t^8-t^{16}+\cdots\right)q^{-2}+\left(t^8+t^{12}+t^{28}+\cdots\right)q^{-1}\\ &+\left(2t^2-t^{12}-t^{564}+\cdots\right)+\left(-t^2+t^{16}+t^{36}+\cdots\right)q\\ &+\left(-t^6-t^{18}-t^{34}+\cdots\right)q^3+\left(-t^{12}-t^{20}-t^{60}+\cdots\right)q^4+\cdots\big].\end{aligned}$$

We note that all the coefficients of q^n in this result are infinite series in t because of the strong indefiniteness of the linking matrix.

For a chosen chamber $\xi = (\xi_1,\xi_2) \in \{\pm 1\}^2$, we know that the vectors $\vec{\ell} = (\ell_1,\ell_2,\ell_3,\ell_4,\ell_5,\ell_6)$ should be of the form

$$\ell_1 = \xi_1(2r+1), \qquad \ell_2 = \xi_2(2s+1), \qquad \ell_i = \pm 1 \quad \text{for } i = 3,4,5,6,$$

and $C_{\xi,n}(t)$ is given by

$$C_{\xi,n}(t) = \sum_{\varepsilon_i\in\{\pm1\}} \xi_1\xi_2\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6 \sum_{r,s=0}^{\infty} t^{2r+2s+2}\Big|_{\frac{3}{4}-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}=\frac{1}{2}+n}. \tag{3.14}$$

According to the algorithm in Appendix A, the nonnegative integer solutions of

$$Q_{\xi,\vec{\varepsilon}} = \frac{3}{4} - \frac{(\vec{\ell},M^{-1}\vec{\ell})}{4} - \frac{1}{2} - n = 0$$

are determined by a finite number of families of solutions, and each family $\{(r_k, s_k)\}_k$ of solutions has the following form:

$$r_k = \alpha_r A^k + \beta_r A^{-k} + \eta_r, \qquad s_k = \alpha_s A^k + \beta_s A^{-k} + \eta_s,$$

where $k \geq k_0 \in \mathbb{Z}_+$. Here the real numbers α_r, α_s, β_r, β_s, η_r, η_s are constants which do depend on the choice of the chamber ξ and the vector $\vec{\varepsilon}$. However, the A does not depend on ξ and $\vec{\varepsilon}$. This is because all QDEs $Q_{\xi,\vec{\varepsilon}} = 0$ from different ξ and $\vec{\varepsilon}$ can be transformed into the same generalized Pell's equation. From the requirement for non-negative solutions, it follows that α_r and α_s are positive real numbers and $A > 1$.

For each family of solutions, we obtain a series in t by (3.14) as following:

$$\sum_{k \geq k_0} t^{2(\alpha_r+\alpha_s)A^k + 2(\beta_r+\beta_s)A^{-k} + 2(\eta_r+\eta_s)}.$$

In order to compute the limit $t \to 1$, we use the substitution $t \to \mathrm{e}^{-\epsilon}$

$$\sum_{k \geq k_0} \mathrm{e}^{-2\epsilon[(\alpha_r+\alpha_s)A^k + (\beta_r+\beta_s)A^{-k} + \eta_r + \eta_s]}.$$

By applying doubly exponential series transformations of Ramanujan [2], we can estimate the above series as follows:

$$\begin{aligned}
&\sum_{k \geq k_0} \mathrm{e}^{-2\epsilon[(\alpha_r+\alpha_s)A^k + (\beta_r+\beta_s)A^{-k} + \eta_r + \eta_s]} \\
&\sim \frac{1}{2} + \frac{1}{\log A}\left[-\log\epsilon - \gamma - \log 2 - \log(\alpha_r + \alpha_s)\right] \\
&-k_0 + O(\epsilon) + 2\sum_{k=1}^{\infty} \sqrt{\frac{\log A}{2k \sinh \frac{2k\pi^2}{\log A}}} \\
&\times \cos\left(\frac{2k\pi}{\log A} \log \frac{k\pi}{\epsilon(\alpha_r + \alpha_s)\log A} - \frac{2k\pi}{\log A} - \frac{\pi}{4} - \frac{B_2 \log A}{4k\pi} + \cdots\right),
\end{aligned} \tag{3.15}$$

where γ is the Euler's constant $\gamma = 0.57721\ldots$ and B_k denotes the kth Bernoulli number defined by

$$\frac{z}{\mathrm{e}^z - 1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} z^k, \qquad |z| < 2\pi.$$

Now let us analyse the terms in the right side of (3.15). First, the second term proportional to $\log\epsilon$ has a singularity at $\epsilon \to 0$, but this term will be cancelled out by taking a sum of $C_{\xi,n}(t)$ over all choices of chambers $\xi \in \mathcal{S}$, which allows us to ignore this term. Next, the infinite series of oscillating terms has a singularity

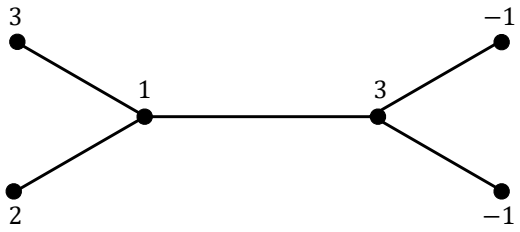


FIGURE 3.7: A non-reduced plumbing graph that realizes the 3-manifold $\overline{\Sigma(2,3,5)}$.

as well. Furthermore, this terms do not disappear by a sum of $C_{\tilde{\zeta},n}(t)$. Thus, we conclude that due to such series of oscillating terms the limit $t \to 1$ is ill-defined and the recovered q-series does not exist.

Example 3.7. In this example, we consider a non-reduced plumbing shown in Figure 3.7, which can be obtained by Neumann moves from the one in Figure 3.3. Hence, the 3-manifold realized by the plumbing graph in Figure 3.7 is homeomorphic to the Brieskorn sphere $\overline{\Sigma(2,3,5)}$. One can easily check that the high-valency vertex with weight 3, placed in the right side of the plumbing, is a pseudo high-valency vertex because the branches are both pseudo.

We have already computed the (q,t)-series (3.5) in the last part of Section 3.1. Now we are going to compute the (q,t)-series for the non-reduced plumbing in Figure 3.7 and compare it with the result in (3.5).

The linking matrix of the plumbing is given by

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 3 & 0 & 0 & 1 & 1 \\ 1 & 0 & 3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix},$$

and its inverse is

$$M^{-1} = \begin{pmatrix} -30 & 6 & 10 & 15 & 6 & 6 \\ 6 & -1 & -2 & -3 & -1 & -1 \\ 10 & -2 & -3 & -5 & -2 & -2 \\ 15 & -3 & -5 & -7 & -3 & -3 \\ 6 & -1 & -2 & -3 & -2 & -1 \\ 6 & -1 & -2 & -3 & -1 & -2 \end{pmatrix}.$$

The formula (3.1) reads

$$\begin{aligned}\hat{Z}_0(q,t) = \frac{1}{4}q^{-\frac{7}{4}}\mathrm{CT}_{\vec{z}}\Bigg\{&\sum_{r=0}^{\infty} t^{2r+1}\left(z_1^{2r+1}-z_1^{-2r-1}\right)\left(z_3-\frac{1}{z_3}\right)\left(z_4-\frac{1}{z_4}\right)\\ &\times\sum_{s=0}^{\infty} t^{2s+1}\left[z_2^{2s+1}\left(z_5t-\frac{1}{z_5t}\right)\left(z_6t-\frac{1}{z_6t}\right)\right.\\ &\left.-z_2^{-2s-1}\left(\frac{z_5}{t}-\frac{t}{z_5}\right)\left(\frac{z_6}{t}-\frac{t}{z_6}\right)\right]\\ &\times\sum_{\vec{\ell}\in 2M\mathbb{Z}^6+\vec{\delta}} q^{-\frac{(\vec{\ell},M^{-1}\vec{\ell})}{4}}\prod_{v=1}^{6} z_v^{\ell_v}\Bigg\},\end{aligned}$$

and its computation gives an interesting result as follows:

$$\begin{aligned}\hat{Z}_0(q,t) = &\frac{1+t^2}{2}\\ &\times q^{-\frac{3}{2}}\left(1-q-q^3-q^7+tq^8+q^{14}+q^{20}+t^2q^{29}-q^{31}-t^2q^{42}+\cdots\right).\end{aligned}$$

Here one might be curious about the result that coefficients of monomials in q are finite polynomials in t even though the plumbing is strongly indefinite. The key to the answer is that there are cancellations between each pair of vectors $\vec{\ell}_1 = (\ldots,+1)$ and $\vec{\ell}_2 = (\ldots,-1)$, where ± 1 denotes the element of the vector $\vec{\ell}$ associated to valency one vertex on any pseudo branch.

We notice that except an extra factor $\frac{1+t^2}{2t}$, this result completely agrees with (3.5), furthermore the recovered q-series by taking $t=1$ are the same.

Let us reiterate that Theorem 3.3 says that the (q,t)-series for a reduced plumbing defined in (3.1) is unchanged by the Neumann moves, thus the recovered q-series also remains unchanged if the limit $t\to 1$ exists. The Example 3.7 tells us that the (q,t)-series for a non-reduced plumbing is changed by Neumann moves, but the recovered q-series might be still preserved. This property indeed holds for non-reduced plumbings with at most two high-valency vertices.

Proposition 3.8. *Assume that the (q,t)-series for a given plumbing can be finitely recovered q-series in a sense that the limit $t\to 1$ is finite. Then the recovered q-series is preserved by arbitrary Neumann moves even for non-reduced plumbings with at most two high-valency vertices.*

Proof. Since the (q,t)-series gets changed only when a Neumann move creates a pseudo high-valency vertex, it is sufficient to show that the recovered q-series remains unchanged for such a Neumann move. Therefore, we are going to

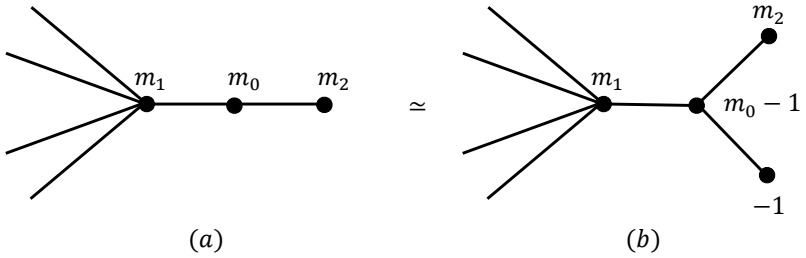


FIGURE 3.8: A reduced plumbing (a) with one high-valency vertex turns out to be a non-reduced plumbing (b) with two high-valency vertices by a Neumann move.

consider a case shown in Figure 3.8, where the left plumbing (a) is a reduced one with one high-valency vertex weighted by m_1 and the right plumbing (b) has an extra pseudo high-valency vertex created by a Neumann move of type (c) in Figure 2.2. The cases for other types of Neumann moves creating a new pseudo high-valency vertex can be handled in a similar way.

