Manuscript: Quantum Graph Kernels, Symmetries, and Speedups

Jue Xu*

June 23, 2022

Abstract

We discuss the quantum analogue of diffusion kernels on graphs. prove the quantum speedup in terms of groups and symmetries. motivation

Contents

*juexu@cs.umd.edu

1 Introduction

group theoretical methods in machine learning by Kondor [Kon08]. diffusion kernel on graphs [KL02]

Many insightful and powerful models, like adiabatic quantum computation [Far+00], quantum random walks [Chi04]

1.1 Preliminary and Notations

a datum $\mathbf{x} := (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ is a vector where d is the number of *features*; dataset $\left\{(\mathbf{x}^{(i)}, y^{(i)})\right\}_{i=1}^{m}$ where the *label* $y \in \Sigma$ with some (discrete set) alphabet/category Σ. for binary classification, we have $\Sigma = \{-1, 1\}$, Ω; computational basis $|z\rangle$ with $z \in \{1, 2, \ldots, 2^n\}$ where n is the number of qubits, the binary representation $|\mathbf{x}\rangle = |x_1, x_2, \ldots, x_n\rangle, x_j \in \{0, 1\}$. $|0\rangle^n$, $|\pm\rangle$. graph G, group $\mathbb{G}$

Definition 1 (Inner product)**.** *inner product* $\langle \mathbf{x}, \mathbf{x}' \rangle$. Dirac notation $\langle \mathbf{x}|\mathbf{x}' \rangle$

'similarity' between two vectors ?

Definition 2 (Hilbert space)**.** *Hilbert space* $\mathcal{H}$ induced by *inner product*

1.2 Terminology: SVM and Kernel trick

supervised learning [1]

Support Vector Machine (SVM)

training stage, classification stage. exponentially large space; *hyperplane* $(\mathbf{w}, b)$ parametrized by a normal vector $\mathbf{w} \in \mathbb{R}^n$ and a *bias* term $b \in \mathbb{R}$. in the (high-dimensional) *feature space*. maximize the *margin* [ref]

$$f^* = \arg\max_f L(y, \tilde{y}) + \|\| \tag{1}$$

where the loss function L, slackness. called *support vector*.

Kernel trick (method)

kernel trick: *feature map* the input data to higher dimension such that the data are linearly separatable in this feature space. only depend on the inner product to avoid the expensive (exponential) calculation. [ref]

Definition 3 (Kernel function)**.** a *kernel function* (mapping) $\mathcal{K} : \Omega \times \Omega \to \mathbb{R}$, is defined as Inner product

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}} \tag{2}$$

w.r.t a Feature map $\Phi(\mathbf{x})$.

Remark 1. A function $\mathcal{K}$ (k) is a valid kernel (in machine learning) if and only if? the matrix $\mathcal{K}(x, x')$ is symmetric and *positive semi-definite* (PSD, p.s.d).

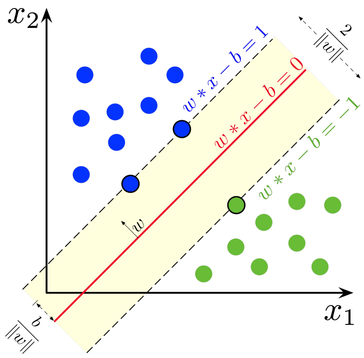
Definition 4 (Feature map)**.** The *feature map* is a function (mapping)

