基于约束 Lattice 的 Invariant Group 的优化算法

黄天域，王梓桐，李可，陈佳梁

January 2025

1 引言

Lattice quantizer problem 是几何学中的一个经典问题，其目标在于找到 n 维空间中的一个 lattice 并最小化一个随机向量到 lattice 中一个点的平均平方距离，也被叫做 Normalized second moments (NSM)。

受到下一节中介绍的文献的启发，我们对 quantizer problem 进行了一些探索：我们不仅复现了论文的结果，还研究并设计了一个新的求 G-invariant lattice 的算法，能够在保证 lattice 满足一些几何性质的前提下优化其 NSM，并得到了一些实验成果。本篇报告会详细介绍我们在这个 project 上做出的尝试，并总结我们得到的成果与目前算法的局限性。我们的代码已在 Github 上开源：`https://github.com/AK-DREAM/ML-quantizer`。

2 文献综述与研究动机

在 Conway 和 Sloane 的 *Sphere Packings, Lattices and Groups* 一书 [1] 中，作者详细介绍了 lattice 的定义和性质，以及 quantizing problem。在第三章的第 4 节中，作者介绍了 lattice 的自同构群（automorphism group），提出了一种使用整数矩阵的表示方式，并指出对这一整数矩阵群 G 的研究很大程度上是对在该群下 invariant 的 lattice 的研究，启发我们对 lattice 的 invariant group 进行进一步的探索。

Agrell，Pook-Kolb 和 Allen 在 [3] 中提出了一种基于随机梯度下降（stochastic gradient descent）的高效算法用于最小化特定维数 lattice 的 NSM，通过在 lattice 的 Voronoi region 中随机选取样本，并计算生成矩阵 B 的梯度来优化 lattice，启发了我们使用机器学习算法来优化

G-invariant lattice 的 NSM。同时，这篇文章也提出了一种使用 theta image 来消除数值误差，将由数值生成的 lattice 转化为其本质的精确形式的方法。

Agrell 与 Allen 在 [2] 中提出了基于低维情形较好 lattice 通过正交拼接 (gluing) 以求得更高维空间中较好 lattice 的技术，结合前面的思考，这启发了我们尝试去探索更为灵活的 gluing 算法，结合 invariant group 的想法，我们认为好的 lattice 的特殊几何结构，或者说对称性应当反映于其 invariant group 中。最终我们提出了下面基于约束 invariant group 的求解优化 lattice 的算法。

3 算法简介

3.1 原始算法复现

我们首先尝试复现了 Agrell，Pook-Kolb 和 Allen 的论文 [3] 中求 n 维无约束最优 lattice 的算法。

在 [3] 中算法的基础上，我们为 CLP 增加了并行计算支持，并使用 mini-batch 的梯度下降方法，以更好地利用计算资源得到问题的解，具体改进可参考 3.5 节。

下表展示了我们的算法以 $batchsize = 128$ 运行 $T = 100\,000$ 轮后的的结果。其中 $\hat{G} \pm 2\hat{\sigma}$ 是我们算法给出的结果，$\hat{G}^* \pm 2\hat{\sigma}^*$ 为 [3] 给出的结果。

表 1: Result of 3.1

n	$\hat{G} \pm 2\hat{\sigma}$	$\hat{G}^* \pm 2\hat{\sigma}^*$	n	$\hat{G} \pm 2\hat{\sigma}$	$\hat{G}^* \pm 2\hat{\sigma}^*$
10	0.070812 ± 0.000003	0.070811 ± 0.000003	17	0.068215 ± 0.000002	-
11	0.070428 ± 0.000003	0.070424 ± 0.000002	18	0.068153 ± 0.000002	-
12	0.070064 ± 0.000003	0.070029 ± 0.000002	19	0.067843 ± 0.000002	-
13	0.069703 ± 0.000003	0.069696 ± 0.000002	20	0.067581 ± 0.000002	-
14	0.069261 ± 0.000002	0.069261 ± 0.000002	21	0.067339 ± 0.000002	-
15	0.068872 ± 0.000002	0.068869 ± 0.000002	22	0.067093 ± 0.000001	-
16	0.068298 ± 0.000002	0.068296 ± 0.000002	23	0.066872 ± 0.000001	-

3.2 理论推导

3.2.1 数学准备

在本小节，我们假定读者有基本的抽象代数基础，为了报告的简洁性，我们省略了如群，群上的二元运算，群同态、群同构等的定义。

为了解释我们的算法，我需要先引入一定的数学概念作为准备，并准备利用这些概念进行一些推导。下面我将介绍本篇文章所需要的数学知识：

首先给出在本篇文章中十分重要的两个群的记号约定：

$$n\text{ 阶一般线性群}\mathrm{GL}_n(\mathbb{K}) = \{A|A\text{是 } n\times n \text{ 阶系数在 } K \text{ 中的可逆矩阵}\}$$

本篇报告中 K 主要取为 $\mathbb{R}, \mathbf{Z}$.

$$n\text{ 阶正交群}\mathrm{O}(n) = \{A|A\text{是 } n\times n \text{ 阶正交矩阵}\}$$

上面两个群的群上的二元运算都是矩阵乘法，容易发现 $\mathrm{O}(n)$ 是 $\mathrm{GL}_n(\mathbb{R})$ 的子群，即 $\mathrm{O}(n) \in \mathrm{GL}_n(\mathbb{R})$ 且两者的群运算是相容的。

