UNIVERSITY OF CALIFORNIA SAN DIEGO

Graph Embedding via Subspace Minimization with Applications to Chip Placement and Semi-Supervised Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
Computer Science and Engineering

in

Doctor of Philosophy
Computer Science and Engineering

by

Chester Holtz

Committee in charge:

Professor Chung-Kuan Cheng, Chair
Professor Albert Chern
Professor Bill Lin
Professor Gal Mishne

2023

The Dissertation of Chester Holtz is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Chung-Kuan Cheng for his support as my advisor and the chair of my committee. I would also like to acknowledge Professor Gal Mishne for her care and collaboration and for her feedback on much of the work I have been doing during my PhD. Their academic guidance and mentorship have been invaluable.

Chapter 1, in part is a composition of two works, one of which has been submitted for publication—Revisiting Semi-Supervised Laplacian Eigenmaps via Alignment, 2023 and one currently being prepared for submission for publication of the material: Minimizing a Quadratic over Stiefel Manifolds.

Chapter 2, in full, is a reprint of material comprising one submitted paper: Revisiting Semi-Supervised Laplacian Eigenmaps via Alignment and three publications: Placement Initialization via Sequential Subspace Optimization with Sphere Constraints, 2023, International Symposium on Physical Design (ISPD) (best paper nomination), Placement Initialization via a Projected Eigenvector Algorithm, 2022, Design Automation Conference (DAC) 2022, and Net Separation-Oriented Printed Circuit Board Placement via Margin Maximization, 2022, 27th Asia and South Pacific Design Automation Conference (ASP-DAC) (best paper award). The dissertation author was the primary investigator and author of these papers.

VITA

2017	Bachelor of Science, University of Rochester, Rochester
2023	Doctor of Philosophy, University of California San Diego

ABSTRACT OF THE DISSERTATION

Graph Embedding via Subspace Minimization with Applications to Chip Placement and Semi-Supervised Learning

by

Chester Holtz

Doctor of Philosophy
Computer Science and Engineering in Doctor of Philosophy
Computer Science and Engineering

University of California San Diego, 2023

Professor Chung-Kuan Cheng, Chair

Recent work has shown that by considering an optimization perspective of the eigenvalues and eigenvectors of graph Laplacians, more efficient algorithms can be developed for tackling many graph-related computing tasks. In this dissertation, we present efficient methods for solving general quadratic programs with nonconvex constraints in the context of very-large-scale integration (VLSI) computer-aided design (CAD) and graph-based semi-supervised learning problems. We propose a general framework for matrix quadratic programming with nonconvex constraints, which is motivated by classic algorithms for solving trust-region subproblems.

We introduce approximate and iterative methods with derived convergence guarantees. We demonstrate the effectiveness of our framework on large-scale numerical test cases, specifically real-world benchmarks. By leveraging analytical VLSI and PCB layout engines, we show that effective initialization using our method consistently improves a variety of pre- and post-detailed placement metrics. Additionally, we introduce a graph semi-supervised learning algorithm based on this framework, which yields strong results across a wide spectrum of label rates.

Introduction

0.1 Optimization on Stiefel Manifolds

Optimization problems involving orthonormal matrices arise in various scientific and engineering applications, such as eigenfunction and electronic structure calculations Absil et al. [2007], Edelman et al. [1998], Townsend et al. [2016]. Although these problems typically have smooth objective functions, they are often ill-conditioned.

Efficient optimization procedures that can handle orthogonality constraints, particularly for large-scale problems on graphs, have been a subject of recent research. One popular framework for optimization on manifolds, including the Stiefel manifold characterized by $X^\top X = I$, was developed in Absil et al. [2007], Edelman et al. [1998]. Notably, the trust-region methods proposed in Absil et al. [2007] generalize traditional Euclidean optimization procedures to the Stiefel manifold, including methods based on gradient descent, non-linear conjugate gradient, and Newton's method.

However, problems with computationally desirable structures such as sparsity particularly benefit from numerical improvements in efficiency and robustness, particularly in the large-scale and ill-conditioned regimes. For example, Newton's method is typically too expensive, necessitating inversion of large matrices and existing first-order methods converge slowly for ill-conditioned problems. Therefore, there is a need for techniques that inherit the desireable rapid convergence properties of Newton's methods, while maintaining practical scalability under such circumstances.

In this dissertation, we address this gap by adapting sequential subspace methods, in particular by generalizing Hager's Sequential Subspace Method (SSM) Hager [2001], Hager and Park [2005], Erway et al. [2009], to the Stiefel manifold. Generalizing these methods involves developing a Sequential Quadratic Programming (SQP) iterate which is specifically designed to deal with the problems that arise on the Stiefel manifold. In addition to introducing efficient algorithms for quadratic optimization on the Stiefel Manifold, we show empirically that SSM is very robust and efficient on a variety of large-scale numerical problems compared to existing open-source implementations of baseline methods.

0.2 Laplacians of Graphs

The numerical study provided in this dissertation focuses on the graph data. Spectral graph theory, a subfield of graph theory, is the study of the combinatorial properties of graphs implied by the eigenvalues and eigenvectors of certain connectivity matrices characterized by the graph structure. Applications of spectral techniques in computer science and machine learning are discussed in Arsic et al. [2012]. In particular, spectral clustering has received much attention in the machine learning community. The original work by Fiedler [1973] provides insight into how spectral methods yield information regarding the topology an unweighted graph and the relevance of the second-smallest eigenvalue of L to associated cluster structure. More concretely, the work of Donath and Hoffman [1973] derived a lower bound on the number of cut-edges for a k-partition in terms of the k largest eigenvalues of a family of matrices that includes the Laplacian, L, and its additive inverse, $-L$. Spectral embeddings were originally introduced by Hall [1970], who showed that the r smallest positive eigenvectors of L can be used to embed the vertices of a graph in r dimensions. This technique is particularly useful for graph visualization and VLSI placement, where it is necessary to determine coordinates in the plane to lay out the vertices of a graph or VLSI netlist.

Spectral graph theory has a long history beyond spectral clustering. A detailed survey of results from the 1950s is presented in Cvetković et al. [1980]. Of particular note is the seminal work of Kirchhoff [1847], often considered the field's birth. Kirchhoff proved that the number of different spanning trees of an unweighted graph is equal to the product of all positive eigenvalues of the Laplacian matrix L, divided by n.

More recently, research has shown that by leveraging an optimization perspective of key spectral properties of the graph Laplacians, such as eigenvalues and eigenvectors, more efficient algorithms can be developed to tackle many graph-related computing tasks Teng [2016]. Spectral methods can potentially lead to much faster algorithms for solving sparse matrices Zhao et al. [2017], Koutis et al. [2014], numerical optimization Christiano et al. [2011], graph analytics Koren [2003], Imre et al. [2020], and machine learning Belkin and Niyogi [2003], Holtz et al., as well as very-large-scale integration (VLSI) computer-aided design (CAD) Hall [1970], Jacobs et al. [2018], Hamada et al. [1992], Hagen and Kahng [1991], Charles J. Alpert [1995].

The work in this dissertation contributes to the development of efficient methods inspired by spectral graph theory for graph-based semi-supervised learning and VLSI layout.

0.3 Outline and Contributions

This dissertation includes three chapters. Chapter 1 describes a scaleable and general framework for quadratic programming on Stiefel manifolds. The method is introduced as a generalization of an existing technique for solving trust-region subproblems. Chapter 2 validates the efficacy of this method on several problems in VLSI design and machine learning. In particular, the scalability of solution quality relative to the state of the art are highlighted. Chapter 3 presents the conclusions of this dissertation and discusses the possible future works to be explored.

Chapter 1

A Sequential Subspace Method for Quadratic Optimization on Stiefel Manifolds

1.1 Quadratic Programming on Stiefel Manifolds

We assume k is a positive integer (much) less than n. Let K denote the set $\{1,2,\ldots,k\}$. Let $I_{n,k}$ denote the sub matrix of the identity matrix I_n, consisting of the first k columns. Let O_k be the orthogonal group, i.e., $Q \in O_k$ if and only if $Q \in \mathbb{R}_{k,k}$ and $Q^\top Q = I_k$.

1.1.1 Formulation

In this chapter, we study algorithms designed to solve matrix optimization problems over a compact Stiefel manifold:

$$\min_X \left\{ \mathcal{F}(X) := \frac{1}{2}\langle X, AX\rangle - \langle B, X\rangle \right\}, \quad \text{subject to } X \in St(n,r) \tag{1.1}$$

Occasionally, we will also consider the more general *reweighted* problem

$$\min_X \left\{ \mathcal{F}(X) := \frac{1}{2}\langle X, AXC\rangle - \langle BC^{1/2}, X\rangle \right\}, \quad \text{subject to } X \in St(n,r) \tag{1.2}$$

Note that when $C = I$, there is an equivalence between the two problems. Typically, A is denoted the *system matrix*, which is a symmetric $n \times n$ matrix and $B \in \mathbb{R}^{m,r}$. In this dissertation, and as we discuss later, the above problem is discussed in the context of applications involving graphs. Notably, A will almost always correspond to a *combinatorial Laplacian*. The set $St(n,r)$ is a Stiefel manifolds and is defined as

$$St(n,r) := \{X \in \mathbb{R}^{n\times r} : X^\top X = I_r\} \tag{1.3}$$

and the associated Euclidean projection is given by

Remark 1.1.1. Given a matrix $X \in \mathbb{R}^{n\times k}$, its projection onto $\mathrm{St}(n,k)$, $[X]_+ := \arg\min\{\|X_s - X\|_F : X_s \in \mathrm{St}(n,k)\}$ is given by

$$[X]_+ = UV^\top, \tag{1.4}$$

where $X = U\Sigma V^\top$ is the singular value decomposition.

Minimization over a Stiefel manifold can be regarded as one generalization of the well-known nonconvex quadratic over the unit ball or sphere, arisen in trust region methods Sorensen [1982], Conn et al. [2000]. There could exist many local solutions in Eq. 1.1. A global minimizer of Eq. 1.1 in general can be be ensured in advance. To overcome this difficulty, we introduce the associated problems,

$$\min_X \left\{\mathcal{F}(X) := \frac{1}{2}\langle X, A'X\rangle - \langle B, X\rangle\right\}, \quad \text{subject to } X \in St(n,r) \tag{1.5}$$

and

$$\min_X \left\{\mathcal{F}(X) := \frac{1}{2}\langle X, A'XC\rangle - \langle BC^{1/2}, X\rangle\right\}, \quad \text{subject to } X \in St(n,r) \tag{1.6}$$

Where A', constructed from A, has the same set of eigenvectors as A, but has eigenvalues $d'_1 = d'_2 = \ldots = d'_r \leq \ldots \leq d'_n$. More precisely, write the eigenvector decompositions of A and A', respectively:

$$A = [v_1, v_2, \dots, v_n]\mathrm{diag}(d_1, \dots, d_n)[v_1, v_2, \dots, v_n]^\top$$

$$A' = A - \sum_{k=2}^{r}(d_k - d_1)v_k v_k^\top = [v_1, v_2, \dots, v_n]\mathrm{diag}(d'_1, \dots, d'_n)[v_1, v_2, \dots, v_n]^\top$$

Note that $d'_j = d_j$ for all $j > r$ and $d'_1 = d'_2 = \dots d'_r = d_1$.

1.1.2 Preliminary results in the spherical case

As previously mentioned, eq. (1.1) is one natural generalization of the following constrained problem discussed in Hager [2001], Hager and Park [2005]

$$\min_x \{x^\top Ax - 2\langle x, b\rangle : \|x\| = 1, x \in \mathbb{R}^n\} \tag{1.7}$$

This problem is related to the trust region subproblem:

$$\min_x \{x^\top Ax - 2\langle x, b\rangle : \|x\| \leq 1, x \in \mathbb{R}^n\} \tag{1.8}$$

The following two propositions describe the global solution of 1.7. See Sorensen [1982]. The condition 1.9 states that the global solution x is the critical point associated with λ bounded above by the smallest eigenvalue d_1 of A.

Proposition 1.1.1 (Hager and Park [2005])**.** *A vector $x \in \mathbb{R}^n$ is a global solution of 1.7, if and only if $\|x\| = 1$, and*

$$A - \lambda I \succcurlyeq 0, \ (A - \lambda I)x = b \tag{1.9}$$

holds for some $\lambda \in \mathbb{R}$

Proposition 1.1.2 (Hager [2001])**.** *Consider the eigenvector decomposition*

$$A = [v_1, v_2, \dots, v_n]diag(d_1, \dots, d_n)[v_1, v_2, \dots, v_n]^T.$$

Let V_1 be the matrix whose columns are eigenvectors of A with eigenvalue d_1. Then, $x = Vc$ is a solution for a vector c chosen in the following way:

- *Degenerate case: suppose $V_1^\top b = 0$ and $c_\perp := \|(A - d_1 I)^\dagger b\| \leq 1$ Then, $\lambda = d_1$ and $x = (1 - c_\perp^2)^{1/2} v_1 + (A - d_1 I)^\dagger b$*

- *Nondegenerate case: $\lambda < d_1$ is chosen so that $x = (A - \lambda I)^{-1} b$ with $\|x\| = 1$.*

Remark 1.1.2 (Hager and Park [2005]). Note that $\|(A - \lambda I)^{-1} b\|$ decreases monotonically with respect to $\lambda \leq d_1$. A proper value of λ meets the condition $\|x\| = 1$. A tighter bound on λ can be estimated from

$$(d_1 - \lambda)\|V_1 b\|^2 \leq 1 = \|(A - \lambda I)^{-1} b\|^2 \leq (d_1 - \lambda)^{-2} \|b\|^2$$

With $\|V_1 b\| > 0$, λ lies in the interval $[d_1 - \|b\|, d_1 - \|V_1 b\|]$.

1.1.3 Optimality conditions

In this section, we characterize the necessary and sufficient conditions for local and global critical points. First, we state a first order condition of eq. (1.1) and an implied definiteness condition. From this definiteness condition, we motivate an approximate method in Section 1.2. We then discuss the sufficient conditions for local solutions of eq. (1.1) and associated global solutions of eq. (1.5). Notably, these conditions are characterized by the eigenvalues of the system matrix A and A' as well as the eigenvalues of the Lagrangian multiplier matrix Λ.

Necessary conditions

We first state a pair of first-order necessary conditions of eq. (1.1):

Remark 1.1.3 (First order condition of eq. (1.1)). Any tangent vector to $\mathrm{St}(n,r)$ at X can be expressed as $X\Omega + Z$ for some skew matrix Ω and $Z \in \mathbb{R}^{n\times r}$ with $Z^\top X = 0$. For a local optimal X, the differential $AX - B$ of $\mathcal{F}$ must lie in the normal space to $\mathrm{St}(n,r)$ at X. This is exactly the first order optimal condition:

$$AX = X\Lambda + B, \quad \lambda \in \mathbb{R}^{r\times r} \tag{1.10}$$

for some symmetric matrix Λ.

Proposition 1.1.3 (Definiteness conditions of $X^\top B$ for eq. (1.1))**.** *Note the first term of F satisfies the invariance $\langle X, AX\rangle = tr(X^\top AX) = tr(\widetilde{X}^\top A\widetilde{X})$, where $\widetilde{X} = XQ$ for any orthogonal $Q \in \mathbb{R}^{2\times 2}$. X is a local minimizer if $X^\top B \succcurlyeq 0$ and symmetric.*

Proof. Note that by assumption, X is feasible—i.e. $X^\top X = I$. Consider the substitution $X \to XQ$ for some orthogonal matrix Q. Since $C = I$, $F(XQ) = \langle XQ, AXQ\rangle - \langle Q, X^\top B\rangle$.

Fix X. Note that the first term satisfies the invariance $\langle X, AX\rangle = \langle XQ, AXQ\rangle$. For any orthogonal $Q \in \mathbb{R}^{k\times k}$. The optimal choice of Q is determined by the affine second term. A standard result from matrix analysis yields its minimizer Horn and Johnson [2013]. Let $X^\top B = UDV^\top$ be the SVD of $X^\top B$. Then, $Q = U_B V_B^\top$ and $\langle XQ, B\rangle = \langle Q, X^\top B\rangle = \langle I, D\rangle = tr(D) \geq 0$. □

In general, there could exist many critical points X fulfilling the aforementioned condition. These points are called stationary points(maximizers, minimizers, or saddle points). We will show that the eigenvalues of the associated matrix Λ, when examined in conjunction with the eigenvalues of the system matrix A, characterize the quality of these critical points and the optimality of X.

Sufficient conditions

Here, we state several sufficient conditions for optimality. Introduce one orthogonal projection $P_j^\perp I - \sum_{k\in\mathcal{R}\setminus\{j\}} x_k x_k^\top$.

Remark 1.1.4 (Min-max theorem). Consider the $n-(r-1)$ dimensional subspace, i.e., orthogonal to span$(x_2, \ldots, x_r)$. Apply the Courant-Fischer min-max theorem. Then the smallest nonzero eigenvalue of $P_1^\perp A P_1^\perp$ is bounded above by d_r, i.e., $d_r \geq \lambda_1$ must hold. Repeat the arguments to other $r-1$ second order conditions. Thus, for a minimizer, eigenvalues of the associated matrix Λ are bounded above by d_r:

$$d_r \geq \max(\lambda_1, \ldots, \lambda_r)$$

Proposition 1.1.4 (Global solutions of eq. (1.1))**.** *Let d_1 be the smallest eigenvalue of A. Let X_0 be a stationary point of eq. (1.1). and let Λ' be the associated multiplier matrix. Suppose*

$$d_1 I \succcurlyeq \Lambda'$$

Then X_0 is a global minimizer. Suppose $d_1 I \succ \Lambda'$. Then X_0 is the unique global minimizer.

Proof. Let $\Lambda \in \mathbb{R}^{r\times r}$ and the Lagrangian

$$\mathcal{L}(X) = \frac{1}{2}\langle X, AX\rangle + \langle B, X\rangle - \frac{1}{2}\langle \Lambda, X^\top X - I\rangle$$

Take any $X \in St(n,r)$ and consider the taylor expansion of the Lagrangian around X':

$$\mathcal{F}(X) = \mathcal{L}(X) = \mathcal{L}(X') + \frac{1}{2}\{\langle (X-X'), A(X-X')\rangle - \langle (X-X'), (X-X')\Lambda'\rangle\} \tag{1.11}$$

$$\geq \mathcal{L}(X) + \frac{1}{2}\langle (X-X'), (X-X')(d_1 I - \Lambda')\rangle \geq \mathcal{F}(X') \tag{1.12}$$

Where the linear term is dropped due to eq. (1.10). To summarize, since Λ satisfies $d_1 I \geq \Lambda$, the Taylor expansion implies

$$\mathcal{F}(X) = \mathcal{L}(X) \geq \mathcal{L}(X') \geq \mathcal{F}(X')$$

for each $X \in St(n,r)$, i.e. X' is a global minimizer of $\mathcal{F}$. On the other hand, suppose $\mathcal{F}(X) = \mathcal{F}(X')$. The condition $d_1 I \geq \Lambda$ implies $(X-X')^\top (X-X') = 0$, i.e. $X = X'$ is unique. □

Generally, eq. (1.10) can fail to hold for any local solution X of $\mathcal{F}$. Algorithmically, we overcome this difficulty by adjusting eigenvalues of the system matrix $A \to A'$ (eq. (1.5)). The modification is motivated by the following remark, for which eigenvalues of A are identical.

Remark 1.1.5 (A special case: $A = I$). When $A = I$, then the function $\mathcal{F}$ in eq. (1.1) reduces to:

$$\mathcal{F}(X) = \frac{1}{2}||X-B||^2 - \frac{1}{2}||B||^2$$

and its minimizer corresponds to the projection of B on $\mathrm{St}(n,r)$. Write the singular value decomposition of $B = V_B D_B U_B^\top$. When B has r positive singular values, the optimal X is uniquely given by $X = V_B U_B^\top$.

Theorem 1 (Local solutions of eq. (1.5) are global solutions)**.** *Consider eq. (1.5). The smallest nonzero eigenvalue of $P_j^\perp A' P_j^\perp$ is d_1 for $j = 1,2,\dots,r$. Any local solution of $\mathcal{F}'$ is a global solution.*

Proof. Let X be an optimal solution. Then, the first order condition $A'X = X\Lambda + B$ holds. Eigenvalues $\lambda_1,\dots,\lambda_r$ of Λ satisfy

$$P_j^\perp A' P_j^\perp \succcurlyeq \lambda_j P_j^\perp.$$

By remark 1.1.4, we have $\lambda_j \leq d_r = d_1$ for all $j = 1,\dots,r$. From Prop. 1.1.4, any local solution is a global solution, which completes the proof. □

1.2 A Sequential Subspace Method

In this section, we introduce approximate and iterative methods to solve eq. (1.1) (eq. (1.5)), the reweighted version of eq. (1.2)(eq. (1.6)). In theory, and as we show, one can start with an arbitrary initialization to obtain a critical point of eq. (1.2) using projected gradient descent, with the descent direction given by the gradient of eq. (1.5) and projection given by eq. (2.21). However, the empirical rate of convergence depends significantly on the initialization of the embedding matrix X. In order to improve convergence of our method, we first introduce and motivate an efficient method based on Procrustes Analysis Wang and Mahadevan [2008] to approximately compute critical points of the *unscaled* objective ($C = I$). Local convergence is derived for eq. (1.2) and eq. (1.6). Assuming $C = I$, global convergence can be guaranteed for the perturbed problem eq. (1.5).

1.2.1 Efficient approximation via orthogonal Procrustes

Here we propose an efficient way to compute approximate critical points of eq. (1.5). First we solve the canonical eigenvalue problem $\min_X \langle X, AX \rangle$ subject to a constraint on the second moment of X: $X^\top X = I$, yields X are the eigenvectors of A. Second, we appropriately transform the solution so that $X^\top B$ is positive definite (i.e. satisfies a necessary condition for first-order optimality eq. (1.1.3), which we restate below).

Proposition 1.2.1 (Definiteness conditions of $X^\top B$ for eq. (1.1))**.** *Assume $C = I$. Note the first term of F satisfies the invariance $\langle X, AX \rangle = tr(X^\top AX) = tr(\widetilde{X}^\top A\widetilde{X})$, where $\widetilde{X} = XQ$ for any orthogonal $Q \in \mathbb{R}^{2\times 2}$. X is a local minimizer if $X^\top B \succcurlyeq 0$ and symmetric.*

A consequence of this is the following algorithm:

1. Take X to minimize the first term of eq. (1.1)—i.e. the eigenvectors of A
2. Let the SVD of $X^\top B = U_B D_B V_B^\top$ and let $Q = U_B V_B^\top$. Projecting X onto Q decreases the objective of eq. (1.1).

In Chapter 2 of this dissertation, we apply this method to both initialize a subsequent refinement step.