Let Γ be the left plumbing and Γ' be the right one in Figure 3.8. We use a prime to denote the quantities for the right plumbing. Let us write a vector $\vec{\ell}$ by

$$\vec{\ell} = (\vec{\ell}_1, 0, \ell_2),$$

where $\vec{\ell}_1$ corresponds to the left part of the graph Γ including the vertex m_1. As we have already seen in the proof of Theorem 3.3, we have corresponding vectors for Γ'

$$\vec{\ell}'_{\pm} = (\vec{\ell}_1, \pm 1, \ell_2, \mp 1),$$

which give the exactly same contribution as $\vec{\ell}$. However, due to the existence of pseudo high-valency vertex in Γ', we have the following extra vectors $\vec{\ell}'$ corresponding to $\vec{\ell}$

$$\vec{\ell}'_{1,\pm} = (\vec{\ell}_1, 2s+1, \ell_2, \pm 1), \qquad \vec{\ell}'_{2,\pm} = (\vec{\ell}_1, -2s-1, \ell_2, \pm 1)$$

for $s \in \mathbb{N}$. It is immediate that those infinitely many extra vectors generate extra t-series. Therefore, we need to show that the limit $t \to 1$ of those extra t-series is equal to zero.

Let us consider the extra t-series coming from the vectors $\vec{\ell}'_1$. The case for the vectors $\vec{\ell}'_2$ is similar. For some $n \in \mathbb{Z}$ and $\Delta_a \in \mathbb{Q}$, the t-series coefficient of $q^{\Delta_a + n}$ from those vectors is determined by

$$\sum_{r=0}^{\infty} \sum_{s=1}^{\infty} \left\{ t^{2r+2s+p_r+2} \Big|_{Q(\vec{\ell}'_{1,+})} - t^{2r+2s+p_r} \Big|_{Q(\vec{\ell}'_{1,-})} \right\},$$

where $p_r = \deg(v_1) - 2$ and $Q(\vec{\ell}')$ denotes the following quadratic equation

$$\frac{3\sigma' - \sum_I m'_I}{4} - \frac{(\vec{\ell}', M'^{-1}\vec{\ell}')}{4} = \Delta_a + n.$$

This means that those extra t-series would be zero after taking the limit $t \to 1$ if the non-negative solutions of two quadratic equations $Q(\vec{\ell}'_{1,\pm})$ are expressed by the families of solutions with the same patterns.

Indeed, this is the case. As we have mentioned in Appendix A, the first step to solve general quadratic Diophantine equation (A.1) is to transform it into the form of generalized Pell's equation (A.3) by using the following transformations

$$D = b^2 - 4ac, \qquad E = bd - 2ae, \qquad F = d^2 - 4af, \qquad N = E^2 - DF.$$

Therefore, the solutions of (A.1) are expressed by the factors D, E and F together with the solutions of the generalized Pell's equation. We have already seen that two quadratic equations $Q(\vec{\ell}'_{1,\pm})$ are transformed into the same generalized Pell's equation. Moreover, it is an algebraic exercise that these equations have the same transforming factors D, E and F. Thus, the solutions of two quadratic equations have the same patterns, which completes the proof. □

Based on Proposition 3.8, we naturally suggest the following conjecture:

Conjecture 3.9. *The recovered q-series, obtained from the (q,t)-series in the sense of $t \to 1$, is an invariant of 3-manifolds realized by arbitrary (i.e., possibly non-reduced) plumbings.*

Remark. As we have seen in Sections 3.1 and 3.2, $\hat{Z}_a(q,t)$ is defined for reduced plumbings and it has a nice property that it is not only an invariant of 3-manifolds realized by reduced plumbings but also a refined version of $\hat{Z}_a(q)$, i.e., $\lim_{t\to 1} \hat{Z}_a(q,t) = \hat{Z}_a(q)$ for weakly negative definite plumbings. Furthermore, it provides a key to define $\hat{Z}_a(q)$ even for strongly indefinite plumbings if its limit as $t \to 1$ exists.

For non-reduced plumbings, it is dependent on the presentation of plumbings and it undergoes changed by the Neumann moves. However, as we demonstrates in Example 3.7 and Proposition 3.8, the limit $\lim_{t\to 1} \hat{Z}_a(q,t)$, if exists, is still an invariant for non-reduced plumbings with at most two high-valency vertices. Thus, if Conjecture 3.9 holds, then we could construct an invariant of all tree plumbed 3-manifolds by obtaining the recovered q-series under the assumption of the existence of the limit $t \to 1$.

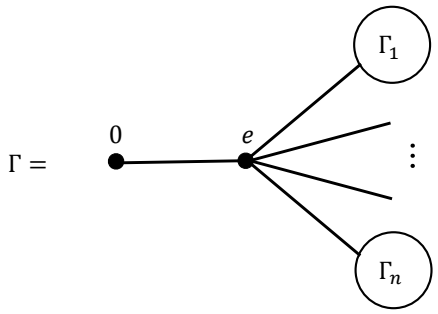


FIGURE 3.9: A plumbing graph Γ that is equivalent to the disjoint union of $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$. Here the weight e can be any integer.

3.3 Connected sum of plumbed 3-manifolds

Let Γ_1 and Γ_2 be plumbing graphs and Y_1, Y_2 be 3-manifolds realized by Γ_1, Γ_2, respectively. It is well-known that the connected sum $Y_1 \sharp Y_2$ can be realized by the disjoint union $\Gamma_1 \sqcup \Gamma_2$. The following proposition [34] says that the disjoint union of finite number of plumbing graphs is equivalent a certain single plumbing graph.

Proposition 3.10. *If a plumbing graph Γ has the form of Figure 3.9, then the 3-manifold $Y(\Gamma)$ is homeomorphic to the connected sum of $Y(\Gamma_1), Y(\Gamma_2), \ldots, Y(\Gamma_n)$.*

Let $\hat{Z}_{a_j}(\Gamma_j; q, t)$ be the (q,t)-series for Γ_j with Spin^c structure a_j, where $j = 1, 2$. Then, according to Proposition 3.10, the connected sum of $Y_1 = Y(\Gamma_1)$ and $Y_2 = Y(\Gamma_2)$ can be realized by a plumbing Γ in the form of Figure 3.9. Just for simplicity, we assume that Γ_j is connected to the vertex e through a valency one vertex of a non-pseudo branch in Γ_j. This assumption is just to avoid creating a new high-valency vertex or a new pseudo bridge.

We now present a formula for the (q,t)-series $\hat{Z}_a(\Gamma; q, t)$.

Proposition 3.11. *The (q,t)-series of Γ can be expressed in terms of the product of $\hat{Z}_{a_j}(\Gamma_j; q, t)$, $j = 1, 2$ as following:*

$$\hat{Z}_a(\Gamma; q, t) = \frac{1}{2} \sum_{r=0}^{\infty} t^{2r+1} (q^{-\frac{2r+1}{2}} - q^{\frac{2r+1}{2}}) \cdot \hat{Z}_{a_1}(\Gamma_1; q, t) \hat{Z}_{a_2}(\Gamma_2; q, t). \tag{3.16}$$

It is worth to notice here that the series

$$\frac{1}{2} \sum_{r=0}^{\infty} (q^{-\frac{2r+1}{2}} - q^{\frac{2r+1}{2}})$$

is the symmetric expansion of a rational function

$$\frac{1}{q^{1/2}-q^{-1/2}},$$

that is just the inverse of the q-series of 3*-sphere.*

Proof. Suppose that the linking matrices of Γ_1 and Γ_2 are given by

$$M_1 = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nn} \end{pmatrix}, \qquad M_2 = \begin{pmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{1m} & \cdots & b_{mm} \end{pmatrix},$$

then the linking matrix of Γ is expressed by

$$M = \begin{pmatrix} a_{11} & \cdots & a_{1n} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nn} & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & 1 & e & 1 & 0 & \cdots \\ 0 & \cdots & 0 & 0 & 1 & b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & b_{1m} & \cdots & b_{mm} \end{pmatrix}. \tag{3.17}$$

Also, (3.17) implies

$$\begin{aligned} \det M &= -\det M_1 \cdot \det M_2, \\ \sigma(M) &= \sigma(M_1) + \sigma(M_2), \\ \pi(M) &= \pi(M_1) + \pi(M_2) + 1. \end{aligned} \tag{3.18}$$

Furthermore, the inverse matrix M^{-1} reads as following:

$$M^{-1} = \begin{pmatrix} A_{11} & \cdots & A_{1n} & -A_{1n} & 0 & & & \\ \vdots & \ddots & \vdots & \vdots & \vdots & & & \\ A_{1n} & \cdots & A_{nn} & -A_{nn} & 0 & & & \\ -A_{1n} & \cdots & -A_{nn} & A_{nn}+B_{11}-e & 1 & -B_{11} & \cdots & -B_{1m} \\ 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ & & & -B_{11} & 0 & B_{11} & \cdots & B_{1m} \\ & & & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & -B_{1m} & 0 & B_{1m} & \cdots & B_{mm} \end{pmatrix}, \tag{3.19}$$

where

$$M_1^{-1} = \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{1n} & \cdots & A_{nn} \end{pmatrix}, \qquad M_2^{-1} = \begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{1m} & \cdots & B_{mm} \end{pmatrix}$$

are the inverse matrices of M_1 and M_2, respectively. From (3.18), we have following relations:

$$\begin{aligned} (-1)^{\pi(M)} &= -(-1)^{\pi(M_1)} \cdot (-1)^{\pi(M_2)}, \\ q^{\frac{3\sigma - \sum_I m_I}{4}} &= q^{\frac{3\sigma_1 - \sum_{I_1} m_{I_1}}{4}} \cdot q^{\frac{3\sigma_2 - \sum_{I_2} m_{I_2}}{4}} \cdot q^{-\frac{e}{4}}. \end{aligned}$$

Now let us write vectors $\vec{\ell}$ as

$$\vec{\ell} = (\ell_1, \ldots, \ell_n \pm 1, \pm 1, \ell_e, \ell'_1 \pm 1, \ldots, \ell'_m),$$

with the entry ℓ_e being for the vertex e and ± 1 being for the vertex weighted by 0. Then, it follows from the explicit expression (3.19) for M^{-1} that we have

$$(\vec{\ell}, M^{-1}\vec{\ell}\,) = (\vec{\ell}_1, M_1^{-1}\vec{\ell}_1) + (\vec{\ell}'_2, M_2^{-1}\vec{\ell}'_2) - e - 2\ell_e,$$

which implies

$$\Theta_a^{-M}(\vec{z}) = \Theta_{a_1}^{-M_1}(\vec{z_1}) \left(z_n - \frac{1}{z_n}\right) \Theta_{a_2}^{-M_2}(\vec{z_2}) \left(z'_1 - \frac{1}{z'_1}\right) \sum_{\ell_e} q^{\frac{e+2\ell_e}{4}} z_e^{\ell_e}.$$

Putting all together into the defining formula (3.1), it is straightforward to get (3.16). □

Chapter 4

Graph Neural Networks and Plumbed 3-Manifolds

In this chapter, we consider the class of 3-manifolds described by plumbing graphs and use Graph Neural Networks (GNN) for the problem of deciding whether a pair of graphs give homeomorphic 3-manifolds. We use supervised learning to train a GNN that provides the answer to such a question with high accuracy. Moreover, we consider reinforcement learning by a GNN to find a sequence of Neumann moves that relates the pair of graphs if the answer is positive. The setting can be understood as a toy model of the problem of deciding whether a pair of Kirby diagrams give diffeomorphic 3- or 4-manifolds.

