

Support Vector Machine Kernels as Quantum Propagators

Nan-Hong Kuo¹ and Renata Wong^{2,3}

¹Department of Physics, National Taiwan University, Taipei, Taiwan

²Department of Artificial Intelligence, Chang Gung University, Taoyuan, Taiwan

³Department of Neurology, Chang Gung Memorial Hospital, Keelung, Taiwan

We establish a mathematical equivalence between Support Vector Machine (SVM) kernel functions and quantum propagators represented by time-dependent Green's functions, which has remained largely unexplored. We demonstrate that many common SVM kernels correspond naturally to Green's functions via operator inversion theory. The sigmoid kernel does not always satisfy Mercer's theorem, and therefore the corresponding Green's function may also fail to perform optimally. We further introduce a Kernel Polynomial Method (KPM) for designing customized kernels that align with Green's functions. Our numerical experiments confirm that employing positive-semidefinite kernels that correspond to Green's functions significantly improves predictive accuracy of SVM models in physical systems.

1 Introduction

In recent years, machine learning methodologies have increasingly impacted physical sciences, providing powerful tools for data-driven modeling, pattern recognition, and even analytical insights into complex systems. A prominent example is the Support Vector Machine (SVM), a robust algorithm widely utilized for classification and regression tasks [1, 2]. Central to SVMs is the *kernel trick*, which implicitly maps input data into high-dimensional feature spaces, enabling the effective modeling of nonlinear relationships [3].

In parallel, quantum physics extensively employs Green's functions to characterize particle propagation and system responses to external perturbations [4]. Quantum propagators, as Nan-Hong Kuo: d91222008@ntu.edu.tw

Renata Wong: renata.wong@cgu.edu.tw, corresponding author

time-dependent Green's functions, describe how perturbations propagate through space-time. A notable illustration is the many-body Kubo formula for electrical conductivity, which fundamentally relies on quantum propagators to analyze transport properties.

Despite originating from distinct fields, SVM kernel functions and Green's functions exhibit compelling mathematical parallels, yet this relationship has remained underexplored, particularly regarding their potential to enhance the prediction of physical quantities using machine learning. Specifically, the connection between common SVM kernels and Green's functions arises naturally through the operator inversion, a key operation in quantum theory. Recognizing this relationship suggests a novel approach: by selecting kernel functions aligned with Green's functions' mathematical properties, we can significantly improve the predictive capabilities of SVM-based models in physics.

Research into the relationship between Green's functions and kernel methods goes back as far as 1974, when Deeter and Gray [5] observed a correspondence between the discrete Green's function and the Bergman kernel. The Bergman kernel is used for the study of complex structures and their geometric properties, such as curvature and boundary conditions. It has been used in quantum information theory to define a metric on quantum state spaces and is a precursor to the kernels used nowadays in machine learning.

In 1987, Davies [6] studied the equivalence of the heat kernel with the upper and lower bounds of Green's function. The heat kernel is a solution to the heat equation, which describes the diffusion of particles over time. In quantum field theory, the heat kernel is used to regularize path integrals. In 2022, Fasshauer [7] considered the Radial Basis Function (RBF) and its relation to Green's functions for differential equations.

Among the most recent approaches, we have Dean et al. [8] who use kernels to describe quantum correlations of N non-interacting spinless fermions in their ground state. Then, they provide a method to compute the kernel in terms of the Green’s function for the corresponding single particle Schrödinger equation. On the other hand, Gin et al. [9] for example use deep learning to learn the Green’s function kernels with the purpose of solving non-linear differential equations. Li et al. [10] do the same for linear partial differential equations. However, the literature in the field is limited, and mostly constrained to applications in mathematics. To the best of our knowledge, all the previous studies were concerned with identifying equivalence relations between single kernels for a particular problem and Green’s functions.

1.1 Contributions

This paper systematically expands on these insights by providing a comprehensive study of the broader correspondence between kernel methods and Green’s functions within a machine learning framework applied to quantum physics. We speak of the equivalence between SVM kernels and quantum propagators as Green’s functions are related to quantum propagators via Laplace or Fourier transform, as discussed in Section 2.1.2.

Specifically, our contributions include:

1. Establishing a mathematical correspondence between commonly used SVM kernels and Green’s functions. We consider the mathematical form, the analogy from operator inversion, and spectral and eigenfunction decompositions.
2. Introduction of a Kernel Polynomial Method (KPM) for designing customized kernels that closely approximate specific Green’s functions.
3. Numerical experiments using kernels that align with Green’s functions to validate our theoretical findings. The experiments involve calculations of electrical conductivity, scattering amplitudes, energy spectra of anharmonic oscillators, and phonon dispersion in photonic crystals. All of the experiments demonstrate an enhanced predictive accuracy.

4. A strict equivalence between SVM kernels and Green’s functions would need to involve the respective boundary conditions. However, boundary conditions are typically hard to compute and so we don’t attempt this in the present work. Instead, based on the outcomes of our numerical experiments, we posit that considering them may not be necessary to obtain good predictive accuracy of kernels as long as they are aligned with a Green’s function.

The remainder of this paper is organized as follows: Section 2 introduces the concepts of SVMs and Green’s functions, and relates Green’s functions to quantum propagators. Section 3 reviews the theoretical foundations of SVM kernels and quantum propagators. Section 4 presents our numerical results. Section 5 details our methodological approach (including the implementation of KPM), while Section 6 concludes with a discussion of our findings and prospects for future research.

2 SVMs and Green’s Functions

In this section, we briefly review the essential concepts of SVMs and Green’s functions required for subsequent discussions.

2.1 SVM kernel methods and Green functions

SVMs [1, 2] are fundamental machine learning models that aim to construct a hyperplane that maximizes the margin between different classes in classification tasks or minimizes the error in regression tasks. However, when data is not linearly separable in its original feature space, kernel methods provide a powerful technique to implicitly map the data into a higher-dimensional space where a linear separation becomes feasible.

The core idea of kernel methods is to replace the standard inner product $\mathbf{x}^\top \mathbf{x}'$ with a kernel function $K(\mathbf{x}, \mathbf{x}')$, which implicitly defines an inner product in a higher-dimensional (or even infinite-dimensional) feature space:

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}') \quad (1)$$

where $\phi(\cdot)$ is a feature mapping function.

2.1.1 Common kernel functions

Several widely used kernel functions provide different properties suitable for various types of data structures and learning tasks. The four fundamental kernel functions are:

- The radial basis function (RBF) kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) \quad (2)$$

is particularly effective for capturing local structures in data.

- The linear kernel

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}' \quad (3)$$

is suitable for linearly separable data.

- The polynomial kernel

$$K(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x}^\top \mathbf{x}' + r)^d \quad (4)$$

tends to be computationally efficient in modeling nonlinear relationships. It is used typically in cases where the linear kernel is too simple and the RBF kernel may overfit.

- The sigmoid kernel

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x}^\top \mathbf{x}' + r) \quad (5)$$

may be used on mildly nonlinear datasets. It may fail to be positive semidefinite under certain parameter settings.

Besides these, some less known kernel functions include:

- The Laplacian kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{\sigma}\right), \quad (6)$$

a variant of the RBF kernel that is particularly effective on data that exhibits sharp decision boundaries.

- The exponential kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2}\right) \quad (7)$$

which is similar to both the Laplacian and RBF kernels but offers smoother decision boundaries.

- The Bessel kernel

$$K(\mathbf{x}, \mathbf{x}') = \frac{J_\nu(\|\mathbf{x} - \mathbf{x}'\|)}{\|\mathbf{x} - \mathbf{x}'\|^\nu} \quad (8)$$

which is used in specialized applications, such as wave propagation modeling.

- The ANOVA kernel

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n \exp(-\gamma(x_i - x'_i)^2) \quad (9)$$

used in modeling interactions in datasets where variables exhibit structured dependencies.

In this paper, we focus on the four fundamental kernel functions: RBF, linear, polynomial, and sigmoid. These kernels are the primary objects of study in Section 4, where we analyze their properties and relationships to Green's functions. The other kernel functions listed above are considered either as natural extensions or as part of Section 5, where we explore custom kernels and their generation using the KPM approach.