接下来我将引入我们使用的核心数学技术：

定义 3.1. 群的实表示 是指一个群 G 和 $\mathbb{R}$ 上的一个有限维向量空间 V 之间的一个同态映射

$$\rho : G \to \mathrm{GL}(V),$$

其中 $\mathrm{GL}(V)$ 表示 V 上的可逆线性变换构成的群（即一般线性群）。换言之，群 G 的每个元素 g 都按照下面的方式“作用”在线性空间 V 上

$$g \cdot v := \phi(g) \cdot v$$

，特别的，若 $\phi(G) \subset \mathrm{O}(V)$，即包含在 V 上的正交群中，那么称这个（实）表示是**正交表示**

对于一个 n 维 lattice $\Lambda = \mathbb{Z}\mathbf{x_1} + \mathbb{Z}\mathbf{x_2} + \ldots \mathbb{Z}\mathbf{x_n}$，其中我们约定 $\mathbf{x_i}$ 均为行向量，称矩阵

$$B = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \cdots \\ \mathbf{x_n} \end{bmatrix}$$

为 lattice Λ 的生成矩阵，它的行向量的整系数组合给出 lattice 中的所有点，而矩阵 $A = BB^T$ 称为 lattice Λ 的 **Gram matrix**，下面我将根据 A, B 定义我们算法中最为核心的两个群。

定义 3.2. ***Invariant Group***，上述 *lattice*Λ 的 *invariant group* G 定义为

$$G = \{g \in \mathrm{GL}_n(\mathbf{Z}) | gAg^T = A\}$$

，若一个 *lattice* ω 的 *Invariant Group* 包含 G，则称 ω ***G-invariant***.

定义 3.3. ***Isometric Group***，上述上述 *lattice*Λ 的 *isometric group* S 定义为

$$S = \{s \in \mathrm{O}(n) | \Lambda * s = \Lambda\}$$

即在 S 中的正交变换的作用下，*lattice* Λ 保持不变。

3.3 方法推导

在本小节, 我们将推导证明支撑我们算法的主要数学结果, 包括如何将约束 Invaariat Group 包含指定群转化为无约束问题，Invariant Group 与 Isometric Group 之间的联系以及如何计算 Invariant Group 等等。

对于 lattice Λ, B, A 分别为其生成矩阵，Gram matrix. 根据 [3]，在生成的 Lattice 是等价的前提下（生成的 lattice 的 Voronoi region 全等），可以不妨假设 B 是上三角矩阵并且对角元大于 0. 那么根据著名的 cholesky 分解定理：

定理 3.4. ***cholesky* 分解定理：**设矩阵 A 是正定对称矩阵，那么 A 存在且恰存在唯一的分解

$$A = LL^T$$

这里 L 为下三角矩阵，主对角元大于 *0*。

根据这个定理，结合我们对 B 的假设，lattice Λ 的 Gram matrix 唯一决定了生成矩阵 B，在此基础上，我们来初步理解 Invariant Group 所代表的几何含义。

对给定的 n 维 lattice Λ 规定 $< \cdot, \cdot >$ 为 $\mathbf{R}^n$ 上的标准内积：$< \mathbf{u}, \mathbf{v} > := \mathbf{u} * \mathbf{v}^T$

那么对于 $\forall group G \subset \mathrm{GL}_n(\mathbb{R})$ lattice Λ G-invariant 当且仅当

$$< \xi B, \xi B > = < \xi g B, \xi g B >, \forall g \in G, \xi \in \mathbb{Z}^n \tag{1}$$

这相当于说，对 lattice 中的任何一对点 $(\xi B, \eta B)$ 其中 $\xi, \eta \in \mathbb{Z}^n$ 为两个点的整系数坐标，在经过 $g \in G$ 变换后，得到的新的一对点 $((\xi g)B, (\eta g)B)$ 之间的距离不变。根据(1)，这等价于

$$gAg^T = A, \forall g \in G \tag{2}$$

对 Gram matrix A，若以 $f(\mathbf{x}) = \mathbf{x}A\mathbf{x}^t, \mathbf{x} \in \mathbb{R}^n$ 记 A 生成的二次型，那么(2)等价于 $f(\mathbf{x}) = f(g\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$。由此引出支撑我们算法正确性的核心定理：

定理 3.5. *对任意正定二次型* $f(\mathbf{x}) = xAx^T$ *以及* $G \subset \mathrm{GL}_n(\mathbb{Z})$，$A$ *诱导的 lattice G-invariant 当且仅当*

$$f(\mathbf{x}) = \frac{1}{\#G} \sum_{g \in G} f(g\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$$

特别的，任取正定二次型 $u(\mathbf{x})$，*我们都有：对正定对称矩阵* C *满足*

$$\mathbf{x}C\mathbf{x}^T = \frac{1}{\#G} \sum_{g \in G} u(g\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$$

都有 C *诱导的 lattice G-invariant.*

定理 3.5 的证明： 首先证明前半部分，先证必要性：若 A 诱导的 lattice G-invariant，那么根据先前的解释以及(2)，$f(\mathbf{x}) = f(g\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$，从而 $\frac{1}{\#G} \sum_{g \in G} f(g\mathbf{x}) = \frac{1}{\#G} \sum_{g \in G} f(\mathbf{x}) = f(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$，结论成立。

再证明充分性：若 $f(\mathbf{x}) = \frac{1}{\#G} \sum_{g \in G} f(g\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$，$\forall h \in G$：

$$\begin{aligned}
f(h\mathbf{x}) &= \frac{1}{\#G} \sum_{g \in G} f(hg\mathbf{x}) \\
&= \frac{1}{\#G} \sum_{\substack{l \in G \\ g = h^{-1}l}} f(hg\mathbf{x}) \\
&= \frac{1}{\#G} \sum_{\substack{l \in G \\ g = h^{-1}l}} f(l\mathbf{x}) \\
&= f(\mathbf{x})
\end{aligned}$$