$$\Phi(\mathbf{x}) : \Omega \to \mathcal{H}_{\mathcal{K}} \tag{3}$$

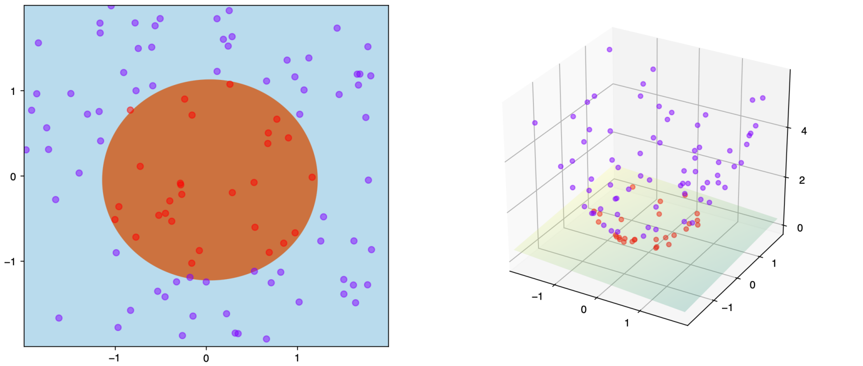
from a low dimensional space non-linearly in to a high dimensional Hilbert space $\mathcal{H}$ which is commonly referred to as the *feature space*. For example, $\Phi(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^n$ with $n \gg d$.

Some common kernels:

[1] unsupervised learning



(a) linearly separable SVM



(b) kernel trick idea: SVM with kernel given by $\phi(\mathbf{x} := (a, b)) = (a, b, a^2 + b^2)$ and thus $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} + \|x\|^2 \|y\|^2$ that only depends on inner product. The training points are mapped to a 3-dimensional space where a separating hyperplane can be easily found. (from Wikipedia: Kernel method)

- polynomial kernel:
- radial:
- *Gaussian (radial basis) kernel*:

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2}\right) \tag{4}$$

Definition 5 (Reproducing Kernel Hilbert Space)**.** It is well known that any continuous, symmetric, positive definite Kernel function has corresponding Hilbert space $\mathcal{H}$, called the *Reproducing Kernel Hilbert Space* (RKHS), which induces a Feature map satisfying Eq. (2).

Theorem 1 (Representer theorem)**.** *?*

2 Graph Kernels and Quantum Random Walks

Largely, this development was driven by the empirical success of supervised learning of vector-valued data or image data. However, in many domains, such as chemo- and bioinformatics, social network analysis or computer vision, observations describe relations between objects or individuals and cannot be interpreted as vectors or fixed grids; instead, they are naturally represented by graphs. [KJM20] graph-structured data is to make use of graph kernels—functions which measure the similarity between graphs. **How similar are two graphs? How similar are two nodes of a graph?**

2.1 Diffusion kernel of a graph (between a pair of vertices)

[KL02]. generalize to regularization: We propose a family of regularization operators (equivalently, kernels) on graphs that include Diffusion Kernels as a special case, and show that this family encompasses all possible regularization operators invariant under permutations of the vertices in a particular sense. [SK03] with Euclidean space $\Omega = \mathbb{R}^m$

Definition 6 (Adjacency matrix)**.** Given a (undirected, unweighted) graph $G(V, E)$, its *adjacency matrix* $\hat{A}$ is defined as

$$\hat{A}(v, v') := \begin{cases} 1, & (v, v') \in E \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where the matrix entry is 1 if the two vertices v, v' (labels of the column and the row) are connected by an edge, otherwise the entry is 0.

Definition 7 (Graph Laplacian)**.** With the adjacency matrix $\hat{A}$, the graph Laplacian is

$$\hat{\mathfrak{L}} := \hat{A} - \hat{D} \tag{6}$$

where $\hat{D}_{v,v} := \mathsf{deg}(v)$ is its diagonal degree matrix. The adjacency matrix and graph Laplacian of weighted graphs can be defined analogously.

Remark 2. Graph Laplacian $\hat{\mathfrak{L}}$ is the discrete version of (continuous) Laplacian operator ∇^2. [Chu97]

Lemma 1. *matrix exponential of any ? $\hat{H}$ i.e., $e^{\beta\hat{H}}$, is a valid kernel (PSD). [to verify for $e^{-\mathrm{i}t\hat{H}}$]*

Definition 8 (Diffusion kernel)**.** The *diffusion kernel* of a graph w.r.t a pair of vertices (v, v') is $\mathcal{K}(v, v') : V(G) \to \mathbb{R}$, i.e.,

$$\mathcal{K}(v, v') := \left[\sum_k \frac{\beta^k}{k!}\hat{\mathfrak{L}}^k\right]_{v,v'} = \left[e^{\beta\hat{\mathfrak{L}}}\right]_{v,v'} \tag{7}$$

where Ω is the set of vertices of graph G.