1.2.2 A gradient projection method

In this section, We introduce a projected gradient-based method. With appropriate step size $\alpha > 0$, PGD produces iterates X_t, $t = 1, 2, \ldots$

$$X_{t+1} = [X_t - \alpha g_t]_+$$

Where g_t is given by the gradient of the objective of eq. (A.1)—i.e. $g_t = AX_tC - BC^{1/2}$ and $[X_t]_+$ is the projection onto the manifold

$$\mathcal{M} := \{X : X \in St(n,k),\ X^\top B \geq 0\}$$

We first describe the projection $X = [X_1]_+$ as a composition of two projections; i.e. $[X_1]_+ = [[X_1]_{St}]_{\mathcal{B}} \in \mathcal{M}$:

$$Z = [X_1]_{St} := \arg\min_{Z}\{\|X_1 - Z\|_F : Z \in St(m,r)\} \tag{1.13}$$

$$X = [Z]_{\mathcal{B}} := ZQ,\ Q = \arg\min_{Q}\{\|Z - BQ^\top\|_F : Q \in O_k\} \tag{1.14}$$

In other words, $Z \in St(n,r)$ and $Q \in O_k$ are chosen to minimize the sum

$$\|X_1 - Z\|_F^2 + \|Z - BQ^\top\|_F^2 = \|X_1Q - ZQ\|_F^2 + \|ZQ - B\|_F^2$$

Take the SVD of X_1, i.e. $X_1 = U_X D_X V_X^\top$. Then, the solution to eq. (1.13) is given by $Z = U_X V_X^\top$. Likewise, $X = ZQ$ for some orthogonal matrix Q chosen to maximize $\langle X, B\rangle = X^\top B$.

Proposition 1.2.2 (Projection onto $\mathcal{B}$)**.** *Consider the solution to the following projection:*

$$[X_1]_+ = \arg\min_{X \in St(n,k)}\{\min_{Q}\|X - X_1Q\|_F^2 : X^\top B \geq 0, Q \in O_k\} \tag{1.15}$$

Suppose the singular values of X_1 and B are positive. Then, the minimizer X is uniquely determined by

$$X = [X_1]_+ = U_1U_2V_2^\top$$

where U_1, V_1, V_2 are determined from the two SVDs,

$$U_1\Sigma_1V_1^\top = X_1,\ U_2\Sigma_2V_2^\top = U_1^\top B$$

Proof. The minimizer X in eq. (1.15) is the maximizer of $\max_X\langle X, X_1Q\rangle$. Note that

$$\langle X, X_1Q\rangle = \langle X, U_1\Sigma_1V_1^\top Q\rangle = \langle U_1^\top XQ^\top V_1, \Sigma_1\rangle \leq tr(\Sigma_1)$$

Note two observations: (1) that $U_1^\top XQ^\top V_1$ lies in O_k, and (2) that equality holds if and only if $U_1^\top XQ^\top V_1 = I_k$ for some $XQ^\top V_1 \in \mathrm{St}(n,k)$, i.e. $XQ^\top V_1 = U_1$, and thus,

$$X = U_1 V_1^\top Q.$$

Furthermore, the condition $X^\top B$ is symmetric and positive definite implies a choice of Q that fulfills

$$X^\top B = Q^\top V_1 U_2 \Sigma_2 V_2^\top,\ \textit{i.e.}, V^\top Q = V_2 U_2^\top$$

Finally, note that since the singular values Σ_1, Σ_2 are distinct and positive, U_1 and V_1 are uniquely determined up to column-sign $Q_1 = \mathrm{diag}(\pm 1, \ldots \pm 1)$. Likewise, U_2 and V_2 are uniquely determined up to $Q_2 = \mathrm{diag}(\pm 1, \ldots \pm 1)$. Hence,

$$X = U_1 Q_1 Q_2^\top U_2 Q_2 Q_2^\top V_2^\top = U_1 U_2 V_2^\top$$

is unique. □

Given this result and the following remark, we will derive convergence of the gradient method with the Armijo rule.

Remark 1.2.1. Consider the function $h(X) = [X]_{\mathrm{St}}$ defined on $\mathbb{R}^{n\times k}$ and $X \in St(n,k)$. The differential $\mathbb{D}h$ at X is the linear map given by

$$\mathbb{D}h(X)[T] - \lim_{\alpha\to 0} \alpha^{-1}(h(X+\alpha T) - h(X)) = (I - XX^\top)T + (-1/2)X(T^\top X - X^\top T)$$

for each $T \in \mathbb{R}^{n\times k}$. When $X \in \mathcal{M}$ and $T = -(AXC - BC^{1/2})$, then

$$\langle T, \mathbb{D}h(X)[T]\rangle = \|(I - XX^\top)T\|_F^2.$$

Proposition 1.2.3. *Let $d_t = -(AX_tC - BC^{1/2})$. Let $\{X_t\}$ be a sequence generated by the gradient* *projection method*

$$X_{t+1} = [x_t + \alpha_t d_t]_+$$

where α_t is chosen according the Armijo rule. Then, every limit point of $\{X_t\}$ is a stationary point.

Proof. The proof is motivated by the proof by contradiction of Prop. 1.2.1 in Bertsekas [1999]. Let $\mathcal{G}(X) = (I - X^\top X)(AXC - BC^{1/2})$ be the projected gradient of the objective of A.1, F at X. We define α given by the Armijo rule—i.e. let $s > 0$, $\sigma \in (0,1)$ and $\beta \in (0,1)$. $\alpha_t = \beta^{m_t} s$, where m_t is the first nonnegative integer m for which

$$F(X_t) - F([X_t + \beta^m s d_t]_+) \geq -\sigma \beta_t s \langle \mathcal{G}(X_t), d_t \rangle$$

Suppose $\hat{X} \in \mathcal{M}$ is a limit point of $\{X_t\}$ with $||\mathcal{G}(\hat{X}|| > 0$. By definition, $\{F(X_t)\}$ is monotonically nonincreasing to $F(\hat{X})$., i.e. $F(X_t) - f(X_{t-1}) \to 0$. By definition, since the α_t, the step sizes are generated via the Armijo rule, a_t satisfies

$$\begin{aligned} F(X_t) - F(x_{t+1}) &\geq F(X_t) - F([X_t + \alpha_t d_t]_+) \\ &\geq -\sigma \langle \mathcal{G}(X_t), d_t \rangle = \sigma \alpha_t ||\mathcal{G}(X_t)||_F^2 \end{aligned} \tag{1.16}$$

Let $\{X_t\}_{\mathcal{T}}$ be a subsequence converging to $\hat{X} \in \mathcal{M}$ Since

$$\lim_{t \to \infty} \sup -\langle \mathcal{G}(X_t), d_t \rangle = ||\mathcal{G}(\hat{X})||^2 > 0,$$

eq. (1.16) implies $\{\alpha_t\}_{\mathcal{T}} \to 0$. From Armijo's rule, for some $t' \geq 0$, the inequality

$$F(X_t) - F([X_t + \alpha_t \beta^{-1} d_t]_+) < -\sigma \alpha_t \beta^{-1} \langle \mathcal{G}(X_t), d_t \rangle \tag{1.17}$$

holds for all $t \geq t'$. By taking a subsequence $\{d_t\}_{\mathcal{T}'}$ of $\{d_t\}_{\mathcal{T}}$ such that $\{d_t\}_{\mathcal{T}'} \to d'$ and $X_t \to X'$, applying the mean value theorem to the left hand side of eq. (1.17), we have that

$$\begin{aligned}
&-\langle L([X_t+\alpha_t' d_t]_+)-B, \mathbb{D}([X_t+\alpha_t' d_t]_+)[d_t]\rangle \\
&= (\alpha_t\beta^{-1})^{-1}(F(X_t)) - F([X_t+\alpha_t\beta^{-1}d_t]_+) \\
&< -\sigma\langle \mathcal{G}(X_t), d_t\rangle
\end{aligned}$$

for some $\alpha_t' \in [0, \alpha_t\beta^{-1}]$. Taking the limit as $k \to \infty$, we have that $\alpha_t' \to 0$ and $\mathbb{D}([X_t+\alpha_t' d_t]_+)[d_t] \to \mathcal{G}(X')$, which implies

$$-\langle \mathcal{G}(X'), d'\rangle \leq -\sigma\langle \mathcal{G}(X'), d'\rangle, \text{ i.e. } -(1-\sigma)\langle \mathcal{G}(X'), d'\rangle \leq 0$$

Since $\sigma < 1$, it follows that

$$-\langle \mathcal{G}(X'), d'\rangle = ||\mathcal{G}(X')||_F^2 \leq 0$$

which contradicts the non-stationarity of X'. Hence, the limit point $\hat{X}$ is a stationary point. □

1.2.3 A sequential subspace method

Here we introduce an efficient iterative method capable of global convergence to high-quality stationary points. We apply this method to refine the approximate solutions produced using the method introduces in Section 1.2.1. Motivated by the similarity between eq. (2.22) and standard trust-region subproblems, we apply the framework of SSM. In the $k = 1$ and $C = I$ case, SSM has been applied to Trust-Region sub-problems with remarkable empirical results Hager [2001] and robust global convergence guarantees Hager and Park [2005], Erway et al. [2009], even for so-called degenerate problems. At a high level, SSM-based algorithms generate a sequence of iterates X_t by solving a series of rescaled quadratic programs (of the same form as eq. (2.22)) in subspaces of dimension much smaller than that of the original problem.

At step t, we introduce a subspace S_t derived from the current iterate X_t, the gradient of the objective of Prob. eq. (2.22) $g_t = AX_tC - BC^{1/2}$, an *SQP* (i.e. Newton's method applied to the first-order optimality system X_t) iterate X_{sqp} derived in Prop. 1.2.4, and the principal eigenvectors of A. The Sequential Quadratic Programming (SQP) framework Nocedal and Wright [1999] is applied to compute X_{sqp}. In theory incorporating the Newton direction as the descent direction can dramatically speed up convergence to a stationary point.

Following the principle of SQP, we introduce the SQP direction Z according to the linearization of eq. (2.24), the first-order conditions of eq. (2.22):

$$(AZC - Z\Lambda) - X\Delta = E := BC^{1/2} - (AXC - X\Lambda)$$
$$X^\top Z = 0$$

Proposition 1.2.4 (SQP iterate of the Lagrangian of eq. (2.23))**.** *Assume Λ is symmetric. Let $P^\perp = I - X^\top X$ be the projection onto the orthogonal complement of the column space of X and $\Lambda C^{-1} = U diag([\lambda_1, \dots, \lambda_k])U^{-1}$ be the eigenvector decomposition of ΛC^{-1}. The Newton direction Z of X via the linearization of the first-order conditions is*

$$Z = OU^\top, \tag{1.18}$$

where each column of O, $o_j = (P^\perp A P^\perp - \lambda_j P^\perp)^\dagger BC^{-1} u_j$.

Proof. Recall the FOC and its associated linearization with respect to descent directions of X, Λ; (Z, Δ):

$$\begin{aligned}&(AZC - Z\Lambda) - X\Delta = E := BC^{1/2} - (AXC - X\Lambda)\\&X^\top Z = 0\end{aligned}$$

Applying the projection $P^{\perp} = I - XX^{\top}$ eliminates the $X\Delta$ term:

$$PAZ - Z\Lambda C^{-1} = PAPZ - ZU\text{diag}([\lambda_1, \ldots, \lambda_k])U^{-1} = PEC^{-1}$$

Equivalently,

$$PAPZU - ZU\text{diag}([\lambda_1, \ldots, \lambda_k]) = PEC^{-1}U.$$

Let $O = ZU = [o_1, \ldots, o_k]$ lie in the range of P. Then,

$$PAw_j - \lambda_j = PEC^{-1}u_j, \text{ so } o_j = (PAP - \lambda_j P)^{\dagger} EC^{-1}u_j. \square$$

Note that λ_j appears in the computation of the pseudo inverse of $P^{\perp}AP^{\perp} - \lambda_j P^{\perp}$. To ensure that $P^{\perp}AP^{\perp} - \lambda_j P^{\perp}$ is positive semi-definite, we introduce a *safeguard* to adjust λ_j manually.

Proposition 1.2.5 (Safeguard for λ_j)**.** *λ_j are chosen to be the eigenvalues of ΛC^{-1}. Let*

$$P_i^{\perp} = P^{\perp} + x_i x_i^{\top} \quad \forall i = 1, 2, \ldots, k$$

Let μ_i be the smallest eigenvalue of $P_i^{\perp}$. Then, $\mu_i \geq \lambda_i$ holds, i.e., we can use μ_i as a safeguard for λ_i. Hence, the "safe" update to λ_j is

$$\tilde{\lambda}_j = \min(\mu_j - \epsilon, \lambda_j)$$

We note that the safeguarding λ_j introduces additional computational cost to our algorithm. In practice, we observe that safeugarding is not typically necessary. In the applications presented later in this dissertation we forgo safeguarding to save computation.

Remark 1.2.2 (Lagrange multipliers). After each update $X \to X + \alpha Z$, we take a projection $[]_+$ to generate X_{k+1}. Hence, the update of Λ does not benefit from the linearaization given in Prop. 1.2.4. For simplicity, we directly employ X_{k+1} to update Λ_{l+1} via the following least squares estimates estimate of the first-order condition:

$$\Lambda_{k+1} = \arg\min_{\Lambda} ||AX_{k+1}C - BC^{1/2} - X_{k+1}\Lambda||^2$$

That is,

$$\Lambda_{k+1} = X_{k+1}^{\top}X_{k+1}\Lambda = X_{k+1}^{\top}(AX_{k+1}C - BC^{1/2})$$

Algorithm 1. SQP Update

Input: System matrix A, affine term B, intermediate feasible iterate X_t, scaling term C
Output: $j-th$ columns of Newton updates—Δ_j, Z_j

```
function SQP($A, \Lambda_t, B, X_t$)
    $\Lambda_t = X_t^{\top}(AX_tC - BC^{1/2})$
    $U\text{diag}([\lambda_1, \ldots \lambda_k])U = \Lambda C^{-1} = C^{-1/2}\Lambda_t C^{-1/2}$
    init $O$, $P^{\perp} = I - X^{\top}X$
    for $j \in [k]$ do
        $o_j = (P^{\perp}AP^{\perp})^{\dagger}BC^{-1}u_j$
    end for
    return $OU^{\top}$
end function
```

Algorithm 1 presents the detailed steps involved in the computation of the Newton directions, i.e. Prop. 1.2.4. In Section 1.2.4, we discuss its computational cost.

Although stationary points can be recovered via an iterative project-descent procedure using gradient-based or SQP-based descent directions, we introduce the Sequential Subspace Method (SSM) in Algorithm 2: a computationally friendly algorithm to address scalability with respect to large problems. Our development of SSM is inspired by the 1-dimensional algorithm of Hager [2001], originally proposed to solve large trust-region subproblems. In other words, instead of solving eq. (2.19) directly as previously mentioned, we instead solve a sequence of quadratic programs in subspaces of much smaller dimension relative to the size of the graph (the dimensions of A). SSM generically involves repeating the following pair of steps:

1. Compute the SQP direction $Z = \mathrm{SQP}(A,\Lambda,B,X)$ as defined in Prop. 1.18 and described in Algorithm 1, line 5. Let V be the orthogonal matrix consisting of columns in S (Algorithm 2, lines 6 and 7), where

$$S = \mathrm{span}(X_t, Z_t, v, g_t). \tag{1.19}$$

2. SSM generates an approximation of (X,Λ) and an approximation of the smallest pair of eigenvalues σ and eigenvectors v of A in the subspace S,

$$[X,\Lambda,v,\sigma] = \mathrm{SSM}(A,B,S)$$

consider the approximation $X = V\widetilde{X}$ for some $\widetilde{X}$. Compute

$$\min_{\widetilde{X}\in\mathrm{St}(\widetilde{n},k)} F_S := \min_{\widetilde{X}} F(\widetilde{X}; V^\top AV, V^\top B). \tag{1.20}$$

Note that eq. (1.20) is solved using the Projected Gradient Method, where the projection is given by eq. (2.21): $[X]_+ = UV^\top$. We update Λ according to the least-squares estimate derived from the first order condition in eq. (2.24) $\Lambda = X^\top(AXC - BC^{1/2})$.

The proof of convergence follows from the associated result for the Projected Gradient Method, Prop. 1.2.3—i.e. applying the Projected Gradient Method with step sizes chosen according to the Armijo rule ensures that any limit point X_* is a stationary point, when $d_t = -(AX_tC - BC^{1/2}) \in S_t$. Let V_t be an isometry, consisting of vectors in S_t computed via a QR-factorization. Let $A_t := V_t^\top AV_t$ and $B_t = V_t^\top B$. Then, $F(\widetilde{X}; A_t, B_t)$ be the corresponding objective in S_t. SSM computes $X_{t+1} = V_t\widetilde{X}$, where

$$\widetilde{X} := \arg\min_{\widetilde{X}} F(\widetilde{X}; A_t, B_t)$$

Algorithm 2. Sequential Subspace Minimization

Input: System matrix A, unit vector v
Output: Solution X

```
function SSM($A, v$)
    Initialize $X$ to Proj($U$), $U = [u_1, \ldots, u_k]$, where $u_k$ is the eigenvector of $A$ corresponding to the $i$-th
    smallest nonzero eigenvalue (Sec 1.2.1).
    while not converged do
        $Z \leftarrow SQP(A, \Lambda, B, X, v)$                                ▷ Eq. 1.18 & Algorithm 1
        $\mathcal{S} \leftarrow \text{span}(X_t, Z_t, v, g_t)$
        $V \leftarrow QR(\text{col}(S))$
        $\widetilde{A} \leftarrow V^\top A V$, $\widetilde{B} \leftarrow V^\top B$
        $\widetilde{X} \leftarrow \min_{X:X^\top X=I} F(X; \widetilde{A}, \widetilde{B})$                   ▷ Solve Eq. 2.22 in $S$
        $X_t \leftarrow V^\top \widetilde{X}$                                         ▷ Lifted coordinates
        $t \leftarrow t+1$
    end while
    return $X_t$
end function
```

Note that the sequence $\{X_1, \ldots, X_t, \ldots\}$, with $X_{t+1} \in V_t$ monotonically reduces F with respect to t:

$$\begin{aligned} F(\widetilde{X}; A_t, B_t) &= \frac{1}{2}\langle X_{t+1}, AX_{t+1}C - 2BC^{1/2}\rangle \\ &\leq \min_{\widetilde{X}}\{\frac{1}{2}\langle V_t\widetilde{X}, AV_t\widetilde{X}C - 2BC^{1/2}\rangle = \frac{1}{2}\langle \widetilde{X}, A_t\widetilde{X}C - 2B_tC^{1/2}\rangle = F(\widetilde{X}; A_t, B_tC^{1/2})\} \\ &\leq \frac{1}{2}\langle X_t, AX_tC - 2BC^{1/2}\rangle = F(X_t; A, B) \end{aligned}$$

For each t, since the columns of X_t and $AX_tC - BC^{1/2}$ lie in S_t, the iterations of the gradient projection method with Armijo rule lie in S_t, and the sequence with decreasing objective reaches a stationary point $\widetilde{X}$, which ensures that the first order condition

$$AX_*C - BC^{1/2} = X_*\Lambda_*$$

holds for some matrix $\Lambda_* \in \mathbb{R}^{k\times k}$, given by

$$\Lambda_* = X_*^\top(AX_*C - BC^{1/2}) = \lim_t X_t^\top(AX_kC - BC^{1/2})$$

In the case $C = I$, the following states that the inclusion of $[v_1, \ldots, v_k]$ in S_t improves the quality of the stationary point X_*, characterized by the eigenvalues of Λ.

Proposition 1.2.6 (Convergence of SSM for eq. (1.1))**.** *Assume $C = I$. Let $X_* := [x_1, \ldots, x_k]$ be a stationary point generated from SSM. Then,*

$$AX_* - X_*\Lambda_*$$

Let $\lambda_1, \ldots, \lambda_k$ be the eigenvalues of Λ_* and let the eigenvalues of A be $d_1 \leq d_2 \leq \ldots \leq d_n$. Then, $\max\{\lambda_1, \ldots, \lambda_k\} \leq d_k$.

Proof. Let $X_t = [x_{1,t}, \ldots, x_{k,t}]$ be a global minimizer in V_{t-1}. let $Y_t := [y_{1,t}, \ldots, y_{k,t}] = V_{t-1}^\top [x_{1,t}, \ldots, x_{k,t}]$. Then,

$$L_{t-1}Y_t - B_{t-1} = Y_t\Lambda_t$$

holds for some Λ_k with eigenvalues $[\lambda_{1,t}, \ldots, \lambda_{k,t}]$. In addition, since $Y_tB_{t-1} = X_tB$, then X_tB is positive semidefinite and symmetric. As $t \to \infty$, $X_*^\top B$ is also positive semidefinite and symmetric. Additionally, let

$$P_j^{\perp,(t)} = I - \sum_{i \in \mathcal{R}-j} y_{i,t}y_{i,t}^\top.$$

The second order condition implies

$$P_j^{\perp,(t)} A P_j^{\perp,(t)} - \lambda_{j,t} P_j^{\perp,(t)} \geq 0.$$

Consider the optimality of $x_{1,t}$. Let $\phi_{1,t}$ by a unit vector orthogonal to $[x_{2,t}, \ldots, x_{k,t}]$ in $span\{v_1, \ldots, v_k\}$.

Then $P_j^{\perp,(t)} V_{t-1}^\top \phi_{1,t} = V_{t-1}^\top \phi_{1,t}$ holds and the second order condition yields

$$\begin{aligned} 0 &\leq \langle V_{t-1}^\top \phi_{1,t}, (P_j^{\perp,(t)} A P_{j,t}^\perp - \lambda_{j,t} P_{j,t}^\perp) V_{t-1}^\top \phi_{1,t} \rangle \\ &= \langle \phi_{1,t}, (L - \lambda_{1,t} I) \phi_{1,t} \rangle \\ &\leq (\min_i d_i - \lambda_{1,t}) \|\phi_{1,t}\|^2, \end{aligned}$$

Which implies $\lambda_{1,t} \leq \min_i d_i$. As $t \to \infty$, a subsequence of $\{x_{1,t}, \ldots, x_{k,t} : t\}$ converges to $[x_1, \ldots, x_k] \in \mathcal{M}$ and $\lambda_{1,t}$ converges to λ_1 Hence, $\lambda_1 \leq \min_i d_i$. Likewise, $\lambda_j \leq \min_i d_i$ by the optimality of $x_{j,t}$ for $j = 2, \ldots, k$. □

The following is the result of Proposition 1.2.6

Theorem 2 (Global convergence of SSM for eq. (1.5))**.** *A limit X_* of $\{X_1, X_2, \ldots, X_t, \ldots\}$ generated by SSM is a global minimizer of eq. (1.5).*

In Chapter 3, we provide a numerical evaluation of the proposed SSM algorithm on large-scale engineering and machine learning problems against standard algorithms and existing open-source frameworks.