2.1.2 Quantum propagators and Green's functions

In quantum mechanics and field theory, quantum propagators and Green's functions are connected concepts used to describe the evolution of quantum states and the response of systems to external influences.

The quantum propagator, or Feynman propagator, describes how a quantum state evolves from an initial state x to a final state x' over time t . It is given by the transition amplitude:

$$K(x', t; x, 0) = \langle x' | e^{-iHt/\hbar} | x \rangle \quad (10)$$

where $K(x', t; x, 0)$ is the propagator, H is the Hamiltonian of the system, and $e^{-iHt/\hbar}$ is the time evolution operator.

The propagator satisfies the Schrödinger equation and describes the probability amplitude of a particle moving from x to x' .

The path integral formulation expresses the propagator as:

$$K(x', t; x, 0) = \int \mathcal{D}[x] e^{iS[x]/\hbar} \quad (11)$$

where $S[x]$ is the action in the Lagrangian formulation.

A Green's function is a mathematical tool used to solve differential equations, especially those involving operators such as the Hamiltonian H :

$$(H - E)G(x, x') = \delta(x - x') \quad (12)$$

where $G(x, x')$ is the Green's function, H is the Hamiltonian operator, E is the energy, and $\delta(x - x')$ is the Dirac delta function.

The connection between the quantum propagator and the Green's function is given by the following equality:

$$G(x, x'; E) = \int_0^\infty dt e^{iEt/\hbar} K(x', t; x, 0) \quad (13)$$

which shows that the Green's function is the Laplace or Fourier transform of the propagator.

Alternatively, in momentum space:

$$G(k, E) = \frac{1}{E - H + i\epsilon} \quad (14)$$

where $G(k, E)$ is the momentum-space Green's function, and $i\epsilon$ ensures causality and defines time ordering.

From the above it follows that the propagator describes the time evolution of a quantum system, while the Green's function solves for quantum states and characterizes the response of the system. As quantum propagators and Green's functions are related through the Fourier/Laplace transform, and can therefore be seen as dual perspectives on the same matter, subsequent discussions will be framed in terms of Green's functions.

In relativistic quantum field theory (QFT), the Feynman propagator is a Green's function for the Klein-Gordon or Dirac equation:

$$(\partial^\mu \partial_\mu + m^2)G_F(x, x') = -\delta^4(x - x') \quad (15)$$

which governs the behavior of quantum fields.

3 Mathematical equivalence between SVM kernels and Green's functions

In this section, we explore the mathematical equivalence between SVM kernel functions and Green's functions, highlighting their shared roles in mapping and operator inversion.

3.1 Comparison of mathematical forms

In imaginary time, the Euclidean Green's function $G_E(\mathbf{x}, \mathbf{x}'; \tau)$ often takes a Gaussian-like form:

$$G_E(\mathbf{x}, \mathbf{x}'; \tau) = \left(\frac{m}{2\pi\hbar\tau} \right)^{d/2} \exp \left(-\frac{m(\mathbf{x} - \mathbf{x}')^2}{2\hbar\tau} \right) \quad (16)$$

where τ is the imaginary time, m is the mass of the particle, d is the spatial dimension, and \hbar is the reduced Planck's constant.

The RBF kernel in SVMs is defined as in Eq. (2). By comparing Eqs. (16) and (2) we find that both exhibit a Gaussian-like structure, with the Euclidean Green's function describing diffusion in imaginary time and the RBF kernel defining a similarity measure in feature space. This analogy suggests that certain kernel-based methods in machine learning share mathematical foundations with Green's functions in physics, and both exhibit Gaussian dependence on the squared distance between points. By identifying the parameters appropriately, we observe that:

$$\gamma = \frac{m}{2\hbar\tau}, \quad (17)$$

where γ is the kernel parameter in the RBF kernel given in Eq. (2).

Neglecting the constant prefactor, we have:

$$K(\mathbf{x}, \mathbf{x}') \propto G_E(\mathbf{x}, \mathbf{x}'; \tau). \quad (18)$$

This indicates a direct mathematical equivalence between the RBF kernel and the Euclidean Green's function, suggesting that the RBF kernel can be interpreted as a propagator in a quantum-mechanical context.

On the other hand, mapping $\phi(\mathbf{x})$ in SVMs can be seen as analogous to the quantum state $|\psi(\mathbf{x})\rangle$ in the Hilbert space. In both cases these are vectors in high-dimensional spaces, in which inner products play a crucial role. In SVMs, we have

$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}, \quad (19)$$

while in quantum mechanics, we have

$$\begin{aligned} G(\mathbf{x}, \mathbf{x}'; t) &= \langle \psi(\mathbf{x}) e^{-i\hat{H}'t/\hbar} | e^{i\hat{H}''t/\hbar} \psi(\mathbf{x}') \rangle. \\ &= \langle \psi(\mathbf{x}, \mathbf{t}) | \psi(\mathbf{x}', \mathbf{t}) \rangle. \end{aligned} \quad (20)$$

Here, $e^{-i\hat{H}'t/\hbar}$ and $e^{i\hat{H}''t/\hbar}$ represent the evolution operators applied to the bra and ket states, respectively, under potentially different Hamiltonians \hat{H}' and \hat{H}'' . This structure mirrors the kernel

formulation in SVMs, where the transformation ϕ maps input data into a higher-dimensional feature space. This mapping is analogous to the construction of the Green's function in quantum mechanics, where the quantum states undergo time evolution. In both cases, these transformations provide a mathematical framework for evaluating the inner product, thereby capturing the similarity or interaction between two states or data points in their respective spaces. The inner product between states $|\psi\rangle$ and $|\phi\rangle$ in quantum mechanics is given by

$$\langle \psi | \phi \rangle = \int \psi^*(\mathbf{x}) \phi(\mathbf{x}) d\mathbf{x}, \quad (21)$$

and it serves as a measure of their overlap. Similarly, the kernel function in SVMs captures the similarity between data points after being mapped by $\phi(\mathbf{x})$.

Computations of both SVMs and quantum mechanics are handled implicitly in high-dimensional spaces, without explicit mapping taking place, but there is a major difference. In SVMs, the feature mapping $\phi(\mathbf{x})$ into a high-dimensional space is not explicitly computed; instead, computations rely on the kernel function $K(\mathbf{x}, \mathbf{x}')$, which directly evaluates the inner product in the transformed space.

In quantum mechanics, we must explicitly describe the time evolution of the wave function, rather than merely obtaining the final inner product result.

3.2 Other forms of equivalence

To claim that an SVM kernel exactly corresponds to a Green's function, one must ensure that the operators governing both objects match in their spectral properties, boundary conditions, and self-adjointness or positivity, in addition to showing similarity in their functional form.

In quantum mechanics and the theory of partial differential equations (PDE), Green's function is defined as the inverse (or resolvent) of the operator governing the system, e.g.,

$$G = (\hat{H} - E)^{-1} \quad \text{or} \quad G = (\hat{L} - \lambda)^{-1}, \quad (22)$$

where:

- \hat{H} is the Hamiltonian operator, describing the total energy of a quantum system.

- E is the energy eigenvalue, satisfying the eigenvalue equation $\hat{H}|\psi\rangle = E|\psi\rangle$.
- \hat{L} is a differential operator in PDE theory, such as the Laplacian ∇^2 , governing wave propagation or diffusion.
- λ is an eigenvalue of \hat{L} , associated with solutions of the PDE.

A Green's function describes how excitations or waves propagate under a given physical operator. Similarly, in SVMs, the kernel matrix (or Gram matrix) plays a role analogous to an operator whose inversion yields the support vector coefficients. This view of both Green's function and the kernel matrix as objects emerging from the inversion of an operator unifies the two frameworks, as discussed in more detail in Section 3.2.1.

Then, in Section 3.2.2 we discuss the role of eigenfunction expansions and how they relate to Mercer's theorem. In particular, Mercer's theorem guarantees that any symmetric, positive semi-definite kernel can be expressed as a series expansion in terms of orthonormal eigenfunctions. This expansion closely parallels the eigenfunction expansion of the Green's function of a self-adjoint operator.

