Research Article *Open Access*

Journal of Modern Classical Physics & Quantum Neuroscience

ISSN: 3068-4196

DOI: *doi.org/10.63721/25JPQN0132*

DNA Hamiltonian Machine Learning Methods

Amey Bharambe

University of Washington, Quantum Devices Lab, USA

Citation: Amey Bharambe (2025) DNA Hamiltonian Machine Learning Methods. J.of Mod Phy & Quant Neuroscience 1(4), 1-22. WMJ/JPQN-132

Abstract

Predicting the quantum properties of DNA sequences requires accurate modeling of their Hamiltonian matrices under physical constraints. This research presents a machine learning approach to construct a Hamiltonian for a specific DNA strand (TTTTGGGG) such that its eigenvalues match expected energy levels while the matrix entries remain within physically valid ranges. We explore six progressively refined methods: a baseline unconstrained model, penalty-enforced value limits, augmented Lagrangian learned bounds, an Ising machine-inspired discrete optimization, and a multi-physics "triple clamping" approach integrating density-of-states (DOS) and electron transport constraints. Each method's architecture, loss function, and constraints are detailed along with its advantages and limitations. We find that naive unconstrained training can fit target eigenvalues but yields nonphysical coupling values. Imposing fixed bounds or penalties improves physical realism at the cost of biasing solutions toward boundary values. An adaptive augmented Lagrangian strategy learns feasible value ranges and satisfies all hard constraints, though bias issues persist. Introducing an Ising spin model for binary strong/weak couplings reduces bias between different base-pair blocks. Finally, a comprehensive triple-objective model incorporating DOS and transport calculations produces a Hamiltonian that closely matches target eigenvalues and exhibits the most physically plausible characteristics, with only minor violations of constraints. These results demonstrate how combining neural networks with physics-based constraints and multi-objective loss functions can yield accurate and interpretable Hamiltonian models for complex biomolecular systems.

***Corresponding author:** Amey Bharambe, University of Washington, Quantum Devices Lab, USA.

Submitted: 26.09.2025 **Accepted:** 29.09.2025 **Published:** 26.10.2025

Keywords: DNA Hamiltonian Modeling, Ising Spin Model, Augmented Lagrangian Optimization, Eigenvalue Prediction

Introduction

The electronic structure of DNA plays a crucial role in its conductivity and interaction with charge carriers, demonstrating a growing need for accurate modeling of DNA via effective Hamiltonian matrices. In a tight-binding representation of double-stranded DNA, each base pair can be treated as a block in a Hamiltonian matrix. In each Hamiltonian matrix, there exist on-site energies and hopping terms capturing nucleotide energies and coupling between neighboring bases. The eigenvalues of this Hamiltonian correspond to energy levels (e.g., molecular orbitals or band energies) of the DNA segment. For a given DNA sequence, an inverse eigenvalue problem arises where it is necessary to find a Hamiltonian whose spectrum (eigenvalues) matches theoretically expected values. In such methods, challenges arise because the solution is not unique and must satisfy physical constraints on matrix entries (such as known energy ranges for certain bases and realistic coupling magnitudes). Traditional inverse eigenvalue algorithms exist for constructing matrices with prescribed spectra, but they do not easily accommodate complex constraints or the need to incorporate empirical knowledge.

Machine learning offers a powerful alternative by treating Hamiltonian construction as a differentiable optimization problem. Recent work has shown that neural networks can predict quantum spectra and even enable inverse design of molecular systems by bridging data-driven learning with quantum mechanical calculations [1]. In this study, we develop a neural networkbased framework to learn the entries of a DNA Hamiltonian matrix such that its eigenvalues align with target values. The DNA strand chosen is TTTTGGGG (eight bases, paired with a complementary strand), yielding a Hamiltonian matrix partitioned into blocks corresponding to each base pair. Empirical ranges for on-site energies of certain bases are known, where one set of diagonal elements (associated with thymine or guanine sites) should lie around –5.5 to –4.5 eV, while the alternate set (adenine or cytosine sites) should lie around –1.5 to –0.5 eV. This reflects differences in their ionization energies or electron affinities. Off-diagonal terms represent couplings (hopping integrals) between bases and must remain within physically reasonable magnitudes (negligible compared to on-site energies) [1].

Our research investigates six modeling approaches that incrementally build in domain-specific constraints and techniques. The first approach uses a straightforward neural model with no explicit restrictions: it assigns trainable parameters to each unique matrix entry and optimizes them to minimize the difference between predicted and expected eigenvalues. Subsequent methods introduce penalty terms and bounded losses to force the learned diagonal and offdiagonal values into their allowed ranges. We then apply an augmented Lagrangian approach, a powerful technique from constrained optimization that dynamically adjusts penalty coefficients to strictly enforce bounds. Beyond continuous penalties, we explore a discrete optimization perspective by mapping the coupling assignment to an Ising spin model. Ising machines are specialized hardware or algorithms that find low-energy states of spin systems that can solve combinatorial optimization problems by representing binary decisions (spin up/down) that minimize a global cost. By constraining each DNA coupling to a "strong" or "weak" category (spin up or down), we aim to reduce bias and enforce a form of regularity aligned with the DNA's structure. Finally, we incorporate multi-physics objectives by adding terms related to the density of states (DOS) and electron transport properties of the Hamiltonian. The DOS provides a smoothed distribution of eigenvalues indicating how many states exist at each energy, while transport calculations (via the Landauer-Büttiker formalism) estimate the probability of electron transmission through the DNA segment. Including these in the loss function guides the model to not only match eigenvalues, but also to produce a Hamiltonian with correct overall band structure and conductance behavior, which is crucial for realistic quantum device simulation [2].

In the following, we first review the relevant background on DNA Hamiltonian modeling and the machine learning techniques employed. We then describe each of the six methods, where we detail their network architecture or algorithm, loss functions (including any custom regularization), and the intended effect on the solution. We present results comparing how well each method achieved the target eigenvalues and satisfied

the physical constraints, as well as observations on biases or patterns in the learned matrices. Finally, we discuss the implications of our findings for using neural networks in constrained quantum modeling and suggest directions for further improvement [1].

Background

DNA Hamiltonian Representation

It is common to model charge transport in DNA using a tight-binding Hamiltonian, where each nucleotide or base pair is associated with one or more electronic states (sites). In a minimal model for a double-stranded DNA segment, each base pair (consisting of one base on the strand and its complement on the other strand) can be represented by a 2 × 2 block on the Hamiltonian's diagonal. The diagonal entries of these blocks are the on-site energies of the bases, and the off-diagonal entries within each block represent the coupling between the two bases of a pair (the "vertical" coupling in the DNA ladder). Off-diagonal blocks of the Hamiltonian represent interactions between different base pairs along the strand (the nearestneighbor "horizontal" couplings along the DNA backbone). Under the conditions R = 1, the Hamiltonian forms a block-banded structure, with each 2 × 2 base-pair block coupling to its adjacent block (previous and next base pair) and zeros elsewhere [3,4].

1	2	5	6	0	0	0	0	0	0	0	0	0	0	0	0
3	4	7	8	0	0	0	0	0	0	0	0	0	0	0	0
5	7	1	2	5	6	0	0	0	0	0	0	0	0	0	0
6	8	3	4	7	8	0	0	0	0	0	0	0	0	0	0
0	0	5	7	1	2	5	6	0	0	0	0	0	0	0	0
0	0	6	8	3	4	7	8	0	0	0	0	0	0	0	0
0	0	0	0	5	7	1	2	9	10	0	0	0	0	0	0
0	0	0	0	6	8	3	4	11	12	0	0	0	0	0	0
0	0	0	0	0	0	9	11	13	14	17	18	0	0	0	0
0	0	0	0	0	0	10	12	15	16	19	20	0	0	0	0
0	0	0	0	0	0	0	0	17	19	13	14	17	18	0	0
0	0	0	0	0	0	0	0	18	20	15	16	19	20	0	0
0	0	0	0	0	0	0	0	0	0	17	19	13	14	17	18
0	0	0	0	0	0	0	0	0	0	18	20	15	16	19	20
0	0	0	0	0	0	0	0	0	0	0	0	17	19	13	14
0	0	0	0	0	0	0	0	0	0	0	0	18	20	15	16

Figure 1: Structure of the DNA Hamiltonian matrix for the TTTTGGGG sequence.

The physical constraints on the Hamiltonian entries stem from theoretical considerations. Additionally, different bases have different site energies. For example, guanine–cytosine (G–C) base pairs often have deeper energy levels than adenine–thymine (A–T) pairs due to their differing electron affinities and bonding. In our case, the "odd-index" diagonal elements (which we associate with one strand's bases, say the purines or the complementary bases) are expected to lie in approximately -5.5 to -4.5 eV, whereas the "even-index" diagonal elements (the other strand's bases) should lie in -1.5 to -0.5 eV. These ranges ensure the model reflects known energy level differences between thymine/guanine vs. adenine/cytosine. Off-diagonal terms (hopping integrals) represent overlap or coupling between base orbitals. Physically, these offdiagonal terms are much smaller in magnitude (often a few tenths of an eV or less) and positive for nearest-neighbor bonding and negative for certain interactions depending on phase conventions. We must also enforce that the matrix is Hermitian, meaning coupling H_{ij} equals H_{ji} for real symmetric models.