1.2.4 Complexity analysis of SSM

In this section, we discuss the computational cost of SSM, dominated by the SQP routine to compute the SQP directions. We claim that the per-iteration complexity of ssm is T_{matrix}, where T_{matrix} is the complexity of each call to a matrix solver. For example, in the case of Laplacians, and more generally *M-matrices*, there exist nearly linear-time solvers Spielman and Teng [2014]. Additionally, the QR-decomposition of $\text{col}(S)$ takes time linear in n. We further note that the SSM procedure itself exhibits quadratic rates of convergence for nondegenerate problems and global convergence with *at least* linear rates, even when the problem exhibits certain degenerate characteristics Hager and Park [2005].

Regarding the approximate method, note that computing the principal eigenpairs of A, specifically the eigenvectors U is typically a cubic operation for dense matrices, can be done quite efficiently for sparse symmetric matrices, such as graph laplacians. Specifically, the locally-optimal block preconditioned conjugate gradient (LOBPCG) method Knyazev [2001] is one state of the art method that we adopt in the numerical experiments presented in Chapter 2. Typically, these methods necessitate matrix-vector products Ax leading to efficient implementations of Ax in time linear in e, $O(e)$ the number of nonzero entries of A. The complexity of each eigen-pair computation for A is then $O(ret)$, where t is a constant equal to the average number of iterations required for the LOPCG algorithm (note t depends on the spectral properties of A and is independent of its size, i.e. n). And aligning U with the observed labels necessitates an SVD of the matrix $U^\top B$ and the projection of U onto Q, both negligeable if r is small.

Generic quadratic programs are NP-hard Pardalos and Vavasis [1991], i.e. it takes super-polynomial time to solve QPs optimally. In the convex case, there are polynomial time interior point algorithms Dikin [1967]. Also, there are approximation algorithms that return local solutions of nonconvex QPs in polynomial time Hager [2001]. Our method falls into the category of algorithms that guarantee local, or block-globally optimal solutions for problems of the form eq. (1.1), eq. (1.2) and globally optimal solutions for associated problems with system matrices that have been appropriately adjusted, of the form eq. (1.5).

Chapter 1, in part is a composition of two works, one of which has been submitted for publication—Revisiting Semi-Supervised Laplacian Eigenmaps via Alignment, 2023 and one currently being prepared for submission for publication of the material: Minimizing a Quadratic over Stiefel Manifolds.

Chapter 2

Applications to Chip Placement and Graph-Based Semi-Supervised Learning

In this chapter, evaluate the framework and algorithm described in Chapter 1 on three problems engineering and machine learning. First, we apply our framework and method as a preprocessing step for VLSI and PCB layout optimization tools. In these contexts, the system matrix A corresponds to an *graph Laplacian matrix*. In particular, VLSI layout problems consider Laplacians with up to several million rows and columns and more than a million nonzero entries. Next, we demonstrate that *graph-based semi-supervised learning* can be formulated as a *rescaled* quadratic optimization problem on the Stiefel Manifold, and that our method can be to solve it, achieving state of the art classification results accross a wide spectrum of label rates. Namely, the *low-label rate regime* is particularly notable as this regime is close to degenerate in a certain sense.

2.1 VLSI Placement Initialization and Optimization

Given a circuit and a region, the placement problem is to assign each circuit module to a specific location in the region. Most state-of-the-art layout algorithms for large-scale VLSI placement rely on solving non-linear problems using iterative first-order optimization algorithms Kahng et al. [2005], Lin et al. [2019], Lu et al. [2015a], Cheng et al. [2018], Lu et al. [2015b]. As a consequence, there are typically few guarantees regarding the convergence of these methods to optimal, or even good, coordinate assignments in a limited time frame and initialization of the variables plays a critical role Lu et al. [2015a]. Despite the importance of initialization, existing methods for placement initialization are primarily based on naive heuristics—including minimizing wirelength without second-order constraints Lu et al. [2015b,a], Kahng et al. [2005], uniformly assigning the cell coordinates to the origin, or assigning coordinates to small random values Gu et al. [2020], Lin et al. [2019].

In this section, we address the following question:

> *Is it possible to improve upon random initialization for large-scale placement engines?*

We investigate a novel fixed node-aware formulation and describe an efficient algorithm to solve it. More concretely, we formulate initialization as a Quadratically Constrained Quadratic Optimization Problem (QCQP) with sphere constraints. Our formulation is aware of fixed nodes via a decomposition of the netlist-graph. Although the QCQP is non-convex, we propose an algorithm that can recover local and block-globally optimal (under certain assumptions) solutions. We validate our technique by demonstrating scalability and convergence to superior post-detailed placement solutions compared to min-wirelength and random initializations using an open source placement flow Gu et al. [2020], Lin et al. [2019]. Furthermore, we propose a statistical test to quantify the preservation of local structures derived from the initialization through global placement.

2.1.1 Contributions

Our contributions are summarized below.

1. We introduce a novel formulation of global placement initialization as a sphere-constrained quadratic programming problem, an extension of a classic Rayleigh Quotient problem Hager [2001] and devise a novel algorithm to solve it.

2. We propose a way to exploit the structure of the QCQP to improve the efficiency of

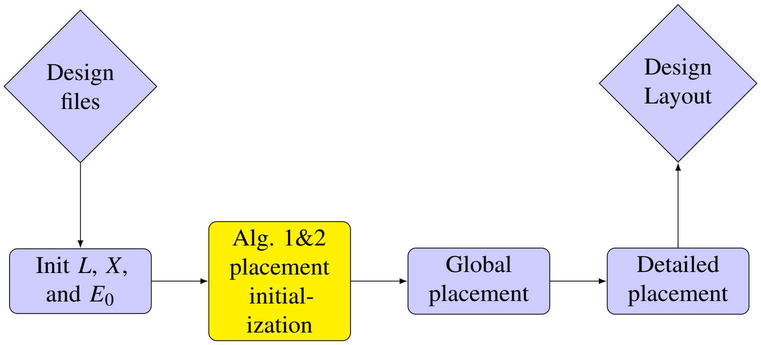


Figure 2.1. Placement flow. Our proposed method is a "placement initialization" stage, highlighted in yellow.

optimization by iteratively solving the problem in a sequence of carefully chosen subspaces.

3. We adapt our approach via iterative reweighting to facilitate direct minimization of Half-Perimeter Wirelength (HPWL).

4. We perform a comparison between various initialization schemes for analytic placement with fixed macros.

2.1.2 Preliminaries

Number of components	$n, n_{\text{free}}, n_{\text{fixed}} \in \mathbb{R}_+$
Placement coordinates	$x, y \in \mathbb{R}^n$
Adjacency, Degree, & Laplacian matrices	$A, D, L \in \mathbb{R}^{n \times n}$
Linear offset terms	$b, d \in \mathbb{R}^n, E_0 = [b : d] \in \mathbb{R}^{n \times 2}$
Cell volumes	$v \in \mathbb{R}^n_+, G = \text{diag}(v) \in \mathbb{R}^{n \times n}_+$
Cell area constraints	$c_i, i \in \{1,2,3,4,5\} \in \mathbb{R}_+$
Lagrange multipliers	$\Lambda \in \mathbb{R}^{2 \times 2}$
Newton update direction	$Z \in \mathbb{R}^{n \times 2}$

Figure 2.2. Notation

Let $x, y \in \mathbb{R}^n$ be vectors corresponding to the coordinates of n components such that the i-th component has coordinates encoded in the i-th row of $[x : y]$; $[x : y]_i$. We aim to assign coordinates so that the resulting layout has small cumulative wirelength.

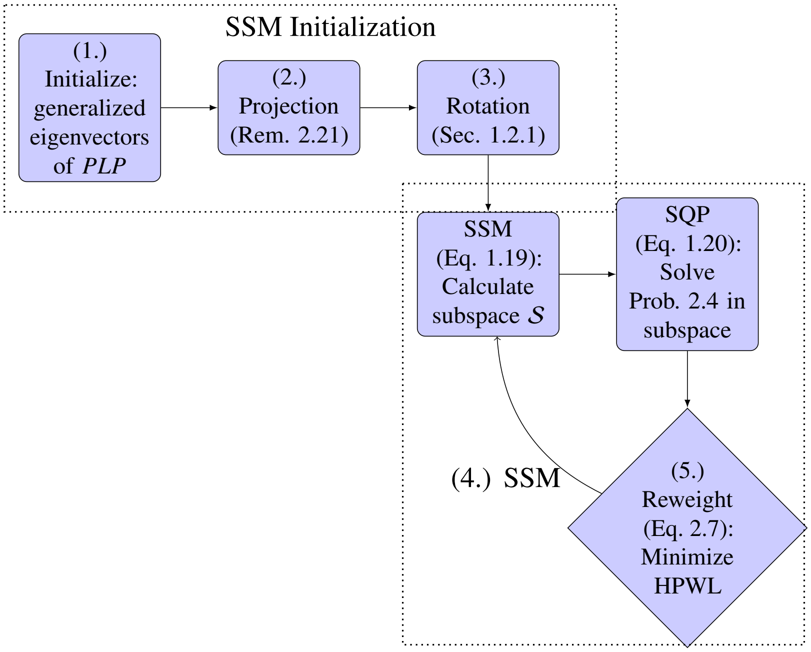

SSM Initialization
(1.) Initialize: generalized eigenvectors of *PLP*
(2.) Projection (Rem. 2.21)
(3.) Rotation (Sec. 1.2.1)
SSM (Eq. 1.19): Calculate subspace $\mathcal{S}$
SQP (Eq. 1.20): Solve Prob. 2.4 in subspace
(4.) SSM
(5.) Reweight (Eq. 2.7): Minimize HPWL

Figure 2.3. QCQP placement initialization with reweighting.

Global analytical placement

Conventional global placement strategies minimize wirelength subject to density constraints. Density constraints are usually integrated into the objective to yield an unconstrained relaxation Cheng et al. [2018], Lin et al. [2019]:

$$\min_{x,y}(\textstyle\sum_{e\in\mathcal{E}} Wl(e;x,y) + \lambda\mathcal{D}(x,y)) \tag{2.1}$$

where $\mathcal{E}$ denotes a set of given nets and $Wl(\cdot;\cdot)$ is a function that takes a net instance e as input and returns the cumulative wirelength and $\mathcal{D}(\cdot)$ is a density penalty. In the context of VLSI placement, the wirelength of a net is commonly modelled with its Half-Perimeter Wirelength (HPWL) or a smooth alternative and $\mathcal{D}$ is a smooth density penalty Lu et al. [2015a].

A typical approach is to represent individual nets as rectangles and to minimize the sum-perimeters over all nets. Repulsion is often applied between overlapping nodes to reduce density. For example, Lin et al. [2019] adopt the smooth and differentiable weighted-average wirelength (WL) model for the wirelength cost Hsu et al. [2013]. The horizontal net-wirelength for net e is given by

$$Wl_x^{(e)} = \frac{\sum_{i\in e} x_i \exp(\frac{x_i}{c})}{\sum_{i\in e} \exp(\frac{x_i}{c})} - \frac{\sum_{i\in e} x_i \exp(-\frac{x_i}{c})}{\sum_{i\in e} \exp(-\frac{x_i}{c})}$$

where c is a parameter that controls the smoothness and approximation error with respect to the HPWL of net e (i.e. $|x_i - x_j|$ for two-pin net $e = (i, j)$). The wirelength of e is:

$$Wl(e; x, y) = Wl_x^{(e)} + Wl_y^{(e)}$$

To model the density term, the placement area is divided into B bins, and the placer seeks to equalize the overlap at each bin via an analogy to an electrostatic system, with cells being modeled as charges, density penalty modeled as potential energy, and density gradient modeled as the electric field.

Overlap constraints are satisfied over the placement process by gradually increasing λ, usually at the cost of increased wirelength. Current state-of-the-art VLSI placement algorithms Lu et al. [2015a], Cheng et al. [2018], Lin et al. [2019] solve Problem 2.1 in this manner.

Global placement initialization

The VLSI placement problem is reduced to a graph layout problem by first collapsing the netlist hypergraph to a component graph via various models (e.g. clique, star, etc.) Viswanathan and Chu [2004]. A matrix-representation of the graph connectivity—the *graph Laplacian* is then derived. The solution to the associated eigenvalue problem approximates the solution to the sparsest cut problem Hall [1970], Alpert and Kahng [1996], and clusters arising out of the vertex-projection into the space spanned by the first nontrivial eigenvalues correspond highly connected components of the graph.

More concretely, we solve a variant of the following problem where x and y are cell coordinates, c_i are constants, v is a vector of cell areas, and L is the graph Laplacian; $L = D - A$, where A is a (weighted) adjacency matrix, and D is the associated degree matrix.

$$\begin{aligned} \min_{x,y}\ x^\top Lx + y^\top Ly \quad \text{s.t. } v^\top x = 0, \qquad v^\top y = 0, \\ x^\top Gx = c_1, \quad y^\top Gy = c_2, \qquad x^\top Gy = c_3 \end{aligned} \tag{2.2}$$

Typically, $G = \operatorname{diag}(v)$. In general, one can recover a reduction to the case $G = I$ via the normalization $[x,y] \leftarrow G^{1/2}[x,y]$, $L \leftarrow G^{-1/2}LG^{-1/2}$ and $[v,b,d] \leftarrow G^{-1/2}[v,b,d]$. Intuitively, the objective is to minimize the weighted squared wirelength of a 2D placement. The linear constraints characterize an origin (i.e. remove translational invariance) and the quadratic constraints spread the layout evenly over the x and y axes (i.e. ensure that the embedding has nonzero constant variance).

Fixed node constraints

Many layouts involve constraints on a subset of the cells—typically large macros and primary input/output pads. We show how such fixed node constraints naturally lead to a decomposition of the x, y and L terms in Eq. 2.2. We denote the coordinates of the fixed nodes x_1, y_1. Likewise, let the movable nodes be x_2, y_2 Then, we can express x, y, and L and the parameters v and G in terms of these indices: $L = \left[\begin{smallmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{smallmatrix}\right]$, with $x_1 \in \mathbb{R}^{n_\text{fixed}}$, $x_2 \in \mathbb{R}^{n_\text{free}}$, $L_{22} \in \mathbb{R}^{n_\text{free} \times n_\text{free}}$, and $L_{12} \in \mathbb{R}^{n_\text{free} \times n_\text{fixed}}$. $x = [x_1, x_2]^\top$ (likewise for y). By considering fixed-node terms (i.e. x_1 and y_1) as constants, Problem 2.2 may be re-written (ignoring constants):

$$\begin{aligned} \min_{x_2,y_2} x_2^\top L_{22} x_2 + y_2^\top L_{22} y_2 + \quad & 2b^\top x_2 + 2d^\top y_2 \\ \text{s.t. } v_2^\top x_2 = c_1', \qquad & v_2^\top y_2 = c_2', \\ x_2^\top x_2 = c_3', \qquad & y_2^\top y_2 = c_4', \qquad x_2^\top y_2 = c_5' \end{aligned} \tag{2.3}$$

with $b = L_{12}x_1$, $d = L_{12}y_1$.

Equivalently, by re-writing the objective defined in Eq. 2.3 (for brevity, writing L_{22} as "L" and v_2 as v). Let $X_2 = [x_2, y_2]$ and $E_0 = [b,d] \in \mathbb{R}^{n_\text{free} \times 2}$ and $X_1 = [x_1, y_1] \in \mathbb{R}^{n_\text{fixed} \times 2}$.

Let $[c_1', c_2']^\top = -r$, where $r := (v_1^\top X_1)^\top$. To eliminate the linear constraint $v^\top X_2 = -r^\top$, we introduce two adjustments: first, let $(X)_i = (X_2)_i + \frac{1}{w} r^\top$ denote a row-wise centering transformation with respect to the fixed nodes, where w is a scale factor proportional to v_2. This yields the constraint $v^\top X = 0$ and implies the quadratic constraint

$$C = \begin{bmatrix} c_3' & c_5' \\ c_5' & c_4' \end{bmatrix} = \left[\begin{smallmatrix} c_1 & c_3 \\ c_3 & c_2 \end{smallmatrix}\right] - X_1^\top X_1 - \frac{1}{w} r r^\top.$$

Second, assuming that v is normalized to be a unit vector, let $P = I - vv^\top$ be the projection onto the subspace orthogonal to vector $v \in \mathbb{R}^{n_{\text{free}}}$, i.e., $v^\top (PX) = [0,0]$. Without loss of generality, replacing E_0 with $P(E_0 - L\frac{1}{n_{\text{free}}}\mathbf{1}r^\top)$, we have $v^\top E_0 = [0,0]$.

$$\min_X \{F(X) = \operatorname{tr}(X^\top (PLPX + 2E_0))\} \tag{2.4}$$

subject to $X^\top X = C$

Motivation and high-level flow

Our motivation for expressing initializations to Prob. 2.1 with Prob. 2.2—i.e. as a QCQP with sphere constraints—is derived from two observations assuming graph-models of netlists: (1.) if the $Wl(\cdot)$ corresponds to the *squared* wirelength, its minimization is equivalent to minimizing a quadratic form defined on a graph Laplacian. And if the $Wl(\cdot)$ corresponds to the *half-perimeter* wirelength, its minimization may be expressed as a sequence of quadratic problems of the same form as Prob. 2.2. For example, we describe such a method in Sec. **??**. (2.) a quadratic equality—a *sphere* —constraint implies constant variance and satisfaction of density-constraints assuming a uniform grid.

We highlight the high-level flow of our framework in Fig. 2.3:

1. **Eigenvector initialization:** The eigenvectors of L_{22}, which correspond to the minimum-squared wirelength solution are computed (Eq. 2.4).

2. **Eigenvector projection:** These eigenvectors are projected to satisfy the linear and quadratic constraints (Prop. 2.1.1).

   Proposition 2.1.1 (Projection)**.** *Let X_1 be an intermediate solution and $C_1 := X_1^\top X_1$ and $C \succ 0$.*

   $$\text{\textit{The projection of} } X_1, [X_1]_+ := \arg\min_X \{F(X) = \|X - X_1\|_F^2 \\ = tr(C) + tr(C_1) - 2\max\langle X, X_1\rangle\}$$

   s.t. $X^\top X = C$. Take the Singular Value Decomposition (SVD) of $C^{1/2}C_1^{1/2}$, $U\Sigma V^\top = C^{1/2}C_1^{1/2}$.

   Then the minimizer $X = [X_1]_+$ is given by

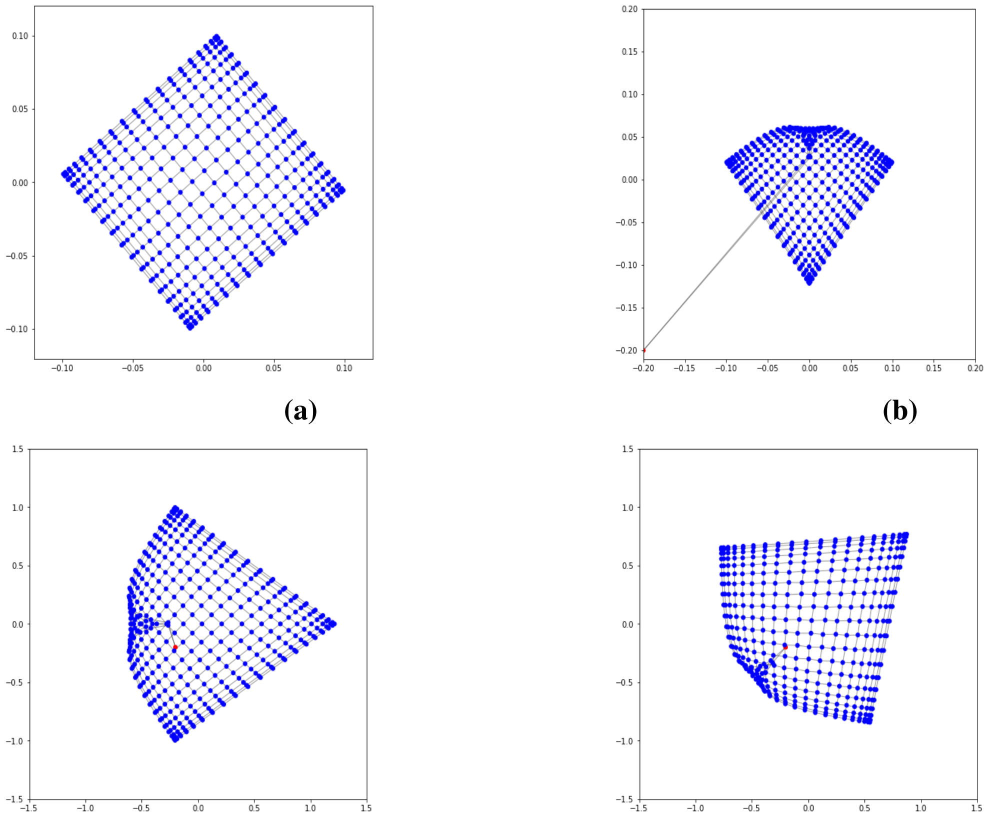
   $$X = X_1 C_1^{-1/2} U V^\top C^{1/2} \tag{2.5}$$

3. **Eigenvector rotation:** An orthogonal transform is applied to the projected eigenvectors to minimize the distance between free and fixed components (Prop. 1.1.3).

4. **Sequential subspace method (SSM):** From these coordinates, an iterative projected-subspace-descent algorithm is applied which results in convergence to a local / block-globally optimal solution (Ch. 1).

5. **Iterative net reweighting:** During iterative descent, L is adjusted (*reweighted*) in order to find a min-HPWL coordinate assignment (Sec. 2.1.2).

In the following section, we describe a sequential subspace method for solving Prob. 2.3. We then show that one can easily adapt this method to facilitate direct minimization of HPWL.



(a) **(b)**

(c) **(d)**

Figure 2.4. Eigenvector method and projection. (**a**): Eigenvectors of full Laplacian L (**b**): Eigenvectors of reduced Laplacian L, ignorant of fixed node (denoted in red) (**c**): Projected eigenvectors of L (Prop. 1) (note the axis scale). (**d**): Orthogonal transform applied to projected eigenvectors (Prop. 2).