2.1.1 Diffusion, heat equation, and random walk

The continuous-time random walk on G is defined as the **solution of the differential equation**

$$\frac{\mathrm{d}}{\mathrm{d}t}p_j(t) = \sum_{k\in V}\hat{\mathfrak{L}}_{jk}\ p_k(t), \tag{8}$$

where $p_j(t)$ denotes the probability associated with vertex j at time t and $\hat{\mathfrak{L}}$ is Graph Laplacian. Since the columns of $\hat{\mathfrak{L}}$ sum to 0, an initially normalized distribution remains normalized: the evolution of the continuous-time random walk for any time t is a *stochastic process*. *random walk* (discrete-space, discrete-time), *heat equation*

$$u_t = \alpha\nabla^2 u \equiv \alpha\Delta u \tag{9}$$

continuous-space (continous-time) the solution to Eq. (9) is called heat kernel $\mathcal{K}(t; x, y) = \frac{1}{4\pi t}e^{-|x-y|^2/4t}$ (Gaussian).

2.2 Continuous-time quantum random walk

The continuous-time quantum random walk [CFG02] is the quantum analogue of classical diffusion (continuous-time random walk). By a direct observation, Eq. (8) is very similar to the time-dependent (evolution) schrodinger equation governed by a Hamiltonian operator $\hat{H}$

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\psi\rangle = \hat{H}|\psi\rangle \tag{10}$$

except that the factor of i$\hbar$.

Definition 9 (Quantum propagator)**.** *Quantum propagator*, *transition amplitude.* Interestingly, this quantity is also called *quantum kernel* (much earlier than the concept of kernel tricks in machine learning),

$$\langle I|e^{-\mathrm{i}t\hat{H}}|F\rangle = \langle I|\hat{U}|F\rangle \tag{11}$$

The quantum propagator can be evaluated by *path integral (Lagrangian) formalism* (see Appendix B.1).

2.3 Graph kernels of a pair of graphs

random walk kernel [Vis+10] quantum kernel [Bai+15] survey [KJM20]

Definition 10 (Graph kernel)**.** Given a pair of graphs (G, G'), a *graph kernel* is a function (mapping) $\mathcal{K}(G, G') : \{G\} \to \mathbb{R}$

[Chu97]

Definition 11 (Product of graphs)**.** *direct product of graphs*;

$$G(V, E) \times G'(V', E') := \left\{(v, v') \in V \times V', ((v, v'), (w, w')) \in E \wedge ??\right\} \tag{12}$$

tensor product of graphs;

$$G \otimes G' := \{\vee\} \tag{13}$$

Cartisan product of graphs;

$$G \times G' = \{\} \tag{14}$$

Kronecker product; *factor graph*; *Hadamard product*;

Remark 3. A natural question to ask is the following: Since diffusion can be viewed as a **continuous time** limit of random walks, can the ideas behind the random walk kernel be extended to diffusion? Unfortunately, the Laplacian of the product graph does not decompose into the Kronecker product of the Laplacian matrices of the constituent graphs; this rules out a straightforward extension. discrete-time quantum random walk (need coin space) [AKR05] [Chi04]; Szegedy's walk formalism. [Sze04]

Remark 4. for diffusion processes on factor graphs the kernel on the factor graph is given by the product of kernels on the constituents, that is

$$k((i, i'), (j, j')) = k(i, j)k'(i', j'). \tag{15}$$

2.4 Discrete-time quantum random walk

Discrete-time quantum random walk is the quantum analogy of random walk

2.5 Relation and examples

2.5.1 Line

(classical) diffusion kernel of a 1-dimensional lattice (line)

$$\mathcal{K}(v, v') = \tag{16}$$

Quantum propagator (quantum kernel)

$$\tilde{\mathcal{K}}(z_F, z_I) = \langle z_F | e^{-\mathrm{i}t\hat{H}_0} | z_I \rangle = \sum_{p=1}^{N} e^{-\mathrm{i}t2\cos\left(\frac{2\pi}{N}p\right) + \mathrm{i}\frac{2\pi}{N}p(z_I - z_F)} \tag{17}$$

$$\approx e^{2\mathrm{i}t}(-\mathrm{i})^d J_d(2t) \tag{18}$$

Remark 5. The random walk on this graph starting from the origin (in either continuous or discrete time) typically moves a distance proportional to $\sqrt{t}$ in time t. In contrast, the quantum random walk spreads as a wave packet with speed 2.