4.1 Supervised Learning

In this section we use supervised learning to decide whether or not two plumbing graphs represent a same plumbed 3-manifold. We build 3 models GEN+GAT, GCN+GCN and GCN+GAT and examine their performance for the task.[1]

4.1.1 Models

All the models are designed to have two convolution operators, one aggregation layer and one classification layer. The models are named by concatenating the names of two convolution operators. For a fair comparison, we use the common aggregation layer and the classification layer, and all the layers have the same dimensions for both input and output.

Since we have already reviewed the convolution operators in Section 2.2, let us now elaborate on the common aggregation layer and classification layer.

[1]Since our task is a graph similarity learning, we had expected that GEN could perform well and we performed GEN combined with various GNNs known in the literature. GEN+GAT performed best among different architectures we tried, and two models GCN+GCN and GCN+GAT are used for a benchmark because GCN and GAT are most commonly used GNNs.

The aggregator computes a graph embedding by aggregating all of its node embeddings, passed from convolution operators. We use the aggregation layer proposed in [27], which is formulated by

$$\mathbf{h}_G = \mathrm{MLP}_G\left(\sum_{i\in V} \mathrm{Softmax}(\mathrm{MLP}_{\mathrm{gate}}(\mathbf{x}_i)) \odot \mathrm{MLP}(\mathbf{x}_i)\right),$$

where $\mathbf{h}_G$ is a graph-level output and $\odot$ denotes element-wise multiplication.

The classification layer plays a role to determine, for a given pair of plumbing graphs, whether or not they are equivalent. This layer has the concatenation of two graph embeddings as its input and classifies into two classes, class 0 and class 1. Here class 1 means two plumbing graphs are equivalent while class 0 denotes they are inequivalent. We implement the classification layer by using MLP with two hidden layers.

A detailed information of the architecture for 3 models are presented in Table 4.1, and the pipeline for GEN+GAT model is depicted in Figure 4.1. For each layer in the table, the first element in the bracket followed by a name of model denotes the dimension of input vectors of the layer while the second one denotes the dimension of output embedding.

TABLE 4.1: The architecture of 3 models with parameter values.

Layers	GEN+GAT	GCN+GAT	GCN+GCN
First convolution	GEN(1, 128)	GCN(1, 128)	
Second convolution	GAT(128, 128)		GCN(128, 128)
Aggregation	Aggregator(128, 32)		
Classification	MLP(64, 2)		

4.1.2 Experimental Settings

For training and validation, we put together datasets including 80,000 random pairs of plumbings generated by algorithms presented in Appendix B. More explicitly, the datasets consists of

- 40,000 pairs of equivalent plumbings generated by EQUIVPAIR, Algorithm 3, with $N_{\max} = 40$.

  To generate a pair of equivalent plumbings, the algorithm starts with

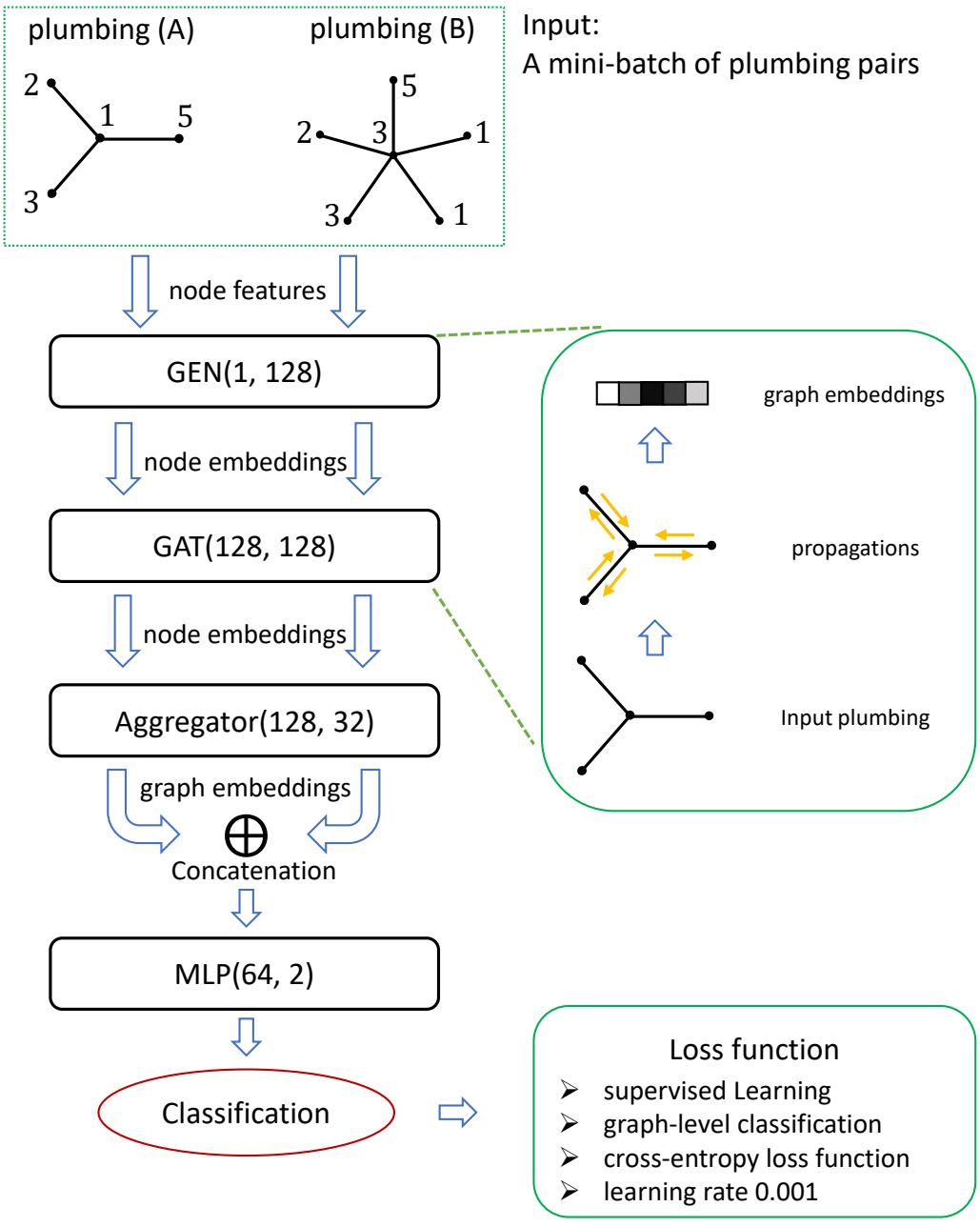


FIGURE 4.1: The architecture of GEN+GAT model.

a random plumbing created by RANDOMPLUMBING, Algorithm 1, and iteratively applies Neumann moves using RANDOMNEUMANNMOVE, Algorithm 2, up to $N_{\max}$ times, to each plumbing in the pair.

- 30,000 pairs of inequivalent plumbings generated by INEQUIVPAIR, Algorithm 4, with $N_{\max} = 40$.
  It has a similar process to EQUIVPAIR, but it starts with a pair of inequivalent plumbings, each of which is separately generated by RANDOMPLUMBING.[2]

- 10,000 pairs of inequivalent plumbings generated by TWEAKPAIR, Algorithm 5, with $N_{\max} = 40$.
  This algorithm generates a pair of inequivalent plumbings, one of which is obtained by tweaking the other. Here, by tweaking a plumbing, we mean that we make a small change of the weight (or node feature) of a randomly chosen node in the plumbing. Since tweaking is different from Neumann moves, this process creates an inequivalent plumbing to the original one. After tweaking, it also applies RANDOMNEUMANNMOVE iteratively up to $N_{\max}$. These pairs are added into the datasets in order for the models to make the decision boundary more accurate, since for a pair generated by INEQUIVPAIR, two plumbings might be quiet different due to random generators.

We divide the datasets into training and validation sets by the ratio 8:2. We train our models on training sets containing 64,000 pairs of plumbings up to 150 epochs. For each model, we use cross-entropy loss for a loss function and Adam for an optimizer with the learning rate 0.001.

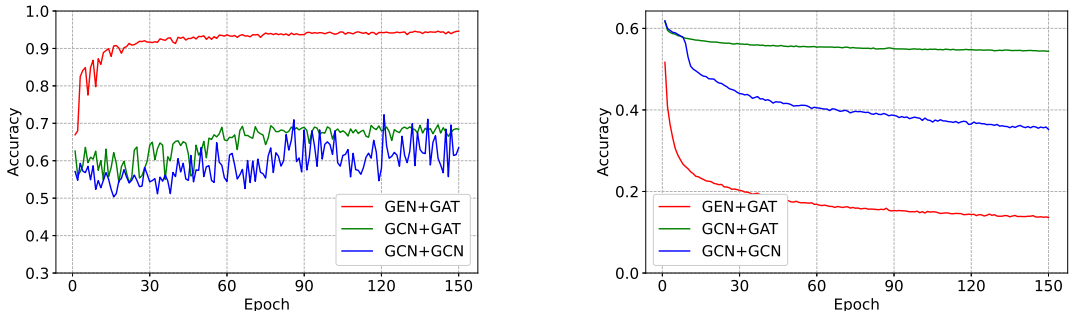
4.1.3 Results

The comparison of the performance between 3 models is plotted in Figure 4.2. We find that GEN+GAT model significantly outperforms the other models GCN+GAT and GCN+GCN. The model GCN+GAT seems to outperform GCN+GCN by few percent, but the performance difference is negligible. [3]

We have also tried other models such as GEN+GEN, GEN+GCN and GAT+GAT to figure out which convolution operators has an important role.

[2] We note that two plumbings, generated in INEQUIVPAIR and TWEAKPAIR by running RANDOMPLUMBING twice, could be accidentally equivalent and this might affect the accuracy of models in training. However, we will ignore this since it is statistically insignificant.

[3] For GCN+GAT and GCN+GCN models, we have checked that increasing weight dimensions and longer training phases did not lead to better performance.



(A) Performance comparison on validation sets.

(B) Loss comparison on validation sets.

FIGURE 4.2: Overview of the performance and loss comparison between GEN+GAT, GCN+GAT and GCN+GCN models.

The model GEN+GCN shows similar performance with GEN+GAT, but slightly underperforms, and the performance of GAT+GAT is somewhere between that of GEN+GAT and GCN+GAT. This means that GEN plays a significant role to evaluate equivalence or inequivalence for a pair of plumbing graphs. However, we found that GEN+GEN does not perform as good as GEN+GAT or GEN+GCN.

We used the following datasets to test our models:

- Test set 1
  It contains 5,000 pairs of equivalent plumbing graphs generated by EQUIVPAIR with $N_{\max} = 40$ and 5,000 pairs of inequivalent plumbings generated by INEQUIVPAIR with $N_{\max} = 40$.

- Test set 2
  This dataset is similar to Test set 1, but with $N_{\max} = 60$.

- Test set 3
  This set is also similar to Test set 1, but with $N_{\max} = 80$.