Minimization of HPWL via re-weighting

In this section, we show how our method may be adapted to facilitate direct minimization of HPWL. A similar method was adopted by the GORDIAN-L cell placement tool Alpert et al. [1998]. Inspired by asymptotically optimal algorithms for lasso-type regression problems Candès et al. [2007], Wipf and Nagarajan [2010], Daubechies et al. [2010], Chartrand and Yin [2008], we solve an equivalent ℓ_1 minimization problem by solving a sequence of re-weighted ℓ_2 minimization problems. In particular, we propose an analogous algorithm for the 2-dimensional case. Note that we now consider the following problem:

$$\sum_{i,j\in\mathcal{E}} w_{ij}(|x_i - x_j| + |y_i - y_j|) \tag{2.6}$$

Informally, the objective is upper bounded by the expression

$$\min_{u_{i,j}>0}\max_{v_{i,j}>0}\left\{\sum_{i,j\in\mathcal{E}}(u_{i,j}|x_i - x_j|^2 + \frac{1}{u_{i,j}} + v_{i,j}|y_i - y_j|^2 + \frac{1}{v_{i,j}})\right\}$$

Crucially, the equality holds if and only if $u_{i,j} = |x_i - x_j|^{-1}$ and $v_{i,j} = |y_i - y_j|^{-1}$ and implies a strategy for solving Prob. 2.6 that involves Prob. 2.4 as a sub-problem:

1. For each $u > 0$, $v > 0$, solve Prob. 2.6 with respect to x, y.

2. For each x, y, solve Prob. 2.6 with respect to u, v.

$$u_{i,j} = |x_i - x_j|^{-1}, \quad v_{i,j} = |y_i - y_j|^{-1} \tag{2.7}$$

In practice, we alter the above algorithm in two ways: (1.) following Alpert et al. [1998], a small adjustment to the denominator of each weight for normalization and to address numerical instability in the situation where two nodes overlap—e.g., $u_{i,j} = 1/(\mathcal{W}\sqrt{(x_i - x_j)^2 + \beta})$, where $\mathcal{W}$ is the width of the placement area (2.) instead of solving Prob. 2.6 (step (1.)) to convergence, we perform incremental 1-step updates—i.e., we perform re-weighting *each* iteration of SSM and compute the subsequent subspace with respect to the new re-weighted matrix L and associated E_0. While the concept of iterative re-weighting for optimization has most commonly been applied to ℓ_1 and ℓ_∞ minimization problems, the framework is quite general and a similar procedure motivates minimization of other kinds of norms-based objectives. Future work includes investigating the efficacy of this reweighting scheme for alternative norm-minimization problems (e.g. robust p-norm minimization) in the context of layout.

2.1.3 Experiments

In this section we describe a set of comprehensive experiments on eight VLSI testcases from the ISPD’05 contest suite Nam et al. [2005]. Summary statistics of the testcases are presented in Table 2.4. Our numerical experiments are aimed at establishing the efficacy of our method with respect to post-detailed placement wirelength. We leverage the DREAMPlace Lin et al. [2019] placement engine and substitute the heuristic initialization schemes with our proposed method.

Table 2.1. Design characteristics. n_{free} =#Free cells and n_{fixed} =#Fixed pins. Max Deg, Avg Deg correspond to characteristics of the graph-models of the design netlists.

Design	#Free cells	#Fixed pins	#Nets	Max Deg	Avg Deg
adaptec1	$211k$	$29k$	$221k$	340	4.2
adaptec2	$255k$	$21k$	$266k$	153	3.9
adaptec3	$452k$	$25k$	$467k$	82	4.0
adaptec4	$496k$	$29k$	$516k$	171	3.7
bigblue1	$278k$	$11k$	$284k$	74	4.1
bigblue2	$558k$	$141k$	$577k$	260	3.5
bigblue3	$558k$	$37k$	$1123k$	91	3.4
bigblue4	$2177k$	$170k$	$2230k$	129	3.7

Table 2.2. Post-detailed place metrics. We report cumulative HPWL and runtime of global and detailed placement and legalization using various initializations. We report the percent improvement over random init. in parenthesis. The best result is **bolded**.

Design	Random		Min-wirelength		Projected Eigenvectors			Projected Eigenvectors + SSM		
	HPWL	GP runtime (s)	HPWL	GP runtime (s)	HPWL	GP runtime (s)	runtime (s)	HPWL	GP runtime (s)	runtime / iter. (s)
adaptec1	73.24	84.39	73.23	74.31	70.36 (3.9%)	63.86	93.6	**70.34** (3.96%)	62.42	26.34
adaptec2	82.51	189.46	82.24	172.91	81.68 (1.0%)	164.37	88.2	**81.21** (1.58%)	162.49	22.56
adaptec3	194.12	314.54	193.87	309.88	189.13 (2.5%)	313.29	181.2	**187.95** (3.18%)	314.01	57.78
adaptec4	174.43	371.72	174.16	354.16	171.73 (1.5%)	372.14	168.6	**171.62** (1.61%)	361.37	47.94
bigblue1	89.43	112.64	89.43	107.56	87.32 (2.3%)	94.11	124.2	**87.04** (2.67%)	94.23	45.71
bigblue2	136.69	387.94	136.69	361.75	132.49 (3.0%)	327.14	150.6	**131.37** (3.89%)	321.86	53.56
bigblue3	303.99	1064.63	303.99	1047.66	298.47 (1.8%)	847.03	369.0	**297.31** (2.20%)	849.23	110.63
bigblue4	743.75	1534.11	743.75	1500.70	726.71 (2.2%)	1372.49	1539.6	**724.78** (2.55%)	1293.10	322.32

Algorithm parameters

To produce graph-layouts of IC netlists we adopt a hybrid net model Viswanathan and Chu [2004]—a combination of the clique and star models. Each net is converted to a star or clique-graph depending on the size of the net—i.e. nets with three or fewer pins are modeled as

Table 2.3. HPWL and structure-preservation test statistic for Prob. 2.3 (min-squared objective) and Prob. 2.3 (HPWL objective).

Design	Squared-wirelength		Direct HPWL	
	HPWL	z	HPWL	z
adaptec1	70.34 (3.96%)	0.131 ± 0.046	**70.12** (4.26%)	0.139 ± 0.052
adaptec2	81.21 (1.58%)	0.069 ± 0.031	**81.12** (1.68%)	0.073 ± 0.038
adaptec3	187.95 (3.18%)	0.072 ± 0.041	**186.61** (3.87%)	0.076 ± 0.043
adaptec4	171.62 (1.61%)	0.126 ± 0.057	**170.34** (2.34%)	0.131 ± 0.061
bigblue1	87.04 (2.67%)	0.063 ± 0.039	**85.72** (4.15%)	0.067 ± 0.041
bigblue2	131.37 (3.89%)	0.079 ± 0.037	**130.19** (4.76%)	0.081 ± 0.044
bigblue3	297.31 (2.2%)	0.074 ± 0.041	**296.04** (2.61%)	0.074 ± 0.043
bigblue4	724.78 (2.55%)	0.081 ± 0.053	**723.77** (2.69%)	0.081 ± 0.054

cliques and nets with four or more pins are modeled as stars, with an associated free *pseudo-pin* variable introduced. To determine v, we first consider the surface area of cells (i.e. $v_i = w_i \times h_i$, where w_i and h_i is the width and height of cell i), scaled such that the distribution is centered about 1. v is then normalized. The c_i are determined according to the free layout space.

Implementation details

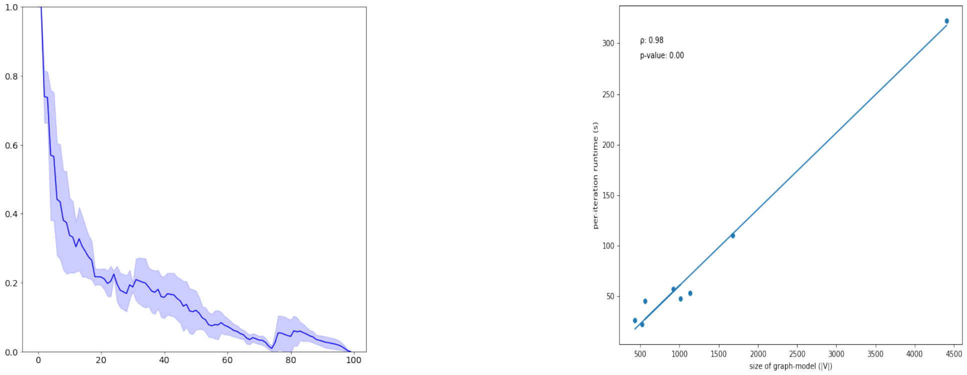
We implemented our algorithms in Python using the JAX framework Bradbury et al. [2018] on a GCP c2-standard-8 machine with 8 virtual CPUs, 32 GB of memory, and a single Nvidia Tesla K80 GPU. In particular, we exploit JAX's capability to vectorize batched computation and compilation to XLA via the *jit* decorator. XLA facilitates hardware acceleration and the entire framework (initialization, global placement, detailed placement / legalization) may exploit GPU and multi-GPU-based parallelism without returning to a Python interpreter.

Numerical results

We applied the proposed method to eight benchmarks from the ISPD'05 contest suite Nam et al. [2005] and measured the cumulative HPWL post-detailed placement. Numerical results are provided in Table 2.2. We find that origin initializations consistently under-perform the other three methods, and that random and min-wirelength exhibit comparable results. However, initialization using the vanilla projected eigenvectors of the reduced Laplacian Chen et al. [2022]

result in superior HPWL—improvement between 1.0% and 3.0% compared to the random and min-wirelength heuristics. Larger gains are achieved when the initialization corresponds to the solution to Prob. 2.3 using SSM without reweighting—between 1.58% and 3.96%. Additionally, improvements in global placement runtime correlate with better initialization. We provide the global placement (DREAMPlace) runtime in Table 2.2. The GP runtime ranges from 62.42s to 1293.10s for the Projected Eigenvectors + SSM method, which is comparable to or less than the other methods.

Reweighted SSM iterations and runtime In Table 2.3, we demonstrate that the directly minimizing HPWL via reweighting yields still further improvements—between 1.68% and 4.76% compared to random and min wirelength initializations. We note that reweighting methods are typically slow to converge Ene and Vladu [2019]. As a consequence, instead of running our algorithm to convergence, we set a hard maximum limit of 100 reweighting / SSM steps. We additionally observe a mean per-iteration wall-time of $26.34-322.32$ and a significant ($\rho = 0.99$, $p = 1.1e-7$) linear correlation with the number of free cells. We plot this trend in Fig. 2.5b. It is likely that further gains could be achieved with a direct method for HPWL minimization.



(a) **(b)**

Figure 2.5. Eigenvector method and projection. (**a**): Mean normalized decay in HPWL of adaptec cases. (**b**): Per-iteration turnaround (seconds) vs. dimension of L_{22}: # free cells + # nets in 10^3 unit.

While the per-iteration runtime of our method is nontrivial, we highlight three key points: (1.) the experiments imply that the proposed QCQP formulation and method can consistently improve placement quality. This evidence incentivises future work to enhance the efficiency of these algorithms—particularly Laplacian solvers to drastically speed up turnaround time, (2.) few iterations are needed to significantly improve the post-detailed placement wirelength (as demonstrated in Fig. 2.5a), (3.) typical placement flows usually involve multiple runs of the global and detailed placement engine to validate different choices of hyperparameters, while our parameter-free initializations need only be computed once.

In Fig. 2.5a, we demonstrate that relatively few iterations are needed to improve the quality of post-detailed placement HPWL. For each testcase, we apply 100 iterations of SSM. Global and detailed placement is performed using each intermediate SSM iterate as the initialization. The HPWL of the post-detailed placement is measured and normalized to lie in the range $[0,1]$. We plot the distribution of normalized post-detailed placement HPWL with the shaded region corresponding to 1 standard deviation in normalized HPWL. We observe that across all testcases, 60% of the improvement in post-detailed placement wirelength is achieved within the first $5-10$ iterations while roughly 80% of the improvement is achieved after the first ~ 20 iterations. Additionally, we emphasize that our method is *parameter free* and yields the same solution across multiple runs. One may only need to generate a single initialization to validate multiple choices of global / detailed placement hyperparameters.

Preservation of initial structure through global placement

In Fig 2.14, we plot intermediate iterations of the global placer, with colors corresponding to clusters of standard cells derived according to physical proximity via Euclidean k-means with $k = 10$. The consistency of the colors (cluster) pre- and post-global placement serves demonstrate that the global placement algorithm preserves the global and local structure induced by the seed layout. Inspired by metrics proposed in Fogaça et al. [2019] to evaluate the quality of a graph partitioning / clustering, we propose to evaluate this hypothesis by proposing a novel two-sample permutation test. We formulate the null (H_0) and alternative (H_a) hypotheses below:

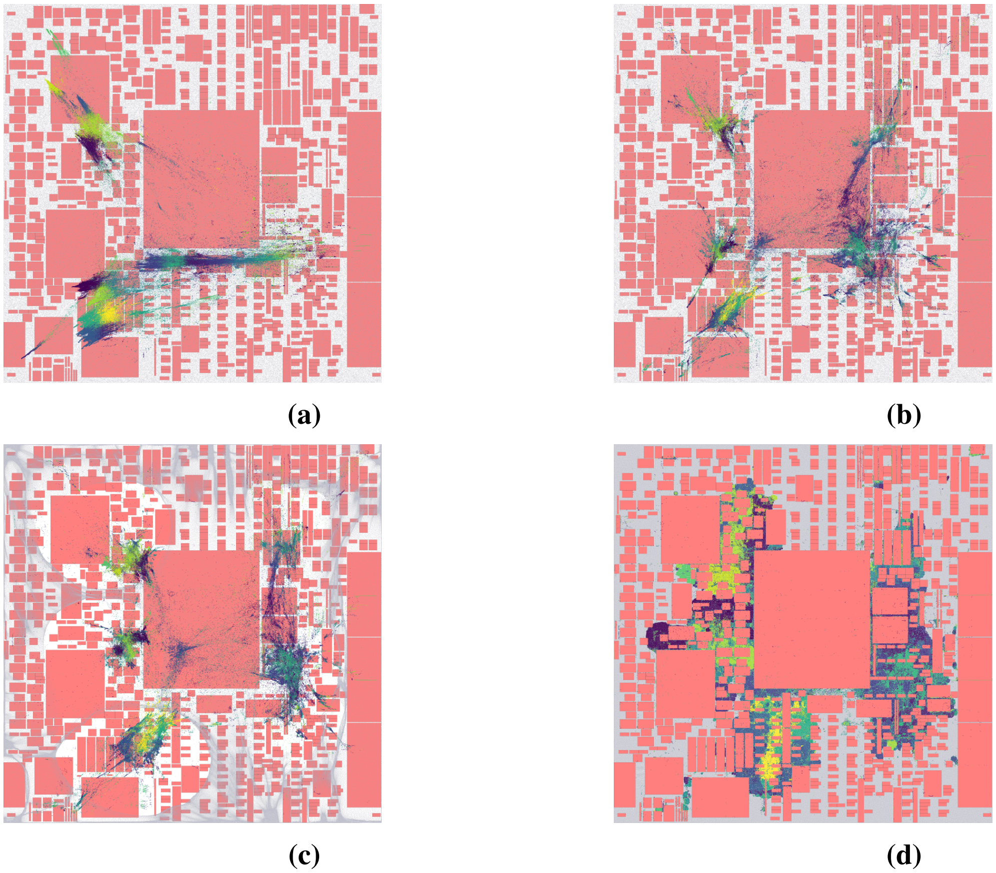


Figure 2.6. Adaptec3 layout. (**a**): Projected eigenvectors for seed layout. Colors denote initial spatial partitions. (**b—d**) Intermediate DREAMPlace results. Note the preservation of cell groups (colors) through global placement.

H_0: *no effect of the initialization on the final layout*

H_a: *there is an effect*

Intuitively, under the null hypothesis, the cells component to any initial *spatial partitioning* (e.g. an arbitrary cell's neighbors) would separate during the global placement process, and a new partitioning *after* global placement would yield very different groups of cells. We consider a partitioning computed based on the initial layout—e.g. we apply Euclidean k-medoids[1] with k = 100. After global placement, we re-partition the final layout using k-means. For each

[1] k-means assigns centers to arbitrary coordinates, k-medoids assigns centers to cells.

centroid-cell c of an initial partition P_c, we find c's partition P'_c in the final layout. The statistic with respect to c is

$$z_c = \frac{|P_c \cap P'_c|}{|P_c| + |P'_c|} \tag{2.8}$$

We consider the mean over all c; $z = \frac{1}{k} \sum_{i \in [k]} z_{c_i}$, as the test statistic for a given initialization. Intuitively, the null-distribution is centered about zero (samples in the initial partition P_c characterized by c may end up arbitrarily far from c after global placement). Likewise, the "ideal" test-static corresponds to 0.5 ($P_c = P'_c$, partitions don't change after global placement). In Table 2.3, we report the z-scores associated with each design (since we find p-values are trivial). We simulate the null-distribution associated with each testcase 1000 times to compute the p-value p_{struct}, the percentage of simulations which result in a test statistic equal to or larger than proposed method's test statistic. We find significance at the 0.01-level for all designs, with the null-distribution close to zero (e.g. $\bar{z}_{\text{null}} = 0.00579$ with standard deviation $< 10^{-5}$ for adaptec3).

Our contributions can be summarized as follows:

1. We propose the NS-Place framework for PCB layout which minimizes net congestion using a support vector machine-like formulation and performs legalization by solving a congestion-aware MILP.
2. We demonstrate that the routed placements produced by our framework have fewer design rule violations and vias, and shorter total metal length compared to manual placements.

In section 2.2, we review previous work. In section 2.2.1, we describe our routability objective. The NS-Place placement framework, including initialization and the MILP-based legalizer is described in Section 2.2.2. In section 2.2.3, we present experimental results on real PCB testcases.

2.2 PCB Placement Initialization

Examples of previous work on the PCB placement problem include Jain and Gea [1996], Ismail et al. [2012], Badriyah et al. [2016], Alexandridis et al. [2017]. These techniques rely on various meta-heuristics to produce non-overlapping layouts while taking into account various metrics such as thermal and power characteristics of components, timing, and tidiness. In general, these methods suffer from drawbacks—e.g. are computationally expensive, incapable of rotating components, or evaluated on synthetic or toy benchmarks. In contrast, our framework is efficient, capable of rotating modules, extensible, and validated on production PCB designs placed by industry experts. We additionally acknowledge the similarity of the PCB placement problem to macro placement, and point the reader to Adya and Markov [2005] for a review of relevant techniques.

A typical method of estimating routing demand is to consider pin or feasible routed-wire density Spindler and Johannes [2007]. Other methods include applying Rent's rule Li et al. [2004], or more sophisticated routing models; for example relying on the construction of rectilinear Steiner trees or the external evaluation of a router Roy and Markov [2007]. State of the art techniques for congestion-aware placement include mPL Li et al. [2004], a multilevel analytical placer based on non-linear optimization and estimating the routing demand based on a two-pin connection routing model, ROOSTER Roy and Markov [2007]: a min-cut placer which models nets by Rectilinear Steiner Minimal Trees, and APlace Kahng et al. [2005], a multilevel analytical placer based on non-linear optimization and stochastic estimates of the routing demand. Similar to our work, Shabbeer et al. [2012] propose to minimize a smooth upper bound on the crossing number to reduce edge crossings in the context of graph visualization, but their formulation is incapable of handling multi-pin nets.

These techniques generally suffer from inadequate estimation or prohibitive computational cost. In contrast, the framework proposed in this work is rigorous and does not rely on rerunning the placement algorithm or applying post-placement optimization.

2.2.1 Net Separation-Oriented Placement

net	$e \in \mathcal{E}$
pin matrices	$A_e \in \mathbb{R}^{p\times 2}$
coordinates and orientation	$x, y \in \mathbb{R}_+^n$, $r \in \{0,1\}^n$
density & net separation cost weight	$\lambda_\text{D}, \lambda_\text{NS} \in \mathbb{R}_+$
convex hull coefficients	$u \in \mathbb{R}^2, \gamma \in \mathbb{R}$
wirelength smoothing parameter	$c \in \mathbb{R}_+$

Figure 2.7. Notation & key terms

Let $x, y \in \mathbb{R}_+^n$ be vectors corresponding of coordinates of n components such that the i-th component has coordinates encoded in the i-th row of $[x : y]$; $[x : y]_i$. Let $\mathcal{E}$ denote a set of m nets. We aim to assign coordinates so that the resulting layout has small cumulative wirelength, layout density, and routing congestion.

Our method may be expressed concisely as the following unconstrained optimization problem given λ:

$$\min_{x,y} \sum_{e\in\mathcal{E}} \left[Wa(e; x, y) + \lambda_\text{NS} Ns(e; x, y)\right] + \lambda_\text{D} D(x, y) \tag{2.9}$$

where Wa and D corresponds to weighted-average wirelength and density terms, and Ns corresponds to the proposed net separation term described in Sec. 2.2.1.D.

Wirelength and density-driven optimization

Many modern techniques for analytic placement rely on quadratic optimization with terms associated with attraction of connected cells, and repulsion of overlapping cells. A typical approach is to represent individual nets as rectangles and to minimize the sum-perimeters over all nets. Repulsion is often applied between overlapping nodes to reduce density. In this work, we adopt the smooth continuous and differentiable weighted-average wirelength (Wa) model Hsu et al. [2011] for wirelength cost. The horizontal net-wirelength for net e is given by

$$Wa_x^{(e)} = \frac{\sum_{i\in e} x_i \exp(\frac{x_i}{c})}{\sum_{i\in e} \exp(\frac{x_i}{c})} - \frac{\sum_{i\in e} x_i \exp(-\frac{x_i}{c})}{\sum_{i\in e} \exp(-\frac{x_i}{c})}$$

where c is a parameter that controls the smoothness and approximation error. We then write the wirelength of e:

$$Wa(e;x,y) = Wa_x^{(e)} + Wa_y^{(e)}$$

The density term corresponds to the mixed-size module bin-based density objective described in Kahng et al. [2005]. The placement area is divided into B bins, and the placer seeks to equalize the overlap at each bin. For a bin b, let x_b be the x-coordinate of the center and w_b be the width. Then the smoothed overlap $\Theta_x(b,i)$ in the x-direction between bin b and module i with width w_i and height h_i is

$$\Theta_x(b,i) = \begin{cases} 1 - 2d_x^2/w_b^2, & \text{if } 0 \leq d_x \leq w_b/2 \\ 2(d_x - w_b)^2/w_b^2, & \text{if } w_b/2 \leq d_x \leq w_b \\ 0 \quad \text{if } w_b \leq d_x \end{cases}$$

where $d_x = |x_i - x_b|$. The overlap in the y-direction is defined similarly. The density function of bin b is then

$$D_b(x,y) = \sum_i C_i \Theta_x(b,i) \Theta_y(b,i) \tag{2.10}$$

where C_i is a normalization factor such that $\sum_b C_i \Theta_x(b,i)\Theta_y(b,i) = w_i h_i$—the area of module i. Finally, $D(x,y) = \sum_b (D_b(x,y) - \frac{\sum_i (w_i h_i)}{B})^2$.