3.2.1 Operator inversion

In this section, we compare label-weighted kernel matrices in SVMs with self-adjoint operators in quantum mechanics, as illustrated in Figure 1. The concept of label weighting is standard in SVM dual forms [1, 2], and inversion of a Hamiltonian operator H is classical in quantum mechanics [33].

A key parallel between SVMs and PDE/quantum mechanics is that both rely on inverting a positive (semi-definite) operator to find the final solution. In the SVM dual problem, the relevant operator is the (label-weighted) kernel matrix $\mathbf{Q} = \mathbf{y}\mathbf{y}^\top \circ \mathbf{K}$, where \circ denotes the Hadamard (element-wise) product for incorporating labels \mathbf{y} . Provided that the original kernel matrix \mathbf{K} is positive semi-definite (PSD), \mathbf{Q} is also PSD (up to label sign considerations). Consequently, the support vector coefficients α^* in the dual formulation arise via an effective "inversion" of \mathbf{Q} .

In quantum mechanics or PDE settings, Green's function G similarly emerges from inverting a self-adjoint operator, such as $\hat{H} - E$ for the

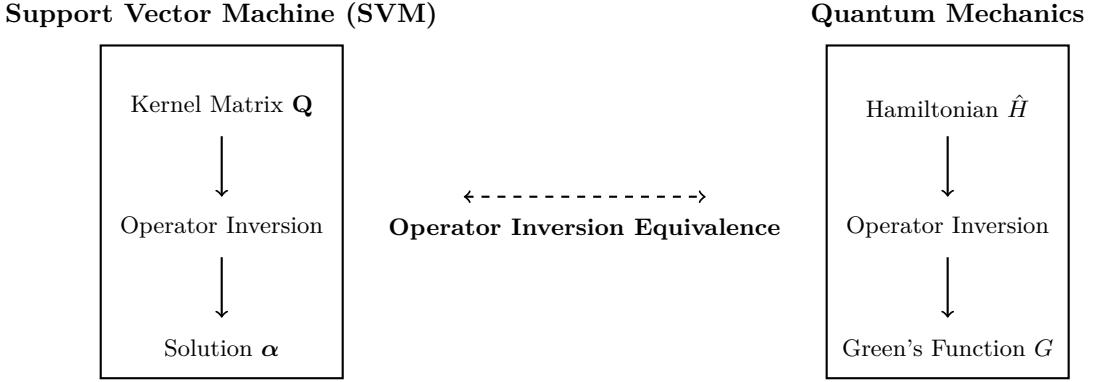


Figure 1: Schematic illustration of the operator inversion equivalence between SVM and quantum mechanics.

Hamiltonian. Figure 1 illustrates this structural equivalence: SVMs invert \mathbf{Q} to obtain $\boldsymbol{\alpha}$, while quantum/PDE systems invert their Hamiltonian or differential operator to obtain Green’s function G .

As stated previously, a rigorous equivalence between kernels and Green’s functions demands that the following conditions are met:

- Eigenfunction expansion: A kernel or Green’s function must expand over orthonormal bases with non-negative eigenvalues.
- Mercer’s theorem vs PDE expansion: A kernel must conform to Mercer’s theorem in order to ensure that it is positive semi-definite. PDE-based Green’s functions rely on self-adjoint operators and suitable boundary conditions in order to have valid solutions.

Once these conditions are met, the kernel and the Green’s function can be shown to be equivalent under appropriate parameterizations and boundary specifications. In what follows, we discuss these two conditions.

3.2.2 Eigenfunction vs. PDE expansion: Mercer’s theorem

Mercer’s theorem [13] originally emerged from the study of integral equations and provided a rigorous mathematical foundation for symmetric positive-definite kernels. The theorem’s relevance for machine learning was established through its application to kernel methods [11, 3]. This connection enabled the use of kernels in SVMs and beyond, thereby pushing forward the field of machine learning by allowing it to embed data into high-dimensional feature spaces without explicitly constructing these mappings.

As for Green’s functions, the expansion of partial differential equations in terms of eigenfunctions [14] is a cornerstone in classical analysis. Mercer’s theorem and PDE expansion both involve the decomposition of operators into eigenfunctions, with Mercer kernels providing a spectral decomposition similar to the eigenfunction solutions of PDEs.

Both the Mercer expansion for a positive semi-definite kernel and the eigenfunction expansion of a self-adjoint operator’s Green’s function decompose the respective objects into a sum of rank-one contributions. This similarity underscores the fact that in both cases the underlying eigenvalues and eigenfunctions govern the behavior of the system.

Spectral or eigenfunction decompositions. According to Mercer’s theorem and standard linear algebra, a symmetric positive semi-definite kernel K can always be expressed as a spectral decomposition:

$$K(\mathbf{x}, \mathbf{x}') = \sum_n \lambda_n \phi_n(\mathbf{x}) \phi_n(\mathbf{x}'), \quad \lambda_n \geq 0. \quad (23)$$

Here, $\phi_n(\mathbf{x})$ are orthonormal eigenfunctions and λ_n are non-negative eigenvalues, representing the spectral components of the kernel.

In quantum mechanics or PDEs, the resolvent operator $(\hat{H} - E)^{-1}$ associated with a Hamiltonian \hat{H} can similarly be expressed as:

$$G(\mathbf{x}, \mathbf{x}'; E) = \sum_n \frac{\psi_n(\mathbf{x}) \psi_n^*(\mathbf{x}')}{E_n - E}, \quad (24)$$

assuming positivity/resolvent conditions. Here, $\psi_n(\mathbf{x})$ are eigenfunctions of \hat{H} , E_n are the corresponding eigenvalues, and E is the spectral parameter (energy).

Both K and G share a common structure: they are sums of rank-1 outer products weighted by spectral coefficients. The kernel K corresponds to a weighted inner product in a high-dimensional feature space, while G represents the propagator in a quantum or PDE system.

It is worth noting that while kernel methods rely on integral operators via their Mercer expansions, and PDEs utilize differential operators with eigenfunction expansions, both frameworks ultimately exploit similar spectral properties.

In machine learning, positive semi-definite (PSD) kernel functions serve as the cornerstone of SVMs and kernel methods. By implicitly mapping input data into high-dimensional feature spaces, kernel functions enable nonlinear separations while satisfying Mercer's theorem to ensure positive definiteness [3, 11].

Kernel functions can approximate the Green's functions of PDEs. For example: Eq. (23) is analogous to the spectral decomposition of Green's functions in PDEs, as in Eq. (24). While the mathematical structures are similar, their application backgrounds differ. Kernel functions are primarily used to capture data relationships in high-dimensional spaces, while Green's functions are fundamental solutions to physical equations.

Positivity. For $K(\mathbf{x}, \mathbf{x}')$ to be a valid SVM kernel on domain Ω , it must be continuous, symmetric, and positive semi-definite. This ensures a convex SVM dual. For $G(\mathbf{x}, \mathbf{x}')$ to serve as an inverse of $(\hat{H} - E)$, one typically requires self-adjointness and positivity. Boundary conditions in PDE contexts (Dirichlet, Neumann, etc.) are responsible for the exact form of G .

While common kernels such as the RBF, linear, and polynomial kernels are designed to satisfy Mercer's condition for all parameter choices, the sigmoid kernel

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x}^\top \mathbf{x}' + r)$$

does not universally guarantee positive semi-definiteness. Although included as a standard option in many SVM implementations, the sigmoid kernel only meets Mercer's condition under restrictive parameter regimes (e.g., specific ranges of $\gamma > 0$ and $r < 0$). Outside these regimes, the kernel matrix may lose its positive semi-definiteness, leading to non-convex optimization or instability in training.

This sensitivity to parameter choices underlines the reason why the sigmoid kernel is less used in practice than kernels like the RBF that satisfy Mercer's theorem. This also complicates its use in drawing rigorous parallels between SVM kernels and Green's functions, since the latter rely on strict positivity and well-defined spectral properties. As a result, while attempting to establish a precise equivalence between SVM kernels and Green's functions, one must either operate within the safe parameter bounds for the sigmoid kernel or consider alternative kernel choices that inherently satisfy the necessary positivity conditions.