- Test set 4
  It contains 64 pairs of plumbings generated in a manual way such that, for each pair, the determinants of adjacency matrices (with weights on the diagonal) of two plumbings are the same. We use this Test set in order to check that graph embeddings from the models are not just functions of the determinant of the adjacency matrix of a plumbing. All types of Neumann moves have the property that it preserves the determinant of the adjacency matrix of the plumbing, which is the order of the first homology group of the corresponding 3-manifold. We wish graph embeddings to not depend on the determinant only, but be more sophisticated (approximate) invariants of plumbed 3-manifolds.

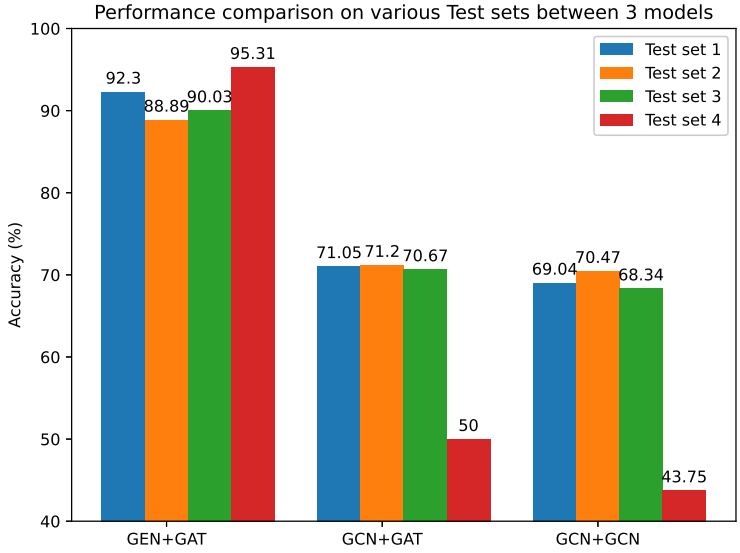


FIGURE 4.3: Performance comparison between 3 models, GEN+GAT, GCN+GAT and GCN+GCN on various Test sets. The error bars are not displayed in the figure since the standard errors on Test set 1, 2, and 3 are too small (smaller than 0.7) to notice. The standard errors on Test set 4 are about 2.64, 6.25, and 6.20 for GEN+GAT, GCN+GAT and GCN+GCN models, respectively.

The results are depicted in Figure 4.3 and they enlighten us with the following two points. The first point is that the accuracy for Test set 2 and Test set 3 is almost the same level as Test set 1 even though Test set 2 and 3 contain plumbing pairs with larger $N_{\max}$ than Test set 1. It is perhaps surprising that such somewhat counter-intuitive property holds even for GCN+GAT and GCN+GCN models, which show less training accuracy than GEN+GAT. The second point is that GEN+GAT model still outperforms the others for Test set 4 and it can distinguish correctly even inequivalent pairs with the same determinants. Since GEN is designed for graph similarity learning and to have a good generalization, we can see that the model GEN+GAT outperforms significantly the others GCN+GCN and GCN+GAT, designed for general classification problems (with a relatively small number of classes), on various Test sets.

4.2 Reinforcement Learning

In this section, we consider reinforcement learning of a neural network that allows, for a given pair of plumbings, not only to recognize whether they are equivalent or not, but also to find out their simplest representations.

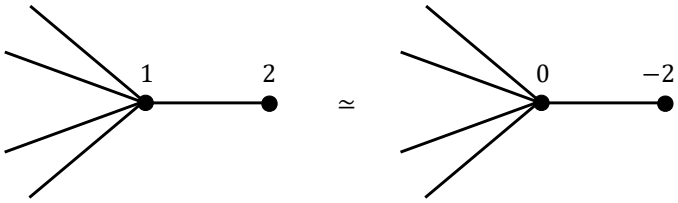


FIGURE 4.4: Two plumbings are equivalent and they have same number of nodes. The right-hand side plumbing is simpler than the left one in the sense of (4.1).

4.2.1 The environment

State space

In our RL environment, the plumbing graph defines the state and the state space is infinity. In order to handle the start state and terminal stats in an easy way, we set the start state for an episode is set to be a plumbing generated by RANDOMPLUMBING, Algorithm 1, with number of nodes equal to 10, then applying Neumann moves $N = 15$ times.

Between two equivalent plumbings, we define a relation as follows: for two equivalent states s_1 and s_2, one state is said to be *simpler* than the other if

$$f(s_1) < f(s_2),$$

where $f(s)$ for a state s is defined by

$$f(s) := 5|V(s)| + \sum_{v \in V(s)} |w(v)|. \tag{4.1}$$

It is easy to check that this relation is well-defined in a set of all equivalent plumbings.

One might think that number of nodes in a state is enough to decide which state is simpler. The reason why we add the sum of the absolute values of the weights of nodes is to make the simplest state generically unique[4]. For example, two plumbings depicted in Figure 4.4 have same number of nodes and it is easy to check that they are equivalent by applying 2 Neumann moves. In this example, we say that the plumbing on the right-hand side is simpler than the other from (4.1).

[4]There still could be specific examples with different plumbings in the same equivalence class that minimize $f(s)$. However, as the results below suggest, such cases are statistically insignificant.

By using this comparison relation, we set the terminal state to be a state equal to or simper than the initial state in the episode. We also terminate each episode after taking 15 time steps.

Action space

An action for the agent in a state is defined to be a Neumann move applied to one of the nodes. There are 8 possible Neumann moves: 5 blow-up moves and 3 blow-down moves. However, blow-down moves are not always available for all nodes and this could raise a problem that there might be too many of such *illegal* actions. Therefore, we incorporate 3 blow-down moves into one such that it takes an available blow-down if the corresponding node satisfies one of three following conditions:

- the degree of the node is 2 and its weight is equal to ± 1,
- the degree of the node is 1 and its weight is equal to ± 1,
- the degree of the node is 1 and its weight is equal to 0.

Then, for a given state, the total number of possible actions is equal to 6 (5 blow-up moves and 1 blow-down move) times number of nodes in the state. If the agent takes an illegal action, then the next state remains the same state as the current state and the agent will be punished with a negative reward, on which we will elaborate soon.

Rewards

Since the goal for the RL agent is to find out the simplest representation for an initial state, it is natural to use $-f(s')$ as a reward (or punishment $+f(s')$) for taking an action in the current state s, where s' denotes the next state obtained by taking an action to the current state s. Since all the rewards are negative and simpler state is less punished, it helps the agent not only make the current representation as simple as possible, but also do this job as fast as possible. It is also important to note that some states must get a new blow-up node in order to be simplified, which means the agent has to sacrifice the immediate reward at some time steps to maximize the total return. As we have seen previously, there are some illegal actions in the action space for each state. The reward for such illegal actions is set to be equal to $-2f(s')$ for the next state s', which remains the same as the current state s as we have discussed above.

We set the discount factor as $\gamma = 0.99$, very close to 1.

4.2.2 The deep RL algorithm

We remind that the RL task is to obtain the simplest representation from a given initial state by using Neumann moves. To accomplish this task, we used Asynchronous Advantage Actor-Critic (A3C) [31] as an RL algorithm, which is the asynchronous version of Actor-Critic (AC) [25], with feedforward GNNs. A3C executes multiple local AC agents asynchronously in parallel to decorrelate the local agent's data into a more stationary process. It also provides practical benefits of being able to use only multi-core CPU, not having to rely on specialized hardware such as GPUs.

The Actor network defines the policy function $\pi(a|s)$, whose output shows the probability of taking action a in state s, while the Critic network is to approximate the value function $V^{\pi}(s)$, which represents the expected return from state s. Since the inputs of the Actor and Critic are plumbing graphs, in the context of GNNs, the Actor network can be thought as the GNNs for node-level action-selection problem and the Critic is for graph-level estimation problem. The architecture of the Actor is designed by using two graph convolutional layers GCN+GCN and two single-layer feedforward neural networks. The Critic has a similar structure, but it has an extra aggregation layer, for which we used a simple mean function. In both Actor and Critic networks, we used skip connections that connect the inputs of the networks to the inputs of a feedforward neural networks in order to accelerate the training process of GNNs [47]. Those two GNNs are depicted in Figure 4.5. We have also tried GEN+GAT and GEN+GCN for the convolutional layers in the Actor and Critic networks. They seemed to perform well, but it takes a bit longer time for training than GCN+GCN. Since the results with GCN+GCN were already pretty good, we ended up using GCN+GCN.

We trained the agents for 8×10^4 episodes using 8 CPU cores and no GPU, which takes around 8 hours. We used Adam optimizer with learning rate 5×10^{-4}. For a comparison, we have also implemented Deep Q-Network (DQN) [32] with feedforward GNNs GCN+GCN with the same settings as those for A3C.

4.2.3 Results

Our RL agents can be used to find the simplest representative in the equivalence class of a given plumbing graph. Furthermore, it also can be used to check whether a pair of plumbing graphs represents the same 3-manifold or not. For the latter purpose, we run the RL agents on a pair of plumbings to get the simplest representations for two plumbings, then we compare those to decide whether two equivalent plumbings are isomorphic or not. This process provides us with

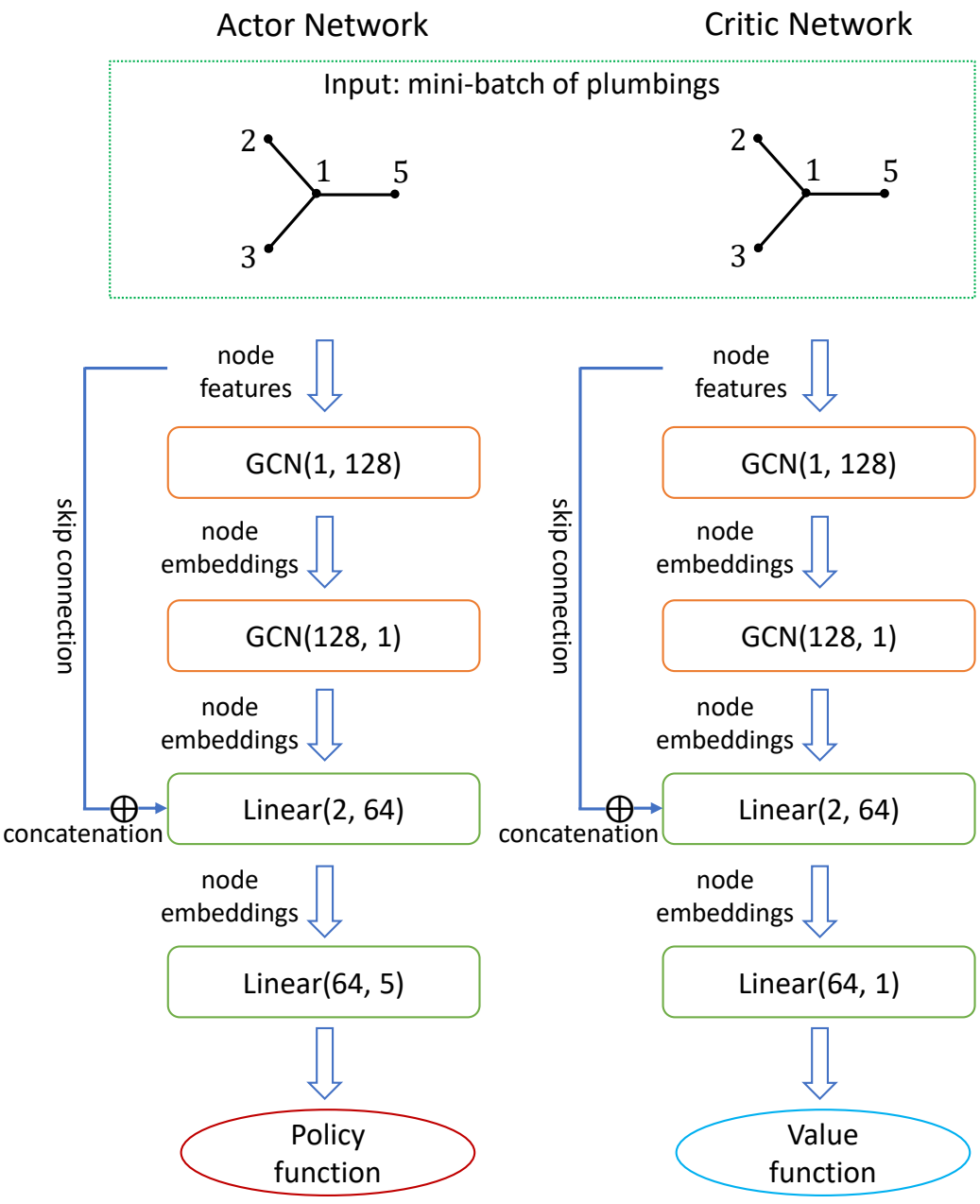


FIGURE 4.5: The architectures of Actor and Critic Networks.