上面用到了群论的基本结果：$\forall h \in G, \{h^{-1}g | g \in G\} = G$，这是因为 G 是有限群，结合 $\{h^{-1}g | g \in G\} \subset G$，只需要证明 $h^{-1}g (g \in G)$ 两两不同，这是显然的，因为若有 $h^{-1}g_1 = h^{-1}g_2$，两边同时乘以 h 即得 $g_1 = g_2$，则此时 $\#\{h^{-1}g | g \in G\} \geq \#G$，结合包含关系就得到了 $\{h^{-1}g | g \in G\} = G$。

最后，若 C 满足 $\mathbf{x}C\mathbf{x}^T = \frac{1}{\#G}\sum_{g\in G} u(g\mathbf{x}), \forall \mathbf{x}\in\mathbb{R}^n$ 记 $\hat{u}(\mathbf{x}) = \mathbf{x}C\mathbf{x}^T$，那么

$$\begin{aligned}
\frac{1}{\#G}\sum_{g\in G}\hat{u}(g\mathbf{x}) &= \frac{1}{\#G}\sum_{h\in G}\left(\frac{1}{\#G}\sum_{g\in G}u(hg\mathbf{x})\right)\\
&= \frac{1}{\#G^2}\sum_{h\in G}\sum_{g\in G}u(hg\mathbf{x})\\
&= \frac{1}{\#G^2}\sum_{l\in G}\sum_{\substack{g,h\in G\\ gh=l}}u(l\mathbf{x})\\
&= \frac{1}{\#G^2}\sum_{l\in G}(\#G * u(l\mathbf{x}))\\
&= \hat{u}(x)
\end{aligned}$$

那么根据前面所推到的结论，C 诱导的 lattice G-invariant，证毕。

利用这个定理，我们可以将约束 lattice 的 Invariant Group 包含指定的群 G，转换为无约束优化问题，具体做法如下：令 L 为一个下三角矩阵，那么正定对称矩阵

$$A = \frac{1}{\#G}\sum_{g\in G}(gL)(gL)^T$$

诱导了一个 G-invariant lattice，于是我们只需要优化 L 便可以间接地优化 G-invariant lattice，基于这些观察，我们设计了我们的算法。

接下来需要解决的问题是如何计算 invariant group，为此我们需要对 Invariant Group 与 Isometric Group 做一些更加深入地讨论，这样也可以让我们更好的理解我们算法的价值。

定理 3.6. *设 n 维 Lattice Λ 的 Invariant Group 和 Isometric Group 分别为 G, H，那么 H 同构于 G 的一个子群。*

定理 3.6 的证明： 利用 [3] 中的结果，两个生成矩阵 $B, B' \in \mathrm{GL}_n(\mathbb{R})$ 生成同一个 lattice 当且仅当 $\exists U \in \mathrm{GL}_n(\mathbb{R}), R \in \mathrm{O}(n)$ 使得 $B' = UBR$。注意到 $\forall h \in H$，$\Lambda h = \Lambda$。从而有

$$\forall h \in H, \exists g_h \in G\ s.t. g_h B = Bh \tag{3}$$

我们说明映射 $\phi : h \to g_h(h \in H)$ 给出 H 到 G 的一个子群的同构。利用(3)：$g_h = B^{-1}hB$ 即为一个相似变换，从而可以直接验证 $\phi(h_1)\phi(h_2) = \phi(h_1h_2)$ 是一个群同态，并且是单射，证毕。

上面的定理 3.6 给出了 Isometric Group 到 Invariant Group 的关系，借此我们可以通过 Invariant Group 寻找 Isometric Group。其计算方式由上面的 ϕ 给出。此外给出一个在计算 Invariant Group 的时比较有用的结论，它对 Invariant Group 中每个元素的阶给出了一个估计。

命题 3.7. 设 $G \subset \mathrm{GL}_n(\mathbb{Z})$ 是一个有限群，则 $\forall g \in G$，元素 g 的阶 $o(g)$ 满足 $\phi(o(g)) \leq n$，这里 ϕ 是欧拉函数，元素 g 的阶指的是使得 $g^o = e$ 为 G 的单位元的最小的正整数 o.

命题 3.7 的证明: 设矩阵 g 作为 $\mathbb{Q}^n$ 上的线性变换的极小多项式为 $f(x) \in \mathbb{Q}[x]$，则 $f(x)|(x^{o(g)} - 1)$，注意到 $o(g) =: k$ 的极小性，$f(x) \nmid (x^m - 1), \forall m < k$，这将导致 $f(x)$ 的根中有一个 n 次本原单位根，即 $\Phi_k(x)|f(x)$，这里 $\Phi_k(x)$ 是第 k 个分圆多项式，利用 $f(x)|\det(xI - g)$，这里 I 是单位矩阵。我们有

$$\phi(k) = deg(\Phi_k(x)) \leq deg(f) \leq n$$

结合 $k = o(g)$，证毕！

这个命题在计算 Invariant Group 的时候有一定作用，因为通常我们只能得到 Invariant Group 的一些生成元，我们需要有关它们阶的分析以尝试计算它们生成的群，例如根据上面的结论，我们可以得到 $\mathrm{GL}_2(\mathbf{Z})$ 中所有有限阶的元素阶只能为 1,2,3,4,6 之一。