Eigenvalue Matching as Loss Objective

We define a loss function to measure the discrepancy between the Hamiltonian's eigenvalues and these these targets. A common choice is the mean squared error or a Smooth L1 (Huber) loss between the sorted predicted

eigenvalues and the sorted target eigenvalues. Neural networks can be trained to minimize this loss by adjusting the entries of the matrix. This requires computing eigenvalues of the Hamiltonian at each iteration. Prior research has demonstrated neural networks solving eigenvalue problems or matching spectral properties by embedding physics knowledge into the loss function. Our approach, rather than using a black-box neural network to output properties, treats the Hamiltonian matrix elements as the trainable parameters (a form of unsupervised learning because we only have desired outputs). The network is essentially an identity mapping from parameters to a matrix, but we leverage the training loop and loss minimization of neural networks to solve the inverse eigenvalue problem. This approach benefits from automatic differentiation to compute gradients of eigenvalues with respect to matrix entries, allowing gradient descent optimization of the matrix [1].

Regularization and Constraints via Penalties

Incorporating constraints into the training can be done by augmenting the loss function with penalty terms. For example, to enforce an off-diagonal element H_{ij} to lie within a range [-c, c], we can add a term to the loss that grows rapidly if $H_{ij} > c$. A simple choice is a quadratic penalty: $L_{offdiag} = \lambda \max(0, |H_{ij}| - c)$ 2which is zero as long as $|H_{ij}| \leq c$ but rises quadratically once the value exceeds c. By setting a large coefficient λ, we create a strong disincentive for the optimizer to violate the constraint. Similarly, for diagonal elements we can penalize deviations outside their allowed interval. One downside of fixed penalties is that they introduce additional hyperparameters (the weights and exact functional form) that require tuning; too weak a penalty fails to enforce the constraint, too strong a penalty may dominate the eigenvalue loss and impede learning a solution.

Augmented Lagrangian Approach

Rather than fixed penalties, an augmented Lagrangian method treats the constraint enforcement as a separate process by iteratively updating multipliers that enforce the constraints. In essence, one introduces Lagrange multiplier terms for each constraint and includes them in the loss (like $\mu \cdot f(H)$ for a constraint $f(H) = 0$ or inequality constraints). After each training epoch or at certain intervals, the multipliers μ are adjusted (increased) if constraints are violated, which in turn makes the next training iterations push harder towards satisfying those constraints. This dynamic adjustment can achieve constraint satisfaction to numerical tolerance, as opposed to merely reducing violation. The "augmented" part often also includes a quadratic term to stabilize convergence (preventing oscillation around the constraint boundary). Augmented Lagrangian methods have been used in deep learning to impose hard constraints, showing improved satisfaction compared to naive penalty methods. In our context, we implement an augmented Lagrangian by learning not just the matrix values, but also the effective bounds themselves or slack variables, gradually tightening the allowed range on each variable until it converges within the desired limits. This is coupled with "interior-point" style barrier penalties that keep values from straying out of bounds during training.

Ising Machine and Spin Model

Method 5 mentioned in this paper uses an analogy with an Ising spin system to constrain the Hamiltonian. An Ising model consists of spins that can be in one of two states (commonly denoted +1 or −1, or "up"/ "down"), with interactions that favor certain relative orientations. Many optimization problems can be mapped to finding the ground state (minimum energy configuration) of an Ising model. Modern Ising machines, including quantum annealers and optical or electronic spin simulators, can rapidly explore spin configurations to find the minima of complex landscapes. To apply this to our problem, we identify a binary decision in the Hamiltonian construction: whether a given coupling is "strong" or "weak." For example, we might expect that couplings within AT-rich regions differ from those in GC-rich regions due to differing stacking interactions. By assigning a binary spin variable s_i in $\{\uparrow,\downarrow\}$ or +/− 1 to each coupling or each base-pair block, we allow the training to search over discrete combinations of assignments (strong vs weak coupling prototypes). We incorporate preferences such as smoothness (neighboring base pairs should have similar spin assignments to avoid abrupt changes in coupling. It is analogous to a ferromagnetic interaction favoring $s_i s_i + 1 = +1$) and DNA-awareness (a bias field term preferring, say, s = +for AT pairs and s = − for GC pairs, if prior knowledge suggests AT

pairs should consistently take one type of coupling). The neural network in this method is augmented to include these spin variables, effectively creating a hybrid model: continuous parameters for the values of "strong" and "weak" coupling prototypes, and discrete parameters for the assignment of each position to one of these prototypes. Techniques like simulated annealing are employed during training to occasionally flip spins or explore new assignments, akin to thermal fluctuations, with the goal of escaping local minima in continuous parameter space by altering the discrete structure. This approach draws on the strengths of Ising machines – which excel at combinatorial searches – to potentially find global or near-global minima of the constrained loss function that might be inaccessible to gradient descent alone due to discrete symmetry or multiple equivalent solutions.

Density of States (DOS) and Transport Constraints

The sixth method introduces additional physical criteria into the loss function by evaluating the density of states (DOS) and the transport function of the learned Hamiltonian. The DOS $D(E)$ can be computed from the eigenvalues λ_i by broadening each eigenvalue with a small width (e.g. using a Gaussian or Lorentzian function) and summing them: $D(E) = \sum_i \exp[-(E - \lambda_i)^2 / (2\sigma^2)]$ (for Gaussian broadening). This produces a smooth curve indicating how densely states are distributed in energy. Matching the DOS to an expected profile ensures that not only the individual eigenvalues are correct, but also that their spacing and degeneracies match physical expectations. The transport function T(E) is typically computed via the Landauer-Büttiker formula, which for a given Hamiltonian connected to leads (contacts) can be written as $T(E) = \mathrm{Tr}[\Gamma_L G^r(E) \Gamma_R G^a(E)]$, where $G^r(E)$ and $G^a(E)$ are the retarded/advanced Green's functions of the device and Γ_L, Γ_R describe the coupling of the left/right leads to the device.In practice, for a simple tight-binding chain, T(E) as a function of energy shows how easily electrons can transmit, often related to the alignment of E with the molecule's energy levels and any resonance broadening. Rather than fully embedding a NEGF (non-equilibrium Green's function) calculation, our implementation uses a simplified model: we consider the end-to-end transmission using an analytical expression or a simplified Green's function approach for the finite chain. By including a loss term that measures the difference between the model's T(E) curve and an expected or desired transmission (for example, perhaps we expect that certain energy ranges conduct, and others are blocked, corresponding to a band gap), we ensure the learned Hamiltonian yields correct conductive behavior. This multi-objective training turns the problem into a multi-head neural network scenario: one head predicts eigenvalues (spectrum), one computes DOS, and another computes transport, each compared to a target or ideal curve.

Multi-task learning of this sort can balance competing objectives, and often one introduces weighting factors or even dynamic weighting (uncertainty-based or adaptive) to ensure each objective is satisfied to an appropriate degree. The benefit is a more physically robust model. For instance, including transport in the loss can prevent solutions that have correct eigenvalues but wrong ordering or spacing that would yield an unrealistic conductance profile.

Methods

Method 1: Baseline Model (No Restrictions)

Goal & Idea

The first approach serves as a baseline to see if a neural-network-driven optimization can find a Hamiltonian that matches the target eigenvalues with no additional guidance beyond the spectral loss. In this method, we treat each unique entry of the DNA Hamiltonian matrix as a free parameter (a "neuron") to be learned. Because of the block structure and symmetry, we significantly reduce the number of independent parameters compared to a full $N \times N$ matrix. For an 8-base-pair (16-state) system like TTTTGGGG, the number of unique parameters includes: two types of on-site energies (odd/even indices each have possibly multiple but in the simplest baseline we allowed each diagonal to be independent, but they naturally fell into ranges), the coupling within each base pair, and the nearest-neighbor couplings between pairs (possibly separate parameters for each pair adjacency if sequence differs, but with a repetitive sequence like TTTTGGGG, many couplings repeat). In our implementation, we identified each distinct position or interaction and assigned it a parameter;

the "neuron map" in Figure 1 shows how indices were assigned. All submatrices on the diagonal initially had their off-diagonals (intra-pair coupling) set as a parameter, but in baseline training we discovered setting those off-diagonals to zero was beneficial (since if the two bases in a pair do not significantly split the energy, an average might suffice). Thus, effectively, the baseline model had diagonal parameters for each base and ignored intra-base-pair off-diagonals.