another advantage that, given two equivalent plumbings, we can get a sequence of Neumann moves that change one plumbing into the other, even though such sequence of Neumann moves is not necessarily the optimal one between two plumbings. From this perspective, we are going to check the performance of the RL agents by running them on pairs of plumbings that represent the same 3-manifolds.

For the initial inputs of the agents, we generate 10,000 random pairs of plumbings by EQUIVPAIR, Algorithm 3, but with a fixed number of Neumann moves $N \in \{20, 40, 60, 80, 100\}$. At each time step, the agents choose a Neumann move and apply it to each plumbing in a pair, then we get another pair of plumbings as the next input for the agents. After taking each action, we compare two plumbings and check if they are isomorphic. If yes, we consider it as the success of finding out a sequence of Neumann moves connecting two plumbings in the initial pair. Otherwise, we move on to the next step and we repeat the process until the number of time steps exceeds $5N$. We define the accuracy of the performance as the ratio the number of successes divided by the number of total episodes. An example of a pair of equivalent graphs with the successful result by the A3C trained agent is shown in Figure 4.9.

The results of the RL agents is presented in Figure 4.6. The plot on the left in Figure 4.6 shows the accuracy comparison between A3C and DQN. The accuracy for A3C tends to slightly decrease as N gets larger, but it's around 93% for all pairs of plumbings. However, the accuracy for DQN drops significantly from around 86% to 42% when N increases from $N = 20$ to $N = 100$.

On the right in Figure 4.6, we show the average number of actions that the agent takes until obtaining a pair of exactly same two plumbings from an initial pair of equivalent plumbings. For A3C agent, the average numbers of actions do not exceed around 1.35 times N, which means the trained A3C agent has a good efficiency to make a plumbing simpler. The DQN agent needs similar number of actions to the A3C for $N = 20$ and $N = 40$. However, it takes almost twice as many number of actions as A3C for larger N.

We have also studied the distribution of Neumann moves (or actions) that the A3C agent performs before and after training to simplify plumbings generated with $N = 100$. In Figure 4.7, we plot the number of each Neumann move taken by the agent divided by N. In the plot, moves 1–5 denote blow-up moves and moves 6–8 denote blow-down moves.

It is natural to observe that all blue dots in Figure 4.7 lay on the line $y = 0.125$, because the untrained agent takes each action equally often from a uniform distribution. On the other hand, red dots for trained agent show that the agent

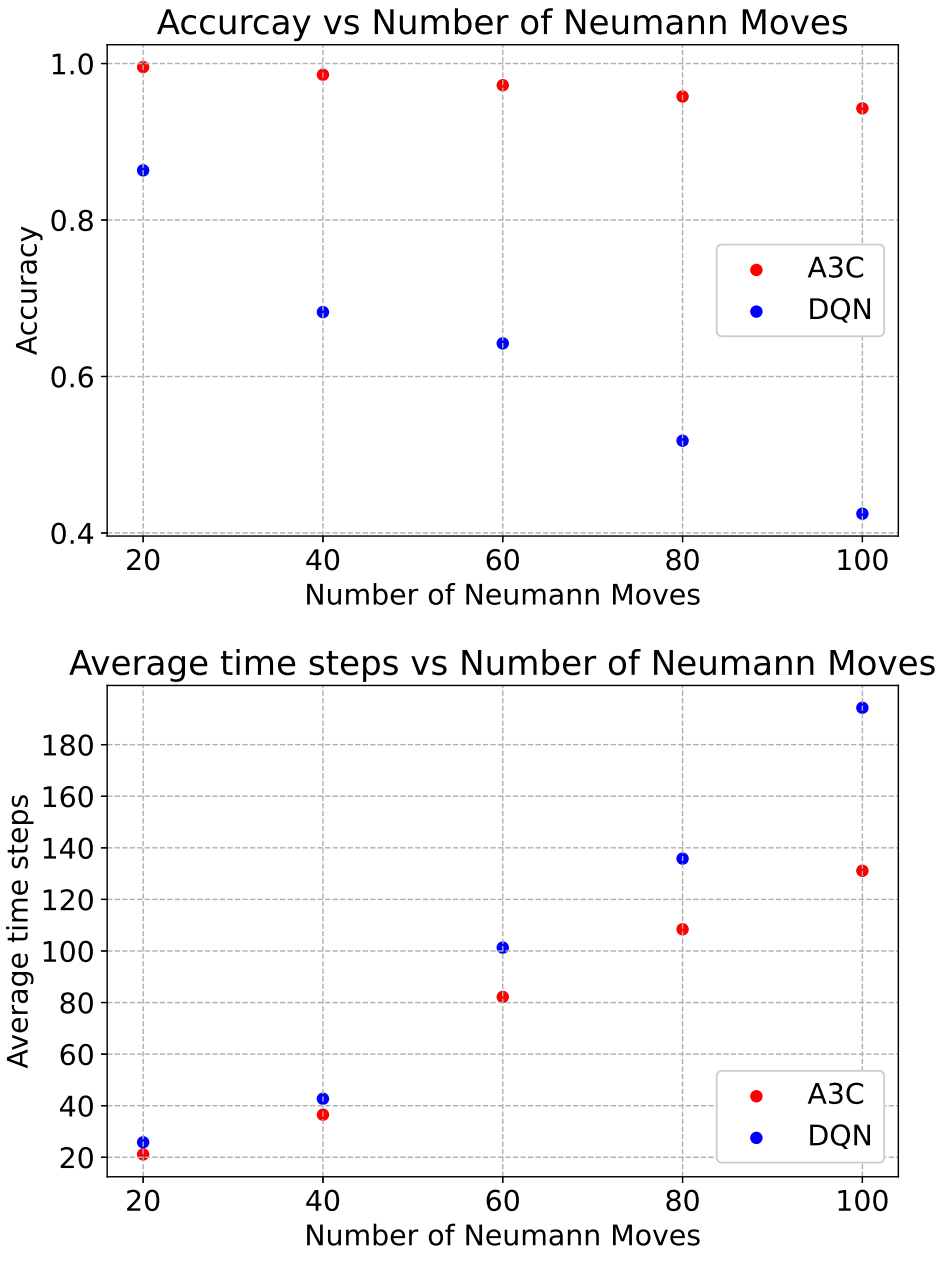


FIGURE 4.6: Performance comparison between A3C and DQN algorithms.

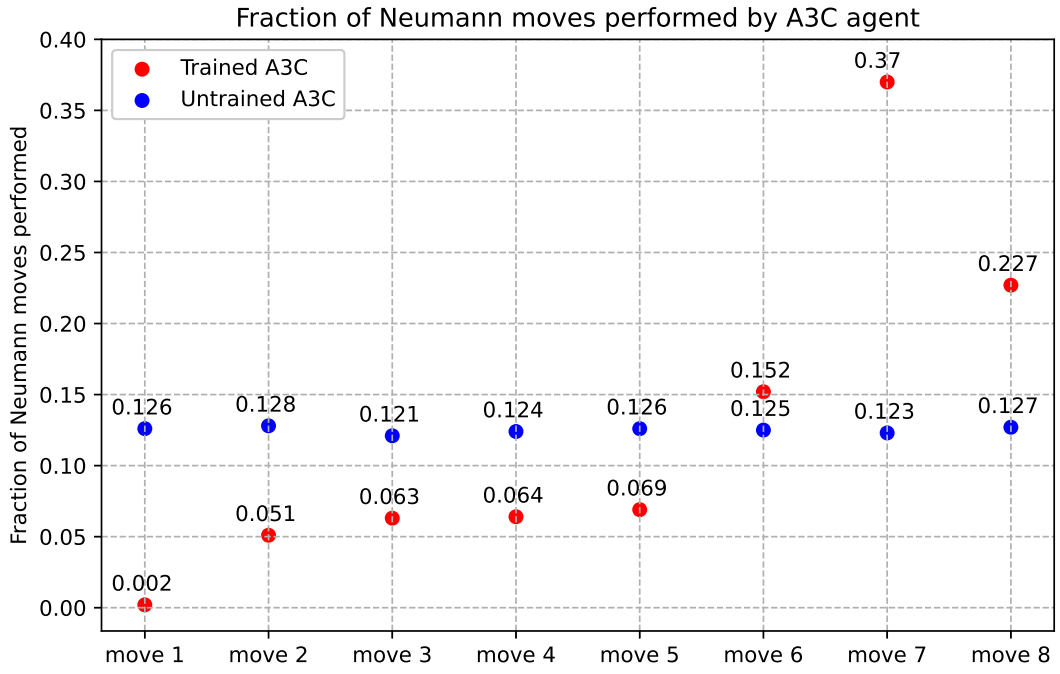


FIGURE 4.7: Comparison of the number of Neumann moves taken by a trained A3C agent and an untrained A3C agent to simplify plumbing. The values shown are the total number of Neumann moves of a given type divided by the total number of actions performed, aggregated over multiple examples.

takes blow-down moves (moves 6–8) with a probability of around 75% and takes blow-up moves (moves 1–5) with the remaining probability. This makes sense from the fact that blow-down moves can actually make the plumbing simpler and get a less punishment than blow-up moves. Especially, we see that the move 7, blow-down move of type (b), is the most frequent action and the move 1, blow-up move of type (a), is the least frequent action. This is explained by the fact that the move 1 is not helpful for the agent to get a simpler plumbing.