最后，鉴于我们的算法只能根据指定的 Invariant Group 生成满足相应约束的 lattice 并进行优化，并不能根据指定的 Isometric Group 生成相应的约束，但是我们可以指出，$\forall G \subset \mathrm{GL}_n(\mathbf{Z})$，$G$ 同构于一个正交群的子群 H，根据**定理 3.6**，这说明 Invariant Group 包含 G 的 lattice 的 Isometric Group 可能包含 H。换言之，如果我们希望找到一个 Isometric Group 包含 H 的 lattice，我们可以通过 $\mathrm{GL}_n(\mathbf{Z})$ 中与之同构的群 G，利用本算法约束生成 lattice 使其 Invariant Group 包含 G，这样的 lattice 可能满足我们的要求。

定理 3.8. 有限群 G 的任意一个实表示均可以实现为一个正交表示，即对于实表示 $\rho : G \to \mathrm{GL}_n(\mathbf{R})$，存在单的群同态 $\eta : \rho(G) \to \mathrm{O}(n)$，则 $\eta \circ \rho : G \to \mathrm{O}(n)$ 给出了这个正交表示的实现。

定理 3.8 的证明: 注意到 $S := \sum_{g \in G} \rho(g)^T \rho(g)$ 是正定对称矩阵，进而存在正交对角化 $S = UDU^T, U \in \mathrm{O}(n)$，$D$ 是对角矩阵，且对角元大于 0，令 $D^{\frac{1}{2}}$ 为 D 的每个对角元开平方根得到的对角矩阵。注意到

$$\rho(h)^T S \rho(h) = \sum_{g \in G} \rho(gh)^T \rho(gh) = S, \forall h \in G$$

令 $eta(h) := D^{\frac{1}{2}} U^T h U D^{-\frac{1}{2}}, \forall h \in \rho(G)$，则我们有 η 是单射且

$$\begin{aligned}
\forall h \in \rho(G), \eta(h)^T \eta(h) &= D^{-\frac{1}{2}} U^T h^T U D^{\frac{1}{2}} D^{\frac{1}{2}} U^T h U D^{-\frac{1}{2}} \\
&= D^{-\frac{1}{2}} U^T h^T S h U D^{-\frac{1}{2}} \\
&= D^{-\frac{1}{2}} U^T S U D^{-\frac{1}{2}} \\
&= I_n
\end{aligned}$$

所以 $\eta(h) \in \mathrm{O}(n) \forall h \in \rho(G)$，证毕！

根据这个定理我们立刻知道 $\mathrm{GL}_n(\mathbb{Z})$ 的每个有限子群都和 $\mathrm{O}(n)$ 的某个有限子群同构。

3.4 理论技术总结

在理论推导中，我们提出了一种将约束 Lattice 的 Invariant Group 包含指定群 G 转换为无约束优化问题的方法并给以证明。同时提出了一个通过 Lattice 的 Isometric Group 来计算其部分 Invariant Group 的方法。并提出了有关 Invariant Group 的计算中可以用上的命题。最后指出通过本算法可以尝试生成 Isometric Group 包含指定群的方法。

在实际应用中，我们的想法是选取现有的每个维数最好的 Lattice 的 Invariant Group，但是由于这些 Lattice 通常可以实现为某些李代数的根系，因此这些群的 Invariant Group 会和李代数的 Weyl 群产生较为深刻的联系，这意味着我们需要尝试分析这些 Weyl 群的整表示，这里所需要的数学知识已经超出了小组成员的能力范围，我们也并未找到相关的参考文献，因此我们只能通过两种方式来获得部分的获得 Invariant Group，其一是根据 Gram matrix 直接猜出部分 Invariant Group，其二是借助 Isometric Group 计算 Invariant Group，因为 Isometric Group 的研究结果较多例如 [7]，这或许是更为可行的策略。

3.5 工程改进

在原论文 [3] 中，其学习率的设置依赖于经验性的假设与实验。在我们的实现中，我们采用了余弦退火的学习率调整方法，通常效果优于原论文的指数下降的学习率调整器，且省去了调整超参数的麻烦。

我们注意到，在维数较高的时候，CLP 部分是运行速度的最大瓶颈。而 CLP 的本质是在 $\mathbb{R}^n$ 空间中进行采样，将样本点归一化到单个 Voronoi Cell 后用于 Monte-Carlo 方法计算 NSM。

这启发我们采用现代较为常见的 batch 方法，一方面可以提升 NSM 以及相应的梯度计算精度，另一方面由于各个采样进程的独立性，可以采用并行计算，从而充分利用计算资源。

在较大规模的 n（如 $n > 40$）的计算中，可以进一步采用如 [6] 中以筛法为基础的方法，使用 GPU，可以进一步提升计算速度和并行水平。在我们的实验中，由于 n 本身并不大，囿于代码水平，我们采用了更为基础的基于 CPU 的多线程并行方式。具体而言，通过使用 numba 库提供的 JIT 技术，我们成功在几乎不损失计算速度的情况下实现了批大小 (batchsize) 为 128 的提升，有效地减轻了 lattice 和 NSM 震荡的情形，提升了收敛速度。这也使得我们在合理的计算时间及计算资源消耗规模的情况下，将原有的计算维数提升至 28 维（见4.1）。

3.6 伪代码与实现

结合前面部分的理论结果，我们在 [3] 中给出的算法上进行修改，得到一个新的基于 5816 SGD 的算法用于找出优秀的 G-invariant 的生成矩阵：

Algorithm 1 G-invariant lattice construction

Input: n: Dimension; G: Invariant group;