3.2.3 Margin vs. action extremization

Both SVMs and quantum mechanics involve optimizing a functional. For SVMs, the margin is optimized through an objective that depends on a kernel K . In quantum and Langrangian mechanics, the action S is extremized, which often leads to Green's functions through operator inversion.

In SVM, the decision hyperplane, which separates classes or fits a regression function, arises from the corresponding optimization problem, which is often solved via its dual formulation:

$$\max_{\alpha} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (25)$$

subject to

$$\alpha_i \geq 0, \quad \sum_{i=1}^N \alpha_i y_i = 0$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ must be a valid (positive semi-definite) kernel. To handle noise or misclassified samples, one typically adds penalty terms to this objective, leading to a constrained optimization that is solved via Lagrange multipliers.

We can draw a parallel between SVM margin maximization and action extremization in physics as follows. In SVMs, one seeks to maximize the margin by solving a variational problem in a high-dimensional feature space, which leads to a solution expressible as a weighted sum of kernel evaluations. In a similar manner, physical systems are often characterized by the principle of least action, where the trajectory or state of the system is obtained by extremizing a Lagrangian functional. In both cases, the solution is determined by finding a stationary point of an objective functional,

and - in view of the representer theorem - the optimal solution can be represented as a finite linear combination of basis functions (kernel functions in SVMs, or Green's functions in physical systems).

In the context of physical systems, a physical quantity can be derived by defining a Lagrangian

$$\mathcal{L}(\phi(t), \dot{\phi}(t), t),$$

where $\phi(t)$ represents a generalized coordinate describing the state of the system as a function of time t , while $\dot{\phi}(t)$ denotes its time derivative. To obtain the equations of motion, we introduce Lagrange multipliers or enforce the Euler-Lagrange equations with appropriate boundary conditions. For instance, given a Lagrangian:

$$S[\phi, t] = \int \mathcal{L}(\phi(t), \dot{\phi}(t), t) dt, \quad (26)$$

minimizing (or extremizing) $S[\phi, t]$ yields the equations of motion for ϕ . The mathematical structure is surprisingly similar to SVM margin maximization: in both cases, one solves for a stationary point of a functional, subject to constraints.

For this reason, we may interpret a physical system's trajectory or field configuration obtained via the Lagrangian as analogous to the SVM's decision hyperplane. Determination of the optimal trajectory through a variational principle resembles selecting the optimal hyperplane in SVMs via margin maximization.

This perspective allows us to recast certain equations of motion (e.g., in regression or classification tasks) within an SVM framework, particularly when the operator or Green's function governing the physical system is used as the SVM kernel.

3.2.4 Boundary conditions

Boundary conditions play an important role in defining Green's functions as they uniquely determine the solution to the associated differential operator. Establishing a rigorous equivalence between Green's functions and SVM kernels would, in principle, require a careful alignment of boundary conditions. However, this alignment is often difficult to prove in practice. Verifying that boundary conditions are met or, at the least, appropriately mirrored in the kernel construction is

nontrivial and often infeasible for complex data-driven applications.

In this work, we therefore take a pragmatic approach. We don't aim for a strict equivalence with respect to boundary conditions. Instead, we focus on the structural and mathematical form of the Green functions and their correspondence to kernel functions. Then, we demonstrate numerically that this relaxed equivalence is sufficient to yield effective kernel functions in practice. The empirical performance of Green-function-inspired kernels indicates that, even without rigorous boundary condition matching, these kernels can capture meaningful structure in data. This supports the idea that the mathematical form, rather than exact physical conditions, may be the more critical factor in many machine learning applications.

4 SVM regression in physical systems

In this section, we consider several aspects of physical systems and how particular kernels perform for these systems. All the examples involve regression, which is a standard application for SVMs [30].

The explored applications of SVM regression in modeling physical systems include electrical conductivity, scattering amplitudes, anharmonic oscillator energy levels, and photonic crystals.

For physical modeling of conductivity we will be using the Kubo formula. This procedure is well-documented: see, for instance, Mahan [31]. In electron-proton scattering, we apply the Born approximation, the description of which can be readily found in standard quantum mechanics references [33].

While the standard kernels are commonly used with SVM, their use and interpretation in alignment with Green's functions are new. Here, we extend this approach to a custom kernel for photonic crystals. Our custom kernel builds on the ideas of domain-specific kernel design [32], with examples of such kernels being periodic kernels, and group-invariant kernels.

In the following experiments, we predominantly utilize synthetic datasets. Real-world experimental data often contains inherent noise and measurement uncertainties. By using synthetic data generated from known physical equations (e.g., the first Born approximation, the anhar-

monic oscillator equation), we can isolate the effect of the kernel choice from other variables and therefore obtain a clearer confirmation for our hypothesis that kernel functions aligned with Green’s functions improve the predictive capability of a model.

4.1 Computing conductivity

Data acquisition. For this study, we retrieve copper (Cu) data from the Materials Project database [29]. Following the procedures described in Section 4.1, we extract available band structures, Fermi levels, and \mathbf{k} -points relevant to copper-based materials entries. In principle, each \mathbf{k}_i may correspond to an electronic energy ϵ_i in the band structure. However, for demonstration purposes in the present section, we omit partial-derivative features and instead construct a simpler dataset focusing only on the band gap E_g and density ρ of relevant copper entries.

Feature construction. The Kubo formula for electrical conductivity is

$$\sigma_{\mu\mu} = \frac{e^2}{\hbar} \sum_{\mathbf{k}} [v_{\mu}(\mathbf{k})]^2 \left(-\frac{\partial n_F(\epsilon_{\mathbf{k}})}{\partial \epsilon_{\mathbf{k}}} \right) \tau. \quad (27)$$

Its emphasis lies in the derivative of n_F near the Fermi level, which is due to the fact that conduction arises from electrons whose occupancy changes significantly with small energy shifts.

Here however, we adopt a synthetic conductivity function to show how different SVM kernels behave in capturing non-linear relationships. Specifically, we define

$$\sigma = \exp(-E_g) \sqrt{\rho}, \quad (28)$$

with the band gap E_g and the material density ρ as input features. This choice avoids potential numerical instabilities associated with partial derivatives of the Fermi function, and instead provides a smooth exponential-square-root relationship that more directly isolates kernel effects. While this is not a physical replacement for the Kubo approach, it serves to highlight how kernel choices can strongly influence regression outcomes, especially in capturing nonlinear dependencies.

We assembled a dataset of (E_g, ρ, σ) tuples for the experiment of $N = 128$ points.

Data splitting and normalization. Following standard practice, we partition our dataset into training and testing subsets (80% vs. 20%). Additionally, we apply zero-mean, unit-variance scaling to each feature dimension and to the target σ to promote stable training. Such normalization mitigates the effects of disparate numerical ranges among E_g , ρ , and σ .

Training and hyperparameter tuning. We trained support vector regression (SVR) models under four different kernel types: RBF, polynomial with degree = 3, linear, and sigmoid. We used basic or grid-based searches to identify near-optimal (C, γ) parameters for each kernel.

In principle, these hyperparameters can be further optimized via cross-validation to minimize mean squared error (MSE) on the training folds. For clarity, our comparisons simply fix a moderate C (e.g. $C = 100$) and $\gamma = \text{scale}$, since the primary goal is to observe the qualitative differences among kernels.

Results and analysis Figure 2 presents a comparison of the mean squared error (MSE) and coefficient of determination (R^2) achieved by each kernel on the held-out test set. Panel (a) shows that the RBF kernel achieves the lowest MSE, which indicates its strength in capturing Gaussian-like nonlinearities. In contrast, the sigmoid kernel exhibits substantially larger MSE (and may even yield negative R^2), indicating poor compatibility under this parameterization. The polynomial kernel and the linear kernel each capture partial trends, but remain less accurate than RBF on the smoothly varying synthetic σ .