Net-separation optimization via margin maximization

As mentioned in Sec. 2.2, typical approaches to model routing congestion rely on estimating or expressing the routed wire density as a function of pin-density and the feasible routing-area (e.g. the pin-bounding box Spindler and Johannes [2007]). The goal is then to minimize this notion of wire-density. In this work, we model the feasible routing region as the convex-hull of the net-pins, and our goal is to separate routing regions. This prevents over-estimation of the routing density as shown in Fig. 2.8. The method consists of two steps:

1. Given two nets, find the max-margin separator h.

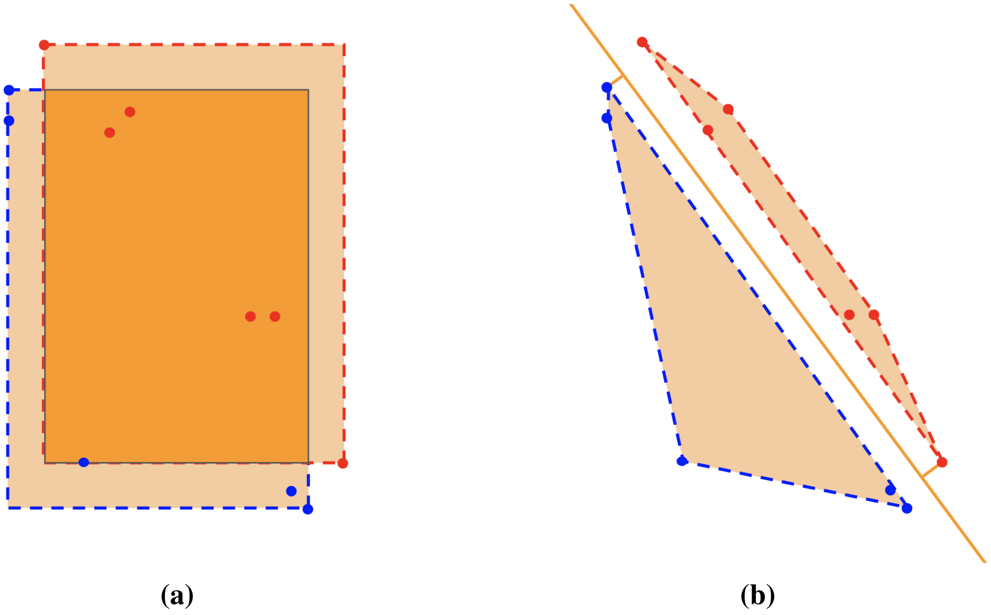


Figure 2.8. Congestion between nets denoted by red and blue pins. **(a)** Rectilinear density metrics are pessimistic. **(b)** An optimistic model of routability. The margin between the convex hulls of nets.

2. For all movable components, find directions (gradients) that maximize the margin with respect to h.

Consider a k-pin net e defined by pins with coordinates $e_1 = [e_{1x}, e_{1y}], e_2 = [e_{2x}, e_{2y}], \ldots, e_k = [e_{kx}, e_{ky}] \in \mathbb{R}^2$. Note that the coordinates of *any* point lying in the convex hull of the net-area may be written as a convex combination of these pin coordinates. Let $A_e^\top = \begin{bmatrix} e_{1x} & e_{2x} & \ldots & e_{kx} \\ e_{1y} & e_{2y} & \ldots & e_{ky} \end{bmatrix}$ be the *pin-matrix* associated with net e. Now consider a k'-pin net e' with associated pin matrix $A_{e'}$. For the convex hulls characterized by the pins to *not* intersect, there necessarily must be a $u \neq 0$ and γ such that $x^\top u - \gamma$ is nonpositive for all x lying in one net, and nonnegative for all x lying in the other net. We denote the hyperplane defined by $\{x | x \in \mathbb{R}^2, x^\top u = \gamma\}$ the *separating hyperplane*, and we want to introduce a regularizer which encourages e and e' to lie in different half-spaces with sufficient margin. Note that e and e' do not intersect if there is no shared point in the interior of the convex hulls characterized by the coordinates of pins in e and e'—i.e. if the following has *no solution*.

$$\begin{aligned}
&\exists \delta_{A_e} \in \mathbb{R}^k, \delta_{A_{e'}} \in \mathbb{R}^{k'} \\
&\text{such that } A_e^\top \delta_{A_e} = A_{e'}^\top \delta_{A_{e'}} \quad \mathbf{1}^\top \delta_{A_e}, \mathbf{1}^\top \delta_{A_{e'}} = 1, \\
&\qquad\qquad \delta_{A_e}, \delta_{A_{e'}} \geq 0
\end{aligned}$$

Namely, duality & Farkas' Lemma provide the conditions that must be satisfied if the convex hulls defined by the pins *do not* intersect:

$$\begin{aligned}
&A_e u \geq \alpha\mathbf{1} \quad A_{e'} u \leq \beta\mathbf{1} \quad \alpha - \beta > 0 \\
&\implies A_e u - \gamma\mathbf{1} \geq \mathbf{1}, \quad A_{e'} u - \gamma\mathbf{1} \leq -\mathbf{1}
\end{aligned}$$

This formulation naturally implies the solution to the following minimization problem. Note that one might alternatively aim to find the *maximum-margin* separator. Due to the equivalence with the SVM optimization problem Cortes and Vapnik [1995], Bennett and Bredensteiner [2000], efficient solvers may be employed to recover γ and u.

$$\begin{aligned}
0 &= \min_{u,\gamma} f(e, e', u, \gamma) \\
&= \min_{u,\gamma} \|(-A_e u + (\gamma + 1)\mathbf{1})_+\|_2 + \\
&\qquad \|(A_{e'} u - (\gamma - 1)\mathbf{1})_+\|_2
\end{aligned} \tag{2.11}$$

Let A_e be the pin matrix corresponding to net e. We define the net-separation regularizer:

$$Ns(\cdot) = \frac{1}{|\mathcal{M}|} \sum_{e' \in \mathcal{E}} \min_{u,\gamma} f(e, e', u, \gamma) \tag{2.12}$$

The gradient of Ns can then be recovered with respect to the i-th row (pin coordinates) of A_e:

$$\frac{\partial Ns}{\partial (A_e)_i} = \mathbb{1}_{(A_e)_i \cdot u \leq \gamma+1} 2\langle u, (\gamma+1)\mathbf{1} - A_e u \rangle$$

Note that due to the reliance of the above gradient on γ, and the reliance of γ on pin coordinates A_e, we adopt an alternating minimization method described in Sec. 2.2.1.B.

2.2.2 NS-Place placement flow

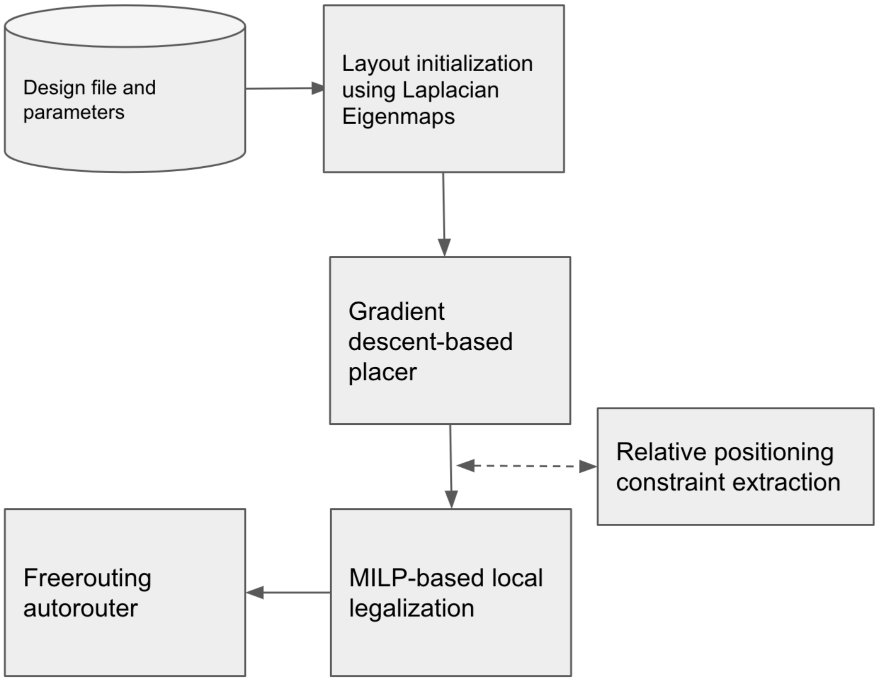

Design file and parameters
Layout initialization using Laplacian Eigenmaps
Gradient descent-based placer
Relative positioning constraint extraction
MILP-based local legalization
Freerouting autorouter

Figure 2.9. Placement procedure with Laplacian Eigenvector initialization, net-separation minimization, and MILP-based legalization with relative positioning constraints.

In this section, we describe the overall flow of our method. The high-level flow is described in Figure 2.9.

Initialization with Laplacian eigenvectors

First-order optimization algorithms are notoriously sensitive to initialization when applied to nonlinear problems. We address this by first collapsing the netlist hypergraph to a component graph via the clique model. We then construct a matrix-representation of the graph connectivity—the *graph Laplacian*. The solution to the associated eigenvalue problem approximates the solution to the sparsest cut problem Hall [1970], and clusters arising out of the vertex-projection into the space spanned by the first nontrivial eigenvalues correspond highly connected components of the graph. We use these coordinates to initialize global placement. More concretely, we solve the following problem, where x and y are coordinates of components, c_i are constants, v is a vector of component areas, $V = \text{diag}(v)$, and L is the normalized graph Laplacian; $L = R^{-\frac{1}{2}} A R^{-\frac{1}{2}}$, where A is an adjacency matrix and R is the diagonal degree matrix.

$$\begin{aligned} &\min\ x^\top L x + y^\top L y \\ &\text{s.t. } v^\top x = 0, \qquad v^\top y = 0, \qquad x, y \neq 0 \\ &\quad x^\top V x = c_1, \quad y^\top V y = c_2, \quad x^\top V y = c_3 \end{aligned} \tag{2.13}$$

Intuitively, the objective is to minimize the weighted squared wire length. The linear and nonequality constraints concentrate the layout about the origin and prevent the trivial solution. The quadratic constraints spread the placement over the x and y axes.

Note that fixed-node constraints may be considered through variable partitioning and the addition of an affine term in the objcetive. However, the problem remains unaware of component orientations. To resolve this, we generate candidate initializations by considering all possible relative component orientations for a given solution to Problem 2.13. The solution with minimal cost with respect to Eq. 2.9 is used to seed the global placer.

Global placement using coordinate descent

In Alg. 3, we present the detailed steps of our iterative method to reduce congestion. Recall the proposed global placement optimization problem described in Eq. 2.9. For brevity, we refer to the cumulative objective as F. Note that the above problem may be solved exactly via quadratic programming. We propose to solve the problem approximately by applying first-order

Algorithm 3. Net crossing minimization

Input: Initial placement $[x : y]$, pin matrices A_e, $A_{e'}$, regularization parameter λ , learning rate α, budget n
Output: Placement $[x : y]$

```
 1: function NsOpt ($[x : y], A_e, A_{e'}; \lambda, \alpha$)
 2:     while Eq. 2.9 not converged do
 3:         Fix $A_e$, $A_{e'}$, compute $u$, $\gamma$ by Eq. 2.11          ▷ separator
 4:         Fix $u$, $\gamma$, compute $\nabla_{p_i} F$, $P$
 5:         $g \leftarrow \lambda_D \nabla_{[x:y]} D + P$          ▷ compute component update
 6:         $[x : y]_{t+1} \leftarrow [x : y]_t + \alpha \cdot g$          ▷ update components
 7:         update $A_e, A_{e'}$ according to $[x : y]_{t+1}$
 8:     end while
 9:     return $[x : y]_n$
10: end function
```

methods in an alternating minimization framework by iteratively solving for u and γ while keeping x and y fixed and visa-versa. Given a layout, we first compute the separating hyperplane characterized by u and γ in Eq. 2.11 (line 4 of Alg. 3). Given u and γ, we can derive the net separation and wirelength gradients associated with individual pins (line 4). To recover the component position update, we introduce the auxiliary variable $P \in \mathbb{R}^{n\times 2}$ corresponding to component update derived from pin-gradients where the i-th row of P is defined to be the average of the ith component's pin gradients—i.e. $P_i = \frac{1}{|\mathcal{P}(i)|} \sum_{p\in\mathcal{P}(i)} \nabla_p F$. The gradients associated with the density and wirelength terms can then be computed, and component positions updated (lines 5-6). Given these new positions, the pin matrices A_e and $A_{e'}$ can then be updated (line 7).

Legalization via mixed integer linear programming

In this section, we introduce a standard MILP-based layout formulation for placing rectilinear components subject to overlap and boundary constraints. We note that this formulation shares similarities with previous work, e.g. Funke et al. [2016] who discuss the optimality of MILP for wirelength-minimal block placements. Additionally, we integrate relative positioning constraints derived from the global placement procedure to preserve net separation while improving scalability.

MILP-based wirelength-minimal layout The objective and constraints are described in the following set of equations. We introduce a second term corresponding to wirelength variance, which we find improves the routability of designs further.

$$\begin{aligned}
&\min_{x,y,r}\left[\sum_{i\in|\mathcal{E}|}\mathrm{hpwl}(i)+\left[\max_{i\in|\mathcal{E}|}\mathrm{hpwl}(i)-\min_{i\in|\mathcal{E}|}\mathrm{hpwl}(i)\right]\right]\\
&\mathrm{hpwl}(i)=(U_x^{(i)}-L_x^{(i)})+(U_y^{(i)}-L_y^{(i)})
\end{aligned}$$

where the hpwl term (given for the x-direction only) is

$$U_x^{(i)}\geq p_j^{(i)}(x),\quad L_x^{(i)}\leq p_j^{(i)}(x)\quad \forall j\in\mathcal{E}_i$$

and the solution is subject to non-overlapping constraints (for brevity, boundary constraints are not included):

$$\begin{aligned}
&x_i+r_ih_i+(1-r_i)w_i\leq x_j+W(p_{ij}+q_{ij}) && \text{i-left-j}\\
&y_i+r_iw_i+(1-r_i)h_i\leq y_j+H(1+p_{ij}-q_{ij}) && \text{i-under-j}\\
&x_i-r_jh_j-(1-r_j)w_j\geq x_j-W(1-p_{ij}+q_{ij}) && \text{i-right-j}\\
&y_i-r_jw_j-(1-r_j)h_h\geq y_j-H(2-p_{ij}-q_{ij}) && \text{i-over-j}\\
&x_i,y_i\geq 0,\quad r_i,q_{ij},p_{ij}\in\{0,1\} && \text{variables}
\end{aligned}$$

Where r_i, q_{ij}, p_{ij} are variables representing orientation and relative positions between modules i and j.

Relative positioning constraints Given a global placement solution, we derive relative position constraints for pairs of components. By doing so, we preserve the global structure of the global placement solution while allowing the MILP placer to make local adjustments.

For each pair of modules, we consider the minimum horizontal and vertical distances between modules (i.e. from boundary to boundary). We then check if the maximum of the horizontal and vertical distances exceeds a pre-defined threshold k which trades off a preference for HPWL-minimal solutions or routability and runtime. A horizontal or vertical relative position constraint is introduced depending on which distance is greater and the associated binary decision variables (q_{ij} or p_{ij}) are removed and the corresponding overlap constraints are simplified or pruned entirely. By integrating these constraints, we preserve the global structure of the analytical solution while eliminating a nontrivial number of decision variables.

2.2.3 Experiments

Table 2.4. Design characteristics. locked are fixed components. layers are layers available for routing.

design	$W \times H(mm^2)$	#comp	#locked.	util.	#nets	#pins	#layers
PCB1	21×14	8	1	0.59	15	40	1
PCB2	51×23	18	5	0.79	34	77	2
PCB3	55×28	34	2	0.44	38	138	2
PCB4	23×60	28	6	0.67	52	140	2
PCB5	41×22	48	2	0.40	54	163	2
PCB6	62×57	48	2	0.17	64	190	2
PCB7	51×23	46	2	0.55	69	211	2
PCB8	57×87	36	2	0.62	70	188	2
PCB9	44×36	58	2	0.60	80	229	2
PCB10	102×54	57	18	0.21	99	319	2
PCB11	89×58	64	2	0.10	134	401	4
PCB12	58×60	58	4	0.31	35	233	4
PCB13	86×72	61	4	0.51	63	314	4
PCB14	86×54	1570	947	0.64	386	1638	4

We applied our method to 14 PCB benchmarks Lin et al. [2021]. Details are provided in Table 2.4. We solve Eq. 2.12 via gradient descent with momentum with $\alpha = 1e^{-3}$, $\lambda_{\text{NS}} = 1$, and $\lambda_{\text{D}} = 1$. GDMILP corresponds to running the NS-Place with $\lambda_{\text{NS}} = 0$, and MILP corresponds to solving the MILP without relative-position constraints. Layout quality is evaluated using the open-source FreeRouting router fre. The routed wirelength, number of design rule violations,

Table 2.5. Pre-route metrics for PCB designs. We report the cumulative HPWL and the net separation cost. The top performing result is **bolded**. The "-" represents that MILP cannot produce a feasible placement in 4 hours.

Design	HPWL (mm)				Net Separation Obj.				Runtime (s)		
	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place	MILP	GDMILP	NS-Place
PCB1	110.22	**64.44 (41.50)**	68.90 (37.4)	72.10 (34.60)	192.13	192.11	192.11	**187.40**	2.70	4.90	5.10
PCB2	362.70	**291.10 (19.70)**	294.32 (18.80)	296.18 (18.30)	497.50	497.40	499.30	**493.10**	6.30	6.20	6.80
PCB3	312.60	**212.70 (32.0)**	252.40 (19.30)	271.30 (13.20)	1427.10	2124.30	1426.40	**1411.40**	14400.00	3793.00	3917.60
PCB4	603.30	-	536.48 (11.10)	**531.19 (12.00)**	3750.20	-	3753.10	**3746.13**	-	9.40	9.90
PCB5	654.10	-	**621.00 (5.10)**	637.00 (2.60)	2400.00	-	2400.00	**2300.00**	-	4231.20	4461.50
PCB6	771.80	-	**649.40 (15.90)**	708.60 (8.40)	1930.00	-	1940.00	**1740.00**	-	4893.10	5072.40
PCB7	2987.90	-	**563.10 (88.50)**	1601.40 (46.40)	1241.34	-	1239.70	**1017.16**	-	5927.30	6008.20
PCB8	766.10	**731.30 (4.60)**	748.40 (2.30)	756.40 (1.30)	98500.00	99500.00	99400.00	**96400.00**	14400.00	4360.90	4732.40
PCB9	714.90	-	677.60 (5.20)	**662.50 (7.30)**	2600.00	-	2600.00	**2400.00**	-	5327.30	5719.80
PCB10	4355.61	-	3317.94 (23.80)	**3315.46 (23.90)**	2117.80	-	2124.10	**2103.60**	-	5314.70	5417.20
PCB11	2941.90	-	**2573.20 (12.50)**	2618.10 (11.00)	9210.00	-	9210.00	**9070.00**	-	4651.90	4782.40
PCB12	972.50	929.30 (4.40)	**932.60 (4.10)**	941.30 (3.20)	10.73	11.31	11.47	**36.92**	14400.00	4619.10	4752.30
PCB13	2644.97	-	**2126.13 (19.60)**	2151.77 (18.60)	2400.00	-	2400.00	**2300.00**	-	3278.60	3416.30
PCB14	7069.26	-	**6432.19 (9.01)**	6691.34 (5.35)	29400	-	29870	**11186**	-	14400.00	14400.00

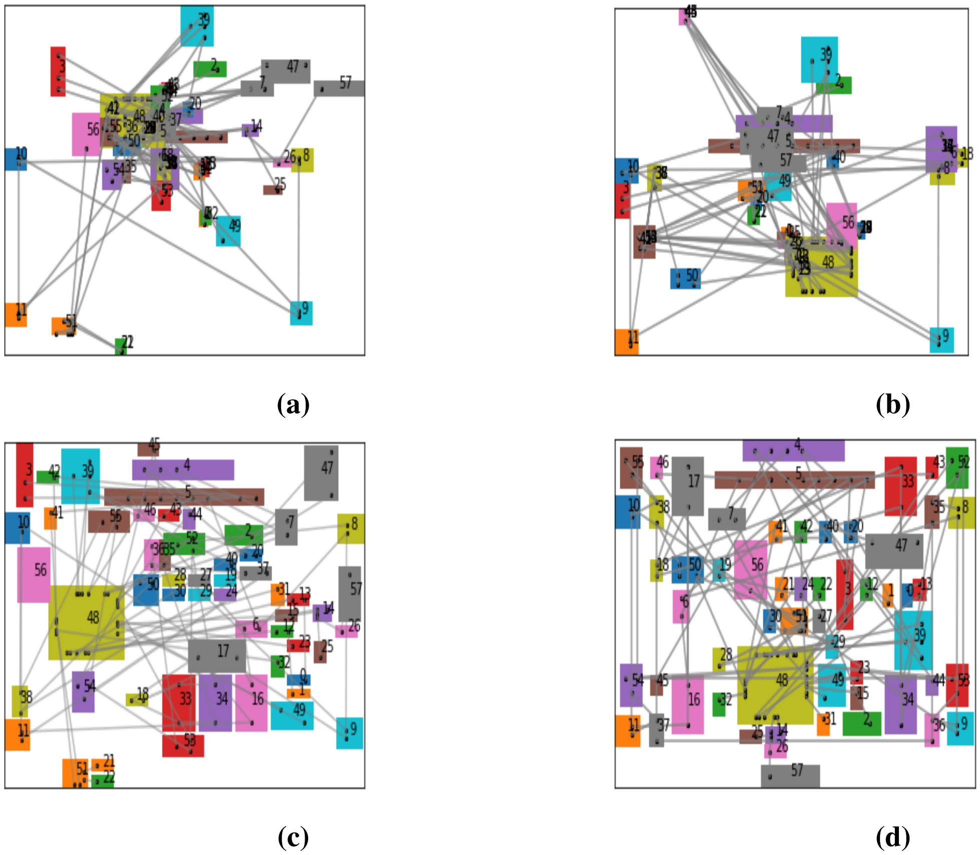
Table 2.6. Post-route metrics for PCB designs. We report routed wirelength using FreeRouting, the number of vias, and the number of DRVs as reported by KiCAD. We report the percent improvement in parenthesis. The best result is **bolded**.