Output: G-invariant generator matrix $\mathbf{B}$

$\mathbf{L} \leftarrow GRAN(n, n)$

$\mathbf{L} \leftarrow |\mathbf{L}|^{-1/n}\mathbf{L}$

for $t = 0$ to n **do**

  Get μ from scheduler;

  $\mathbf{A} \leftarrow \frac{1}{|G|}\sum_{g\in G}\mathbf{gL}(\mathbf{gL})^T$

  $\mathbf{B} \leftarrow CHOL(\mathbf{A})$

  $\mathbf{z} \leftarrow URAN(n)$

  $\mathbf{y} \leftarrow CLP(\mathbf{B}, \mathbf{zB})$

  $\mathbf{e} \leftarrow \mathbf{yB}$

  $\nabla\mathbf{B} \leftarrow B\_DIFF(\mathbf{B}, \mathbf{y}, \mathbf{e})$

  $\nabla\mathbf{A} \leftarrow A\_DIFF(\mathbf{B}, \nabla\mathbf{B})$

  $\nabla\mathbf{L} \leftarrow L\_DIFF(\mathbf{G}, \mathbf{L}, \nabla\mathbf{A})$

  $\mathbf{L} \leftarrow \mathbf{L} - \mu\nabla\mathbf{L}$

  if $(t \bmod T_r) = T_r - 1$ **then**

    $\mathbf{L} \leftarrow |\mathbf{L}|^{-1/n}\mathbf{L}$

  end if

end for

$\mathbf{A} \leftarrow \frac{1}{|G|}\sum_{g\in G}\mathbf{gL}(\mathbf{gL})^T$

$\mathbf{B} \leftarrow CHOL(\mathbf{A})$

$\mathbf{B} \leftarrow |\mathbf{B}|^{-1/n}\mathbf{B}$

$GRAN, URAN, CLP$ 的定义与实现与 [3] 中相同。

$CHOL$ 为 Cholesky 分解，给定正定对称矩阵 A，返回唯一的 B 满足 $A = BB^T$。我们使用了 numpy 库中的实现。

B_DIFF 函数中我们使用 [3] 中的计算结果来直接求得 B 关于 NSM 的梯度 ∇B。

A_DIFF 函数将 ∇B 对 Cholesky 分解反向传播来得到 ∇A，我们使用了 Murray 的论文 [5] 中的算法及代码。

L_DIFF 函数直接通过如下公式计算最终 L 关于 NSM 的梯度 ∇L：

$$\nabla L = \frac{1}{|G|}\sum_{g\in G} 2g^T \nabla A g L$$

我们也使用 pytorch 编写了自动求梯度的版本，与上述版本效果一致但运行效率略低。

值得注意的是，简单地修改这一算法可以只对 n 阶生成矩阵的前 m 个行向量进行 G-invariant 的限制。具体的，令 G 是一个 m 阶矩阵组成的 invariant group，将算法第 5 行改为

$$A \leftarrow \frac{1}{|G|}\sum_{g\in G} \mathbf{gL_1}(\mathbf{gL_1})^T$$

其中 L_1 为 L 的前 m 行构成的 $m\times n$ 矩阵。将第 6 行改为

$$B \leftarrow CHOL(A) + L_2$$

这里 + 表示按行拼接，L_2 为 L 的后 $n-m$ 行，即令 B 为 $CHOL(A)$ 得到的 $m\times m$ 矩阵补零补为 $m\times n$ 矩阵后与 L_2 按行拼接。

同理对算法第 10 ~12 行的反向传播部分进行对应修改，这样就在限制前 m 行向量的同时不对后 $n-m$ 行向量做出限制。

4 实验与结果

4.1 $G=\{I_n\}$ 的无约束情形

注意到当 $G=\{I_n\}$ 时，我们的算法可以退化为 [3] 中的算法，进而我们对其进行了实验，验证我们的算法的结果与之相符合。同时，我们也对于更小的 n 和更大的 n 进行了实验，乙方验证我们的算法确实可以在 n 较小的时候收敛到当前的最优解，同时在更大的 n 上创造了 SOTA 的 NSM。具体的计算结果见4.1。在表格中，所有数据若没有特别标注，均引用自 [2]，并且 [3] 中的数据也已考虑在内。由于我们的 NSM 的计算的方差在 $3*10^{-6}$ 左右，而且优化的次数并不算多，故我们认为误差在 $1*10^{-5}$ 之内都可以看作是收敛于已知的最佳格点，此时在”Better?” 一栏中用 $\approx$ 标记，而如果我们优化的格点优于已知的最好的结果，则用 $<$ 标记。

囿于计算资源，我们在 $n\in\{2,3,\cdots,20\}$ 时采用了 $1.6*10^6$ 次迭代，取得了更为精细的 NSM 以及 theta-image，而在 $n\in\{21,22,\cdots,26\}$ 时采用了 $4*10^5$ 次迭代，在 $n\geq 27$ 时，仅进

行了 10^5 次迭代。在 n 较小的第一种情况中，我们得到了比较精确的 theta-image，且可以通过 [3] 中的方法将 theta-image “锐化”，使得原本模长“几乎相等”的格点变成模长完全相等的格点。而在 $n > 16$ 时，NSM 的收敛速度显著快于格点 theta-image 的收敛速度，进而此时生成的 theta-image 并不理想，其中并没有包含充分有效的信息，故我们略去不提。