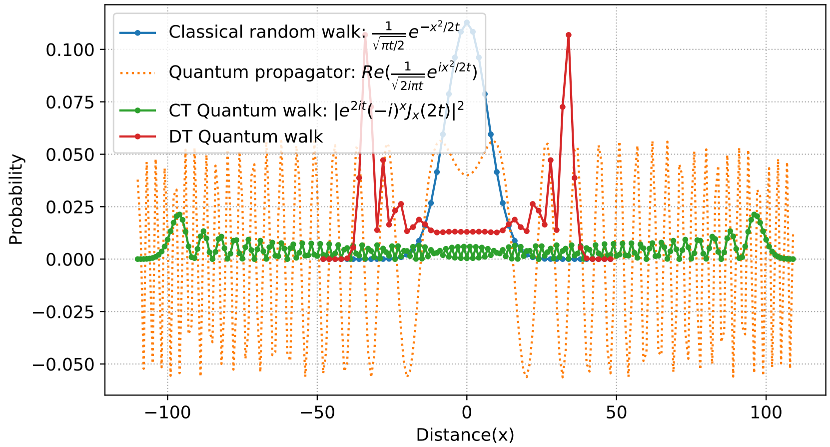


Figure 2

2.5.2 Tree

2.5.3 Hyercube

2.5.4 Cayley graphs

Definition 12 (Cayley graph)**.** Cayley graph is a graph that encodes the abstract structure of a group.

3 Quantum Advantages and Speedups

A quantum version of this approach has already been proposed in [RML14], where an exponential improvement can be achieved if data is provided in a coherent superposition.

Remark 6. However, when data is provided in the conventional way, i.e. from a classical computer, then the methods of [15] cannot be applied.

input model, quantum RAM; quantum-inspired [Tan19]

3.1 Related works

3.1.1 Quantum SVM and kernel tricks

Quantum version of SVM was proposed [RML14] to exploit the power of quantum computer. Naturally, the *quantum kernel estimation* [SK19] [Hav+19] is studied to enhance the performance of quantum SVM.

Definition 13 (Quantum feature map)**.** the quantum state space (Hilbert space) as the feature space to still obtain a quantum advantage mapping the input data non-linearly to a quantum state (density matrix)

$$\Phi(\mathbf{x}) : \Omega \to |\Phi(\mathbf{x})\rangle\langle\Phi(\mathbf{x})| \,, \tag{19}$$

the direct quantum analogy of classical Feature map. On quantum computers, the quantum feature map $\Phi(\mathbf{x})$ is realized by applying a unitary quantum circuit $\hat{U}_{\Phi(\mathbf{x})}$ to a reference state $|0^n\rangle$.

3.1.2 Explicit method (Quantum Variational Classification)

variational quantum circuit: generates a separating hyperplane in the quantum feature space

1. $\mathbf{x} \in \Omega$ (feature) mapped to a quantum state by applying a unitary circuit $U_{\Phi(\mathbf{x})}$ to a reference (initial) state $|0\rangle^n$
2. a short depth quantum circuit $W(\theta)$
3. for binary classification, apply a binary measurement $\{M_y\} = 2^{-1}(\mathbb{1} + y\mathbf{f})$?
4. to obtain the empirical distribution $p_y(\mathbf{x})$, perform repeated measurement shots. then assign the label according to p_y?

3.1.3 Implicit method (Quantum Kernel Estimation)

estimate the kernel function quantumly and implement a conventional SVM. Rather than using a variational quantum circuit to generate the separating hyperplane, we use a classical SVM for classification.