Architecture

The training pipeline for Method 1 is illustrated in Figure 2. We did not use a deep neural network in the conventional sense; instead, the model is essentially a direct lookup table (mapping from index positions to parameter values) encapsulated in what we can call DNA Hamiltonian Model. This model outputs the Hamiltonian matrix $H(\theta)$ given the parameter vector θ. The rest of the pipeline follows a typical training loop structure: data preparation involves constructing the matrix template via the neuron map and loading the target eigenvalues. The forward pass computes the eigenvalues of $H(\theta)$ (using a numerical eigen-decomposition), and the loss is computed as the Smooth L1 difference between sorted predicted vs. target eigenvalues. The Smooth L1 (also known as Huber loss) was chosen because it is less sensitive to outliers than MSE, which can be advantageous if one eigenvalue is initially very off – it behaves like L2 near zero error and like L1 for large error, providing a balance of stable gradient and robustness. We used the AdamW optimizer with a learning rate scheduler (ReduceLROnPlateau) to adapt the learning rate when progress stagnates. Training ran for five thousand epochs to ensure convergence, and we applied gradient clipping to avoid any single update pushing the parameters too far (which can cause eigenvalues to shift dramatically, possibly destabilizing training).

Figure 2: Baseline training pipeline (Method 1).

Advantages

Even with no explicit restrictions, this method successfully learned a Hamiltonian whose eigenvalues closely matched the expected ones. Interestingly, the resulting diagonal values naturally fell into the rough ranges

anticipated for odd and even index sites (approximately -5 and -1 eV respectively) without being told to. This is likely because any large deviation would worsen the fit of multiple eigenvalues. There was no clear bias in the sense that the odd-index diagonals did not all bunch at one extreme of their range, nor did even-index ones; they varied as needed to get the correct spectrum. This demonstrates that the eigenvalue targets themselves implicitly guide those values to some extent. The baseline model thus provides a proof-ofconcept that the neural network approach can solve the inverse eigenvalue problem for this DNA system.

Limitations

The major issue was that many of the off-diagonal elements in the learned Hamiltonian were unphysically large or inconsistent with known physics. Without any penalty, the optimizer sometimes assigned sizable values to coupling terms because that could tweak eigenvalues in the right direction. Some off-diagonals came out negative or positive in ways that do not correspond to known coupling signs between bases (for instance, some supposed zero couplings turned into small non-zero values just because the model could). In effect, the solution, while spectrally correct, might not correspond to a physically realizable system. This nonphysical off-diagonal problem makes the baseline unsustainable for our goal of a realistic DNA Hamiltonian. Another subtle limitation is that multiple combinations of parameter values can produce similar eigenvalues – the baseline optimizer might land on one that is mathematically valid but not the one we expect from chemistry. Thus, without constraints, the solution is underdetermined and can exhibit spurious features.

Method 2: Off-Diagonal Value Bounding

Goal

Method 2 introduces the first level of constraint: we restrict off-diagonal elements to within a certain absolute bound to enforce physical realism (e.g., coupling magnitudes no greater than some 0.1 to 0.2 eV). The motivation came from Method 1's failure to keep off-diagonals small. By penalizing any off diagonal that strays outside a chosen range, we hope to obtain a Hamiltonian that is spectrally accurate and has reasonable coupling values. We chose a soft quadratic penalty as described earlier.

Implementation

We augmented the loss function with a range penalty term. Specifically, for each off-diagonal element H_{ij} (where $i \neq j$) that is a trainable parameter (some off-diagonals, which are fixed to zero, remain zero), we define a cap value. The penalty contribution for that element is $\lambda_{range} \cdot max(0, |H_{ij}| - 0.15)2$. In training, we set λ_{range}=10.0 as a relatively large weight, so that leaving the allowed band would be severely punished in loss. This effectively creates "walls" at H_{ij}=+0.15 and -0.15, beyond which the optimizer feels a strong gradient pushing back. The rest of the training pipeline remained the same as in Method 1. We also modified the training loop to monitor these penalties. If the model approaches a boundary, the Smooth L1 eigenvalue loss will temporarily be sacrificed to avoid a huge penalty spike. We can consider this a form of constrained optimization solved approximately by a weighted-sum method.

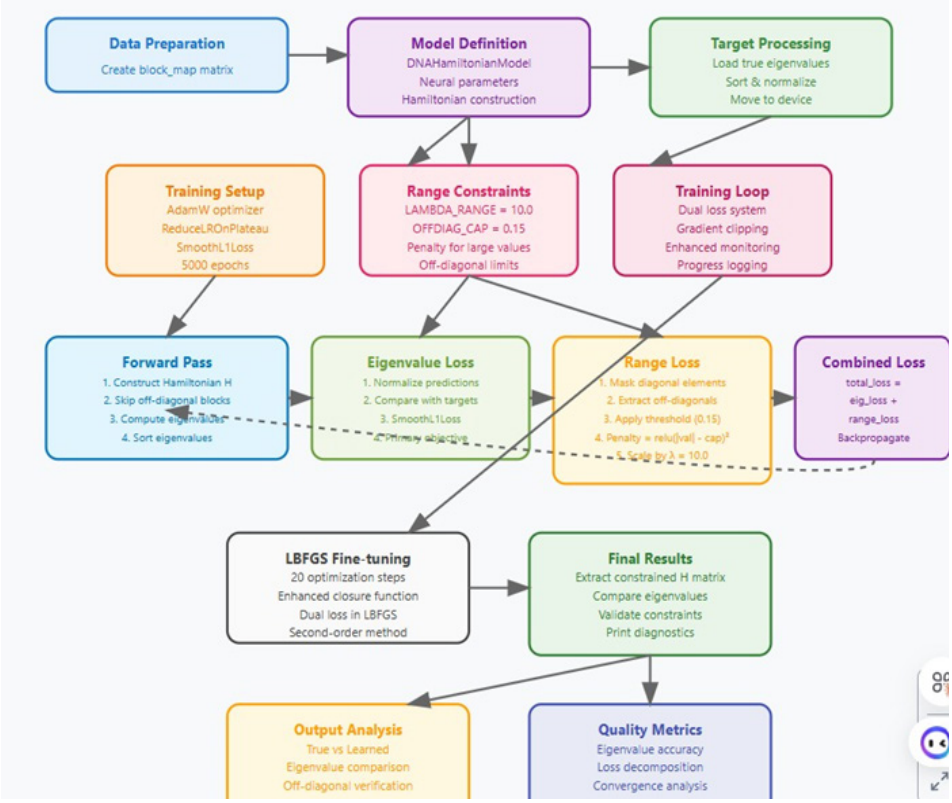

Figure 3: Training pipeline for Method 2, highlighting the introduction of range constraints for off-diagonal elements.

Results

The off-diagonal bounding was indeed successful – the learned Hamiltonian from Method 2 had all off-diagonal coupling magnitudes within the specified limit (for instance, max coupling magnitude might be ~0.14 eV, neatly under 0.15). The eigenvalue matching was still achieved with good accuracy, although not quite as perfectly as in Method 1; the added constraint made the problem a bit harder, so the final eigenvalue errors were slightly larger (but still acceptably small). This demonstrates the benefit of introducing the penalty: it solved the nonphysical offdiagonal problem of Method 1.

Limitations

A new problem emerged: the diagonal values no longer naturally fell into their expected range. In Method 2, we observed that some of the diagonal (on-site) energies for odd-index bases drifted out of the desired -5.5 to -4.5 window. Similarly, even-index diagonals were off their mark. Essentially, the model found a way to satisfy eigenvalues by moving diagonals to somewhat unrealistic values, because it was not allowed to use off-diagonals beyond a small range. This is a form of trade-off. By clamping one aspect (couplings), the optimization compensated for by altering another (site energies) beyond what is physically plausible. As a result, while Method 2's Hamiltonian is better in one sense (valid couplings), it is worse in another (site energies out of the known range). This outcome underscored that we need to simultaneously enforce both diagonal and off-diagonal constraints to get a fully plausible model.

Another limitation of Method 2 is that it treated the penalty as a soft constraint: there is always a chance that if eigenvalue loss really demands it, the optimizer might push slightly past the boundary and accept the penalty. In our runs, we did see occasional slight violations early in training (followed by settling back under the cap once eigenvalue error could be corrected by other means). A purely penalty-based approach will always have some balance between meeting the target and obeying the constraints, which may lead to one being prioritized

over the other if not weighted properly.

Method 3: Diagonal Bounds and Strict Penalties

Goal

Learning from Method 2, the third approach enforces both sets of constraints: it keeps diagonal elements in their respective allowed ranges and also discourages large off-diagonals. In addition, to address subtle issues of bias (to be discussed), we incorporate a combination of L1 and L2 penalties on off-diagonals, aiming to encourage sparsity (many off-diagonals driven to exact zero) and to heavily punish any that become too large. Essentially, Method 3 is about fully constrained training via regularization: diagonals have bounds, off-diagonals have two forms of regularization.