Design	Routed Wirelength (mm)				#Vias				#DRVs + #unrouted nets			
	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place	manual	MILP	GDMILP	NS-Place
PCB1	129.00	123.00 (4.70)	194.00 (-50.4)	**121.00 (6.20)**	**0**	10	9	4	6	**0**	**0**	**0**
PCB2	**354.00**	632 (-78.50)	507.00 (-43.20)	421.00 (-18.90)	**3**	6	8	5	13	26	24	23
PCB3	638.00	682.00 (-6.90)	771.00 (-20.80)	**616.00 (3.40)**	17	21	24	**15**	11	2	3	**0**
PCB4	809.00	-	857.00 (-5.90)	**806.00 (0.40)**	**31**	-	59	54	17	-	4	**0**
PCB5	558.00	-	**538.00 (3.60)**	541.00 (3.00)	**32**	-	84	49	7	-	5	**3**
PCB6	1007.00	-	**854.00 (15.90)**	906.00 (10.00)	23	-	47	**19**	34	-	29	**0**
PCB7	3735.00	-	3140.00 (15.90)	**2949.00 (21.00)**	161	-	141	**93**	23	-	**0**	**0**
PCB8	913.00	854.30 (6.30)	713.40 (21.90)	**711.90 (22.00)**	57	73	89	**43**	27	6	5	**0**
PCB9	1069.00	-	**749.00 (29.90)**	772.00 (27.80)	**38**	-	87	64	17	-	7	**0**
PCB10	5043.00	-	5294.00 (-1.00)	**4914.00 (2.60)**	129	-	136	**97**	12	-	3	**0**
PCB11	3460.00	-	3271.90 (5.40)	**3107.80 (10.20)**	131	-	286	**109**	23	-	57	**13**
PCB12	1790.00	1960.00 (-9.40)	1930.00 (7.80)	**1720.00 (3.90)**	49	62	54	**45**	4	9	8	**0**
PCB13	3150.00	-	2903.00 (7.80)	**2897.00 (8.00)**	161	-	98	**83**	11	-	9	**0**
PCB14	9017.00	-	9643.00 (-6.94)	**8873.00 (1.60)**	976	-	1007	**942**	104	-	139	**92**

and the number of vias are reported. Experiments are performed with an 3.4GHz Intel i7-4770 CPU and 31GB RAM. If no optimal solution is found after 4 hours, we report results for the best-so-far (with respect to the MILP objective) feasible solution. If no feasible solution is found, associated entries are marked with "-".

Main experiments

PCB placement metric comparison We provide manual, MILP, GDMILP, and NS-Place placement results in table 2.5. We show that NS-Place reduces the net separation cost by 77% and 41% on average compared to manual and GDMILP. This implies that although the MILP-



(a)

(b)

(c)

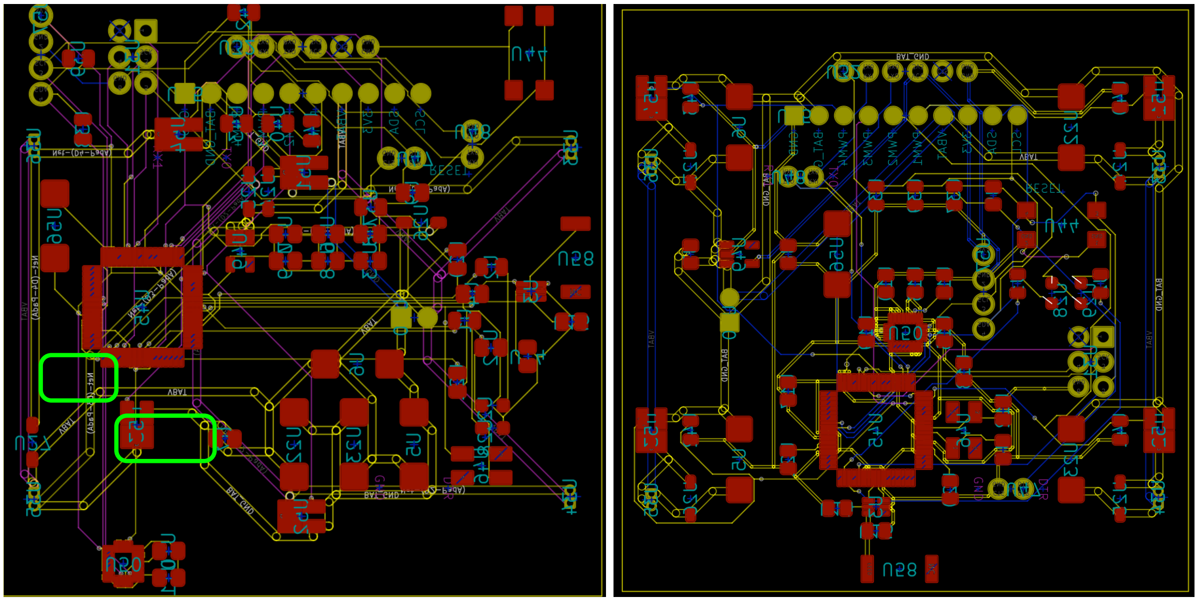
(d)

Figure 2.10. PCB12 layouts. (**a**): Seed placement produced from Laplacian eigenvectors. (**b**) Global placement to minimize net crossings. (**c**) MILP-based legalization. (**d**) Manual layout.

based fine-tuning step does not optimize net separation explicitly, satisfaction of relative position constraints results in preservation of the global structure produced by the net separation step. Furthermore, the inclusion of the net separation term does not result in a large increase in HPWL. Although the HPWL of NS-Place solutions increase 9% on average compared to GDMILP, NS-Place still achieves 20% lower HPWL compared to manual layouts.

We report the cumulative runtime to complete a placement in Table 2.5. Note that methods employing relative positioning constraints (i.e., GDMILP, NS-Place) strictly improve over vanilla MILP. We see that NS-Place closely matches the performance of the GDMILP flow with only around 4% increase on average runtime. This implies that the net separation regularizer imposes low overhead. Fig. 2.10 (a), (b), and (c) show the PCB placement results of PCB12 of each stage of automatic placement flow. Fig. 2.10 (d) is the manual layout of PCB12. Compared to Fig. 2.10 (d), we observe that NS-Place produces placement with fewer net crossings as shown in Fig. 2.10 (c).

PCB routing metric comparison We evaluate routability in Table 2.6 by reporting the routed wirelength, the number of DRVs and vias, and the number of unrouted nets using FreeRouting and Kicad. Compared with manual, MILP, and GDMILP, NS-Place reduces the #DRVs and #unrouted nets by roughly 80%, 70%, and 75% on average.



(a) **(b)**

Figure 2.11. PCB12 P&R result in KiCAD. (**a**) Solution produced by our method. Routable regions are emphasized with green rectangles. (**b**) Routed manual placement.

NS-Place additionally reduces the routed wirelength by 10% and 6% on average compared to manual and GDMILP respectively and reduces the #Vias by 18% and 39%. Moreover, for PCBs 7, 10, 11, and 13, which have #components larger than 60 and #pins larger than 300, NS-Place achieves 34% #Vias reduction on average compared to manual. An example routed result is given in Fig. 2.11. Compared to the manual layout, NS-Place successfully improves routability via net separation (i.e., the routable regions near the center of the board). Note that NS-Place fails to produce a superior solution for PCB2. We attribute this to PCB2's high utilization, with only about 20% of the board area available to the global placer as whitespace.

In summary, we demonstrate that NS-Place significantly reduces #DRVs, #unrouted nets, and #Vias for PCB reliability compared to manual, MILP, and GDMILP with an extensive study on 14 PCBs.

2.3 Graph Learning

Semi-supervised learning is an important field in machine learning and statistics. Semi-supervised methods leverage both labeled and unlabeled data. In semi-supervised learning (SSL), we are given a partially-labeled training set consisting of labeled examples and unlabeled examples. The goal in this setting is to learn a predictor that is superior to a predictor that is trained using the labeled examples alone. This setup is motivated by the high cost of obtaining annotated data on practical problems. Consequently, we are typically interested in the regime where the number of labeled examples is significantly smaller than the number of training points. For problems where very few labels are available, the geometry of the unlabeled data can be used to significantly improve the performance of classic machine learning models. For example, a seminal work in graph-based semi-supervised learning is Laplace learning Zhu et al. [2003], which seeks a harmonic function that extends provided labels over the unlabeled vertices. Laplace learning, and its variants have been widely applied in semi-supervised and graph-structured learning Zhou et al. [2005, 2003], Ando and Zhang [2006].

In this work, we improve upon the state-of-the-art for graph-based semi-supervised learning at very low label rates. Classical Laplace learning and label propagation algorithms yield poor classification results Nadler et al. [2009], Alaoui [2016] in this setting. This is typically attributed to the fact that the solutions develop localized spikes near the labeled vertices and are nearly constant for vertices distant from labels. In other words, Laplace learning-based algorithms often fail to adequately propagate labels over the graph. To address this issue, recent work has suggested imposing small adjustments to classical Laplace learning procedure. For example, p-Laplace learning Alaoui [2016], Slepčev and Thorpe [2019], Calder [2018, 2019] for $p > 2$, and particularly for $p = \infty$, often yields superior empirical performance compared to Laplace learning at low label rates Flores et al. [2019]. Other relevant methods for addressing low label rate problems include higher-order Laplacian regularization Zhou and Belkin [2011] and spectral classification Belkin and Niyogi [2002], Zhou and Srebro [2011].

Other recent approaches have aimed to derive various approximations by iteratively re-weighting the graph more heavily near labels. For example, Weighted Nonlocal Laplacian (WNLL) Shi et al. [2017] amplifies the weights of edges immediately connected to labeled nodes. The WNLL algorithm achieves better results at moderately low label rates and exhibits desirable limiting characteristics, but still performs poorly at very low label rates Flores et al. [2019]. To address this, Calder and Slepčev [2020] proposed a Properly Weighted Laplacian, which re-weights the graph in a way that is well-posed at arbitrarily low label rates.

In this work, we propose to solve a natural semi-supervised extension of Laplacian Eigenmaps, which is well-posed in the limit of unlabeled data. Our extension is initially motivated by the optimization-based perspective of Laplacian Eigenmaps as a Rayleigh Quotient minimization problem over all labeled and unlabeled vertices. We show that a natural partitioning of the problem yields a more general quadratically constrained quadratic program over the unlabeled vertices. We then propose to generalize the sequential subspace (SSM) framework commonly used to solve similar problems in $\mathbb{R}^n$ to $\mathbb{R}^{n \times k}$.

To summarize, our contributions are:

1. We introduce a natural formulation of graph semi-supervised learning as a rescaled quadratic programming problem on a compact Stiefel Manifold, i.e. a generalization of the *Trust-Region Subproblem.*

2. We describe scalable approximate methods and globally convergent iterative methods and demonstrate robustness in a variety of label rate regimes.

3. We compare our approach to competing semi-supervised graph learning algorithms and demonstrate state-of-the-art performance in low, medium, and high label rate settings on

MNIST, Fashion-MNIST, and CIFAR-10.

The rest of the paper is organized as follows. In Section 2.3.1 we first briefly introduce Laplacian Eigenmaps and our supervised variant, and then provide a detailed motivation for the algorithm.

2.3.1 Preliminaries and Laplace learning

We assume the data can be viewed as lying on a graph, such that each vertex is a datapoint. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$ denote the M vertices of the graph with edge weights $w_{ij} \geq 0$ between v_i and v_j. We assume that the graph is symmetric, so $w_{ij} = w_{ji}$. We define the degree $d_i = \sum_{j=1}^{n} w_{ij}$. For a multi-class classification problem with k classes, we let the standard basis vector $e_i \in \mathbb{R}_k$ represent the i-th class (i.e. a "one hot encoding"). Without loss of generality, we assume the first m vertices $v_1, v_2, \ldots, v_m$ are given labels $y_1, y_2, \ldots, y_m \in \{e_1, e_2, \ldots, e_k\}$, where $m \ll M$. Let n denote the number of unlabeled vertices, i.e. $n = M - m$. The problem of graph-based semi-supervised learning is to smoothly propagate the labels over the unlabeled vertices $v_{m+1}, v_{m+2}, \ldots, v_M$.

The Laplace learning algorithm Zhu et al. [2003] extends the labels by solving the following problem

$$\left.\begin{aligned} x(v_i) &= y_i, \quad \text{if } 1 \leq i \leq m \\ \mathcal{L}x(v_i) &= 0, \quad \text{if } m+1 \leq i \leq M \end{aligned}\right\} \tag{2.14}$$

where $\mathcal{L}$ is the unnormalized graph Laplacian given by $\mathcal{L} = D - W$, where D is a diagonal matrix whose elements are the node degrees, and $x : \mathcal{V} \to \mathbb{R}^k$. The prediction for vertex v_i is determined by the largest component of $x(v_i)$

$$\ell(v_i) = \underset{j \in \{1,\ldots,k\}}{\arg\max} \{x_j(v_i)\}. \tag{2.15}$$

We note that Laplace learning is also called *label propagation (LP)* Zhu [2005], since the Laplace equation eq. (2.14), can be solved by repeatedly replacing $x(v_i)$ with the weighted average of its neighbors.

The solution of Laplace learning eq. (2.14), is the minimizer of the following gradient regularized variational problem with label constraints $x(v_i) = y_i$:

$$\min_{x\in\ell^2(\mathcal{V})} \left\{ ||\nabla x||^2_{\ell^2(\mathcal{V}^2)} | x(v_i) = y_i, 1 \leq i \leq m \right\} \tag{2.16}$$

2.3.2 Motivation via graph cut & Laplacian Eigenmaps

Suppose that we have two disjoint sets of vertices on the graph; $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathcal{V}$, such that $\mathcal{V}_2 = \mathcal{V}_1^c$. A graph cut:

$$\text{cut}(V_1) = \sum_{i\in\mathcal{V}_1, j\in\mathcal{V}_2} w_{ij}$$

is the total summation of the edge weights whose two vertices are in different sets. In particular, $\text{cut}(\mathcal{V}) = 0$. Note that finding partitions that minimize this cut is likely to give us unbalanced clusters/partitions.

As an alternative, consider the ratio-cut: $\text{rcut}(\mathcal{V}_1, \mathcal{V}_2) = \frac{\text{cut}(\mathcal{V}_1,\mathcal{V}_2)}{|\mathcal{V}_1|} + \frac{\text{cut}(\mathcal{V}_1,\mathcal{V}_2)}{|\mathcal{V}_2|}$. Minimizing the ratio cut can be approximated by an eigenvector problem. First note that the cut is directly related to quadratic form associated with the weighted graph Laplacian. Denote the indicator of $\mathcal{V}_i$: $\mathbf{1}_{\mathcal{V}_i}$. Then,

$$\text{cut}(\mathcal{V}_1, \mathcal{V}_2) = \mathbf{1}_{\mathcal{V}_1}^\top L \mathbf{1}_{\mathcal{V}_1} = \mathbf{1}_{\mathcal{V}_2}^\top L \mathbf{1}_{\mathcal{V}_2}$$

Likewise, the ratio-cut can be succinctly written

$$\text{rcut}(\mathcal{V}_1, \mathcal{V}_2) = \langle L, Z \rangle$$

Where $Z = \frac{1}{|\mathcal{V}_1|}\mathbf{1}_{\mathcal{V}_1}\mathbf{1}_{\mathcal{V}_1}^\top + \frac{1}{|\mathcal{V}_2|}\mathbf{1}_{\mathcal{V}_2}\mathbf{1}_{\mathcal{V}_2}^\top$. Note that Z is a PSD projection, that is, $Z = UU^\top$. One way to relax the ratio cut is to optimize over $V \in \mathbb{R}^{n\times 2}$:

$$\min \langle L, U^\top \rangle \quad \text{s.t. } U^\top U = I$$

In Laplacian Eigenmaps Belkin and Niyogi [2003], one seeks an embedding of the vertices via the eigenfunctions of the Laplacian corresponding to the smallest nontrivial eigenvalues. Equivalently, this can be expressed as the following *Quadratically Constrained Quadratic Program* (QCQP) over the vertices of the graph:

$$\min_{X_0} \langle X_0, \mathcal{L} X_0 \rangle \quad \text{s.t. } X_0^\top X_0 = I,\ \mathbf{1}^\top X_0 = 0 \tag{2.17}$$

where $\langle A, B \rangle$ is is the trace of the matrix $A^\top B$ and $\mathbf{1}$ is the all-ones vector. The notation $X_0 \in \mathbb{R}^{M \times k}$ is the mapping of the M vertices to a k-dimensional space. In the case where $k = 1$, eq. (2.17) is also known in the numerical analysis literature as a *Rayleigh quotient minimiziation problem* Golub and Van Loan [1996]. Despite its nonconvexity, a unique (up to orthogonal transformations) global solution is given by the set of eigenvectors of $\mathcal{L}$ corresponding to the smallest k nontrivial (nonzero) eigenvalues of $\mathcal{L}$.

2.3.3 Semi-supervised graph learning

We first extend these frameworks with supervision, motivated by Laplace learning in eq. (2.16). Additionally, to facilitate the supervised decomposition, we rescale I uniformly by $p = M/k$, the balanced proportion of samples associated with each class:

$$\begin{gathered} \min_{X_0} \langle X_0, \mathcal{L} X_0 \rangle \\ \text{s.t. } X_0^\top X_0 = pI,\ \mathbf{1}^\top X_0 = 0,\ (X_0)_i = y_i,\ i \in [m] \end{gathered} \tag{2.18}$$

We denote the objective F. The associated prediction is then $\ell(x_i) = \arg\max_{j \in \{1,\dots,k\}} (X_0)_{ij}$. Next, we show how supervision naturally leads to a partitioning of the problem. We denote the submatrices of X_0 and $\mathcal{L}$ corresponding to the n unlabeled and m labeled vertices as X_u, X_l and L_u, L_l, respectively. More concretely, $\mathcal{L} = \begin{bmatrix} L_l & L_{lu} \\ L_{ul} & L_u \end{bmatrix}$ and $X_0 = \begin{bmatrix} X_l \\ X_u \end{bmatrix}$. Then, in conjunction with considering X_l fixed, the problem in eq. (2.17) may be expressed as

$$\begin{aligned}&\min_{X_u} \langle X_u, L_u X_u\rangle - \langle X_u, B_0\rangle \\ &\text{s.t. } X_u^\top X_u = C_u,\ \mathbf{1}^\top X_u = -r^\top\end{aligned} \tag{2.19}$$

where $B_0 = 2 \cdot L_{ul} X_l$, $C_u = pI - X_l^\top X_l = (p - \widetilde{p})I$, where $\widetilde{p} = m/k$, and $r = (\mathbf{1}^\top X_l)^\top = \widetilde{p}\mathbf{1}^\top \in \mathbb{R}^k$ are fixed parameters, and X_u are the decision variables.

In general, the quadratic equality constraint poses a significant challenge from an optimization standpoint. We propose to address this problem directly by solving an equivalent rescaled problem. By introducing terms to eliminate the linear constraint, we show how the problem may be rescaled and efficiently and robustly solved as a quadratic program on a compact *Stiefel Manifold*, denoted

$$\mathrm{St}(n,k) = \{X \in \mathbb{R}^{n\times k} : X^\top X = I\}. \tag{2.20}$$

The associated solution to this problem can then be used to determine the labels of the unlabeled vertices, as in Laplace learning eq. (2.16).

Remark 2.3.1. Given a matrix $X \in \mathbb{R}^{n\times k}$, its projection onto $\mathrm{St}(n,k)$, $[X]_+ := \arg\min\{\|X_s - X\|_F : X_s \in \mathrm{St}(n,k)\}$ is given by

$$[X]_+ = UV^\top, \tag{2.21}$$

where $X = U\Sigma V^\top$ is the singular value decomposition.

To eliminate the linear constraint, we introduce two adjustments: first, let $(X)_i = (X_u)_i + \frac{1}{n}r^\top$ denote a row-wise centering transformation with respect to the labeled nodes. This yields the new constraint $\mathbf{1}^\top X = 0$ and yields $C = C_u - \frac{1}{n}rr^\top$ for X. Second, we introduce the projection $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ onto the subspace orthogonal to the vector $\mathbf{1} \in \mathbb{R}^n$, i.e., $\mathbf{1}^\top(PX) = 0$, which projects vectors onto the set of mean-zero vectors. To obtain a solution limited to this subspace, we introduce the substitutions $B = P(B_0 - L_u\frac{1}{n}\mathbf{1}r^\top)$ and $L = PL_uP$ which implies $\mathbf{1}^\top B = 0$. Consider the substitution $X \leftarrow XC^{1/2}$. The equivalent, rescaled problem is then

$$\min_{X:X\in \mathrm{St}(n,k)} \langle X, LXC\rangle - \langle X, BC^{1/2}\rangle \tag{2.22}$$

Note that this problem is exactly the generalization of well-known problems that arise in trust-region methods, optimization of a nonconvex quadratic over a unit ball or sphere Sorensen [1982], Conn et al. [2000], i.e. problems of the form

$$\min_{x\in\mathbb{R}^n:\|x\|=1} \langle x, Lx\rangle - \langle x, b\rangle$$

We define the Lagrangian of eq. (2.22) where $\Lambda \in \mathbb{R}^{k\times k}$ are the Lagrange multipliers:

$$\langle X, LXC\rangle - \langle X, BC^{1/2}\rangle - \langle \Lambda, (X^\top X - I)\rangle. \tag{2.23}$$

The first-order condition is then

$$LXC = BC^{1/2} + X\Lambda \tag{2.24}$$

for some Λ. Solutions X that satisfy eq. (2.24) are *critical points* or *stationary points*. In general, there could exist multiple critical points that satisfy this condition. In the appendix we show that at these “stationary points” (maximizers, minimizers, or saddle points), (1.) the eigenvalues of Λ characterize the optimality of X and (2.) finding good critical points necessitates computation of the eigenvalues of L.

2.3.4 Efficient approximation via Procrustes Alignment

This step described in Sec. 1.2.1 can be interpreted as an alignment step, where an orthogonal transformation Q is found that aligns unlabeled vertices with their neighboring labeled vertices, and apply this transformation to all unlabeled vertices. We briefly describe the connection with Orthogonal Procrustes Analysis Wang and Mahadevan [2008]. Let X be feasible, i.e. $X^\top X = I$. The approximation $C \approx I$ is appropriate in the limit of few labeled examples or unlimited unlabeled data. Since $C = (p - \widetilde{p})I - \frac{\tilde{p}^2}{n}\mathbf{1}\mathbf{1}^\top$, then $C \to pI$ in the limit of $\frac{n}{m} \to 0$. Note that the invariance is nothing but $\mathrm{tr}(X^\top LX) = \langle X, LX\rangle = \langle XQ, LXQ\rangle$ for any orthogonal Q. Thus,

$$\begin{aligned} &\mathop{\arg\min}_{Q:Q\in\mathrm{St}(k,k)} \langle XQ, LXQ\rangle - \langle XQ, B\rangle \\ =\ &\mathop{\arg\max}_{Q:Q\in\mathrm{St}(k,k)} \langle XQ, B\rangle \qquad (2.25) \\ =\ &\mathop{\arg\min}_{Q:Q\in\mathrm{St}(k,k)} \|XQ - B\|_F^2. \end{aligned}$$

The problem stated in the last line is the canonical Orthogonal Procrustes problem in $\mathbb{R}^{k\times k}$ in the context of finding an alignment between the axis-aligned labeled vertices and their neighborhood of unlabeled vertices. We demonstrate the effect of this procedure in Figure 2.12. In Figure 2.12a—b, we plot the first pair of eigenvectors corresponding to the smallest two nonzero eigenvalues associated with a barbell graph. In Figure 2.12c, we pick a random pair of vertices v_i and v_j with coordinates x_i and x_j from each clique and assign labels $y_i = x_i$ and $y_j = x_j$. Under this labeling, we say that the embedding is *inconsistent*. In other words, a linear predictor in the space spanned by the embedding would be incapable of recovering a separating hyperplane that accurately classifies the unlabeled samples. We then show that by applying the approximate method based on Procrustes Analysis introduced in Section 1.2.1, we recover an embedding which is *consistent* with the labels.