1. the kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}')$ is estimated on a quantum computer
2. the quantum computer is used a second time to estimate the kernel for a new datum (test) $\mathbf{s} \in S$ with all the support vectors.

The kernel entries are the fidelities between different feature vectors. The overlap can be estimated directly from the transition amplitude

Definition 14 (Quantum kernel estimation)**.** The *quantum kernel estimation* is the *Hilbert-Schmidt* Inner product between density matrices

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \mathrm{Tr}\left(|\Phi(\mathbf{x})\rangle\langle\Phi(\mathbf{x})| \cdot \left|\Phi(\mathbf{x}')\right\rangle\left\langle\Phi(\mathbf{x}')\right|\right) = \left|\left\langle\Phi(\mathbf{x})\middle|\Phi(\mathbf{x}')\right\rangle\right|^2 = \left|\langle 0^n|\hat{U}^{\dagger}_{\Phi(\mathbf{x})}\hat{U}_{\Phi(\mathbf{x}')}|0^n\rangle\right|^2. \tag{20}$$

exactly the Quantum propagator.

measure the final state in the Z-basis R-times and record the number of $|0^n\rangle$. The frequency of this string is the estimate of the transition probability. The kernel entry is obtained to an additive sampling error of $\tilde{\epsilon}$ when $\mathcal{O}(\tilde{\epsilon}^{-2})$ shots are used.

Theorem 2 ([Chi+03])**.** *There exists exponential (classiacl-quantum) separation with respect to query complexity under the adjacency matrix (graph) model. (glued tree)*

[Zhe+22]; robust, provable speedup [LAT21]

3.1.4 Quantum graph kernel

quantum graph kernel defined in terms of Jensen Shannon [Bai+15]

3.1.5 Quantum diffusion map

Inspired by random walk on graphs, *diffusion map* (DM) is a class of **unsupervised** machine learning that offers automatic identification of **low-dimensional data structure hidden in a high dimensional dataset** [Sor+21]. Though both are machine learning techniques, diffusion map (for dimension reduction) is the 'reverse direction' of the kernel method.

3.2 Provable: Symmetries, graph properties

structure is required for quantum speedup [AA14] symmetric functions rule out exponential speedup [Ben+20]

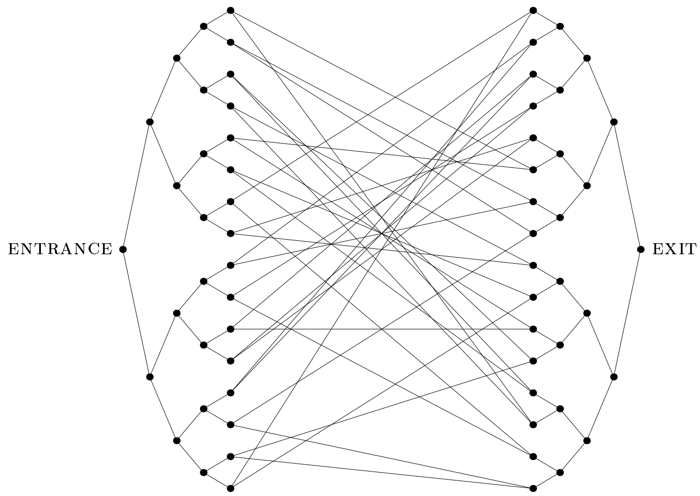


Figure 3: [Chi+03]

3.2.1 Permutation, symmetry, and speedup

3.2.2 FFT and QFT

[KSB09]; quantum linear algebra (qMAT) [ref]: HHL, matrices multiplication, matrix inversion, matrix exponentiation, FFT by quantum algorithms [Sor+21]

3.3 Heuristic: Groups, invariance, symmetries of physical systems

hidden subgroup problem solved by quantum Fourier transformation[CvD10]; rigorous and robust quantum speedup with Discret Logarithm (DLOG) problem [LAT21]