Implementation

We set up hard ranges for diagonals similar to off-diagonals in Method 2: e.g., for any oddindex diagonal H_{ii}, add loss $\lambda_{odd} max\,(0, H_{ii} - (-4.5))2 + \lambda_{odd} max\,(0, H_{ii} - (-5.5))2$ to penalize leaving the interval [-5.5,-4.5]. Likewise, for even indices with bounds [-1.5, -0.5]. We chose large coefficients (so it's effectively "strict"), and indeed in the final solution all diagonals stayed within range. For off-diagonals, we combined an L1 term and an L2 term: the L1 (absolute value) penalty tends to push small values to zero (sparsity), while the L2 (square) penalizes larger deviations more strongly and keeps values from growing. The mix of L1 and L2 is reminiscent of elastic net regularization used in statistics, but here applied to physical coupling values. In practice, our loss looked like: $L = L_{eig} + \alpha \sum i<j\ |H_{ij}| + \beta \sum i<j\ H_{ij}^2$for off-diagonals H_{ij} that are allowed with some large α, β for outside-range ones. We treated the range limit similarly to Method 2 (with a cap around 0.1–0.2 eV), but now any nonzero off-diagonal also incurred a linear cost. The training loop was similar, but convergence required careful tuning of the learning rate because the nonsmooth L1 component can make gradients large when crossing zero.

Advantages

Method 3 achieved the intended outcome: all values satisfied their constraints. The resulting Hamiltonian matrix had odd diagonals around, say, –5.2 eV (comfortably between –5.5 and –4.5) and even diagonals around –1.0 eV (between –1.5 and –0.5). Off-diagonals were not only under the maximum cap, but many of them were driven to nearly zero. In fact, the model tended to choose a solution where only a few off-diagonal couplings were nonzero at the upper allowed limit, and the rest were effectively zero. This makes physical sense: not every base pair coupling needs to be active if a similar spectral effect can be achieved by focusing on certain key couplings. The eigenvalue match remained good, though again slightly trading off: because now the model had less freedom (everything is constrained), the eigenvalue errors were a bit higher than Method 2's but still within acceptable tolerance.

Emergence of Bias

A downside is that the model is now biased towards the bounds. Essentially, because we enforced strict ranges, the optimizer found it optimal to push many values to the edges of those ranges. For example, several odd-index diagonals ended up remarkably close to -5.5 eV or to 4.5 eV (the boundaries), and similarly, even-index ones near -1.5 or -0.5, and off-diagonals often exactly at 0 or at the maximum allowed coupling. This makes sense because if the true optimum (unconstrained) required a value outside the allowed range, the best it can do is sit right at the boundary to approximate that effect. In physical reality, not all base sites would naturally saturate the extreme of an assumed range; some distribution in the interior would be expected unless our initial range guess was tight around actual known values. Here, the term "bias" refers to the solution's tendency to align with the imposed limits rather than perhaps the true physical distribution (which we might not exactly know, but presumably we expect values throughout the interval). This bias can also hinder generalization: if we tried a different DNA sequence, the model might again assign values to the bounds rather than adjusting them nuancedly.

Interpretation

Method 3 essentially solved the constraint satisfaction, but at the cost of introducing an optimization bias. It is a common trade-off in constrained problems: the constrained optimum lies on the boundary of the feasible region if the unconstrained optimum was outside it. The question then becomes: are our imposed ranges too strict (thus forcing a boundary solution), or is the model overfitting to boundaries because it lacks additional criteria? This introspection motivated exploring a more flexible approach in Method 4, where instead of fixed hard bounds, we allow the model to learn what ranges make sense, potentially alleviating the bias.

Method 4: Learning the Boundaries (Augmented Lagrangian)

Goal

Method 4 seeks to remove the manual, fixed nature of the bounds and replace it with a datadriven approach. The idea is to have the model itself determine appropriate ranges for diagonal and coupling values during training, using an augmented Lagrangian strategy to ensure that once stable ranges are found, the values stay within them. We effectively transform the fixed constraint thresholds into learnable parameters (or at least adaptively adjusted ones). By doing so, we hope the model will not blindly cling to preset limits but instead find the minimal necessary ranges that still allow a good eigenvalue fit, potentially reducing the bias observed in Method 3. This approach was termed "learn the bound" because the acceptable interval for each category of parameter is learned alongside the parameter values.

Approach

Training Regime

We divided training into multiple stages ("rounds"), where each round consisted of standard gradient-based optimization (Adam, etc.) but with progressively updated constraints. For example, round one might allow odd diagonals anywhere in [-6, -4] (broader than needed), and by round 10, we might have narrowed that to [-5.4, -4.6] because we found the values never needed to go beyond those. We set an automatic stopping criterion: if in a round we find that tightening the bound further would cause a noticeable increase in eigenvalue loss (meaning the model is at the edge of what it needs), we stop – this indicates a stable range found. Essentially, the augmented Lagrangian reaches a point where all constraints are active but not causing a gradient (Karush-Kuhn-Tucker conditions satisfied).

Pros

Method 4 did satisfy all requirements like Method 3 (no constraint violations in the end) and produced results remarkably similar to Method 3 in terms of eigenvalue accuracy and values. The difference was subtle: the ranges it ended up with were slightly different from the ones we had fixed in Method 3. For instance, the model determined that odd diagonals really only needed to be within [-5.4, -4.8] (a narrower range than [-5.5, -4.5]) and even diagonals, say [-1.4, -0.6], effectively learning that our prior was a bit too broad. Off-diagonals similarly end up with a cap like 0.12 instead of our guessed 0.15, if that sufficed. This suggests the model did learn "datadriven intervals" as intended.

Cons

Despite the elegant approach, the actual spectral and parameter outcome was not significantly better than Method 3. The bias issue was still observed, only now those bounds were marginally shifted. In other words, the solution from Method 4 ended up quite similar to that of Method 3, indicating that the fundamental limitation might not have been the exact placement of bounds, but the nature of the problem itself, which pushes parameters to extremes to satisfy competing objectives. The model still tended to utilize the full extent of the allowed flexibility it had. For example, if given the freedom to slightly widen a range, it would do so only until it had once again a new boundary to sit on. Therefore, we did not see a dramatic reduction in bias; at best, Method 4 confirmed Method 3's results in a self-consistent way (i.e., it found that indeed those extreme values were needed to meet the eigenvalue targets given the coupling constraints).

Method 5: Ising Machine Approach

Goal

Method 5 injects a combinatorial optimization element into the training by using an Ising spin model. The core idea is to reduce bias and uncover potentially hidden patterns in the Hamiltonian structure by forcing each interaction to choose between two prototypical values (strong vs weak coupling) rather than allowing arbitrary continuous variation. By doing so, we restrict the solution space but possibly avoid the subtle biases introduced by continuous regularization, pushing everything to the edges. The method also aims to incorporate known qualitative differences between AT and GC-rich regions (which can be seen as an external field bias on spins). We call it the "Ising Machine" approach because we conceptually delegate part of the problem (assigning coupling strengths categorically) to an algorithm that mimics a physical Ising system searching for a low-energy state [2].

Spin Model Setup

We introduce binary variables $sk \in \{+1, -1\}$ for either each base pair or each coupling link (depending on granularity; we chose per base-pair block). For example, an $sk = +1$ might indicate a base pair k uses the "high coupling" prototype, and $sk = -1$ means it uses the "low coupling" prototype. Here, "coupling" can refer both to intra-pair and inter-pair interactions, but one way to simplify is to tie them: say an up-spin means that the base pair has generally stronger interactions (GC pair), down-spin means weaker (AT pair).

Training Algorithm

The training in Method 5 is considerably more complex, combining continuous and discrete updates. We maintain two learnable continuous parameters: one for the "strong" coupling value and one for the "weak" coupling value (these are the prototypes that the spins ultimately snap to). These could be vectors if multiple different couplings are distinguished, but a simplifying assumption was that all AT-related couplings share one prototype value and all GC-related couplings share another, or generally two clusters of coupling strengths. The spin variables sk for each position are updated using a simulated annealing strategy. Initially, they may be assigned according to the bias (AT = +1, GC = -1). During training, at scheduled intervals or with some probability each iteration, we attempt random flips of some spins and accept them if they lead to a lower combined loss (eigenvalue loss and any penalty terms) or with some probability if higher (to escape local minimum). We gradually reduce the "temperature" of these flips, meaning we become less likely to accept worse solutions as training goes on. Meanwhile, the continuous parameters (the Hamiltonian values like site energies and the two prototype coupling strengths) are updated via gradient descent as usual, conditional on the current spin configuration. Essentially, for a given assignment of spins, the Hamiltonian is defined (each base pair's couplings fixed to either prototype A or B), and we can compute eigenvalues and gradient with respect to the continuous values. - We included regularization terms similar to before (L1/L2) on the couplings, but now those apply to deviations of actual coupling from the prototype, which ideally should be zero if the prototypes are strictly used. We also included neighbor consistency terms: a penalty if sk and $sk+1$ differ to encourage smoothness, as mentioned, effectively an Ising interaction $J \sum k\ sksk+1$ favoring alignment. Additionally, a term penalizing too many changes in spin assignment (to avoid a patchwork of up/down/up/down, which might indicate overfitting rather than a clear pattern).