Before we jump into the conclusion, it is interesting to check whether or not the trained A3C agent is indeed maximizing the total return instead of immediate rewards by a simple example depicted in 4.8. The left plumbing in Figure 4.8 is a standard representation that realizes a 3-manifold known as a Brieskorn 3-sphere $\overline{\Sigma(2,3,5)}$, while the plumbing on the right represents a homeomorphic 3-manifold which can also be considered as the boundary of the E_8 manifold. As one can see immediately, the plumbing on the right in Figure 4.8 does not have nodes available for blow-down moves. Therefore, in order to get the left plumbing from the right one, the RL agent should take appropriate blow-up moves first, then taking available blow-down moves. This is why we take this

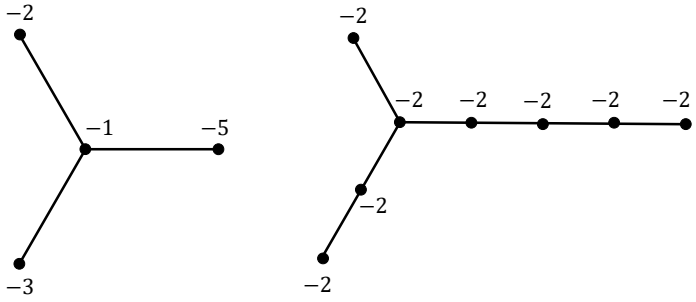


FIGURE 4.8: Two equivalent representations of a plumbed 3-manifold $\overline{\Sigma(2,3,5)}$.

example for the test. We notice that 6 actions are needed to turn one plumbing into the other in an optimal way.

The trained A3C agent successfully simplify the E_8 plumbing to the plumbing $\overline{\Sigma(2,3,5)}$ by taking 16 actions, while the trained DQN does not find a solution until the number of actions exceeds 50. This test ensures that the A3C agent indeed pursues not short-term rewards, but its maximal long-term return.

4.3 Conclusion and Future Work

4.3.1 Conclusion

In this chapter we have examined the GNN approach to the problems in the 3-dimensional topology, which ask whether two given plumbing graphs represent a same 3-manifold or not, and whether or not it is possible to find out the sequence of Neumann moves that connects two plumbings if they are equivalent.

In Section 4.1, we used supervised learning to solve the binary classification of whether or not a pair of plumbings is equivalent. We built 3 models by combining graph convolution operators GEN, GCN and GAT, together with a certain graph aggregation module and an MLP as a classifier. We found that GEN+GAT model outperformed GCN+GCN and GCN+GAT models on randomly generated training datasets with maximal number $N_{\max} = 40$ of applied Neumann moves. GEN+GAT achieved about 95% accuracy while accuracy for the others is below 80%. We also tested those 3 models on randomly generated testsets with larger $N_{\max} = 60$ and $N_{\max} = 80$. Even though those models were trained by a training sets with $N_{\max} = 40$, it is an interesting point that, on such testsets, they still performed on a similar level to their training performance.

In Section 4.2, we utilized reinforcement learning to find out the sequence of Neumann moves that relates to a given pair of equivalent plumbings. We trained the agent such that it could find the simplest representation of a plumbing by using Neumann moves as its actions. We define the simplicity as a certain linear combination of number of nodes and sum of the absolute value of node features. We ran the trained agent on each of two equivalent plumbings until it arrived at two isomorphic plumbings. In this way, we can construct a sequence of Neumann moves connecting two equivalent plumbings. Using A3C algorithm, we see that the agent can find a sequence of Neumann moves in over 90% of randomly generated equivalent plumbing pairs even with $N_{max} = 100$. This outperforms the DQN agent by a factor of around 1.5 when $N = 60$, and by more than a factor of 2 when $N = 100$.

4.3.2 Future work

In this chapter we have used Geometric Deep Learning, GNN in particular, in the problem of classification of 3-manifolds up to homeomorphisms. We restricted to a special simple class of 3-manifolds corresponding to tree plumbing graphs. We hope to apply similar neural network models for more general 3-manifolds and also 4-manifolds in the future. One direct generalization would be considering 3-manifolds corresponding to general plumbing graphs described in [34], possibly disconnected, with loops, and with non-trivial genera assigned to the vertices[5]. This, in particular, would involve considering extra features associated to the vertices and also to the edges of graphs, as well as additional set of moves relating equivalent graphs. A more interesting generalization would be considering general Kirby diagrams for 3-manifolds. A Kirby diagram of a 3-manifold is a planar diagram of a link with an integer framing number assigned to each link component. The 3-manifold corresponding to the diagram is then obtained by performing Dehn surgery on this framed link. Two diagrams produce homeomorphic 3-manifolds if and only if they can be related by a sequence of Reidemeister moves (that do not change the isotopy class of the link) together with the so-called Kirby, or equivalently, Fenn-Rourke moves that do change the link but not the resulting 3-manifold (up to homeomorphism). Such a diagram can be understood as a 4-regular plane graph with additional data specifying the types of crossings in the link diagram and the framings of the link components. Alternatively, one can

[5]In such a more general setting, a vertex with weight w and genus g corresponds to a circle fibration of Euler class w over a closed oriented surface of genus g. The case considered in this chapter is recovered when $g = 0$ for each vertex.

consider Tait graph associated to a checkboard coloring of the link diagram. For practical purposes, this presentation most likely will be more efficient. The Reidemeister, as well as Kirby/Fenn-Rourke moves then can be understood again as certain local operations on graphs associated with Kirby diagrams. The main new challenge would be incorporating the structure of the planar embedding of the graph in GNN. This can be done, for example, by specifying the cyclic order of edges at each vertex, or cyclic order of edges for each face of the plane graph. This additional structure should be taken into account in the layers of the network. This is not considered in most standard GNN architectures. A further step would be the problem of recognizing whether a pair of Kirby diagrams for 4-manifolds produces a diffeomorphic pair. Such Kirby diagrams are again framed link diagrams that also contain special "dotted" link components. There is a corresponding set of local Kirby moves that relate diagrams realizing diffeomorphic 4-manifold. For a comprehensive reference about the Kirby diagrams of 3- and 4-manifold we refer to [12].

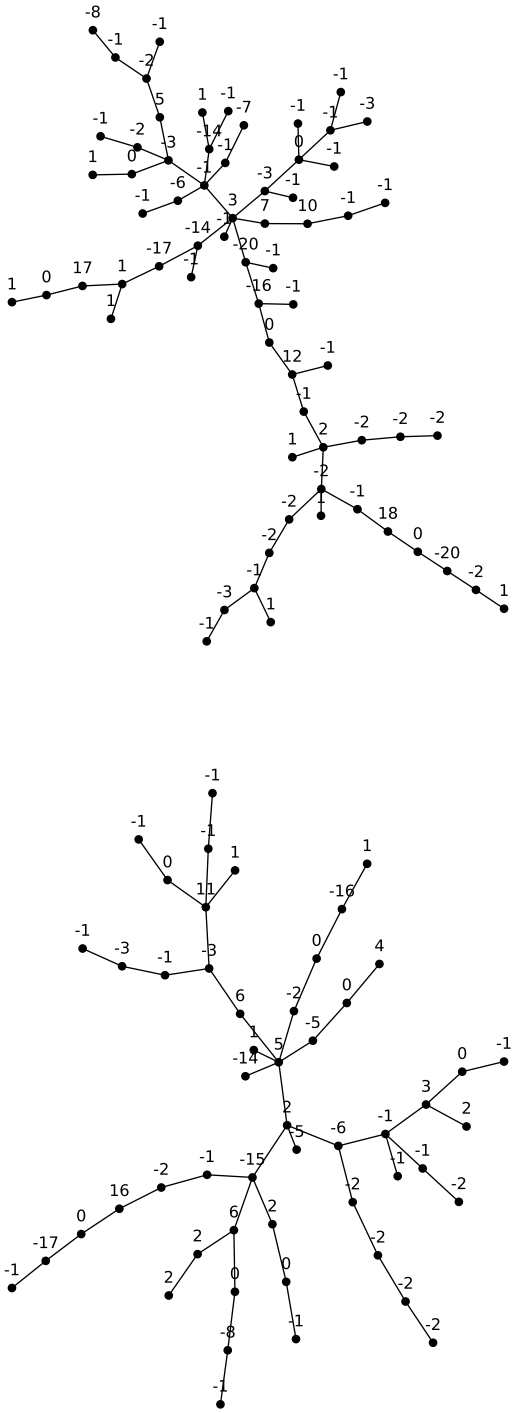


FIGURE 4.9: An example of a pair of equivalent plumbing graphs generated by EQUIVPAIR with the number of Neumann moves fixed to $N = 40$. The graphs are successfully recognized as equivalent both by the RL agent trained by A3C algorithm considered in Section 4.2 and the GEN+GAT neural network considered in Section 4.1.

Appendix A

Quadratic Diophantine equation in two variables

We are going to review here how to solve the general quadratic Diophantine equation

$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{A.1}$$

where the coefficients are all integers, i.e., $a, b, c, d, e, f \in \mathbb{Z}$. We assume that not all of a, b, c are zero. The classical method of solving (A.1) was firstly given by Lagrange over 300 years ago.

Let $D = b^2 - 4ac$, $E = bd - 2ae$ and $F = d^2 - 4af$. We will examine (A.1) in several cases, mainly depending on the values of D.

At first, suppose $D = 0$, then without loss of generality, we can assume $a \neq 0$. Equation (A.1) can be written as

$$Y^2 = 2Ey + F,$$

where $Y = 2ax + by + d$. In the case of $E = 0$ and $F = 0$, then there could be either a line of integer solutions for (A.1) if $\gcd(2a, b)$ divides d, or no solution otherwise. It is clear that there is no integer solution of (A.1) if $E = 0$ and $F < 0$ or if $E = 0$ and F is positive but not a perfect integral square. If $E = 0$ and F is a perfect square, then one can obtain either a line of integer solutions if $\gcd(2a, b)$ divides $\sqrt{F} \pm d$, or no solution otherwise. In the case of $E \neq 0$, (A.1) boils down to the congruence

$$Y^2 \equiv F \mod |2E|. \tag{A.2}$$

Clearly, there are no integer solutions for (A.1) if (A.2) does not have integer solutions. In the case that (A.2) has integer solutions, it is not difficult to observe that the integer solutions (x, y) of (A.1) lie on a parabola if they exist.

Now suppose that $D \neq 0$. Then by putting $N = E^2 - DF$, (A.1) turns out to be

$$X^2 - DY^2 = N, \tag{A.3}$$

where $X = Dy + E$. It is clear that (A.3) has a finite number of solutions if $D < 0$, or if D is a positive perfect integral square, or if $N = 0$. Therefore, the only remaining non-trivial case happens when $N \neq 0$ and $D > 0$ is not a perfect square. In this case, (A.3) is called a *generalized Pell's equation* and it has infinite number of solutions if it has at least one.