3.3.1 Groups, symmetries in physics

covariant [Gli+21] group theory, [Kon08]; symmetries in physics [Bog+20] [Bog+22]; equivariant CNN [Zhe+22]. (classical) machine learning (neural network) for quantum many-body physics: determining the phase (transition) [CM17] [CT17]

3.3.2 Group theory and machine learning

Definition 15 (Covariant)**.** *covariant*

Definition 16 (Equivariance)**.** A map $f : X \to Y$ is said to be *equivariant* w.r.t. the actions $\rho : \mathbb{G} \times X \to X$ and $\rho' : \mathbb{G} \times Y \to Y$ of a group $\mathbb{G}$ on X and Y if

$$\forall x \in X, g \in \mathbb{G}, f(\rho(g, x)) = \rho'(g, f(x)) \tag{21}$$

4 Experiments

4.1 Datasets and benchmark

preliminary experiment

4.1.1 Artificial data

we generate artificial data that can be fully separated by our feature map.

4.1.2 Real-world dataset

UCI [KL02], protein, JET? [Bog+20]; quantum many-body physics (phase transition) [CM17]

5 Discussion and Conclusion

References

[AA14] Scott Aaronson and Andris Ambainis. "The Need for Structure in Quantum Speedups". Feb. 6, 2014. arXiv: `0911.0996 [quant-ph]` (cit. on p. 7).

[AKR05] Andris Ambainis, Julia Kempe, and Alexander Rivosh. Coins Make Quantum Walks Faster. Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '05. USA: Society for Industrial and Applied Mathematics, Jan. 23, 2005, pp. 1099–1108. arXiv: `quant-ph/0402107` (cit. on p. 5).

[Bai+15] Lu Bai et al. A Quantum Jensen–Shannon Graph Kernel for Unattributed Graphs. *Pattern Recognition* 48.2 (Feb. 2015), pp. 344–355 (cit. on pp. 5, 7).

[Ben+20] Shalev Ben-David et al. Symmetries, Graph Properties, and Quantum Speedups. *2020 IEEE 61st Annu. Symp. Found. Comput. Sci. FOCS* (Nov. 2020), pp. 649–660. arXiv: `2006.12760` (cit. on p. 7).

[Bia+17] Jacob Biamonte et al. Quantum Machine Learning. *Nature* 549.7671 (Sept. 2017), pp. 195–202. arXiv: `1611.09347` (cit. on p. 11).

[Bog+20] Alexander Bogatskiy et al. Lorentz Group Equivariant Neural Network for Particle Physics. Proceedings of the 37th International Conference on Machine Learning. International Conference on Machine Learning. PMLR, Nov. 21, 2020, pp. 992–1002 (cit. on pp. 8, 9).

[Bog+22] Alexander Bogatskiy et al. Symmetry Group Equivariant Architectures for Physics. Mar. 11, 2022. arXiv: `2203.06153 [astro-ph, physics:hep-ex, physics:hep-ph]` (cit. on p. 8).

[CFG02] Andrew M. Childs, Edward Farhi, and Sam Gutmann. An Example of the Difference between Quantum and Classical Random Walks. *Quantum Inf. Process.* 1.1/2 (2002), pp. 35–43. arXiv: `quant-ph/0103020` (cit. on p. 4).

[Chi+03] Andrew M. Childs et al. Exponential Algorithmic Speedup by Quantum Walk. *Proc. Thirty-Fifth ACM Symp. Theory Comput. - STOC 03* (2003), p. 59. arXiv: `quant-ph/0209131` (cit. on pp. 7, 8).

[Chi04] Andrew MacGregor Childs. "Quantum Information Processing in Continuous Time". Thesis. Massachusetts Institute of Technology, 2004 (cit. on pp. 2, 5).

[Chu97] Fan R. K. Chung. Spectral Graph Theory. Regional Conference Series in Mathematics no. 92. Providence, R.I: Published for the Conference Board of the mathematical sciences by the American Mathematical Society, 1997. 207 pp. (cit. on pp. 4, 5).

[CM17] Juan Carrasquilla and Roger G. Melko. Machine Learning Phases of Matter. *Nature Phys* 13.5 (5 May 2017), pp. 431–434. arXiv: `1605.01735` (cit. on pp. 8, 9).