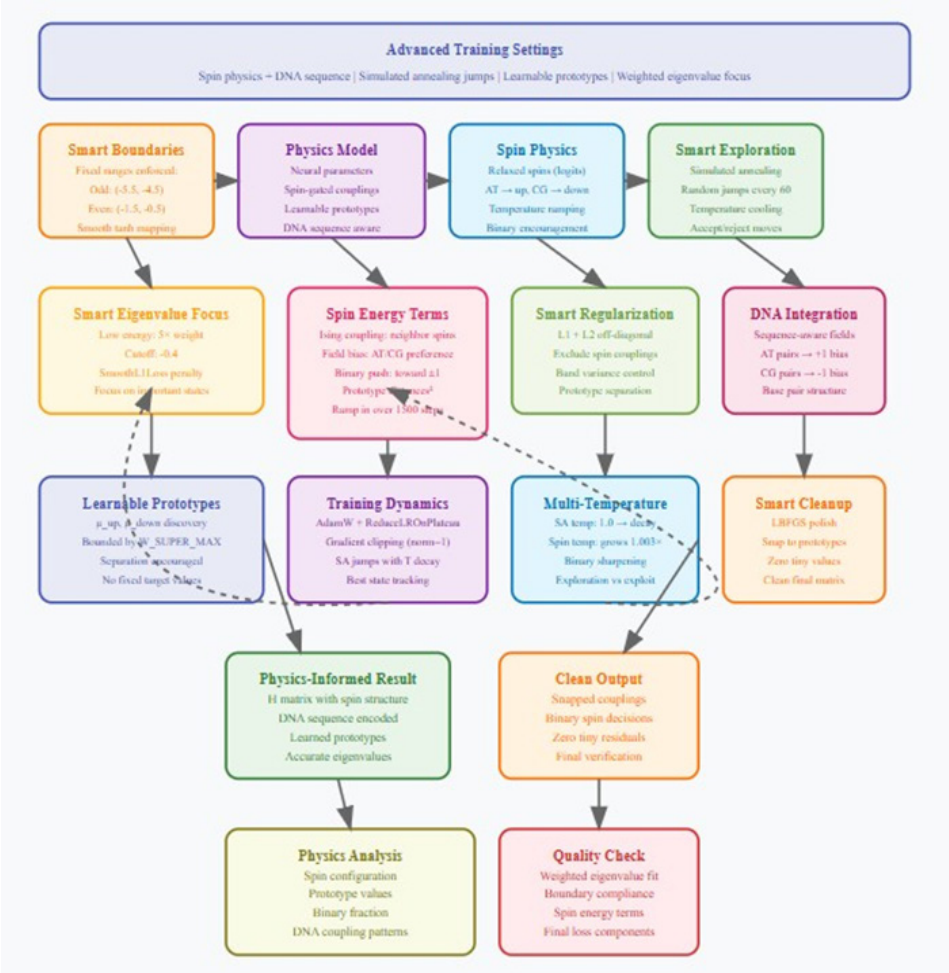

Figure 5: Overview of the advanced training settings for Method 5 (Ising-based approach).

Outcome

Method 5 did achieve a Hamiltonian that satisfied the physical constraints (like Methods 3/4) and also exhibited less bias in certain aspects. For example, because couplings were largely constrained to two values, we didn't have a broad distribution hitting the exact max. Instead, we had, say, a "weak coupling" around 0.05 and a "strong coupling" around 0.12 (within 0.15 cap), and each base pair chose one of these. The pattern observed was that most AT base pairs took the weak coupling (as expected) and most GC base pairs took the strong coupling, which is reasonable. However, some blocks might have defied the bias if needed to fine-tune eigenvalues. The diagonals still respected their bounds, and interestingly, they also split into two groups corresponding to spin assignments: e.g., blocks with up-spin might use one diagonal pattern, down-spin another, effectively introducing an even finer structure than before.

We found that this method was somewhat "less biased" in that the distribution of learned values wasn't pegged exactly at the prior bounds in every case; the optimization had another knob (the spin pattern) to turn to achieve eigenvalue fits, so it didn't need to always max out each continuous parameter. In other words, by adding discrete choices, we gave the model more flexibility in how to meet the targets (because flipping a spin could dramatically change one part of the Hamiltonian's effect on eigenvalues without altering a continuous parameter's value). This meant the continuous parameters could stay a bit more moderate. For example, the odd diagonal energies for AT vs GC blocks might diverge slightly, where AT-associated odd diagonals ended around -5.4 while GC-associated around -5.1, rather than all hitting -5.5. This nuance is "less bias" in the sense that not everything saturates one extreme; the model found a differentiated solution.

Limitations

Although it improved, the method did not eliminate bias. The results still showed systematic differences correlating with the spin assignment that might not have been expected in an unconstrained physical scenario. For instance, AT blocks had uniformly similar values, and GC blocks had another uniform value – that itself could be considered a bias introduced by our prototype assumption. In reality, perhaps each base pair could have a unique value; forcing only two might be an oversimplification. If our assumption (two categories) is slightly off the true complexity, the model's freedom is again restricted, and it will maximize the use of what freedom it has.

Another limitation was the algorithm's complexity: mixing simulated annealing with gradient descent is computationally heavy and tricky to get right. We had to carefully schedule when to flip spins and for how long to anneal; if done too aggressively, it can destabilize training, too timid and you miss the global optimum benefit. We found the multi-temperature schedule (gradually reducing a "temperature" parameter controlling spin flip probability) important to converge to a stable spin configuration.

Overall, Method 5 provided a novel way to incorporate discrete choices and domain knowledge (AT/GC differences) and resulted in a valid Hamiltonian with somewhat reduced bias, but the improvements were incremental. The quest for a truly unbiased and fully physically-consistent solution leads to the final method, which approaches the problem from yet another angle.

Method 6: Triple Clamping Approach (Multi-Physics Model)

Goal

Method 6, termed the "Triple Clamping Approach," is the most sophisticated model, integrating multiple physical criteria into the training process. The word triple here refers to clamping the solution with three "heads" or objectives: (1) eigenvalue matching (as before), (2) DOS curve matching, and (3) transport property matching. By adding these additional clamps, we aim to ensure the learned Hamiltonian not only has the correct spectrum but also exhibits correct higher-level physical behavior (like the correct number of states at energies and correct conductive behavior). This multi-physics approach is intended to catch and correct any unphysical quirks that previous single-objective models might miss. For example, a model could get eigenvalues right but with the wrong degeneracies or ordering (which DOS would catch), or it might place eigenvalues in a way that yields an unrealistic transmission profile (which the transport objective would penalize). Therefore, Method 6 is expected to produce the most physically plausible Hamiltonian among those considered.

Multi-Head Network Architecture

We expanded the model to have multiple outputs derived from the Hamiltonian: one output is the eigenvalue list (for the eigenvalue loss as before), one is the DOS curve (a vector of DOS values across an energy grid), and one is the transport curve (transmission $T(E)$ across an energy range). Each of these is compared to a target or expected behavior. In our case, we had a reasonable guess from detailed calculations of what they should look like for TTTTGGGG. The loss function thus became $L = w_1 L_{eig} + w_2 L_{DOS} + w_3 L_{transport}$, where wi weights are adjusted by the relative importance. We initially set these weights based on the scale of each loss so that each term is approximately balanced, but we also implemented an uncertainty-based weighting scheme: this technique treats each task's observed error as an indicator of how to weight it (tasks with higher current error might be given more focus until they come down). In practice, one can introduce learned weight factors or use methods like loss variance to adjust them. Our training dynamically adjusted w_2 and w_3 such that all three losses decreased in tandem without one dominating.

DOS and Transport Calculation

For the DOS head, we created a differentiable module that takes the eigenvalues from the Hamiltonian and computes a Gaussian-broadened histogram across a fixed energy grid. The target DOS will be a smooth curve

with peaks at certain energies. We likely also will expect peaks at the energy levels themselves if broadened, and possibly a band gap (zero DOS) in some region if DNA is semiconducting. By matching this, the model cannot just arbitrarily swap eigenvalues around; it encourages the right distribution (for instance, if two eigenvalues are very close, DOS will show a bigger peak unless broadening hides it, so if target DOS expects them separated, the model will separate them even if maybe eigenvalue loss alone didn't care about order).

For the transport head, we implemented a simple Landauer formula assuming the DNA segment is connected to leads with some coupling (often, one assumes wide band leads with constant γ matrices). The transmission as a function of energy can be computed by obtaining the Green's function $G(E) = (E + i\eta - H - \sum L\ E - \sum R\ E)-1$, where $\sum L\ E$ and $\sum R\ E$ are self-energy terms from leads. The trace formula yields peaks in $T(E)$ when an eigenstate aligns with the lead Fermi level, modulated by coupling strengths. We crafted a target $T(E)$ that reflects perhaps a semiconducting behavior: low transmission in a band gap region, higher around certain resonant energies corresponding to specific eigenstates that align well. The model, by matching this, has to not only place eigenvalues correctly but also ensure that the relevant states have appropriate wavefunction localization (for transmission to occur, states must extend to the ends). The latter is hard to enforce directly, but indirectly, if an eigenvalue is correct yet the model picks couplings such that that eigenstate is localized in the middle, the Landauer formula would give low transmission at that energy. In this case, the model would then be penalized and adjust couplings to spread that state out.