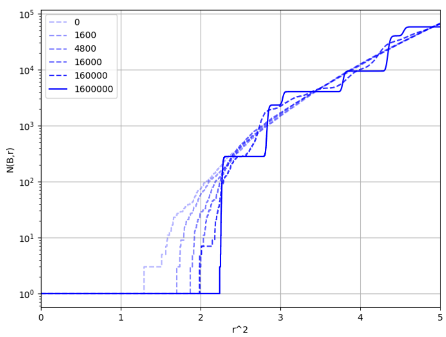

图 1: n=15

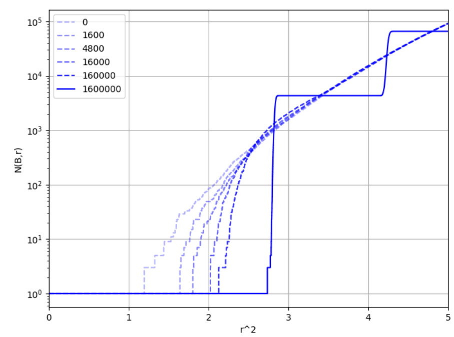

图 2: n=16

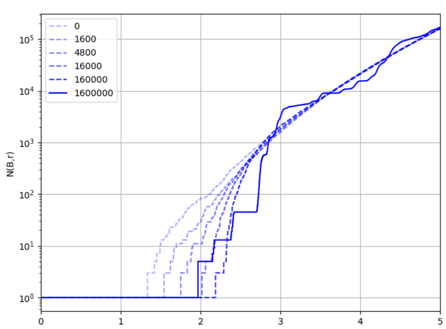

图 3: n=18

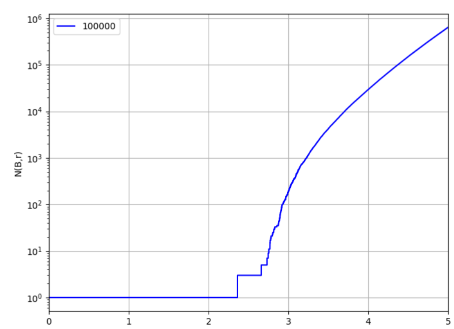

图 4: n=28

表 2: Result of $G = \{I_n\}$

	Best Previously Reported		Generic Bounds		Ours	
n	NSM	Lattice	Lower[8]	Upper[9]	NSM	Better?
2	0.08019	A_2	0.08019	0.08027	0.08019	$\approx$
3	0.07854	A_3^*	0.07787	0.07972	0.07855	$\approx$
4	0.07660	D_4	0.07609	0.07882	0.07660	$\approx$
5	0.07562	D_5^*	0.07465	0.07873	0.07563	$\approx$
6	0.07424	E_6^*	0.07347	0.07778	0.07435	$\approx$
7	0.07312	E_7^*	0.07248	0.07686	0.07312	$\approx$
8	0.07168	E_8	0.07164	0.07565	0.07169	$\approx$
9	0.07162	AE_9	0.07090	0.07555	0.07162	$\approx$
10	0.07081	D_{10}^+	0.07026	0.07486	0.07081	$\approx$
11	0.07043	A_{11}^3	0.06969	0.07402	0.07043	$\approx$
12	0.07003	Glued $D_6 \times D_6$	0.06918	0.07310	0.07004	$\approx$
13	0.06970	[3]	0.06872	0.07240	0.06970	$\approx$
14	0.06926	[3]	0.06830	0.07167	0.06926	$\approx$
15	0.06887	Λ_{15}^*[3]	0.06793	0.07101	0.06888	$\approx$
16	0.06830	Λ_{16}	0.06759	0.07040	0.06830	$\approx$
17	0.06910	$\Lambda_{16} \otimes \mathbb{Z}$	0.06727	0.06989	0.06819	$<$
18	0.06953	$\Lambda_{16} \otimes A_2$	0.06698	0.06940	0.06793	$<$
19	0.06982	$\Lambda_{16} \otimes A_3^*$	0.06671	0.06895	0.06782	$<$
20	0.06988	$\Lambda_{16} \otimes D_4$	0.06645	0.06854	0.06754	$<$
21	0.06998	$\Lambda_{16} \otimes D_5^*$	0.06622	0.06817	0.06731	$<$
22	0.06987	$\Lambda_{16} \otimes E_6^*$	0.06600	0.06782	0.06708	$<$
23	0.06973	$\Lambda_{16} \otimes E_7^*$	0.06580	0.06750	0.06685	$<$
24	0.06577	Λ_{24}	0.06561	0.06721	0.06664	
25	0.07566	A_{25}^*	0.06543	0.06693	0.06643	$<$
26	0.07568	A_{26}^*	0.06526	0.06668	0.06624	$<$
27	0.07571	A_{27}^*	0.06509	0.06644	0.06610	$<$
28	0.07574	A_{28}^*	0.06494	0.06622	0.06593	$<$

4.2 求由两组互相正交的基形成的 lattice

利用 3.5 中的算法，可以求一些满足特殊性质的 lattice，比如求由两组互相正交的基形成的 lattice。具体的，给定 n, m，我们希望求出一个 n 维 lattice 的生成矩阵 B，使得前 m 行的向量与后 $n-m$ 行的向量正交。