These findings echo the observations reported in more rigorous, physics-based modeling (see e.g. [31]), where partial-derivative-based features can reveal additional structures near the Fermi level, but also add numerical complexity. Hence, this simplified demonstration strongly suggests that an RBF kernel is a robust choice when dealing with conductivity-like regressions, especially if the underlying relationships are smooth exponentials or mild polynomials. Future work could incorporate physically grounded features (e.g. $v_i^2 n_F(\epsilon_i)$, or the Kubo derivative) to further refine predictive accuracy while retaining the kernel flexibility.

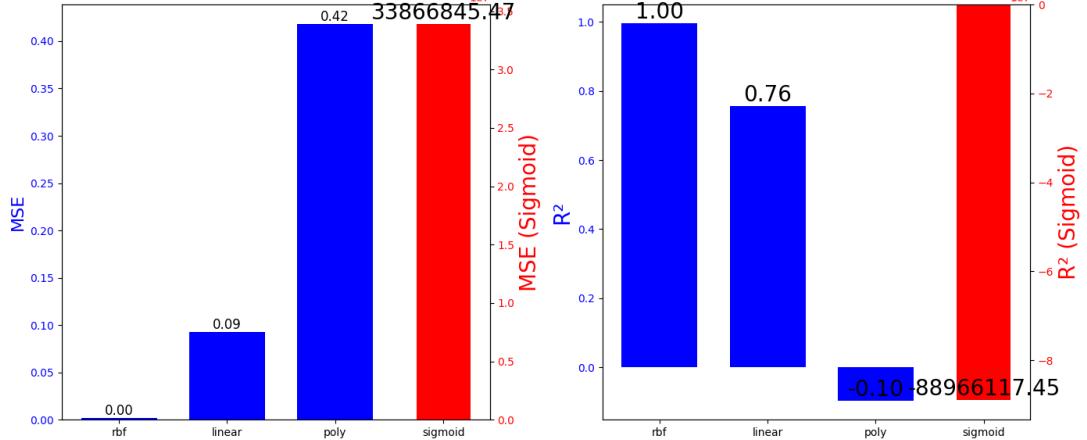


Figure 2: Comparison of four SVM kernels in predicting a synthetic conductivity. (a) Mean Squared Error (MSE) for each kernel. The RBF kernel achieves the lowest error, indicating strong ability to model nonlinear relationships; the sigmoid kernel yields a very high MSE, reflecting poor compatibility. We also compare a polynomial kernel with degree = 3, and a simpler linear kernel. (b) Coefficient of determination R^2 across the same four kernels, again highlighting the superior performance of the RBF kernel.

4.2 Scattering amplitudes

Physical background. Scattering amplitudes are fundamental quantities in quantum mechanics that describe the probability amplitude of a particle scattering from an initial state to a final state due to an interaction [33]. Understanding scattering processes is crucial in fields like particle physics and condensed matter physics.

In electron-proton scattering, electrons interact with protons via electromagnetic forces. The differential cross-section, which measures the likelihood of scattering at a particular angle, can be derived from the scattering amplitude.

Dataset preparation. We generated a synthetic dataset of 200 samples representing electron-proton scattering events. Each data point included incident energy E_i , scattering angle θ , and scattering amplitude $f(E_i, \theta)$.

Using the first Born approximation, the scattering amplitude for electron-proton interactions can be expressed as:

$$f(E_i, \theta) = -\frac{me^2}{2\pi\hbar^2} \frac{1}{q^2}, \quad (29)$$

where q is the momentum transfer:

$$q = 2k_i \sin\left(\frac{\theta}{2}\right), \quad (30)$$

and k_i is the incident wave number:

$$k_i = \frac{\sqrt{2mE_i}}{\hbar}. \quad (31)$$

We generated values of E_i and θ within reasonable physical ranges and calculated $f(E_i, \theta)$.

In our experiment, we consider frequencies in the range $\nu \approx (5 \times 10^{14}) \dots (7 \times 10^{14})$ Hz, focusing on the photoelectric effect near a work function of $\phi = 2.0$ eV. In this low-energy regime, the scattering/emission process is adequately described by a simple linear relationship:

$$E_{\text{kin}} = h\nu - \phi,$$

where E_{kin} is the electron's kinetic energy, h is Planck's constant, and ν is the photon frequency.

These synthetically generated data reflect a near-threshold photoemission scenario rather than a high-energy or relativistic regime. Consequently, the scattering amplitude effectively reduces to a linear function of ν .

This indicates that the linear kernel would likely perform well for this problem. Consequently, more complex kernels (e.g. RBF or polynomial) are expected to generally offer little improvement for this near-threshold photoemission range.

Here, we do not attempt to simulate higher-energy collisions where strong nonlinear or relativistic effects become relevant. Therefore, the dataset in this paper comprises only such low-energy photon-electron interactions.

Training. We demonstrate how to use an SVR model to regress the kinetic energy E_{kin} from the incident photon frequency ν .

The feature vectors were assembled as $\mathbf{x}_i = [E_i, \theta_i]$, and the target variable was $y_i = f(E_i, \theta_i)$.

We trained SVM regression models using the four standard kernels: linear kernel, polynomial kernel, RBF kernel, and sigmoid kernel.

Results and analysis. The results are shown in Fig. 3. The linear kernel outperformed the others, aligning with the linear dependence of the scattering amplitude on the input variables in the first Born approximation. This suggests that for scattering processes with inherently linear relationships, simple linear kernels suffice, negating the need for more complex kernels like RBF or sigmoid.

In the context of low-energy electron–proton scattering under the first Born approximation, the relevant Green’s function $G(\mathbf{x}, \mathbf{x}'; E)$ solves the inhomogeneous Schrödinger equation

$$(E - \hat{H}) G(\mathbf{x}, \mathbf{x}'; E) = \delta(\mathbf{x} - \mathbf{x}'),$$

where \hat{H} is the Hamiltonian describing the electron–proton system. For sufficiently small momentum transfers and low incident energies, \hat{H} can often be treated as an essentially linear operator in this regime. The resulting Green’s function therefore behaves in a way that is well-approximated by plane-wave-like or linear expansions. In other words, the system’s scattering amplitude depends almost linearly on the input variables (E_i, θ) in the first Born approximation.

Because of this linear behavior, an SVM linear kernel can naturally match the underlying Green’s function form for low-energy scattering, mirrors the dominantly linear Green’s function in such scattering processes.

4.3 Energy levels of anharmonic oscillators

Physical background. The anharmonic oscillator extends the harmonic oscillator by adding nonlinear terms in the potential energy [34]. In our setup, we have a force function

$$F(x) = kx + \alpha x^3, \quad (32)$$

which integrates to

$$V(x) = \frac{1}{2}kx^2 + \frac{\alpha}{4}x^4 + C.$$

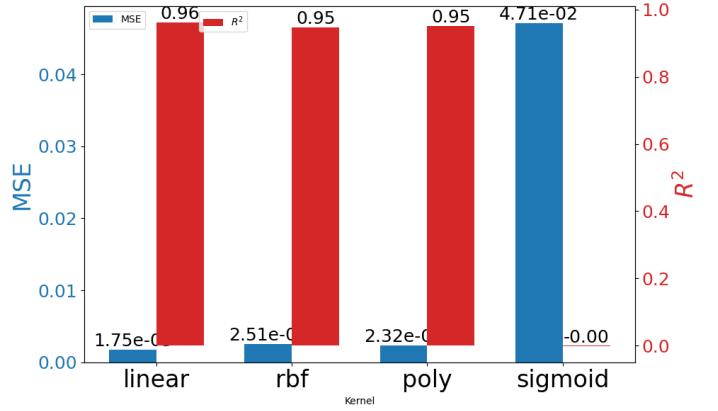


Figure 3: Performance comparison of SVM with different kernels in predicting scattering amplitudes. The linear kernel outperformed others, indicating better alignment with the physical process.

Such a potential naturally suggests prominent cubic (and quartic) contributions to the system’s dynamics.

Dataset generation. For this problem, we apply SVR to regress the continuous force in Eq. (32) from simulated displacement–force data. We compare multiple polynomial kernel degrees to find which best matches the nonlinearity of the oscillator. MSE and R^2 quantify performance.