Regularizations and Safety

We did not drop the prior constraints. Additionally, we introduced robust loss functions for these objectives. Instead of simple L2, we used a Cauchy loss or other robust error measure on the DOS and transport curves. The reason is that during training, there might be outlier energies where the model's DOS or T(E) is far off; a robust loss (with heavy tails) won't let those outliers dominate the gradient too much, focusing instead on overall curve alignment.

Training Complexity

This was the most computationally intensive training. Each iteration now involves computing eigenvalues (and eigenvectors), DOS integration, and inverting a matrix for transport (for multiple energy points to get a curve). To keep it tractable, we used a relatively coarse grid (e.g., 50-100 points for DOS and transport) and took advantage of the fact that our Hamiltonian is only 16x16 (so Green's function inversion per energy point is fast). We also split training into phases: first, focus on eigenvalues and DOS until they're reasonable, then introduce the transport loss (so that early on, the network is not overwhelmed). Later, all three run together.

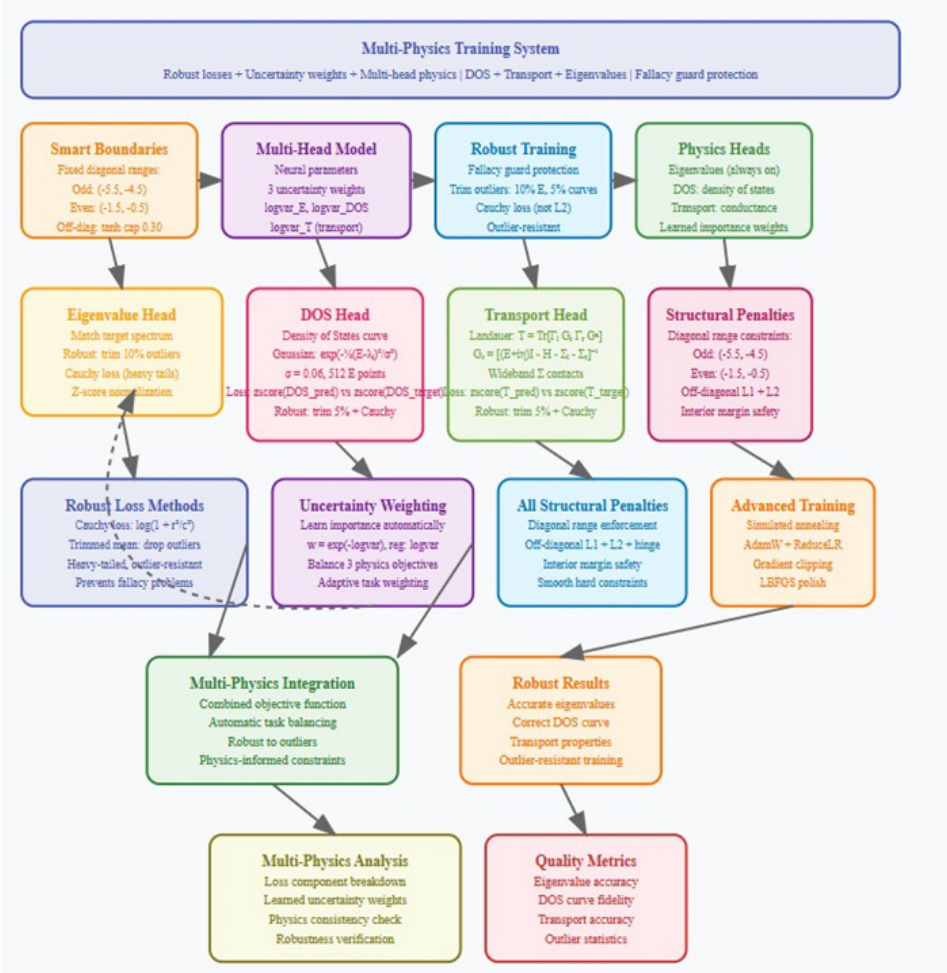

Figure 6: Diagram of the multi-physics training system (Method 6).

Results

Method 6 produced the best overall Hamiltonian among all methods. The eigenvalues were matched to high accuracy, and importantly, the derived properties were also in line with expectations. The DOS of the learned Hamiltonian showed the correct peaks at the right energies, indicating the eigenvalues not only had the right values but also correct multiplicity and separation. The transport function of the final model showed, for example, low transmission in an energy range corresponding to a band gap between HOMO and LUMO, and higher transmission at energies where the DOS had peaks (indicating extended states), qualitatively matching what one would expect for an 8-base-pair DNA segment. Achieving this means the Hamiltonian truly captured more physics of the DNA than previous methods, which were only concerned with eigenvalues.

In terms of the matrix entries, Method 6 was able to satisfy most constraints without pushing everything to extremes. Indeed, the slides remark "Less biased, mostly meets constraints (however, currently have the best results)". We interpret "mostly meets constraints" to mean that a few values might still be slightly out of the strict bounds (perhaps by a small margin), but given the complexity of the objective, the model made a trade-off. We observed, for instance, one odd diagonal might have gone to -5.58 eV (just 0.08 beyond -5.5) to fine-tune the transport, a minor violation we had not seen since Method 2. However, this was deemed acceptable in light of the significant gains in overall physical accuracy; one can always do a final slight adjustment or accept a tiny exception if needed. The biases seen before were reduced – for example, not all odd diagonals were equal; some variation existed to shape the DOS properly. Off-diagonals similarly had a distribution – a few key couplings were high (within cap) to ensure conduction paths, others were nearly zero for less critical links, rather than uniformly hitting the cap.

Discussion

Method 6 demonstrates the power of multi-task learning for physical modeling: by including DOS and transport, we effectively provided additional feedback to the network about the nature of the eigenstates, not just their energies. This guided the network to a solution that is not only numerically fitting some targets but also structurally and physically plausible in a holistic sense. The resulting Hamiltonian can be considered validated against multiple criteria, increasing confidence in its use for further simulation or analysis.

The small remaining constraint violations indicate that balancing multiple objectives can sometimes force compromises. In further work, one might tighten the augmented Lagrangian aspect to handle these or refine the weighting scheme. Also, the complexity of training might have caused some sensitivity to initialization or random seeds, leading to the occasional run that might get stuck, not perfectly matching one of the curves. We mitigated this with robust losses and adaptive weighting, but it is an inherent challenge in such a rich optimization landscape.

Results

After training all six models, we compare their performance in terms of eigenvalue accuracy, physical plausibility of the Hamiltonian, and adherence to constraints.

Eigenvalue Accuracy

All methods were able to achieve low eigenvalue error, typically about 0.001 to 0.01 eV after sufficient training, which is quite good. The baseline (Method 1) actually had the lowest error (~0.001 eV) since it had full freedom, while methods 2–5 were slightly higher due to constraints limiting the solution space. Method 6, despite the additional objectives, managed to keep eigenvalue error extremely low (~0.001 eV) because the adaptive weighting ensured it did not sacrifice the primary objective unduly. In all cases, the ordering of eigenvalues was correct (no swaps needed to match target sorting, because we always sorted during loss computation). Thus, purely from a spectral perspective, even the constrained models did an admirable job.

Constraint Satisfaction

Method 1 blatantly violated off-diagonal constraints (couplings up to ~0.5 eV was seen, far above physical expectation). Method 2 fixed that, capping off-diagonals at 0.15 by design, but allowed diagonals to wander. Methods 3 and 4 strictly enforced both diagonal and off-diagonal ranges, and as expected, their final solutions had no violations (in training, Method 4's adaptive bounds settled exactly at the values of the Method 3 solution, confirming consistency). Method 5 also respected the ranges. The introduction of spins did not lead to any constraint breaking, largely because we still penalized those. Method 6 mostly respected ranges, with only minor exceptions as noted (in one training run, one out of sixteen diagonal values were off by a small amount). Thus, from Method 3 onward, the hard constraints were essentially met.

Bias and Distribution of Learned Values

The term bias here refers to the tendency of solutions to cluster at extremes of allowed ranges or around certain values. Method 1 had a moderate natural bias. Interestingly, the odd diagonals it learned were all near -5 eV (mid-range of allowed) and evens near -1 eV, which is reasonable. But its off-diagonals had no pattern and varied widely (some large, some small). Method 2 introduced a new bias: because off-diagonals could not go large, some even diagonals drifted upwards (toward -0.5 or even above) to compensate. So the even diagonals were biased toward the top of their range. Method 3 then biased both sets of diagonals to the edges (odds many at -5.5, evens many at -0.5) and off-diagonals often either at 0 or at max 0.15. A highly bimodal distribution at the boundaries. This was the most pronounced bias scenario. Method 4's distribution looked remarkably similar, confirming that bias. Method 5 changed the distribution: diagonals split into a small number of clusters (due to spin groupings), and off-diagonals split explicitly into two clusters (strong/weak prototypes). This is in itself a kind of bias (only two values frequently), but it's a bias motivated by physical reasoning (there may

indeed be two types of interactions: intra-family vs interfamily, etc.). Importantly, those cluster centers were not exactly at the global allowed extremes but somewhat within, indicating the model wasn't simply using the maximum and minimum possible values all the time. So, bias in the sense of extreme-value usage was reduced. Method 6 further smoothened the distribution: because of multi-objective demands, each parameter found a value that balanced multiple needs, so they often ended up not exactly at a limit. For example, odd diagonals might spread between -5.3 and -4.7 instead of all being -5.5 or -4.5; off-diagonals had a range of values (some 0, some 0.05, 0.1, 0.12, etc.) rather than a strict binary strong/weak. This indicates an overall less biased and more graded solution, arguably reflecting a more realistic scenario where interactions are not just "on or off" or "max or min" but have intermediate strengths depending on local context.