考虑 G-invariant 的定义，

$$\forall g \in G,\ \forall x \in \mathbb{Z}^{1\times n},\ xAx^T = (xg)A(xg)^T$$

其中 A 为生成矩阵 B 的 Gram matrix，$A = BB^T$。

将 xAx^T 视为二次型，

$$f(x) = \sum_{i,j} A_{i,j} x_i x_j$$

则要求相当于 $f(x) = f(xg)$ 对所有 x, g 成立。

当 $G = \{I_n, \mathrm{diag}(-I_m, I_{n-m})\}$ 时，一个 lattice 是 G-invariant 的当且仅当其对应的 Gram matrix A 满足

$$\sum_{i,j} A_{i,j} x_i x_j = \sum_{i,j} A_{i,j} (xg)_i (xg)_j$$

也即当 $i \le m, j > m$ 或 $i > m, j \le m$ 时要求 $A_{i,j} = 0$，而 $A_{i,j}$ 是生成矩阵 B 的第 i 行与第 j 行的内积，因此该条件相当于 B 的前 m 行与后 $n-m$ 行正交。

故将 G 设置为该大小为 2 的群，可以使用上述算法来计算满足要求的 B 并优化其 NSM。

我们在 $n = 2 \sim 22$, $m = \lceil \frac{n}{2} \rceil \sim n-1$ 上进行实验，每组迭代 10^5 轮，下表是部分结果，其中 NSM 一列为算法得到的结果，NSM* 一列为 [2] 的 Table I 中给出的 product lattice 的参考结果：

表 3: Result of 4.2

n	m	NSM	NSM*	n	m	NSM	NSM*
2	1	0.083315	0.083333	12	11	0.071426	0.071420
3	2	0.081205	0.081222	13	12	0.070975	0.071034
4	2	0.080181	-	14	12	0.071392	0.071455
4	3	0.079729	0.079714	15	12	0.071679	0.071709
6	5	0.076857	0.076858	15	14	0.070600	-
7	6	0.075476	0.075478	16	12	0.071616	0.071668
8	4	0.076599	-	16	15	0.069827	-
8	6	0.075760	-	18	16	0.069533	0.06953
8	7	0.074326	0.074321	19	16	0.069836	0.06982
10	9	0.072716	0.072715	20	16	0.069881	0.06988
12	8	0.073474	-	22	16	0.069875	0.06987

由于计算 NSM 所用 grader 的总轮数较少（10^5 轮），结果存在一些误差。总的来说，使用该算法能够成功找到 [2] 中给出的 Best product 结果，而 $n = 13 \sim 16, m = 12$ 时得到的结果略优于该论文，与 [4] 中的发现相符。

4.3 非平凡的 Invariant Group 的探索

因为能够计算 Invariant Group 的 lattice 较为有限，我们主要实验了 lattice E_6, K_{12} 的 Invariant Group 约束的效果，其中对于 K_{12} 我们取出了其中一个 6 维子空间的部分 Invariant Group 进行约束。相比于 [2] 中的正交 gluing 策略取得了较好的效果。

根据 [1]，E_8 的 Isometric Group 中有元素 $h = \begin{bmatrix} H_4 & 0 \\ 0 & H_4 \end{bmatrix}$ 其中

$$H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

根据这个以及 E_8 的生成矩阵 B，$g = BhB^{-1}$ 在 E_8 的 Invariant Group 中。我们使用 g 在 $\mathrm{GL}_8(\mathbb{Z})$ 中生成的群作为约束，使用我们的算法在 $n = 9, 10$ 的时候约束其中 8 个基向量生成的 lattice 满足 g 的约束，剩下两个没有约束要求，得出的结果如下。

表 4: Result of E_8 gluing

	Best Gluing Lattice [2]		Ours	
n	NSM	Lattice	NSM	Better?
9	0.072891732	$E_8 \otimes \mathbb{Z}$	0.071634753	yes
10	0.072715487	$AE_9 \otimes \mathbb{Z}$	0.071306335	yes

注意到 K_{12} 的 Gram Matrix 为

$$A = \begin{bmatrix} 4 & 0 & 0 & -2 & 0 & 0 & 2 & -1 & -1 & -1 & 2 & -1 \\ 0 & 4 & 0 & 0 & -2 & 0 & 2 & -1 & -1 & -1 & -1 & 2 \\ 0 & 0 & 4 & 0 & 0 & -2 & 2 & 2 & 2 & -1 & -1 & -1 \\ -2 & 0 & 0 & 4 & 0 & 0 & -1 & -1 & 2 & 2 & -1 & -1 \\ 0 & -2 & 0 & 0 & 4 & 0 & -1 & 2 & -1 & 2 & -1 & 2 \\ 0 & 0 & -2 & 0 & 0 & 4 & -1 & -1 & -1 & 2 & 2 & -1 \\ 2 & 2 & 2 & -1 & -1 & -1 & 4 & 0 & 0 & -2 & 0 & 0 \\ -1 & -1 & 2 & -1 & 2 & -1 & 0 & 4 & 0 & 0 & -2 & 0 \\ -1 & -1 & 2 & 2 & -1 & -1 & 0 & 0 & 4 & 0 & 0 & -2 \\ -1 & -1 & -1 & 2 & 2 & 2 & -2 & 0 & 0 & 4 & 0 & 0 \\ 2 & -1 & -1 & -1 & -1 & 2 & 0 & -2 & 0 & 0 & 4 & 0 \\ -1 & 2 & -1 & -1 & -1 & 2 & 0 & 0 & -2 & 0 & 0 & 4 \end{bmatrix}$$

注意到 A 的前 6 行前 6 列形成的矩阵对应的 lattice 在 $\begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}$ 下不变，我们根据此设计了群 G 并进行实验。结果如下