[CT17] Giuseppe Carleo and Matthias Troyer. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science* 355.6325 (Feb. 10, 2017), pp. 602–606. arXiv: `1606.02318` (cit. on p. 8).

[CvD10] Andrew M. Childs and Wim van Dam. Quantum Algorithms for Algebraic Problems. *Rev. Mod. Phys.* 82.1 (Jan. 15, 2010), pp. 1–52. arXiv: 0812.0380 (cit. on p. 8).

[Dir45] P. A. M. Dirac. On the Analogy Between Classical and Quantum Mechanics. *Rev. Mod. Phys.* 17.2-3 (Apr. 1, 1945), pp. 195–199 (cit. on p. 12).

[Far+00] Edward Farhi et al. "Quantum Computation by Adiabatic Evolution". Jan. 28, 2000. arXiv: quant-ph/0001106 (cit. on p. 2).

[FHS10] Richard P. Feynman, Albert R. Hibbs, and Daniel F. Styer. Quantum Mechanics and Path Integrals. Emended ed. Mineola, N.Y: Dover Publications, 2010. 371 pp. (cit. on p. 12).

[Gli+21] Jennifer R. Glick et al. "Covariant Quantum Kernels for Data with Group Structure". May 7, 2021. arXiv: 2105.03406 [quant-ph] (cit. on p. 8).

[Hav+19] Vojtech Havlicek et al. Supervised Learning with Quantum Enhanced Feature Spaces. *Nature* 567.7747 (Mar. 2019), pp. 209–212. arXiv: 1804.11326 (cit. on p. 6).

[KJM20] Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. A Survey on Graph Kernels. *Appl Netw Sci* 5.1 (Dec. 2020), p. 6. arXiv: 1903.11835 [cs, stat] (cit. on pp. 3, 5).

[KL02] Risi Imre Kondor and John Lafferty. Diffusion Kernels on Graphs and Other Discrete Structures (2002), p. 8 (cit. on pp. 2, 3, 9).

[Kon08] Risi Kondor. "Group Theoretical Methods in Machine Learning". Thesis. Columbia University, 2008 (cit. on pp. 2, 8).

[KSB09] Risi Kondor, Nino Shervashidze, and Karsten M. Borgwardt. The Graphlet Spectrum. Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09. The 26th Annual International Conference. Montreal, Quebec, Canada: ACM Press, 2009, pp. 1–8 (cit. on p. 8).

[LAT21] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A Rigorous and Robust Quantum Speed-up in Supervised Machine Learning. *Nat. Phys.* 17.9 (Sept. 2021), pp. 1013–1017. arXiv: 2010.02174 [quant-ph] (cit. on pp. 7, 8).

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* 113.13 (Sept. 25, 2014), p. 130503. arXiv: 1307.0471 [quant-ph] (cit. on p. 6).

[SK03] Alexander J. Smola and Risi Kondor. Kernels and Regularization on Graphs. Learning Theory and Kernel Machines. Ed. by Bernhard Schölkopf and Manfred K. Warmuth. Red. by Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen. Vol. 2777. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 144–158 (cit. on p. 3).

[SK19] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* 122.4 (Feb. 1, 2019), p. 040504. arXiv: 1803.07128 [quant-ph] (cit. on p. 6).

[Sor+21] Apimuk Sornsaeng et al. Quantum Diffusion Map for Nonlinear Dimensionality Reduction. *Phys. Rev. A* 104.5 (Nov. 10, 2021), p. 052410. arXiv: 2106.07302 [quant-ph] (cit. on pp. 7, 8).

[Sze04] Mario Szegedy. "Spectra of Quantized Walks and a \$\sqrt{\delta\epsilon}\$ Rule". Jan. 9, 2004. arXiv: quant-ph/0401053 (cit. on p. 5).

[Tan19] Ewin Tang. A Quantum-Inspired Classical Algorithm for Recommendation Systems. *Proc. 51st Annu. ACM SIGACT Symp. Theory Comput.* (June 23, 2019), pp. 217–228. arXiv: 1807.04271 (cit. on p. 6).