Alternatively, the projection and Q-transform can be interpreted as performing orthogonal multivariate regression in the space spanned by the first k nontrivial eigenvectors of L:

$$Q = \mathop{\arg\min}_{Q:Q\in\mathrm{St}(k,k)} \sum_{i\in[m]} \|x_i Q - y_i\|_2^2,$$

where $Q \in \mathbb{R}^{k\times k}$ and predictions are given by XQ. Note that this is similar in principal to the Semi-Supervised Laplacian Eigenmaps (SSL) algorithm Belkin and Niyogi [2002], which solves

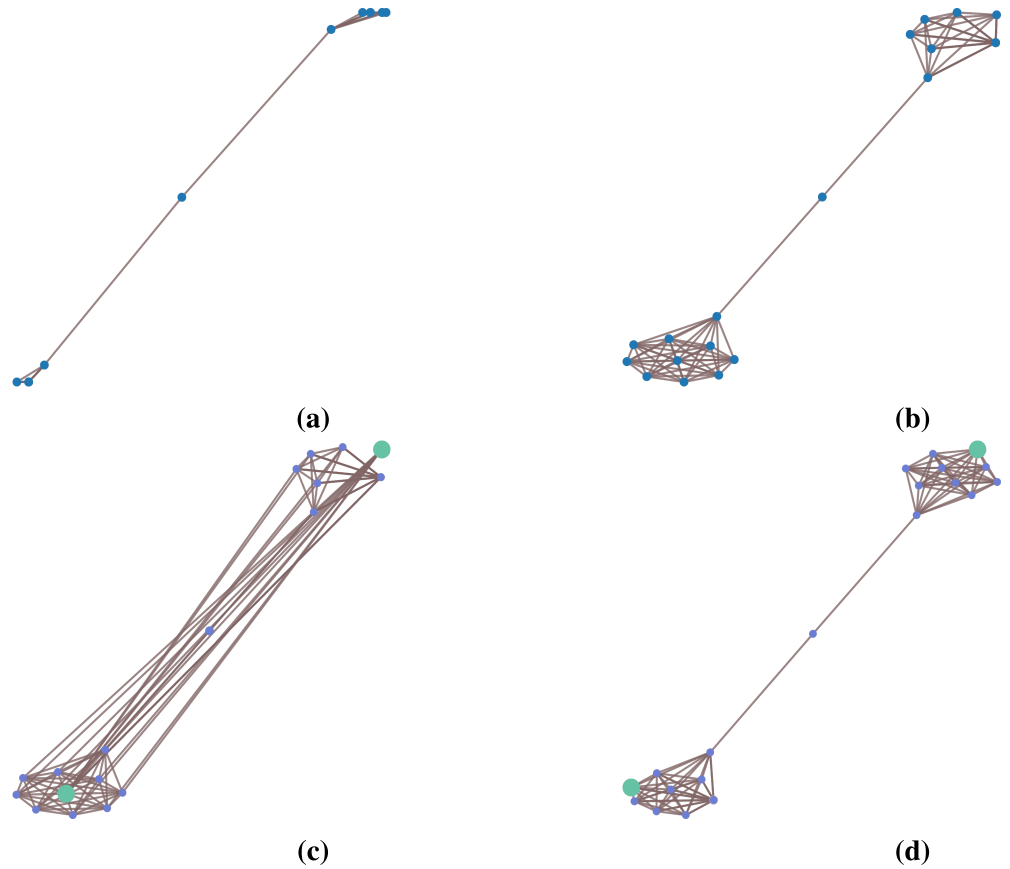


Figure 2.12. Eigenvector method and projection example on the barbell graph. (**a**): Embedding via Laplacian Eigenmaps. (**b**): Several iterations of gradient-based repulsion are applied to remove vertex overlaps for better visualization. (**c**): Consider taking an arbitrary vertex from each clique and assigning it a label (green vertices) such that the embedding is *inconsistent* with the labels. (**d**): Orthogonal transform Q derived from Prop. 1.1.3 and applied to X; XQ resolves the discrepancy between the embeddings and the labeled vertices.

an ordinary least squares problem using eigenvectors X of the Laplacian as features:

$$Q = \arg\min_{Q} \sum_{i\in[m]} \|x_i Q - y_i\|_2^2.$$

Crucially, the orthogonality constraint on Q ensures that the solution remains feasible, i.e. that $XQ \in \mathrm{St}(n,k)$. Furthermore, we show in our experiments that this feasability significantly improves generalization at very low label rates in comparison to standard Laplacian Eigenmaps SSL. These interpretations serve to motivate our initialization and subsequent refinement. In particular, Zhou and Srebro [2011] consider the limiting behavior of Laplacian Eigenmaps SSL

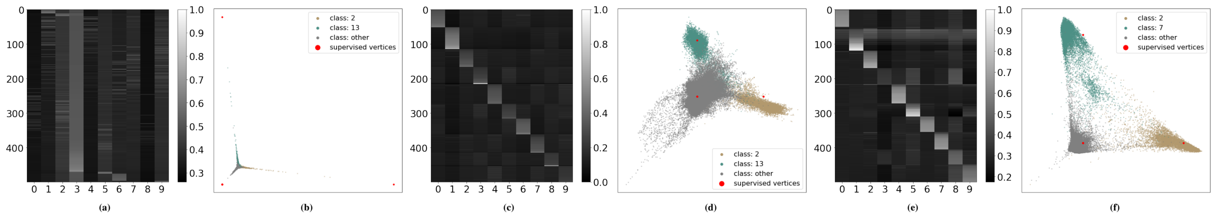


Figure 2.13. Barcode plots of MNIST predictors (left) and embeddings of samples for digits ‘2’ and ‘7’ (right). Learning is performed with 1 label per class. In the barcode plots, the rows are the samples, ordered by their class. Ordering of the columns was obtained by iteratively sorting the columns of the embedding matrices X. (**a—b**) Laplace learning exhibits degeneracy in the limit of unlabeled data. (**c—d**) Embeddings derived using Procrustes Analysis (Section 1.2.1) demonstrates no degeneracy but mixed samples form different classes together. (**e—f**) Our method exhibits good classification performance (a block diagonally dominant barcode) while respecting the geometry of unlabeled examples.

and show that it is non-degenerate in the limit of infinite unlabeled data.

2.3.5 Experiments

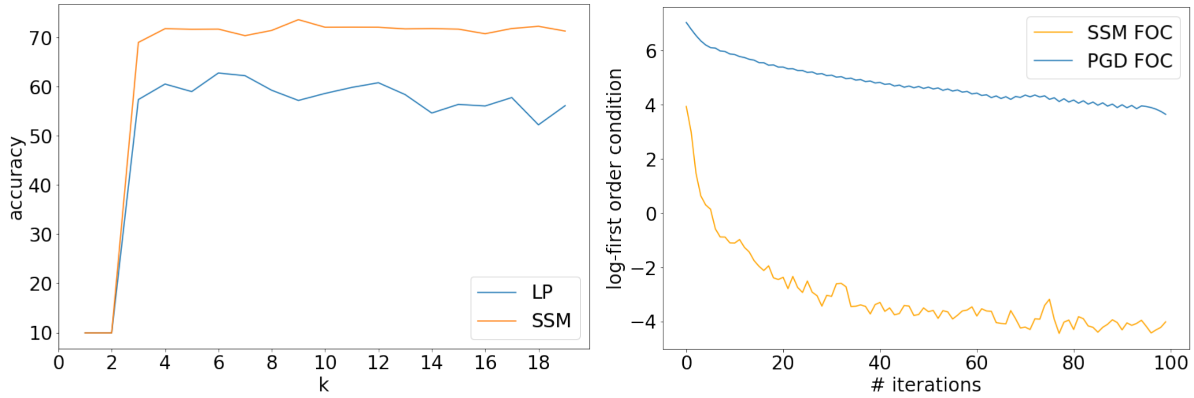
Table 2.7. MNIST: Average accuracy scores over 100 trials with standard deviation in brackets. Best is bolded.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP Zhu et al. [2003]	16.1 (6.2)	28.2 (10.3)	42.0 (12.4)	57.8 (12.3)	69.5 (12.2)
Nearest Neighbor	55.8 (5.1)	65.0 (3.2)	68.9 (3.2)	72.1 (2.8)	74.1 (2.4)
Random Walk Zhou et al. [2004]	66.4 (5.3)	76.2 (3.3)	80.0 (2.7)	82.8 (2.3)	84.5 (2.0)
WNLL Shi et al. [2017]	55.8 (15.2)	82.8 (7.6)	90.5 (3.3)	93.6 (1.5)	94.6 (1.1)
p-Laplace Flores et al. [2019]	72.3 (9.1)	86.5 (3.9)	89.7 (1.6)	90.3 (1.6)	91.9 (1.0)
Poisson Calder et al. [2020]	90.2 (4.0)	93.6 (1.6)	94.5 (1.1)	94.9 (0.8)	95.3 (0.7)
LE-SSL Belkin and Niyogi [2002]	43.1 (0.2)	87.4 (0.1)	88.2 (0.0)	90.5 (0.1)	93.7 (0.0)
Procrustes-SSL	87.0 (0.1)	89.1 (0.0)	89.1 (0.0)	89.6 (0.1)	91.4 (0.0)
SSM	**90.6 (3.8)**	**94.1 (2.1)**	**94.7 (1.6)**	**95.1 (1.1)**	**96.3 (0.9)**

In this section, we present a numerical study of our algorithm applied to image classification in three domains at low label rates. We additionally explore medium and large label rates in comparion to recent state-of-the-art methods.

Table 2.8. FashionMNIST: Average accuracy scores over 100 trials with standard deviation in brackets.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP Zhu et al. [2003]	18.4 (7.3)	32.5 (8.2)	44.0 (8.6)	52.2 (6.2)	57.9 (6.7)
Nearest Neighbor	44.5 (4.2)	50.8 (3.5)	54.6 (3.0)	56.6 (2.5)	58.3 (2.4)
Random Walk Zhou et al. [2004]	49.0 (4.4)	55.6 (3.8)	59.4 (3.0)	61.6 (2.5)	63.4 (2.5)
WNLL Shi et al. [2017]	44.6 (7.1)	59.1 (4.7)	64.7 (3.5)	67.4 (3.3)	70.0 (2.8)
p-Laplace Flores et al. [2019]	54.6 (4.0)	57.4 (3.8)	65.4 (2.8)	68.0 (2.9)	68.4 (0.5)
Poisson Calder et al. [2020]	60.8 (4.6)	66.1 (3.9)	69.6 (2.6)	71.2 (2.2)	72.4 (2.3)
LE-SSL Belkin and Niyogi [2002]	22.0 (0.1)	51.3 (0.1)	62.0 (0.0)	65.4 (0.0)	63.2 (0.0)
Procrustes-SSL	50.1 (0.1)	55.6 (0.1)	62.0 (0.0)	63.4 (0.0)	61.3 (0.0)
Procrustes-SSL + SSM	**61.2 (5.3)**	**66.4 (4.1)**	**70.3 (2.3)**	**71.6 (2.0)**	**73.2 (2.1)**
# Labels per class	**10**	**20**	**40**	**80**	**160**
Laplace/LP Zhu et al. [2003]	70.6 (3.1)	76.5 (1.4)	79.2 (0.7)	80.9 (0.5)	82.3 (0.3)
Nearest Neighbor	62.9 (1.7)	66.9 (1.1)	70.0 (0.8)	72.5 (0.6)	74.7 (0.4)
Random Walk Zhou et al. [2004]	68.2 (1.6)	72.0 (1.0)	75.0 (0.7)	77.4 (0.5)	79.5 (0.3)
WNLL Shi et al. [2017]	74.4 (1.6)	77.6 (1.1)	79.4 (0.6)	80.6 (0.4)	81.5 (0.3)
p-Laplace Flores et al. [2019]	73.0 (0.9)	76.2 (0.8)	78.0 (0.3)	79.7 (0.5)	80.9 (0.3)
Poisson Calder et al. [2020]	75.2 (1.5)	77.3 (1.1)	78.8 (0.7)	79.9 (0.6)	80.7 (0.5)
LE-SSL Belkin and Niyogi [2002]	67.1 (0.0)	68.8 (0.0)	70.5 (0.0)	70.9 (0.0)	66.6 (0.0)
Procrustes-SSL	65.3 (0.0)	66.2 (0.0)	68.3 (0.0)	69.6 (0.0)	64.5 (0.0)
Procrustes-SSL + SSM	**76.4 (1.4)**	**78.1 (1.3)**	**79.4 (0.9)**	**80.3 (0.7)**	**82.6 (0.4)**



(a) **(b)**

Figure 2.14. Robust performance of SSM on F-MNIST. (**a**) robustness to different numbers of neighbors k used to construct the graph, averaged over 10 trials, 5 labels per-class. (**b**) The log-first order condition, i.e. empirical rate of convergence of Projected gradient method and SSM on F-MNIST with 5 labels per-class.

Implementation Details

We implemented our algorithms in Python using the JAX framework Bradbury et al. [2018] with the GraphLearning API Calder [2022] on a single RTX 2080 Ti. In particular, we exploit JAX's capability to vectorize batched computation and compilation to XLA via the *jit* decorator. XLA facilitates hardware acceleration and the entire framework may exploit GPU and multi-GPU-based parallelism without returning to a Python interpreter.

Experimental Setup

We evaluated our method on three datasets: MNIST Lecun et al. [1998], Fashion-MNIST Xiao et al. [2017] and CIFAR-10 Krizhevsky and Hinton [2009]. As in Calder et al. [2020], we used pretrained autoencoders as feature extractors. For MNIST and Fashion-MNIST, we used variational autoencoders with 3 fully connected layers of sizes (784,400,20) and (784,400,30), respectively, followed by a symmetrically defined decoder. The autoencoder was trained for 100 epochs on each dataset. The autoencoder architecture, loss, and training are similar to Kingma and Welling [2014].

For each dataset, we constructed a graph over the latent feature space. We used all available data to construct the graph, giving $n = 70,000$ nodes for MNIST and Fashion-MNIST, and $n = 60,000$ nodes for CIFAR-10. The graph was constructed as a K-nearest neighbor graph with Gaussian weights given by

$$w_{ij} = \exp\left(-4|x_i - x_j|^2/d_K(x_i)^2\right),$$

where x_i represents the latent variables for image i, and $d_K(x_i)$ is the distance in the latent space between x_i and its K^{th} nearest neighbor. We used $K = 10$ in all experiments. The weight matrix was then symmetrized by replacing W with $\frac{1}{2}(W + W^\top)$.

We compare our SSM approach and alignment-based approximation (Procrustes-SSL) against Laplace learning Zhu et al. [2003], Poisson learning Calder et al. [2020], lazy random walks Zhou et al. [2004, 2003], weighted nonlocal Laplacian (WNLL) Shi et al. [2017], *p*-Laplace learning Flores et al. [2019], and Laplacian Eigenmaps SSL (LE-SSL)Belkin and Niyogi [2002]. In all experiments, we evaluated our method on 10 trials with randomly chosen labeled data points.

Numerical results

Table 2.9. CIFAR-10: Average accuracy scores over 100 trials with standard deviation in brackets.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP Zhu et al. [2003]	10.4 (1.3)	11.0 (2.1)	11.6 (2.7)	12.9 (3.9)	14.1 (5.0)
Nearest Neighbor	31.4 (4.2)	35.3 (3.9)	37.3 (2.8)	39.0 (2.6)	40.3 (2.3)
Random Walk Zhou et al. [2004]	36.4 (4.9)	42.0 (4.4)	45.1 (3.3)	47.5 (2.9)	49.0 (2.6)
WNLL Shi et al. [2017]	16.6 (5.2)	26.2 (6.8)	33.2 (7.0)	39.0 (6.2)	44.0 (5.5)
p-Laplace Flores et al. [2019]	26.0 (6.7)	35.0 (5.4)	42.1 (3.1)	48.1 (2.6)	49.7 (3.8)
Poisson Calder et al. [2020]	40.7 (5.5)	46.5 (5.1)	49.9 (3.4)	**52.3 (3.1)**	53.8 (2.6)
LE-SSL Belkin and Niyogi [2002]	16.2 (0.1)	36.5 (0.1)	44.4 (0.1)	43.0 (0.0)	46.1 (0.0)
Procrustes-SSL	36.2 (0.1)	40.6 (0.1)	44.8 (0.1)	42.9 (0.0)	45.6 (0.0)
Procrustes-SSL + SSM	**40.9 (6.1)**	**47.3 (5.9)**	**50.2 (4.3)**	52.1 (4.3)	**54.7 (3.4)**

# Labels per class	**10**	**20**	**40**	**80**	**160**
Laplace/LP Zhu et al. [2003]	21.8 (7.4)	38.6 (8.2)	54.8 (4.4)	62.7 (1.4)	66.6 (0.7)
Nearest Neighbor	43.3 (1.7)	46.7 (1.2)	49.9 (0.8)	52.9 (0.6)	55.5 (0.5)
Random Walk Zhou et al. [2004]	53.9 (1.6)	57.9 (1.1)	61.7 (0.6)	65.4 (0.5)	68.0 (0.4)
WNLL Shi et al. [2017]	54.0 (2.8)	60.3 (1.6)	64.2 (0.7)	66.6 (0.6)	68.2 (0.4)
p-Laplace Flores et al. [2019]	56.4 (1.8)	60.4 (1.2)	63.8 (0.6)	66.3 (0.6)	**68.7 (0.3)**
Poisson Calder et al. [2020]	58.3 (1.7)	61.5 (1.3)	63.8 (0.8)	65.6 (0.6)	67.3 (0.4)
LE-SSL Belkin and Niyogi [2002]	47.9 (0.0)	50.4 (0.0)	46.5 (0.0)	45.0 (0.0)	46.7 (0.0)
Procrustes-SSL	46.1 (0.0)	50.0 (0.0)	46.9 (0.0)	45.5 (0.0)	46.9 (0.0)
Procrustes-SSL + SSM	**59.4 (2.3)**	**62.4 (1.7)**	**64.9 (1.1)**	**66.6 (0.4)**	68.4 (0.4)

Tables 2.7, 2.9, 2.11 and 2.8 show the average accuracy and standard deviation over all 100 trials for various label rates. We demonstrate consistent improvement over the state-of-the-art. In particular, our method strictly improves over relevant methods on all datasets at a variety of label rates ranging from low (1 label) to low-medium (160) and high (4000).

In particular, on all datasets, the proposed method matches or exceeds the performance of relevant methods, except in two instances: Poisson learning Calder et al. [2020] on CIFAR-10

Table 2.10. Scaling behavior as the number of labeled vertices increases beyond the low label rate regime: Average accuracy scores over 10 trials. Note that standard deviation across trials approaches zero for all methods. We use the publicly available implementation of Poisson Learning Calder [2019] with default parameters.

# Labels per class	**500**	**1000**	**2000**	**4000**
MNIST				
Laplace/LP	**97.8**	97.9	**98.1**	**98.3**
Poisson	97.1	96.9	96.3	93.8
Procrustes-SSL + SSM	**97.8**	**98.0**	**98.1**	98.2
FashionMNIST				
Laplace/LP	84.0	84.7	85.3	85.8
Poisson	82.2	78.9	74.9	58.6
Procrustes-SSL + SSM	**84.3**	**84.8**	**85.9**	**86.1**

Table 2.11. Tool comparison: wall time per-iteration, # iterations to reach |grad| <= 10e-5 (– denotes no convergence), and accuracy using MNIST digits restricted to 0-5 with 1 label / class.

	wall time / iter	# iter to crit. point	accuracy
SSM	6.1	7	0.99
TR	145.5	10	0.94
RG	3.4	–	0.75

with 4 labels per class and P-Laplace Calder [2018] on CIFAR-10 with 160 labels per class. We see that while Laplacian Eigenmaps SSL achieves better performance at higher label rates relative to Procrustes-SSL, at lower label rates Procrustes Analysis is significantly more accurate. Future work includes exploring the effect of taking powers of L_0 on the alignment. We highlight the discrepancy between the approximate method we introduce (Procrustes-SSL) and our SSM-based refinement. This indicates the importance of refinement for recovering good critical points of eq. (2.22).

We additionally evaluate the scaling behavior of our method at intermediate and high label rates. In Table 2.11, we compare our method to Laplace learning and Poisson learning on MNIST and Fashion-MNIST with 500, 1000, 2000, and 4000 labels per class. We see significant degradation in the performance of Poisson Learning, however, our method maintains high-quality predictions in conjunction with Laplace learning. These results imply that while Laplace learning suffers degneracy at low label rates and Poisson Learning seemingly degrades at large label rates, our framework is able to reliably in both regimes—covering the spectrum of low and high supervised sampling rates.

Figure 2.14a shows the accuracy of SSM at 5 labels per class as a function of the number of neighbors K used in constructing the graph, showing that the algorithm is not significantly sensitive to this choice. In Figure 2.14b, we demonstrate the convergence behavior of SSM and the projected gradient method discussed previously by plotting the norm of the first order condition: $||LX_tC - BC^{1/2} - X_t\Lambda_t||$. Note that while both methods are guaranteed to monotonically reduce the objective of eq. (2.22) via line search, SSM rapidly converges to a critical point, while the projected gradient method fails to converge, even after 100 iterations.

Chapter 2, in full, is a reprint of material comprising one submitted paper: Revisiting Semi-Supervised Laplacian Eigenmaps via Alignment and three publications: Placement Initialization via Sequential Subspace Optimization with Sphere Constraints, 2023, International Symposium on Physical Design (ISPD) (best paper nomination), Placement Initialization via a Projected Eigenvector Algorithm, 2022, Design Automation Conference (DAC) 2022, and Net Separation-Oriented Printed Circuit Board Placement via Margin Maximization, 2022, 27th Asia and South Pacific Design Automation Conference (ASP-DAC) (best paper award). The dissertation author was the primary investigator and author of these papers.