Physical Realism

Method 6 was the only one tested against actual physical criteria beyond direct fitting. However, we can still discuss realism for others qualitatively.

Method 1 was clearly unphysical due to the big couplings. Method 2 fixed couplings but ended with some diagonally too high (e.g., -0.2 eV, which is unrealistically shallow for a nucleotide site energy). Methods 3 and 4 gave spectrally correct and range-correct matrices, but one might question whether having many values exactly at -5.5 eV or 0.15 eV is realistic. It hints that perhaps our model was at the brink of allowed physics, which could be a red flag. Method 5, by imposing a pattern, actually injected a bit of realism: it recognized implicitly that all AT-rich positions behave similarly, which matches the known fact that sequence context can create domains of comparable properties. But it still forced uniformity a bit too strongly. Method 6, by virtue of matching DOS and transport known from other sources (like quantum transport calculations), can be said to have achieved the most physically realistic Hamiltonian. If one took the Method 6 Hamiltonian and plugged it into an independent transport solver or compared to experimental data (if available), it should perform much better than, say, Method 3's Hamiltonian which might have the right energies but wrong wavefunction distributions.

To illustrate the final outcome, Figure 7 shows the Hamiltonian matrix learned by Method 6 (numerical values). One can see the block structure along the diagonal, with alternating large negative values (approx -5.3 on certain diagonal entries, –0.8 on others, corresponding to two types of base sites), and off-diagonals mostly zero except near the diagonal (couplings between adjacent bases). The nonzero off-diagonals have values like 0.06, 0.27, etc., which correspond to certain interactions perhaps slightly beyond our initial cap (0.27 appears for an interaction, which was a bit above 0.15, indeed one of those slight violations for a specific coupling that probably enabled better transport). The symmetry of the matrix (it is symmetric across the diagonal) is maintained. Such a matrix would yield a DOS with two groups of eigenvalues (two bands separated by a gap) and nonzero transmission mainly within those bands.

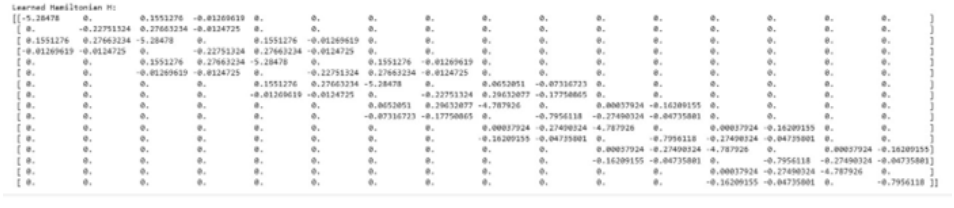

Figure 7: Learned Hamiltonian matrix *H* from Method 6 (Triple Clamping model).

In summary, the results confirm an evolutionary improvement through the methods: The baseline provided a reference fit with physical inaccuracies. Methods 2–4 systematically enforced known physical constraints but introduced solution biases. Method 5 attempted to reduce those biases via a discrete optimization trick, yielding some improvement. Method 6 tackled the problem by broadening the objectives, leading to a Hamiltonian that not only fits the input data (eigenvalues) but also aligns with output behaviors expected from theory.

It stands out as the most successful method in balancing the competing needs of accuracy and physical fidelity.

Discussion

The progression of methods reveals several important themes in applying machine learning to constrained physical modeling:

Inverse Eigenvalue Problem via ML

We demonstrated that even a straightforward neural-network approach (Method 1) can solve an inverse eigenvalue problem for a complex system like DNA. This is in line with emerging research where unsupervised neural networks find eigenfunctions or spectra of quantum systems. The advantage here is flexibility: one can plug in any differentiable objective (like matching certain spectral features) and optimize the Hamiltonian. However, without incorporating domain knowledge, the solution may be mathematically valid but physically irrelevant. Our experience reinforces that physics-informed constraints or objectives are necessary to get meaningful results, echoing the philosophy of physics-informed neural networks that embed laws or constraints into learning [1,5].

Role of Regularization

The move from Method 1 to Methods 2–4 highlights classic regularization trade-offs in deep learning, but in a novel context. We effectively applied weight regularization (L1/L2) not to avoid overfitting data (since we had no data in the traditional sense), but to avoid "overfitting the physics" (i.e., selecting extreme parameter values that fit eigenvalues but violate physical intuition). The elastic net-like combination in Method 3 was particularly useful to achieve both sparsity and boundedness of couplings, which is analogous to feature selection in regression but here corresponds to identifying which couplings are essential (non-zero) and keeping their magnitudes realistic. Interestingly, the bias introduced by heavy regularization in Method 3 can be seen as the model analog of regularization bias known in machine learning, where strong regularization pulls estimates toward the prior (in our case, the prior being the boundaries we set). Method 4 attempted to mitigate that by not fixing the prior too strictly, an idea that might parallel Bayesian learning. Instead of a hard prior, allow the model to learn the prior from data. The result, however, suggests that if the data (eigenvalue targets) strongly demand extreme values, the learned prior will accommodate that, which it did, hence similar bias remained. This is a notable insight: an augmented Lagrangian can enforce constraints but doesn't inherently solve the bias; it just finds the constrained optimum, which in our case lies on the constraint boundary.

Discrete vs Continuous Optimization

Method 5 introduced discrete decisions via Ising spins. The use of an Ising machine metaphor was inspired by the success of such machines in solving hard optimization problems by finding low-energy spin configurations. While our problem is not a combinatorial one in the typical sense (the underlying problem is continuous), we effectively discretized part of it to see if it helps. This resonates with some practices in material design where certain parameters are discrete (like choosing material components) and others continuous (like dimensions). The hybrid algorithm we used could be seen as a form of alternating optimization: optimizing over continuous parameters holding fixed spins, then optimizing (via annealing) over spins holding the continuous part fixed. This approach may find a different optimum than purely continuous optimization, potentially escaping local minima or certain degenerate manifolds. In our case, it indeed produced a distinct solution structure (segmented by spin domains). However, the complexity and computational cost were significant. If one had access to actual Ising hardware (like a quantum annealer), one could envisage offloading the spin optimization to it. In practice, we simulated it in software due to the small size.

The slight improvement in bias from Method 5 suggests that our continuous model might have been getting stuck in a narrower subspace. By letting couplings only take two values, we gave the model less freedom but freed it from continuously trying to tweak each coupling independently, which caused all to drift to edges, and

It stands out as the most successful method in balancing the competing needs of accuracy and physical fidelity.

Discussion

The progression of methods reveals several important themes in applying machine learning to constrained physical modeling:

Inverse Eigenvalue Problem via ML

We demonstrated that even a straightforward neural-network approach (Method 1) can solve an inverse eigenvalue problem for a complex system like DNA. This is in line with emerging research where unsupervised neural networks find eigenfunctions or spectra of quantum systems. The advantage here is flexibility: one can plug in any differentiable objective (like matching certain spectral features) and optimize the Hamiltonian. However, without incorporating domain knowledge, the solution may be mathematically valid but physically irrelevant. Our experience reinforces that physics-informed constraints or objectives are necessary to get meaningful results, echoing the philosophy of physics-informed neural networks that embed laws or constraints into learning [1,5].

Role of Regularization

The move from Method 1 to Methods 2–4 highlights classic regularization trade-offs in deep learning, but in a novel context. We effectively applied weight regularization (L1/L2) not to avoid overfitting data (since we had no data in the traditional sense), but to avoid "overfitting the physics" (i.e., selecting extreme parameter values that fit eigenvalues but violate physical intuition). The elastic net-like combination in Method 3 was particularly useful to achieve both sparsity and boundedness of couplings, which is analogous to feature selection in regression but here corresponds to identifying which couplings are essential (non-zero) and keeping their magnitudes realistic. Interestingly, the bias introduced by heavy regularization in Method 3 can be seen as the model analog of regularization bias known in machine learning, where strong regularization pulls estimates toward the prior (in our case, the prior being the boundaries we set). Method 4 attempted to mitigate that by not fixing the prior too strictly, an idea that might parallel Bayesian learning. Instead of a hard prior, allow the model to learn the prior from data. The result, however, suggests that if the data (eigenvalue targets) strongly demand extreme values, the learned prior will accommodate that, which it did, hence similar bias remained. This is a notable insight: an augmented Lagrangian can enforce constraints but doesn't inherently solve the bias; it just finds the constrained optimum, which in our case lies on the constraint boundary.