表 5: Result of part of K_{12} gluing

	Best Gluing Lattice in [2]		Ours	
n	NSM	Lattice	NSM	Better?
13	0.071034583	$K_{12} \otimes \mathbb{Z}$	0.069811712	yes
14	0.071455542	$K_{12} \otimes A_2$	0.069409005	yes
15	0.071709124	$K_{12} \otimes A_3^*$	0.069130726	yes

可以看到确实比正交拼接的效果要好不少。

4.4 训练结果可视化

我们定义 $N(B,r) = |\{u \in \mathbb{Z}^n : ||uB|| \le r\}|$，称 $N(B,r)$ 关于 r^2 的图像为 theta-image。

theta-image 能够可视化 lattice 的结构，同时算法总是会倾向于阶梯状的 theta-image。绘制训练过程中的 theta-image 能够可视化算法的收敛速度。下图展示了 $n = 12$ 时原算法、$G = \{I_{12}\}$ 和 $G = \{I, \mathrm{diag}(-I_6, I_6)\}$ 时的图像：

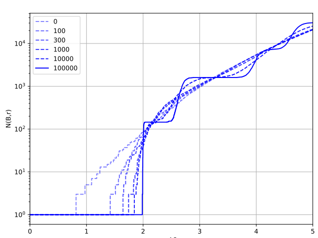

图 5: 原算法

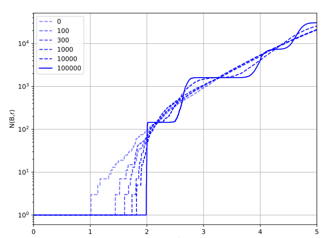

图 6: $G = \{I_{12}\}$

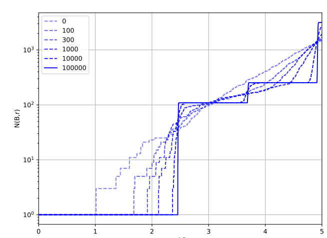

图 7: $G = \{I, \mathrm{diag}(-I_6, I_6)\}$

可以看到我们的做法取 $G = \{I_{12}\}$ 时能够生成与原算法类似的 lattice，同时收敛速度快于原算法。而当 $G = \{I, \mathrm{diag}(-I_6, I_6)\}$ 算法也能够收敛到符合条件的局部最优 lattice，但是很难达到全局最优点。

5 当前算法的不足以及可以进行的进一步探索

我们当前的算法仍然有一些不足，目前该算法仅能控制一组基向量生成的 lattice 的 Invariant Group，最理想的情况是应当可以将基向量分为任意多组并分别控制它们的 Invariant Group，但限制于 cholesky 分解只在对应维数存在唯一，这导致我们控制多组 Invariant Group 存在困难。

可以看到我们的算法仍然有许多潜力等待着开发，例如在特定情形下，当我们对 Voronoi 有一定先验知识因而需要开发具有特定的对称性的 Lattice 时，可以通过设计 Invariant Group 以完成该需求。

6 总结

我们基于数学推导设计了一个用于求得 G-invariant lattice 的数值优化算法，并通过实验验证了该算法的有效性。相对于原先只能求无约束 lattice 的算法，我们的算法可以对 lattice 的几何性质进行约束，有更高的灵活性，同时也保证了有效性，能够高效地收敛到之前的论文中得出的最优解，并在一些场景下能得到更优的结果。同时，除了报告中所举的几个实验例子之外，我们认为这一算法也可能会有更广泛的应用场景。

参考文献

[1] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. NewYork, NY:Springer, 1999. https://doi.org/10.1007/978-1-4757-6568-7

[2] E. Agrell and B. Allen, “On the best lattice quantizers”, *IEEE Trans. Inf. Theory*, vol. 69, no. 12, pp. 7650–7658, Dec. 2023. https://doi.org/10.1109/TIT.2023.3291313

[3] E. Agrell, D. Pook-Kolb and B. Allen, “Optimization and identification of lattice quantizers”, arXiv:2401.01799, Jun. 2024. https://doi.org/10.48550/arXiv.2401.01799

[4] E. Agrell, D. Pook-Kolb, and B. Allen, “Glued lattices are better quantizers than K_{12},” *IEEE Trans. Inf. Theory*, 2024, to appear. https://doi.org/10.1109/TIT.2024.3398421

[5] I. Murray, ”Differentiation of the Cholesky decomposition”, arXiv:1602.07527, Feb. 2016. https://doi.org/10.48550/arXiv.1602.07527

[6] Ducas, L., Stevens, M., van Woerden, W. (2021). Advanced Lattice Sieving on GPUs, with Tensor Cores. In: Canteaut, A., Standaert, FX. (eds) Advances in Cryptology –EUROCRYPT 2021. EUROCRYPT https://doi.org/10.1007/978-3-030-77886-6_9

[7] W. PLESKEN, B. SOUVIGNIER, Computing Isometries of Lattices, Journal of Symbolic Computation, Volume 24, Issues 3–4, 1997, Pages 327-334, ISSN 0747-7171, https://doi.org/10.1006/jsco.1996.0130.

[8] J. Conway and N. Sloane, ”A lower bound on the average error of vector quantizers (Corresp.),” in IEEE Transactions on Information Theory, vol. 31, no. 1, pp. 106-109, January 1985, https://doi.org/10.1109/TIT.1985.1056993.

[9] ”Reformulation of the covering and quantizer problems as ground states of interacting particles” in Phys. Rev. E, vol. 82, pp. 22, Nov 2010, https://link.aps.org/doi/10.1103/PhysRevE.82.056109