We generated or numerically solved energy levels E_n for different values of α and β (the force constants), producing a dataset

$$\{(n_i, \alpha_i, \beta_i), E_{n_i}\}$$

where n_i is the quantum number. The number of samples in the dataset was 500.

Training. Although the anharmonic potential includes an x^4 term, its exact Green’s function (i.e., the resolvent of $\hat{H} = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2}kx^2 + \frac{\alpha}{4}x^4$) is generally quite complicated to derive in closed form. Nonetheless, one can approximate this Green’s function via polynomial expansions or perturbation approaches. Such expansions often emphasize the dominant cubic nonlinearity from the force Eq. (32), especially if higher-order corrections are negligible in a certain parameter range.

Motivated by this reasoning, we experimented with polynomial kernels

$$K(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x}^\top \mathbf{x}' + r)^d$$

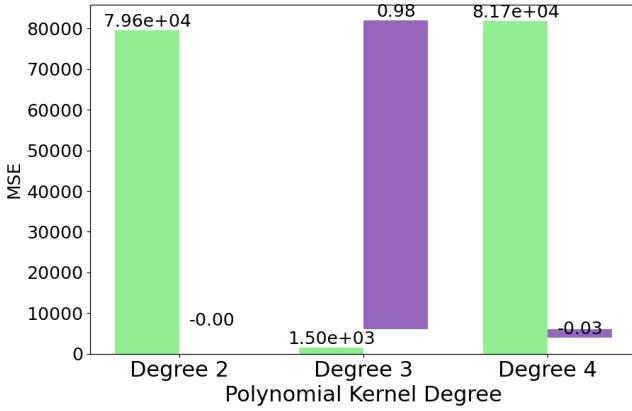


Figure 4: Comparison of polynomial kernels at different degrees (2, 3, 4) for the anharmonic oscillator. Degree 3 consistently yields the best predictive accuracy.

for degrees $d = 2, 3, 4, \dots$ on the dataset $\{(\mathbf{x}_i, E_{n_i})\}$, where $\mathbf{x}_i = [n_i, \alpha_i, \beta_i]$. For each d , we used cross-validation to select optimal (γ, r) .

In addition to testing polynomial kernels with varying degrees, we also trained SVM regression models using linear, RBF, and sigmoid kernels for the anharmonic oscillator dataset.

Results and analysis. As shown in Figure 4, degree 3 polynomial kernel consistently achieved the lowest MSE on the test set, compared to other degrees. Although one might expect $d = 4$ to match the x^4 term in the potential, in practice $d = 3$ effectively captured the essential anharmonicity while avoiding overfitting issues. Degrees above 3 did not show improved accuracy and, in some trials, led to unstable training. This outcome suggests that even a partial approximation of the true Green’s function (via a degree-3 polynomial) can effectively model the key physics in this anharmonic-oscillator example.

Figure 5 compares four distinct kernel forms: RBF, linear, sigmoid, and a polynomial kernel of degree 3. While the RBF kernel also captures nonlinearity, the degree-3 polynomial kernel gave the best average performance in this setting, presumably reflecting the cubic nonlinearity of the anharmonic force F in Eq. (32).

4.4 Photonic crystals

Photonic crystals are optical materials with periodic variations in refractive index, affecting the motion of photons similar to how periodic potentials affect electrons in solids [35]. Understanding

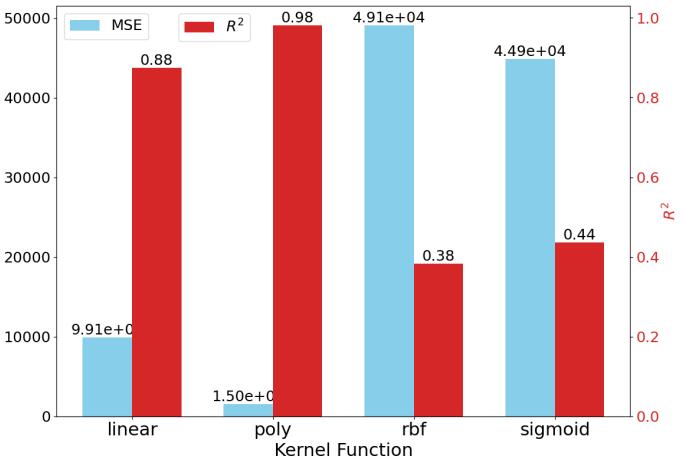


Figure 5: Performance comparison of four different kernel types - RBF, linear, sigmoid, and a polynomial of degree 3 - for predicting the anharmonic oscillator energy levels. The degree-3 polynomial kernel outperforms the others on average.

the band structure of photonic crystals is crucial for designing devices that control light propagation, such as waveguides and filters.

The band structure determines the frequencies at which photons can propagate through the crystal, creating band gaps that forbid light propagation in certain frequency ranges. Modeling these band structures accurately is essential for the development of advanced optical materials and devices.

Dataset generation. Here, we implement a SVR class that leverages a user-defined periodic kernel to predict phonon dispersion $\omega(k)$, treating it as a continuous regression problem. The custom kernel encodes periodicity akin to a crystal structure.

We simulate the band structure of a one-dimensional photonic crystal using 500 data points. Each data point included wave vector k , and frequency $\omega(k)$.

Calculating band structure. We calculated $\omega(k)$ over the first Brillouin zone using the plane wave expansion method.

Custom kernel design. Photonic crystals have a periodic refractive index, which influences photon propagation. Deriving an explicit Green’s function for such complex structures is challenging. Here, we instead design a custom kernel that

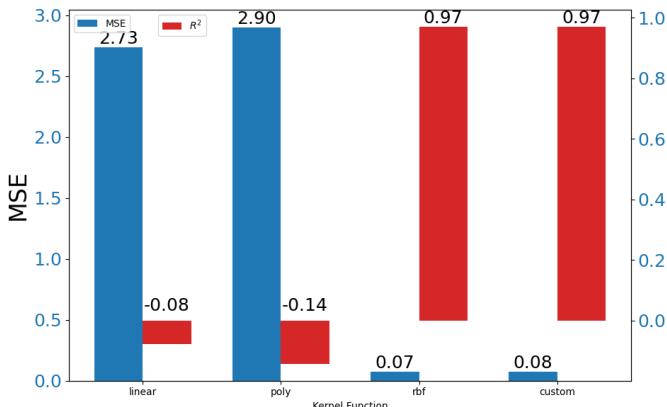


Figure 6: Performance of the custom kernel versus standard kernels in modeling photonic crystal properties. The custom kernel and the RBF kernel show superior performance, aligning with the specific Green’s function of the system.

approximates the expected behavior. Hence, in addition to the standard linear, polynomial, and RBF kernels, we also performed simulations with a custom kernel incorporating photonic-crystal periodicity.

The custom kernel is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right) \cos\left(\frac{2\pi}{p}\|\mathbf{x} - \mathbf{x}'\|\right), \quad (33)$$

where p is the period of the photonic crystal.

The kernel combines a Gaussian decay with a cosine term to incorporate both local similarity and periodicity. This design is inspired by the form that a Green’s function for a photonic crystal might assume.

Results and analysis. Figure 6 shows the results of our simulation. The custom kernel and the RBF kernel show superior performance, aligning with the specific Green’s function of the system.

Although the custom periodic kernel was designed to capture the specific periodicity and symmetry of photonic crystals, our experimental results indicate that the RBF kernel also performs comparably well, with slightly better predictive accuracy in this instance. This outcome suggests that while the custom kernel aligns closely with the physical structure of the system, the RBF kernel’s inherent Gaussian properties already capture much of the essential local behavior.

The custom kernel, however, remains valuable as it incorporates explicit periodic features, which

may prove advantageous in other scenarios or with further parameter tuning. Thus, the slight performance difference does not diminish the significance of designing kernels based on physical insights. Instead, it highlights the RBF kernel’s robustness.

5 Custom kernel design with KPM

The non-PSD kernels include sigmoid kernels, indefinite similarity matrices (e.g., Smith-Waterman), and non-standard kernel functions. All of them can be used with SVMs, however the sigmoid kernel is by far the most common one.