Chapter 3

Conclusion

3.1 Conclusion

In this dissertation, we propose a practical and principled framework for large-scale quadratic optimization on Stiefel Manifolds. Our iterative method minimizes the quadratic over a sequence of varying subspaces, and we demonstrate global convergence for this method and a projected gradient method. We also introduce an effective initialization scheme based on the eigenvectors of the system matrix and a transformation motivated by the first-order necessary condition.

We conduct a detailed numerical study of our framework applied to graph problems in the context of chip placement and semi-supervised learning. Specifically, we show that applying our framework as an initialization step for downstream layout engines consistently outperforms the state-of-the-art placements with respect to post-detailed placement layout quality for eight VLSI designs and 12 PCB designs. We also explore an extension of our method that facilitates the minimization of linear wirelength.

In addition, we introduce a formulation of Laplacian Eigenmaps with label constraints as a non-convex Quadratically Constrained Quadratic Program, motivated by the robustness of semi-supervised Laplacian Eigenmaps in low label rate regimes. Application of our framework to this reformulation results in classification solutions that consistently outperform the state-of-the-art with respect to semi-supervised accuracy in low, medium, and high label rate settings.

3.1.1 Future Directions

Future work includes a more careful analysis of the problem in eq. (2.22) and our generalization of SSM—particularly verification of convergence rates from Hager [2001], Hager and Park [2005], Absil et al. [2007]. Continuing to improve scalability is another notable direction; e.g. by exploring ways that we might relax the necessity of computing Z in eq. (1.2.4) exactly. Other directions includes investigation of methods to *query* labeled points—i.e. active learning, and imbalanced label settings. For example, there might be many to potential ways to select "the best" vertices to label. E.g., to better condition the optimization problem, one may want to select vertices that result in the columns of B being correlated with the eigenvectors of L:

For simplicity, take $k = 2$. And denote λ_1, λ_2 the eigenvalues of Λ. We can prove that for any local minimizer X, if the associated eigenvalues λ_1, λ_2 are both less than d_1, the smallest nonzero eigenvalue of L, that X is a global minimizer. As we have shown, with proper initialization, our method recovers a local minimizer X such that λ_1, λ_2 are both less than d_2. There are several potential avenues to prove that any local minimizer is also a global minimizer. For example we could impose an eigenvector-preserving transformation to L or restrict our analysis to graphs such that $\lambda_i = \lambda_k$ where $\lambda_i \geq 0$. An alternative method is to ensure a non-degenerate condition on B as in canonical trust-region methods. Briefly, let $v_1, \ldots, v_k$ be the eigenvectors of L corresponding to k smallest nonzero eigenvalues $d_1 \leq \ldots \leq d_k$. At a high level, we need to ensure that the columns of B are not nearly orthogonal to $v_1, \ldots, v_r$

Proposition 3.1.1. *Let $V = [v_1, v_2] \in \mathbb{R}^{n \times 2}$ be the eigenvectors of L corresponding to the smallest two nonzero eigenvalues $d_1 \leq d_2$. Let X be a local solution satisfying the first order condition*

$$AX = X\Lambda + B$$

and second order condition $\lambda_1, \lambda_2 \leq d_2$. Let s_1 be the smallest singular value of $V^\top B$. Suppose

$$s_1 > d_2 - d_1$$

Then, the eigenvalues λ_1, λ_2 of the Lagrangian multipliers Λ are less than d_1 and X is a global minimizer.

Proof. Recall that the second order condition is

$$\lambda_j \leq d_2 \text{ for } j = 1, 2.$$

Consider the first order condition $LX = X\Lambda + B$. Taking an inner product with v_2 and v_1 yields

$$d_2 v_2^\top X = v_2^\top LX = (v_2^\top X)\Lambda + v_2^\top B,$$
$$d_1 v_1^\top X = v_1^\top LX = (v_1^\top X)\Lambda + v_1^\top B.$$

Let u_1 and u_2 be eigenvectors of Λ associated with eigenvalues λ_1, λ_2. For $j = 1, 2$, taking the product with u_j on the r.h.s yields

$$(d_2 - \lambda_j) v_2^\top X u_j = v_2^\top B u_j, \ (d_1 - \lambda_j) v_1^\top X u_j = v_1^\top B u_j$$

Since $\|V\| = 1 = \|X\|$, then $\|V^\top X u_j\| \leq \|X u_j\| \leq 1$. Thus,

$$|d_2 - d_1| < s_1 \leq \|V^\top B u_j\| \leq \min(|d_2 - \lambda_j|, |d_1 - \lambda_j|)\|V^\top X u_j\| \leq \min(|d_2 - \lambda_j|, |d_1 - \lambda_j|)$$

which implies $\lambda_j < d_1$.

Appendix A

A.1 Additional Derivations

$$\min_{X:X\in\mathrm{St}(n,k)} \langle X, LXC\rangle - \langle X, BC^{1/2}\rangle \tag{A.1}$$

Quadratic minimization over the Stiefel manifold is a generalization of the well-known nonconvex quadratic over the unit ball or sphere. These problems often arise in trust region methods Sorensen [1982], Conn et al. [2000]. Notably, there could exist many local solutions in A.1. We will demonstrate convergence to critical points of two iterative methods: a gradient projection method and the Sequential Subspace Method proposed in this work. Furthermore, when the subspaces component to the SSM algorithm contain the span of the first k nontrivial eigenvectors of L, we show that the quality of the stationary point is characterized by the eigenvalues of the Lagranage multipliers Λ.

First denote the the eigenvector decomposition of L be given by

$$L = [v_1, v_2, \ldots, v_n]\mathrm{diag}(d_1, \ldots, d_n)[v_1, v_2, \ldots, v_n].T$$

To summarize, in Chapter 1 of this dissertation, we have shown the following:

- We propose a gradient projection method to solve A.1. We prove convergence with the Armijo rule in Prop. 1.2.3.

- We analyze the proposed iterative method in the framework of SSM, constructing a sequence of subspaces to reach one approximation of local solutions, where the subspace consists of columns of X_t, $LX_tC - BC^{1/2}$ and eigenvectors associated with d_i, $i = 1, \dots, k$. Each sub-problem can be solved by the aforementioned gradient projection method. We demonstrate global convergence to high-quality critical points. Theoretically, when $C = I$ and the multiplicity of the first nontrivial eigenvalue is greater than or equal to k, a local solution is actually a global solution.

A.1.1 Computational results

Derivation of B and C

Given $\mathbf{1}^\top X = 0$ and $X^\top X = pI$, we have that $\mathbf{1}^\top X_u = -r^\top$ and $X_u^\top X_u = pI - X_l^\top X_l = (p - \widetilde{p})I$, where $r = \mathbf{1}^\top X_l$.

Let $(\widetilde{X_u})_i = (X_u)_i + \frac{1}{n}r$. We have the following:

1. $\mathbf{1}^\top \widetilde{X_u} = 0$

2. $X_u^\top X_u = \widetilde{X_u}^\top \widetilde{X_u} + \frac{1}{n} r^\top r$

Thus, $\widetilde{X_u}^\top \widetilde{X_u} = (p - \widetilde{p})I - \frac{1}{n} r^\top r$

Consider $\langle X_0, LX_0 \rangle = X_u^\top L_u X_u + 2X_u^\top L_{ul} X_l + X_l^\top L_l X_l$. Let $R = \frac{1}{n}\begin{bmatrix} r^\top \\ \vdots \\ r^\top \end{bmatrix} \in \mathbb{R}^{n \times k}$ We have that

$$
\begin{aligned}
\langle X_0, LX_0 \rangle &= \widetilde{X_u}^\top L_u \widetilde{X_u} - 2\widetilde{X_u}^\top L_u R \\
&+ R \top L_u R + 2\widetilde{X_u}^\top L_{ul} X_l - 2R^\top L_{ul} X_l + X_l^\top L_l X_l \\
&= \widetilde{X_u}^\top L_u \widetilde{X_u} + 2\widetilde{X_u}^\top (-L_u R + L_{ul} X_l) \\
&+ R^\top L_u R - 2R^\top L_{ul} X_l + X_l^\top L_l X_l
\end{aligned}
$$

Bibliography

Freerouting. URL https://freerouting.org/.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, USA, 2007. ISBN 0691132984.

S. N. Adya and Igor L. Markov. Combinatorial techniques for mixed-size placement. *ACM Trans. Des. Autom. Electron. Syst.*, 10, 2005. ISSN 1084-4309. doi: 10.1145/1044111.1044116.

A. El Kacimi Alaoui. Asymptotic behavior of ℓ_p-based laplacian regularization in semi-supervised learning. In *Annual Conference Computational Learning Theory*, 2016.

Alex Alexandridis, Evangelos Paizis, Eva Chondrodima, and Marios Stogiannos. A particle swarm optimization approach in printed circuit board thermal design. *Integ. Comp.-Aided Eng.*, 24, 2017. ISSN 1069-2509. doi: 10.3233/ICA-160536.

C.J. Alpert and A.B. Kahng. Simple eigenvector-based circuit clustering can be effective. In *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, volume 4, pages 683–686, 1996.

C.J. Alpert, T.F. Chan, A.B. Kahng, I.L. Markov, and P. Mulet. Faster minimization of linear wirelength for global placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(1):3–13, 1998.

Rie Ando and Tong Zhang. Learning on graph with laplacian regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/file/d87c68a56bc8eb803b44f25abb627786-Paper.pdf.

Branko Arsic, Dragos Cvetkovic, Slobodan Simic, and Milan Škarić. Graph spectral techniques in computer sciences. *Applicable Analysis and Discrete Mathematics*, 6:1–30, 04 2012. doi: 10.2298/AADM111223025A.

T. Badriyah, F. Setyorini, and N. Yuliawan. The implementation of genetic algorithm and routing lee for pcb design optimization. In *Int. Conf. on Informatics and Computing*, 2016.

Mikhail Belkin and Partha Niyogi. Using manifold structure for partially labelled classification. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, page 953–960, Cambridge, MA, USA, 2002. MIT Press.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

Kristin P. Bennett and Erin J. Bredensteiner. Duality and geometry in svm classifiers. In *Int. Conf. on Machine Learning*, 2000. ISBN 1558607072.

D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Jeff Calder. The game theoretic p-laplacian and semi-supervised learning with few labels. *Nonlinearity*, 32(1):301, dec 2018. doi: 10.1088/1361-6544/aae949. URL https://dx.doi.org/10.1088/1361-6544/aae949.

Jeff Calder. Consistency of lipschitz learning with infinite unlabeled data and finite labeled data. *SIAM Journal on Mathematics of Data Science*, 1(4):780–812, 2019. doi: 10.1137/18M1199241. URL https://doi.org/10.1137/18M1199241.

Jeff Calder. Graphlearning python package, January 2022. URL https://doi.org/10.5281/zenodo.5850940.

Jeff Calder and Dejan Slepčev. Properly-weighted graph laplacian for semi-supervised learning. *Appl. Math. Optim.*, 82(3):1111–1159, dec 2020. ISSN 0095-4616. doi: 10.1007/s00245-019-09637-3. URL https://doi.org/10.1007/s00245-019-09637-3.

Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepčev. Poisson learning: Graph based semi-supervised learning at very low label rates. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Emmanuel Candès, Michael Wakin, and Stephen Boyd. Enhancing sparsity by reweighted l1 minimization. *J. of Fourier Analysis and Applications*, 14:877–905, 2007.

So-Zen Yao Charles J. Alpert. Spectral partitioning: The more eigenvectors, the better. In *32nd Design Automation Conference*, pages 195–200, 1995. doi: 10.1109/DAC.1995.250089.

Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages

3869–3872, 2008. doi: 10.1109/ICASSP.2008.4518498.

Pengwen Chen, Chung-Kuan Cheng, Albert Chern, Chester Holtz, Aoxi Li, and Yucheng Wang. Placement initialization via a projected eigenvector algorithm: Late breaking results. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, page 1398–1399, 2022.

C. Cheng, A. B. Kahng, I. Kang, and L. Wang. Replace: Advancing solution quality and routability validation in global placement. *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 2018. ISSN 0278-0070. doi: 10.1109/TCAD.2018.2859220.

Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 273–282, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306911. doi: 10.1145/1993636.1993674. URL https://doi.org/10.1145/1993636.1993674.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. doi: 10.1137/1.9780898719857. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719857.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3), 1995.

Dragoš M. Cvetković, Michael Doob, and Horst Sachs. *Spectra of Graphs: Theory and Application*. Academic Press, New York, 1980. ISBN 0121951502.

Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C. Sinann Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Sov. Math., Dokl.*, 8:674–675, 1967. ISSN 0197-6788.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973. doi: 10.1147/rd.175.0420.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954. URL https://doi.org/10.1137/S0895479895290954.

Alina Ene and Adrian Vladu. Improved convergence for ℓ_∞ and ℓ_1 regression via iteratively reweighted least squares. *arXiv*, abs/1902.06391, 2019.

Jennifer Erway, Philip Gill, and Josh Griffin. Iterative methods for finding a trust-region step.

SIAM Journal on Optimization, 20:1110–1131, 01 2009. doi: 10.1137/070708494.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2): 298–305, 1973. URL http://eudml.org/doc/12723.

Mauricio Flores, Jeff Calder, and Gilad Lerman. Analysis and algorithms for ℓ_p-based semi-supervised learning on graphs, 2019. URL https://arxiv.org/abs/1901.05031.

Mateus Fogaça, Andrew B. Kahng, Ricardo Reis, and Lutong Wang. Finding placement-relevant clusters with fast modularity-based clustering. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, ASPDAC, page 569–576. Association for Computing Machinery, 2019. ISBN 9781450360074.

J. Funke, S. Hougardy, and J. Schneider. An exact algorithm for wirelength optimal placements in VLSI design. *Integration*, 52, 2016. doi: 10.1016/j.vlsi.2015.07.001.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

Jiaqi Gu, Zixuan Jiang, Yibo Lin, and David Z. Pan. Dreamplace 3.0: Multi-electrostatics based robust vlsi placement with region constraints. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2020.

L. Hagen and A. Kahng. Fast spectral methods for ratio cut partitioning and clustering. In *1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers*, pages 10–13, 1991. doi: 10.1109/ICCAD.1991.185177.

William Hager and Soonchul Park. Global convergence of ssm for minimizing a quadratic over a sphere. *Math. Comput.*, 74:1413–1423, 07 2005. doi: 10.1090/S0025-5718-04-01731-4.

William W. Hager. Minimizing a quadratic over a sphere. volume 12, 2001.

Kenneth M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3), November 1970. doi: 10.1287/mnsc.17.3.219.

T. Hamada, C. . Cheng, and P. M. Chau. An efficient multilevel placement technique using hierarchical partitioning. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, 39(6), 1992. doi: 10.1109/81.153634.

Chester Holtz, Pengwen Chen, Alexander Cloninger, Chung-Kuan Cheng, and Gal Mishne. Revisiting semi-supervised laplacian eigenmaps via alignment.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge; New York, 2nd edition, 2013. ISBN 9780521839402.

M. Hsu, Y. Chang, and V. Balabanov. Tsv-aware analytical placement for 3d ic designs. In *ACM Trans. Des. Autom. Conf.*, volume 48, 2011.

Meng-Kai Hsu, Valeriy Balabanov, and Yao-Wen Chang. Tsv-aware analytical placement for 3-d ic designs based on a novel weighted-average wirelength model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(4):497–509, 2013. doi: 10.1109/TCAD.2012.2226584.

Martin Imre, Jun Tao, Yongyu Wang, Zhiqiang Zhao, Zhuo Feng, and Chaoli Wang. Spectrum-preserving sparsification for visualization of big graphs. *Computers & Graphics*, 87:89–102, 2020. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2020.02.004. URL https://www.sciencedirect.com/science/article/pii/S0097849320300212.

Fatimah Sham Ismail, Rubiyah Yusof, and Marzuki Khalid. Optimization of electronics component placement design on pcb using self organizing genetic algorithm. *J. of Intell. Manuf.*, 21, 2012. doi: 10.1007/s10845-010-0444-x.

Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoḡlu. Auction dynamics: A volume constrained mbo scheme. *Journal of Computational Physics*, 354:288–310, 2018. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2017.10.036. URL https://www.sciencedirect.com/science/article/pii/S0021999117308033.

Sakait Jain and Hae Chang Gea. PCB Layout Design Using a Genetic Algorithm. *J. of Electronic Packaging*, 118(1), 1996. ISSN 1043-7398. doi: 10.1115/1.2792119.

Andrew B. Kahng, Sherief Reda, and Qinke Wang. Aplace: A general analytic placement framework. In *Proc. of the Int. Sym. on Phys. Des.*, 2005. ISBN 1595930213. doi: 10.1145/1055137.1055187.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Gustav R. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 148:497–508, 1847.

Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001. doi: 10.1137/S1064827500366124. URL https://doi.org/10.1137/S1064827500366124.

Yehuda Koren. On spectral graph drawing. In Tandy Warnow and Binhai Zhu, editors, *Computing and Combinatorics*, pages 496–508, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

ISBN 978-3-540-45071-9.

Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving sdd linear systems. *SIAM Journal on Computing*, 43(1):337–354, 2014. doi: 10.1137/110845914. URL https://doi.org/10.1137/110845914.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. (0), 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Chen Li, Min Xie, Cheng-Kok Koh, J. Cong, and P.H. Madden. Routability-driven placement and white space allocation. In *IEEE/ACM Int. Conf. on Computer Aided Des.*, 2004. doi: 10.1109/ICCAD.2004.1382607.

Ting-Chou Lin, Devon Merrill, Yen-Yi Wu, Chester Holtz, and Chung-Kuan Cheng. A unified printed circuit board routing algorithm with complicated constraints and differential pairs. In *Proc. of the 26th Asia and South Pacific Design Automation Conf.*, 2021. ISBN 9781450379991. doi: 10.1145/3394885.3431568.

Yibo Lin, Shounak Dhar, Wuxi Li, Haoxing Ren, Brucek Khailany, and David Z. Pan. Dreamplace: Deep learning toolkit-enabled gpu acceleration for modern vlsi placement. In *Proc. of the 56th Annual Design Automation Conf.*, 2019. ISBN 978-1-4503-6725-7. doi: 10.1145/3316781.3317803.

Jingwei Lu, Pengwen Chen, Chin-Chih Chang, Lu Sha, Dennis Jen-Hsin Huang, Chin-Chi Teng, and Chung-Kuan Cheng. eplace: Electrostatics-based placement using fast fourier transform and nesterov's method. *ACM Trans. Des. Autom. Electron. Sys.*, 20, 2015a. ISSN 1084-4309. doi: 10.1145/2699873.

Jingwei Lu, Hao Zhuang, Pengwen Chen, Hongliang Chang, Chin-Chih Chang, Yiu-Chung Wong, Lu Sha, Dennis Huang, Yufeng Luo, Chin-Chi Teng, and Chung-Kuan Cheng. eplace-ms: Electrostatics-based placement for mixed-size circuits. *IEEE TCAD*, 34(5):685–698, 2015b.

Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, page 1330–1338, Red Hook, NY, USA, 2009. Curran Associates Inc. ISBN 9781615679119.

Gi-Joon Nam, Charles J. Alpert, Paul Villarrubia, Bruce Winter, and Mehmet Yildiz. The ispd2005 placement contest and benchmark suite. In *ISPD*, 2005.

Jorge Nocedal and Stephen J. Wright, editors. *Sequential Quadratic Programming*, pages 526–573. Springer New York, New York, NY, 1999. ISBN 978-0-387-22742-9. doi: 10.1007/0-387-22742-3_18. URL https://doi.org/10.1007/0-387-22742-3_18.

Panos M. Pardalos and Stephen A. Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *J. of Global Optimization*, 1:15–22, 1991.

Jarrod A. Roy and Igor L. Markov. Seeing the forest and the trees: Steiner wirelength optimization in placement. *IEEE Trans. on Comp.-Aided Design of Integrated Circuits and Systems*, 26(4), 2007. doi: 10.1109/TCAD.2006.888260.

Amina Shabbeer, Cagri Ozcaglar, and Kristin P. Bennett. Crossing minimization within graph embeddings, 2012.

Zuoqiang Shi, Stanley Osher, and Wei Zhu. Weighted nonlocal laplacian on interpolation from sparse data. *Journal of Scientific Computing*, 73(2-3), 4 2017. ISSN 0885-7474. doi: 10.1007/s10915-017-0421-z. URL https://www.osti.gov/biblio/1537761.

Dejan Slepčev and Matthew Thorpe. Analysis of p-laplacian regularization in semisupervised learning. *SIAM Journal on Mathematical Analysis*, 51(3):2085–2120, 2019. doi: 10.1137/17M115222X. URL https://doi.org/10.1137/17M115222X.

D. C. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982. doi: 10.1137/0719026. URL https://doi.org/10.1137/0719026.

Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, jan 2014. ISSN 0895-4798.

P. Spindler and F. M. Johannes. Fast and accurate routing demand estimation for efficient routability-driven placement. In *2007 Design, Automation Test in Europe Conf.*, 2007. doi: 10.1109/DATE.2007.364463.

Shang-Hua Teng. 2016.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *J. Mach. Learn. Res.*, 17(1):4755–4759, jan 2016. ISSN 1532-4435.

Natarajan Viswanathan and Chris Chu. Fastplace: Efficient analytical placement using cell shifting, iterative local refinement, and a hybrid net model. pages 26–33, 2004.

Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In

Proceedings of the 25th International Conference on Machine Learning, ICML '08, page 1120–1127, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390297. URL https://doi.org/10.1145/1390156.1390297.

David Wipf and Srikantan Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE J. of Selected Topics in Signal Processing*, 4:317–329, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.

Zhiqiang Zhao, Yongyu Wang, and Zhuo Feng. Samg: Sparsified graph-theoretic algebraic multigrid for solving large symmetric diagonally dominant (sdd) matrices. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 601–606, 2017. doi: 10.1109/ICCAD.2017.8203832.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

Dengyong Zhou, Bernhard Schölkopf, C.E. Rasmussen, Heinrich Bülthoff, and Martin Giese. Learning from labeled and unlabeled data using random walks. volume 3175, 08 2004. ISBN 978-3-540-22945-2. doi: 10.1007/978-3-540-28649-3_29.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 1036–1043, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102482. URL https://doi.org/10.1145/1102351.1102482.

Xueyuan Zhou and Mikhail Belkin. Semi-supervised learning by higher order regularization. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 892–900, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/zhou11b.html.

Xueyuan Zhou and Nathan Srebro. Error analysis of laplacian eigenmaps for semi-supervised learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 901–908, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/zhou11c.html.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 912–919. AAAI Press, 2003.

ISBN 1577351894.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.