A.1 Pell's equation

A *Pell's equation* is a Diophantine equation of the form

$$T^2 - DU^2 = 1, \tag{A.4}$$

where D is a positive integer that is not a perfect square. The trivial solutions are $(T, U) = (\pm 1, 0)$. Among all non-trivial solutions, the pair (t, u) is called the *fundamental solution* if t and u are both positive and they are minimal. It is well-known from the work of Lagrange in 1768 that any Pell's equation has a non-trivial solution. Furthermore, once we find a solution, there is a way to generate all the solutions of (A.4). A pair (T, U) of positive integers is a solution of (A.4) if and only if there exists $n \in \mathbb{N}$ such that

$$T + U\sqrt{D} = \left(t + u\sqrt{D}\right)^n. \tag{A.5}$$

Sometimes it might be much more convenient to express (A.5) in a matrix form as follows:

$$\begin{aligned}
\begin{pmatrix} T_n \\ U_n \end{pmatrix} &= \begin{pmatrix} t & Du \\ u & t \end{pmatrix}^n \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{D} & -\sqrt{D} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \left(t + u\sqrt{D}\right)^n & 0 \\ 0 & \left(t - u\sqrt{D}\right)^n \end{pmatrix} \begin{pmatrix} \sqrt{D} & -\sqrt{D} \\ 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}\left(\left(t + u\sqrt{D}\right)^n + \left(t - u\sqrt{D}\right)^n\right) \\ \frac{1}{2\sqrt{D}}\left(\left(t + u\sqrt{D}\right)^n - \left(t - u\sqrt{D}\right)^n\right) \end{pmatrix},
\end{aligned}$$

where in the second line we have diagonalized the matrix

$$\begin{pmatrix} t & Du \\ u & t \end{pmatrix} = \begin{pmatrix} \sqrt{D} & -\sqrt{D} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} t+u\sqrt{D} & 0 \\ 0 & t-u\sqrt{D} \end{pmatrix} \begin{pmatrix} \sqrt{D} & -\sqrt{D} \\ 1 & 1 \end{pmatrix}^{-1}$$

by using its eigenvectors and eigenvalues. Thus, any positive solution of (A.4) has the form of

$$\begin{aligned} T_n &= \frac{1}{2}\left(\left(t+u\sqrt{D}\right)^n + \left(t-u\sqrt{D}\right)^n\right), \\ U_n &= \frac{1}{2\sqrt{D}}\left(\left(t+u\sqrt{D}\right)^n - \left(t-u\sqrt{D}\right)^n\right), \end{aligned} \tag{A.6}$$

and clearly $(\pm T_n, \pm U_n)$ will also be solutions.

A.2 Generalized Pell's equation

So far we have seen that Pell's equation (A.4) has an infinite number of solutions. Using those solutions one can describe a method to obtain all solutions of a generalized Pell's equation (A.3). Indeed, Lagrange showed that every solution of (A.3) can be expressed by a power of $t+u\sqrt{D}$ times $X+Y\sqrt{D}$, that is,

$$X_n + Y_n\sqrt{D} = \left(X+Y\sqrt{D}\right)\left(t+u\sqrt{D}\right)^n \qquad \text{for some } n \in \mathbb{Z}, \tag{A.7}$$

where (t,u) are the fundamental solution of the corresponding Pell's equation $T^2 - DU^2 = 1$ and (X,Y) is a solution of (A.3) such that

$$|X| \le \sqrt{|N|}\sqrt{t+u\sqrt{D}}/2 \qquad \text{and} \qquad |Y| \le \sqrt{|N|}\sqrt{t+u\sqrt{D}}/\left(2\sqrt{D}\right).$$

In other words, there exists a finite set $\mathcal{S}$ of solutions for (A.3) such that every solution (X_n, Y_n) can be obtained by (A.7) with some $(X,Y) \in \mathcal{S}$. It follows from (A.6) and (A.7) that

$$\begin{aligned} X_n &= XT_n + DYU_n = \frac{1}{2}\left[\left(X+Y\sqrt{D}\right)A^n + \left(X-Y\sqrt{D}\right)A^{-n}\right], \\ Y_n &= XT_n + DYU_n = \frac{1}{2\sqrt{D}}\left[\left(X+Y\sqrt{D}\right)A^n - \left(X-Y\sqrt{D}\right)A^{-n}\right], \end{aligned} \tag{A.8}$$

where we denoted $A = t+u\sqrt{D} > 1$.

A.3 Quadratic Diophantine equation in two variables: revisited

Now we circle back to the solutions of our Diophantine equation (A.1). We have already seen that in the case when $N \neq 0$ and $D > 0$ is not a perfect square, the solutions of (A.1) is related to those of generalized Pell's equation (A.3). Once we obtain all solutions of (A.3), we need to identify for each $(X, Y) \in \mathcal{S}$, those values of $n \in \mathbb{Z}$ such that

$$\begin{cases} X_n \equiv E \mod D, \\ Y_n \equiv b(X_n - E)/D + d \mod 2a, \end{cases}$$

or equivalently,

$$\begin{cases} X_n \equiv E \mod D, \\ DY_n \equiv bX_n - bE + Dd \mod 2aD. \end{cases}$$

The efficient way [39] is to check a necessary condition

$$2a \mid X - bY. \tag{A.9}$$

If (A.9) and

$$D \left| \frac{dD - bE - DY + bX}{2a} \right.$$

hold, then (A.8) produces solutions for (A.1) for all even n. If (A.9) and

$$D \left| \frac{dD - bE - DYt + bXt}{2a} \right.$$

hold, then (A.8) produces solutions for (A.1) for all odd n. If none of these holds, then there is no solution of (A.1).

Appendix B

Algorithms

In this appendix, we provide details of the algorithms which have been used to generate datasets for training and testing both SL and RL models in Section 4.1 and Section 4.2.

- RANDOMPLUMBING
  This algorithm generates a random plumbing tree by creating a random array for node features and building an adjacency matrix. It starts to choose a random integer as a number of nodes between 1 and 25. In general, there are N^{N-2} different plumbing trees with N nodes if we don't consider node features. Therefore, the upper limit 25 is large enough to generate around 10^6 random plumbing tress with statistically insignificant overlapping plumbings. The array of node feature is also created by randomly choosing an integer in the interval $(-20, 20)$ for each node. Then we define the adjacency matrix for the plumbing tree, and the algorithm returns a pair of node feature array and adjacency matrix as data for the output plumbing. Note that all random process is done by using a uniform distribution.

- RANDOMNEUMANNMOVE
  The role of this algorithm is to apply a randomly chosen Neumann move to a random node of the input plumbing, then returns the resulting plumbing. A random Neumann move is characterized by 3 variables, i.e., $type$, $updown$, and $sign$. Here $type \in \{1, 2, 3\}$ denotes 3 types of Neumann moves depicted in Figure 2.2, $updown \in \{1, -1\}$ points out blow-up ($updown = 1$) or blow-down ($updown = -1$), and $sign \in \{1, -1\}$ denotes the sign of the new vertex for blow-up Neumann moves of type (b) and (c). Notice that other moves does not require $sign$.

  The algorithm first takes a random node of the input and fixes a random tuple $(type, updown, sign)$ from a uniform distribution. Then it builds new node feature array and adjacency matrix for the plumbing obtained

by applying the Neumann move to the chosen node. If the Neumann move determined by a tuple $(type, updown, sign)$ is an illegal move, the output plumbing is the same as the input. The algorithm also returns another variable $done \in \{\text{TRUE}, \text{FALSE}\}$, which makes it possible to notice whether the Neumann move to be applied is legal ($done = \text{TRUE}$) or illegal ($done = \text{FALSE}$). This variable $done$ will be used to decide the rewards of actions in Section 4.2.

- EQUIVPAIR and INEQUIVPAIR
  These are used to generate an equivalent plumbing pair (EQUIVPAIR) or an inequivalent plumbing pair (INEQUIVPAIR). At the first step, EQUIVPAIR generates an initial pair of isomorphic plumbings, while INEQUIVPAIR generates two inequivalent plumbings, by using RANDOMPLUMBING. Then they have the same process, in which they apply Neumann moves iteratively up to $N_{\max}$ times to each plumbing in the initial pair. Then they return the resulting pair as well as a variable, named $label$, which will be used for classification problem in Section 4.1. Notice that $label = 1$ for EQUIVPAIR and $label = -1$ for INEQUIVPAIR.

- TWEAKPAIR
  This algorithm generates an inequivalent pair of plumbings, but with the same graph structure. One plumbing is generated by RANDOMPLUMBING, and the other is obtained by tweaking a copy of the first plumbing, i.e., by making a small change to a feature of a randomly chosen node. These two plumbings form an initial pair. Since the adjacency matrices of two plumbings are same, they have the same graph structure. However, due to the small change, two plumbings are inequivalent. Then the algorithm has the same structure as in EQUIVPAIR and INEQUIVPAIR to apply random Neumann moves iteratively to each plumbing in the initial pair.

Algorithm 1 RANDOMPLUMBING

$n \leftarrow$ random integer between 1 and 25 ▷ number of nodes
$\mathbf{x} \leftarrow$ array of n random integers between -20 and 20 ▷ node features
$\mathbf{a} \leftarrow n \times n$ matrix of zeros ▷ initialize the adjacency matrix
for $i = 2$ to n **do** ▷ construct the adjacency matrix
  $j \leftarrow$ random integer between 1 and $i - 1$
  $\mathbf{a}_{i,j}, \mathbf{a}_{j,i} \leftarrow 1$
end for
$G \leftarrow (\mathbf{x}, \mathbf{a})$ ▷ G defines the plumbing
return G

Algorithm 2 RANDOMNEUMANNMOVE

Require: a plumbing G
$v \leftarrow$ a random node of G
$type \leftarrow$ a random choice in $\{1, 2, 3\}$
$updown \leftarrow$ a random choice in $\{1, -1\}$
if $updown = 1$ **then** ▷ blow-up move
  if $type = 1$ **then**
    $G' \leftarrow$ a plumbing applied a blow-up move of type (a) to the node v
  else
    $sign \leftarrow$ a random choice in $\{1, -1\}$
    $G' \leftarrow$ a plumbing applied a blow-up move determined by $(type, sign)$
  end if
  $done \leftarrow$ TRUE
else ▷ blow-down move
  if v can be removed by a blow-down move **then**
    $G' \leftarrow$ a plumbing applied a blow-down move to the node v
    $done \leftarrow$ TRUE
  else
    $G' \leftarrow G$ ▷ returns the input plumbing for a forbidden move
    $done \leftarrow$ FALSE
  end if
end if
return $(done, G')$

Algorithm 3 EQUIVPAIR

Require: $N_{\max} \in \mathbb{Z}^+$
$G \leftarrow$ a plumbing by RANDOMPLUMBING
$G_1 \leftarrow G$
$n_1 \leftarrow$ a random integer between 1 and $N_{\max}$
for $i = 1$ to n_1 **do** ▷ Apply Neumann moves n_1 times
  $G_1 \leftarrow$ RANDOMNEUMANNMOVE(G_1)
end for
$G_2 \leftarrow G$
$n_2 \leftarrow$ a random integer between 1 and $N_{\max}$
for $j = 1$ to n_2 **do** ▷ Apply Neumann moves n_2 times
  $G_2 \leftarrow$ RANDOMNEUMANNMOVE(G_2)
end for
$label \leftarrow 1$
return $G_1, G_2, label$