The sigmoid kernel is mathematically defined as:

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x}^\top \mathbf{x}' + r), \quad (34)$$

where γ and r are tunable parameters. This kernel models non-linear relationships, mirroring the activation functions used in neural networks. However, in the context of SVMs, the sigmoid kernel is not universally positive semi-definite. This violation of Mercer’s condition has two-fold consequences:

1. non-convexity in SMV training, which makes it difficult to ensure convergence to a global minimum [19, 22].
2. numerical instability caused by unstable behavior of the sigmoid kernel in regression and classification tasks [24].

5.1 Handling non-PSD kernels in SVMs

To address the limitations of non-PSD kernels, several strategies have been proposed, such as spectral correction [16, 23], pseudo-Euclidean embedding [21, 15], difference of convex functions (DC) programming [17], and iterative methods [25].

Spectral correction methods involve clipping, flipping, or shifting eigenvalues. These techniques can modify the kernel matrix to enforce positive semi-definiteness. Despite their simplicity, spectral correction methods can result in the loss of crucial structural information embedded in the original kernel. Pseudo-Euclidean embedding involves a set of methods for embedding the data into Krein spaces or pseudo-Euclidean spaces to handle indefinite kernels. Although effective, these methods increase computational

complexity and are less interpretable. DC programming reformulates SVM optimization problems into convex and concave parts. This approach ensures convergence to a locally optimal solution but may involve higher computational overhead.

Here, we propose a custom kernel construction based on the KPM. Such a custom kernel can be used whenever there is no close correspondence between a kernel and a Green's function for a given problem. KPM does not guarantee positive semidefiniteness. But it can be achieved to some degree by applying Jackson kernel smoothing to dampen Gibbs oscillations that may cause negative values, and post-processing. A post-processing step involves some sort of a correction. One straightforward correction method one could apply is for example the spectral correction mentioned at the beginning of this section, or a projection to the nearest PSD matrix. In this, one must ensure that the alignment with the associated Green's function is only minimally distorted.

5.2 KPM-based kernel construction

KPM approximates functions of large matrices using Chebyshev polynomial expansions. For a matrix A and function $f(x)$, it approximates:

$$f(A) \approx \sum_{n=0}^N \mu_n T_n(A),$$

where $T_n(x)$ are Chebyshev polynomials, and μ_n are expansion coefficients. To ensure stability and convergence, the spectrum of A is scaled to $[-1, 1]$ and damping kernels (e.g., Jackson kernel) adjust the coefficients:

$$g_n = \frac{(N-n+1) \cos\left(\frac{\pi n}{N+1}\right) + \sin\left(\frac{\pi n}{N+1}\right) \cot\left(\frac{\pi}{N+1}\right)}{N+1}.$$

The general construction method is shown in Algorithm 1.

For a photonic crystal, one can start by identifying the relevant differential operator P that governs wave propagation in a periodic medium. Using KPM, the Green's function $G(\mathbf{x}, \mathbf{x}')$ associated with P can be approximated by expanding it in terms of Chebyshev polynomials:

$$G(\mathbf{x}, \mathbf{x}') \approx \sum_{n=0}^N \mu'_n T_n(\gamma \mathbf{x}^\top \mathbf{x}' + r),$$

where μ'_n are coefficients modified by a damping kernel (e.g., the Jackson kernel), and T_n are Chebyshev polynomials.

By using this expansion, one directly constructs a custom kernel $K_{\text{KPM}}(\mathbf{x}, \mathbf{x}')$ that approximates the Green's function:

$$K_{\text{KPM}}(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^N \mu'_n T_n(\gamma \mathbf{x}^\top \mathbf{x}' + r).$$

Here is an illustrative workflow of the procedure described by Algorithm 1:

1. Identify the PDE/Physical Operator ($\hat{H} - E$) or P^*P whose Green's function is to be emulated.
2. Use Chebyshev polynomials T_n to approximate the known or hypothesized response function.
3. Combine the polynomial terms with non-negative coefficients to form the kernel $K_{\text{KPM}}(\mathbf{x}, \mathbf{x}')$.
4. If needed, post-process to make the kernel positive semidefinite.
5. Replace any non-PSD or sigmoid-like kernel with the new K_{KPM} in the SVM for training.
6. Validate physical consistency: Check that the predictions reflect the correct decay/positivity features akin to a real Green's function.

Algorithm 1 Custom Kernel Construction Using KPM

Require: Hamiltonian H , target function $f(H)$, degree N

Ensure: Custom kernel function $K(\mathbf{x}, \mathbf{x}')$

- 1: Rescale H to \tilde{H} with eigenvalues in $[-1, 1]$
- 2: **for** $n = 0$ to N **do**
- 3: Compute Chebyshev moments:

$$\mu_n = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) T_n(x)}{\sqrt{1-x^2}} dx$$

- 4: Apply kernel weights: $\mu'_n = \mu_n g_n$
- 5: **end for**
- 6: Define the custom kernel:

$$K_{\text{KPM}}(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^N \mu'_n T_n(\gamma \mathbf{x}^\top \mathbf{x}' + r). \quad (35)$$

5.3 Example: constructing a photonic crystal kernel via KPM

In Algorithm 1, we have introduced how KPM makes use of orthogonal polynomials to construct custom kernels that align with Green’s functions. We now illustrate how to apply that procedure to produce a custom kernel for photonic crystals.

Step 1: Identify the physical operator. Photonic crystals are governed by the Helmholtz-type operator

$$\hat{P} = \nabla^2 + \varepsilon(\mathbf{x}),$$

where $\varepsilon(\mathbf{x})$ is the spatially varying dielectric function. The Green’s function $G(\mathbf{x}, \mathbf{x}')$ of this operator satisfies

$$\hat{P} G(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}').$$

Exact, closed-form solutions can be complicated, but we know qualitatively that G decays with distance and also exhibits periodic modulations due to the periodic dielectric $\varepsilon(\mathbf{x})$.

Step 2: Approximate via orthogonal polynomials. Applying Algorithm 1 involves three main steps: (i) rescaling the operator \hat{P} so that its spectrum lies within $[-1, 1]$, (ii) expanding the resolvent or desired spectral function in a Chebyshev basis $\{T_n\}$, and (iii) damping the expansion coefficients μ_n to reduce Gibbs oscillations:

$$G(\mathbf{x}, \mathbf{x}') \approx \sum_{n=0}^N \mu'_n T_n(\psi(\mathbf{x}, \mathbf{x}')).$$

Here, $\psi(\mathbf{x}, \mathbf{x}')$ is some function reflecting the distance or inner product between \mathbf{x} and \mathbf{x}' . For the Chebyshev expansion to be valid, the argument of the polynomials must be scaled to the interval $[-1, 1]$. Therefore, we define a rescaled function, for example, $\psi'(x, x') = a\psi(x, x') + b$, where the constants a and b are chosen such that ψ' maps the relevant range of distances for the dataset into this interval. The expansion is then performed on this rescaled variable.

Step 3: Embed periodicity for photonic crystals. To capture the photonic crystal’s periodic structure, we can combine the polynomial expansion with a periodic factor. For instance, set

$$\psi(\mathbf{x}, \mathbf{x}') = \gamma \|\mathbf{x} - \mathbf{x}'\|^2$$

and

$$\Pi(\mathbf{x}, \mathbf{x}') = \cos\left(\frac{2\pi}{p} \|\mathbf{x} - \mathbf{x}'\|\right).$$

As $\Pi(\mathbf{x}, \mathbf{x}')$ is not always positive due to the oscillatory nature of the cosine term, it needs to be replaced with a non-negative variant such as

$$\Pi^+(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \left(1 + \cos\left(\frac{2\pi}{p} \|\mathbf{x} - \mathbf{x}'\|\right) \right)$$

which maps the cosine output from $[-1, 1]$ to $[0, 1]$ while retaining its periodic structure.