[Vis+10] S.V.N. Vishwanathan et al. Graph Kernels. *J. Mach. Learn. Res.* 11.40 (2010), pp. 1201–1242. arXiv: 0807.0093 (cit. on p. 5).

[Xu21] Jue Xu. "On Lagrangian Formalism of Quantum Computation". Dec. 7, 2021. arXiv: 2112.04892 [quant-ph] (cit. on p. 11).

[Zhe+22] Han Zheng et al. "Speeding up Learning Quantum States through Group Equivariant Convolutional Quantum Ans\"atze". Jan. 20, 2022. arXiv: 2112.07611 [math-ph, physics:quant-ph, stat] (cit. on pp. 7, 8).

A Machine Learning and Group Theory

A.1 Machine learning

training set, test set.

A.1.1 SVM and kernel tricks

objective (cost function): *empirical risk* (error rate, loss function)

$$R_{emp}(\theta) = \frac{1}{|T|} \sum_{\mathbf{x} \in T} \mathbb{P}(\tilde{y} \neq y) \tag{22}$$

the dual quadratic program that (only uses access to the kernel) we maximize

$$L_D(\alpha) = \sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{t} y_i y_j \alpha_i \alpha_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{23}$$

subject to $\sum_{i=1}^{t} \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for each i?. construct the classifier

$$\tilde{m}(\mathbf{s}) := \operatorname{sign}\left(\sum_{i=1}^{t} y_i \alpha_i^* \mathcal{K}(\mathbf{x}_i, \mathbf{s}) + b \right) \tag{24}$$

A.1.2 Quantum machine learning

[Bia+17]; neural network

A.2 Group theory and symmetries

group $\mathbb{G}$

A.2.1 Representation theory

B Symmetries in physics

B.1 Lagrangian formalism

B.1.1 Path integral and quantum computing

[Xu21] In optics, Fermat's principle states that the path taken by a ray between two given points is the path that can be traveled in the least (extremum) time. A similar argument, *principle of least action*, was developed in classical mechanics:

Axiom 1 (Principle of least action)**.** *The actual path $q(t)$ taken by a classical system is the path that yields an extremum of its action $\mathcal{S}$. So, this principle is also called principle of stationary action. The action of the dynamics is the integral of Lagrangian over time*

$$\mathcal{S}[q(t)] := \int_{t_I}^{t_F} \mathcal{L}(q(t), \dot{q}(t); t)\, \mathrm{d}t \tag{25}$$

where $\mathcal{L}(q, \dot{q})$ is the Lagrangian in terms of generalized coordinate q and velocity $\dot{q}$ at certain time t.

The notion $\mathcal{S}[\cdot]$ reminds that action is a functional that takes a function (path) $q(t)$ as input. By varying the action, one have the equation of motion (Eq.**??**) called *Euler-Lagrange equation.* This Lagrangian formalism was extended by Dirac [Dir45] and Feynman [FHS10] to explain quantum mechanics.

Axiom 2 (Path integral)**.** *The amplitude (probability) of a quantum system evolving from $|q_I\rangle$ to $|q_F\rangle$ in a time interval can be evaluated by (functional) integrating over all possible paths with fixed initial and final position*

$$\langle q_F | e^{-\mathrm{i}t\hat{H}/\hbar} | q_I \rangle = \int_{q(t_I)=q_F}^{q(t_F)=q_I} \mathcal{D}q \; e^{\mathrm{i}\mathcal{S}[q]/\hbar} \tag{26}$$

where the action defined in classical mechanics as Eq. (25).

The Larangian (path integral) formalism of quantum mechanics is proved to be equivalent to the well-known Schrödinger equation Eq. (10) [FHS10, Chp4] which is a differential equation determining the evolution of quantum state. In the classical limit (Planck's constant $\hbar \to 0$), Path integral reduces to Principle of least action because only the paths around the stationary point of the action contribute (the other paths' contributions frequently oscillate and cancel out).

B.2 Symmetries with Lagrangian