Algorithm 4 INEQUIVPAIR

Require: $N_{\max} \in \mathbb{Z}^+$
$G_1 \leftarrow$ a plumbing by RANDOMPLUMBING
$G_2 \leftarrow$ another plumbing by RANDOMPLUMBING
$n_1 \leftarrow$ a random integer between 1 and $N_{\max}$
for $i = 1$ to n_1 **do** ▷ Apply Neumann moves n_1 times
  $G_1 \leftarrow$ RANDOMNEUMANNMOVE(G_1)
end for
$n_2 \leftarrow$ a random integer between 1 and $N_{\max}$
for $j = 1$ to n_2 **do** ▷ Apply Neumann moves n_2 times
  $G_2 \leftarrow$ RANDOMNEUMANNMOVE(G_2)
end for
$label \leftarrow -1$
return $G_1, G_2, label$

Algorithm 5 TWEAKPAIR

Require: $N_{\max} \in \mathbb{Z}^+$
$G_1 \leftarrow$ a plumbing by RANDOMPLUMBING
$G_2 \leftarrow G_1$
$v \leftarrow$ a random node in G_2
$t \leftarrow$ a random integer between -3 and 3, not 0.
$\mathbf{x} \leftarrow$ node feature of G_2
$\mathbf{a} \leftarrow$ adjacency matrix of G_2
$\mathbf{x}_v \leftarrow \mathbf{x}_v + t$
$G_2 \leftarrow$ a plumbing with $(\mathbf{x}, \mathbf{a})$
$n_1 \leftarrow$ a random integer between 1 and $N_{\max}$
for $i = 1$ to n_1 **do** ▷ Apply Neumann moves n_1 times
  $G_1 \leftarrow$ RANDOMNEUMANNMOVE(G_1)
end for
$n_2 \leftarrow$ a random integer between 1 and $N_{\max}$
for $i = 1$ to n_2 **do** ▷ Apply Neumann moves n_2 times
  $G_2 \leftarrow$ RANDOMNEUMANNMOVE(G_2)
end for
$label \leftarrow -1$
return $G_1, G_2, label$

Bibliography

[1] Rostislav Akhmechet, Peter K. Johnson, and Vyacheslav Krushkal. "Lattice cohomology and q-series invariants of 3-manifolds". In: *J. Reine Angew. Math.* 796 (2023), pp. 269–299. DOI: 10.1515/crelle-2022-0096. eprint: 2109.14139. URL: https://doi.org/10.1515/crelle-2022-0096.

[2] Bruce C. Berndt and James Lee Hafner. "Two remarkable doubly exponential series transformations of Ramanujan". In: *Proc. Indian Acad. Sci. Math. Sci.* 104.1 (1994), pp. 245–252. DOI: 10.1007/BF02830888. URL: https://doi.org/10.1007/BF02830888.

[3] Michael M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. 2021. arXiv: 2104.13478 [cs.LG].

[4] Wenming Cao et al. "A comprehensive survey on geometric deep learning". In: *IEEE Access* 8 (2020), pp. 35929–35949.

[5] Miranda C. N. Cheng et al. "3d modularity". In: *J. High Energy Phys.* 2019.10 (2019), no. 10, 010, 93 pages. DOI: 10.1007/jhep10(2019)010. eprint: 1809.10148. URL: https://doi.org/10.1007/jhep10(2019)010.

[6] Sungbong Chun et al. "3d-3d correspondence for mapping tori". In: *Journal of High Energy Physics* 2020.9 (Sept. 2020). DOI: 10.1007/jhep09(2020)152. URL: https://doi.org/10.1007%2Fjhep09%282020%29152.

[7] Francesco Costantino, Sergei Gukov, and Pavel Putrov. "Non-semisimple TQFT's and BPS q-series". In: *SIGMA* 19 (2023), 010, 71 pages. DOI: 10.3842/SIGMA.2023.010. eprint: 2107.14238. URL: https://doi.org/10.3842/SIGMA.2023.010.

[8] Jessica Craven et al. "Learning knot invariants across dimensions". In: *SciPost Phys.* 14.2 (2023), p. 021. DOI: 10.21468/SciPostPhys.14.2.021. arXiv: 2112.00016 [hep-th].

[9] Alex Davies et al. "Advancing mathematics by guiding human intuition with AI". In: *Nature* 600.7887 (2021), pp. 70–74.

[10] Tobias Ekholm et al. "$\widehat{Z}$ at large N: from curve counts to quantum modularity". In: *Comm. Math. Phys.* 396.1 (2022), pp. 143–186. DOI: 10.1007/s00220-022-04469-9. eprint: 2005.13349. URL: https://doi.org/10.1007/s00220-022-04469-9.

[11] Matthias Fey and Jan E. Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019. URL: https://arxiv.org/abs/1903.02428.

[12] Robert E Gompf and András Stipsicz. *4-manifolds and Kirby calculus*. Vol. 20. Graduate Studies in Mathematics. American Mathematical Soc., 1999.

[13] Sergei Gukov and Ciprian Manolescu. "A two-variable series for knot complements". In: *Quantum Topol.* 12.1 (2021), pp. 1–109. DOI: 10.4171/qt/145. eprint: 1904.06057. URL: https://doi.org/10.4171/qt/145.

[14] Sergei Gukov, Pavel Putrov, and Cumrun Vafa. "Fivebranes and 3-manifold homology". In: *J. High Energy Phys.* 2017.7 (2017), no. 7, 071, 81 pages. DOI: 10.1007/JHEP07(2017)071. eprint: 1602.05302. URL: https://doi.org/10.1007/JHEP07(2017)071.

[15] Sergei Gukov et al. "BPS spectra and 3-manifold invariants". In: *J. Knot Theory Ramifications* 29.2 (2020), 2040003, 85 pages. DOI: 10.1142/S0218216520400039. eprint: 1701.06567. URL: https://doi.org/10.1142/S0218216520400039.

[16] Sergei Gukov et al. "Learning to Unknot". In: *Mach. Learn. Sci. Tech.* 2.2 (2021), p. 025035. DOI: 10.1088/2632-2153/abe91f. arXiv: 2010.16263 [math.GT].

[17] Sergei Gukov et al. *Searching for ribbons with machine learning*. 2023. arXiv: 2304.09304 [math.GT].

[18] Yang-Hui He, Elli Heyes, and Edward Hirst. "Machine Learning in Physics and Geometry". In: (Mar. 2023). arXiv: 2303.12626 [hep-th].

[19] Mark C Hughes. "A neural network approach to predicting and computing knot invariants". In: *Journal of Knot Theory and Its Ramifications* 29.03 (2020), p. 2050005.

[20] Vishnu Jejjala, Arjun Kar, and Onkar Parrikar. "Deep Learning the Hyperbolic Volume of a Knot". In: *Phys. Lett. B* 799 (2019), p. 135033. DOI: 10.1016/j.physletb.2019.135033. arXiv: 1902.05547 [hep-th].

[21] LH Kauffman, NE Russkikh, and IA Taimanov. "Rectangular knot diagrams classification with deep learning". In: *Journal of Knot Theory and Its Ramifications* 31.11 (2022), p. 2250067.

[22] Abdullah Khan, Alexei Vernitski, and Alexei Lisitsa. "Untangling braids with multi-agent q-learning". In: *2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. 2021, pp. 135–139.

[23] Mikhail Khovanov. "A categorification of the Jones polynomial". In: *Duke Math. J.* 101.3 (2000), pp. 359–426. DOI: 10.1215/S0012-7094-00-10131-7. eprint: math.QA/9908171. URL: https://doi.org/10.1215/S0012-7094-00-10131-7.

[24] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[25] Vijay Konda and John Tsitsiklis. "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[26] Greg Kuperberg. "Algorithmic homeomorphism of 3-manifolds as a corollary of geometrization". In: *Pacific Journal of Mathematics* 301.1 (2019), pp. 189–241.

[27] Yujia Li et al. *Gated Graph Sequence Neural Networks*. 2017. arXiv: 1511.05493 [cs.LG].

[28] Yujia Li et al. *Graph Matching Networks for Learning the Similarity of Graph Structured Objects*. 2019. arXiv: 1904.12787 [cs.LG].

[29] W. B. R. Lickorish. "A representation of orientable combinatorial 3-manifolds". In: *Ann. of Math.* 76 (1962), pp. 531–540. DOI: 10.2307/1970373. URL: https://doi.org/10.2307/1970373.

[30] Alexei Lisitsa, Mateo Salles, and Alexei Vernitski. "Supervised Learning for Untangling Braids". In: *15th International Conference on Agents and Artificial Intelligence (ICAART 2023)*. 2023.

[31] Volodymyr Mnih et al. *Asynchronous Methods for Deep Reinforcement Learning*. 2016. arXiv: 1602.01783 [cs.LG].

[32] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG].

[33] Yuya Murakami. *A proof of a conjecture of Gukov-Pei-Putrov-Vafa*. 2023. arXiv: 2302.13526 [math.GT].

[34] Walter D. Neumann. "A calculus for plumbing applied to the topology of complex surface singularities and degenerating complex curves". In: *Trans. Amer. Math. Soc.* 268.2 (1981), pp. 299–344. DOI: 10.2307/1999331. URL: https://doi.org/10.2307/1999331.

[35] Peter Ozsváth and Zoltán Szabó. "On the Floer homology of plumbed three-manifolds". In: *Geom. Topol.* 7 (2003), pp. 185–224. DOI: 10.2140/gt.2003.7.185. eprint: math.SG/0203265. URL: https://doi.org/10.2140/gt.2003.7.185.

[36] Sunghyuk Park. "Higher rank $\hat{Z}$ and F_K". In: *SIGMA* 16 (2020), 044, 17 pages. DOI: 10.3842/SIGMA.2020.044. eprint: 1909.13002. URL: https://doi.org/10.3842/SIGMA.2020.044.

[37] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[38] N. Reshetikhin and V. G. Turaev. "Invariants of 3-manifolds via link polynomials and quantum groups". In: *Invent. Math.* 103.3 (1991), pp. 547–597. DOI: 10.1007/BF01239527. URL: https://doi.org/10.1007/BF01239527.

[39] R. E. Sawilla, A. K. Silvester, and H. C. Williams. "A new look at an old equation". In: *Algorithmic Number Theory*. Vol. 5011. Lecture Notes in Comput. Sci. Springer, Berlin, 2008, pp. 37–59. DOI: 10.1007/978-3-540-79456-1_2. URL: https://doi.org/10.1007/978-3-540-79456-1_2.

[40] Franco Scarselli et al. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[41] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. "Graph neural networks in particle physics". In: *Machine Learning: Science and Technology* 2.2 (2020), p. 021001.

[42] Petar Veličković et al. "Graph Attention Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[43] Alexei Vernitski, Alexei Lisitsa, et al. "Reinforcement learning algorithms for the Untangling of Braids". In: *The International FLAIRS Conference Proceedings*. Vol. 35. 2022.

[44] Andrew H. Wallace. "Modifications and cobounding manifolds". In: *Canadian J. Math.* 12 (1960), pp. 503–528. DOI: 10.4153/CJM-1960-045-7. URL: https://doi.org/10.4153/CJM-1960-045-7.

[45] Edward Witten. "Quantum field theory and the Jones polynomial". In: *Comm. Math. Phys.* 121.3 (1989), pp. 351–399. DOI: 10.1007/BF01217730. URL: https://doi.org/10.1007/BF01217730.

[46] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[47] Keyulu Xu et al. *Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth*. 2021. arXiv: 2105.04550 [cs.LG].

[48] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012.