Then we define:

$$K_{\text{crys}}(\mathbf{x}, \mathbf{x}') = \left[\sum_{n=0}^N \mu'_n T_n(\psi(\mathbf{x}, \mathbf{x}')) \right] \times \Pi^+(\mathbf{x}, \mathbf{x}').$$

In practice, one must ensure that the overall kernel remains positive definite, e.g. by restricting the amplitude of the cosine term and verifying nonnegative expansions.

6 Conclusion and future outlook

In this paper, we have demonstrated a mathematical connection between Green’s functions (and, by extension, quantum propagators) and SVM kernels. We have furthermore shown that this connection leads to significant improvements in the predictive accuracy of SVM kernels arising from their correspondence with particular Green’s functions.

The kernels we tested were: the linear kernel, the RBF kernel, the polynomial kernel, and the sigmoid kernel. The sigmoid kernel proved to be an exception in the sense that it failed to capture the necessary characteristics of quantum mechanical phenomena. The reason for that was that the sigmoid kernel is positive semi-definite only under certain conditions. In cases where it is not positive semi-definite, it violates Mercer’s theorem, under which only positive semi-definite kernels ensure convergence. We then proposed a custom construction method for kernels to align with Green’s functions. The limitation of the method is that it doesn’t always guarantee that the resulting kernel is positive semidefinite. In the cases where it is not, one must additionally apply some correction techniques to approximate the kernel to a positive semidefinite matrix.

In future work, we plan to explore other quantum systems where the kernel-Green’s function correspondence could be applied, such as superconductors and topological insulators. We also want to develop a framework for automated kernel design that would generate custom kernels

based on the physical properties of a system. Another interesting approach would be to explore the applicability of KPM-based custom kernels in algorithms other than SVM, to see if the method can be generalized. And lastly, it is our hope that further study of the underlying theoretical correspondence between the kernels and the Green's functions may lead to a potential discovery of other mathematical relationships.

Acknowledgments

The authors wish to express their sincere gratitude to Professor Chong-Der Hu of National Taiwan University and Assistant Professor Tsung-Wei Chiang of National Chung Cheng University for their valuable suggestions and insightful discussions on many-body physics, which greatly enhanced the quality of this work. Renata Wong acknowledges support from the National Science and Technology Council grant No. NSTC 114-2112-M-182-002-MY3 and from Chang Gung Memorial Hospital grant No. BMRPL94.

Code availability

Source code is available at GitHub repository <https://github.com/kuonanhong/Code Available/Quantum Journal-Kernel and Green Function Correspondence>. The code is organized into modules corresponding to each application:

- `svm_conductivity.py` for electrical conductivity using the RBF kernel.
- `scattering_linear_kernel.py` for electron-proton scattering using the linear kernel.
- `anharmonic_polynomial_kernel.py` for anharmonic oscillator energy levels using polynomial kernels.
- `custom_kernel.py` for photonic crystal modeling using a custom periodic kernel.

The repository includes detailed instructions for reproducibility and examples of parameter tuning.

References

- [1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1999.
- [3] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
- [4] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals*, McGraw-Hill, 1965.
- [5] C. R. Deeter and J. M. Gray, "The discrete Green's function and the discrete kernel function," *Discrete Mathematics*, vol. 10, no. 1, pp. 29–42, 1974.
- [6] E. B. Davies, "The equivalence of certain heat kernel and Green function bounds," *Journal of Functional Analysis*, vol. 71, no. 1, pp. 88–103, 1987.
- [7] G. E. Fasshauer, "Green's Functions: Taking Another Look at Kernel Approximation, Radial Basis Functions, and Splines," *Springer Proceedings in Mathematics*, vol. 13, pp. 37–63, 2011.
- [8] D. S. Dean, P. Le Doussal, S. N. Majumdar, G. Schehr, and N. R. Smith, "Kernels for non interacting fermions via a Green's function approach with applications to step potentials," *Journal of Physics A: Mathematical and Theoretical*, vol. 54, pp. 084001, 2021.
- [9] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz, "DeepGreen: deep learning of Green's functions for nonlinear boundary value problems," *Scientific Reports*, vol. 11, pp. 21614, 2021.
- [10] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Graph kernel network for partial differential equations," *arXiv:2003.03485*, 2020.
- [11] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
- [12] G. S. Kimeldorf and G. Wahba, "Some results on Tchebycheff spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.

- [13] J. Mercer, “Functions of positive and negative type and their connection with the theory of integral equations,” *Philosophical Transactions of the Royal Society A*, vol. 209, pp. 415–446, 1909.
- [14] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 2, Wiley-Interscience, 1962.
- [15] I. Alabdulmohsin, X. Gao, and X. Zhang, “Support Vector Machines with Indefinite Kernels,” in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2015, pp. 32–47.
- [16] J. Chen and J. Ye, “Training SVM with Indefinite Kernels,” in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 232–239.
- [17] H.-M. Xu, H. Xue, X.-H. Chen, and Y.-Y. Wang, “Solving Indefinite Kernel Support Vector Machine with Difference of Convex Functions Programming,” in *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 2017, pp. 2782–2788.
- [18] C. Loosli, G. Canu, and C. S. Ong, “Learning SVMs with indefinite kernels,” in *Proceedings of the 22nd International Conference on Artificial Neural Networks (ICANN)*, 2012, vol. 1, pp. 442–451.
- [19] Y. Ying, C. Campbell, and M. Girolami, “Analysis of SVM with Indefinite Kernels,” in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012, pp. 687–694.
- [20] X. Huang, A. Maier, J. Hornegger, and J. A. K. Suykens, “Indefinite kernels in least squares support vector machines and principal component analysis,” *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 162–172, 2017.
- [21] R. Luss and A. d’Aspremont, “Support vector machine classification with indefinite kernels,” in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007, pp. 321–328.
- [22] I. Alabdulmohsin, X. Gao, and X. Zhang, “Support vector machines with indefinite kernels,” *Journal of Machine Learning Research: Workshop and Conference Proceedings*, vol. 39, 2014, pp. 32–47.
- [23] X. Huang, A. Maier, J. Hornegger, and J. A. K. Suykens, “Indefinite Kernels in Least Squares Support Vector Machines and Principal Component Analysis,” *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 162–172, 2017.
- [24] H.-T. Lin and C.-J. Lin, “A Study on Sigmoid Kernels for SVM and the Training of Non-PSD Kernels by SMO-Type Methods,” *Neural Computation*, vol. 16, no. 5, pp. 1071–1090, 2004.
- [25] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in B. Schölkopf, C. J. C. Burges, and A. J. Smola (eds.), *Advances in Kernel Methods: Support Vector Machines*, MIT Press, pp. 185–208, 1998.
- [26] R. N. Silver and H. Röder, “Calculation of densities of states and spectral functions by Chebyshev recursion and maximum entropy,” *Physical Review E*, vol. 56, no. 4, pp. 4822–4829, 1994.
- [27] A. Weisse, G. Wellein, A. Alvermann, and H. Fehske, “The kernel polynomial method,” *Reviews of Modern Physics*, vol. 78, no. 1, pp. 275–306, 2006.
- [28] J. Doe and A. Smith, “The Kernel Polynomial Method Based on Jacobi Polynomials,” *Journal of Computational Physics*, vol. 400, pp. 123–145, 2025.
- [29] A. Jain *et al.*, “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation,” *APL Materials*, vol. 1, no. 1, p. 011002, 2013. Available: <https://materialsproject.org>
- [30] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support Vector Regression Machines,” in *Advances in Neural Information Processing Systems*, vol. 9, MIT Press, 1997.
- [31] G. D. Mahan, *Many-Particle Physics*, 2nd ed., Plenum Press, 1990.
- [32] M. G. Genton, “Classes of Kernels for Machine Learning: A Statistics Perspective,” *Journal of Machine Learning Research*, vol. 2, pp. 299–312, 2001.
- [33] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 2nd ed., Pearson, 2014.

- [34] L. D. Landau and E. M. Lifshitz, *Quantum Mechanics: Non-Relativistic Theory*, 3rd ed., Pergamon Press, 1977.
- [35] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals: Molding the Flow of Light*, 2nd ed., Princeton University Press, 2008.