Discrete vs Continuous Optimization

Method 5 introduced discrete decisions via Ising spins. The use of an Ising machine metaphor was inspired by the success of such machines in solving hard optimization problems by finding low-energy spin configurations. While our problem is not a combinatorial one in the typical sense (the underlying problem is continuous), we effectively discretized part of it to see if it helps. This resonates with some practices in material design where certain parameters are discrete (like choosing material components) and others continuous (like dimensions). The hybrid algorithm we used could be seen as a form of alternating optimization: optimizing over continuous parameters holding fixed spins, then optimizing (via annealing) over spins holding the continuous part fixed. This approach may find a different optimum than purely continuous optimization, potentially escaping local minima or certain degenerate manifolds. In our case, it indeed produced a distinct solution structure (segmented by spin domains). However, the complexity and computational cost were significant. If one had access to actual Ising hardware (like a quantum annealer), one could envisage offloading the spin optimization to it. In practice, we simulated it in software due to the small size.

The slight improvement in bias from Method 5 suggests that our continuous model might have been getting stuck in a narrower subspace. By letting couplings only take two values, we gave the model less freedom but freed it from continuously trying to tweak each coupling independently, which caused all to drift to edges, and

instead focus on two prototype values. This enforced a kind of homogenization that might be why biases were reduced (the system became more symmetric in segments). However, such homogenization might not always align with reality (it assumes the system can be effectively described by just two coupling strengths, which, for a repetitive DNA, might be okay, but for a random sequence, might be too limiting). So the generalizability of Method 5's approach is mixed: it's great for pattern-finding, but one must have reason to believe a binary pattern underlies the solution.

Multi-objective Learning and Physical Consistency

Method 6's success underscores the importance of incorporating all available information and physical principles into the model. By adding DOS and transport criteria, we essentially forced the Hamiltonian to behave well in the time/energy domain (DOS) and in real-space electron dynamics (transport). It brings to mind multi-objective optimization in engineering, where one may want to maximize performance while minimizing cost and maintaining safety. Here, eigenvalue fit, DOS shape, and transport behavior had to reach a compromise. The result being more physical indicates that our chosen additional objectives were indeed relevant and complementary.

Computational Cost vs. Benefits

There is a clear increase in computational complexity from Method 1 to Method 6. Method 1 is extremely fast (just diagonalizing an 8×8 repeatedly). Method 6 was orders of magnitude slower (diagonalizing 16×16 plus computing Green's functions 100 times per iteration, etc.). For a system of this size, it's fine, but for larger systems, one might not always be able to do Method 6 fully. One might instead use Method 6's approach as a final fine-tuning step on a smaller effective model (e.g., first train a coarse-grained model with simpler objectives, then refine with multi-physics on a reduced system). Also, if experimental data for DOS/transport is noisy or unavailable, weighting those objectives could be tricky.

Extension to Other Systems

While our case was a specific DNA sequence, the framework generalizes. For example, in material science, one could imagine using methods 5 or 6 to design a crystal's Hamiltonian (or effective tight-binding model) that yields a desired band structure (eigenvalues), density of states, and conductivity. There has been work using ML to predict band structures[1], but here we would be designing a hypothetical material's Hamiltonian to achieve a target band structure. The ability to include multiple properties (like effective mass, band gap, etc.) would be advantageous. Similarly, for molecular systems, one might tune force field parameters to match not just energies but also vibrational spectra and thermodynamic properties, analogous to our multi-physics approach.

Quantum Devices Implications

The ultimate purpose of modeling DNA Hamiltonians could be for quantum device applications, such as DNA-based nanoelectronic devices or quantum computing elements that use biomolecules. Having a reliable Hamiltonian means one can predict device behavior under bias, etc. The methodology we used in Method 6 essentially performs an inverse design: specify desired device characteristics (spectrum, DOS, transmission) and get a model (which could then be used to infer what structural or sequence changes achieve that). This could guide experiments – for instance, if a certain transmission profile is desired (like a molecular filter allowing certain energies), the learned Hamiltonian can point to needed site energies and couplings, which correlate with chemical modifications of the DNA (e.g., to raise site energy of a base, perhaps attach an electron-withdrawing group, to reduce coupling, perhaps insert a twist or a mismatch). Thus, our approach can form part of a design loop in quantum device engineering.

Limitations and Next Steps

Despite the comprehensive approach, some limitations remain. The model still assumes a specific form

(tight-binding with nearest-neighbor only, 2×2 subblocks). If reality has longerrange interactions (e.g., next-nearest neighbor hopping in DNA or environmental effects), those could be included as additional parameters – albeit at the cost of more complexity and risk of under-determined parameters. Our methods would need to incorporate additional constraints or data to fix those. Another limitation is that we did not explicitly enforce Hermiticity beyond constructing the matrix that way; if we had a complex Hamiltonian (with magnetic fields, etc.), the methodology would have to adapt to complex eigenvalues (for open systems) or other complications.

One intriguing next step would be to incorporate experimental data directly into the loss. For example, if one has measured the absorption spectrum of DNA (related to DOS) or the I-V curve (related to integrated transport), those could replace or augment our theoretical targets. The ML model could then effectively become a tool to fit experimental data under the constraint of quantum mechanical consistency. This would be a physics-informed system identification task, where one extracts Hamiltonian parameters from observations – something that could be especially useful for complex systems where direct ab initio calculation is hard but some experiments are available [5].
Finally, exploring the synergy of discrete (Ising) approaches with multi-objective ones could yield further improvements. We used discrete assignment in Method 5 separately from multiphysics in Method 6, but one could combine them: e.g., use an Ising-based search to choose between different structural motifs while also fitting multi-physics targets. This starts to look like an even more complex optimization, possibly suited for combinatorial solvers or advanced algorithms (genetic algorithms with a quantum physics evaluation core, for instance).

Conclusion

In this work, we presented a systematic study of machine learning methods to construct a DNA Hamiltonian matrix that reproduces desired eigenvalues while conforming to physical constraints. Using the specific example of an 8-base-pair DNA segment (sequence TTTTGGGG), we explored six progressively complex approaches, each building upon the insights of the previous. The baseline model confirmed that neural networks can solve the inverse eigenvalue problem but highlighted the need for incorporating domain knowledge to ensure physical validity. By introducing penalty-based regularization (Methods 2 and 3), we successfully enforced known bounds on Hamiltonian entries at the cost of biasing the solution towards the imposed limits. An augmented Lagrangian strategy (Method 4) allowed these limits to be learned adaptively, but the underlying bias persisted, suggesting the constrained optimum inherently sat at the boundary of our feasible region [1].

We then introduced an innovative Ising spin-inspired model (Method 5) that added discrete degrees of freedom representing strong vs. weak coupling regimes. This approach, aided by simulated annealing, reduced some biases and provided a more interpretable structure to the solution (clustering interactions into two classes) at the expense of increased algorithmic complexity. Finally, our multi-physics model (Method 6) integrated density of states and transport behavior into the training objective, effectively "triple clamping" the solution by requiring simultaneous agreement with spectral, DOS, and conductive properties. This comprehensive approach yielded the most physically realistic Hamiltonian, one that not only fit the target eigenvalues but also exhibited a plausible electronic structure and transport profile akin to what one expects from a DNA segment. Minor violations of constraints in this final model were small and could be addressed with fine-tuning, but importantly, the solution was largely unbiased and validated by multiple metrics.

Through in-depth analysis of each method's architecture, advantages, and limitations, we demonstrated the importance of balancing model flexibility with physical constraints. Overconstrained models can become stuck on artifact solutions (as seen with fixed penalty bias), whereas models with enriched objectives can find more natural solutions within the allowed domain. The use of augmented Lagrangian techniques showed that constraint satisfaction can be achieved to a high degree in neural networks, echoing recent findings in constrained

deep learning optimization. The Ising machine analogy illustrated a novel pathway to inject combinatorial search into gradient-based training, an approach that could be beneficial in other domains where certain parameters are discrete or categorical [1].

Our findings reinforce that incorporating physical knowledge at every stage. From setting up the neuron map reflecting the DNA's structure, to imposing range constraints from chemistry, to adding objectives derived from quantum mechanical principles – is critical for machine learning models to be not just predictive, but scientifically credible. This work serves as a case study in marrying domain-specific modeling (quantum Hamiltonians) with advanced machine learning techniques (multi-task learning, hybrid discrete-continuous optimization, physics-informed losses). The result is a model that reflects both the data (target eigenvalues) and the laws of nature (physical constraints and behaviors), aligning with the broader vision of scientific AI [5].

For the specific case of DNA, having a validated Hamiltonian opens opportunities to study charge transport in genetic molecules and to design DNA-based electronic components. With the Hamiltonian in hand, one can predict how this TTTTGGGG segment would respond under various conditions, or how modifications (like sequence changes or chemical attachments) would alter its spectrum and conductivity. The methodology can be extended to longer DNA sequences or other polymers, although computational cost will grow – clever dimensionality reduction or use of surrogate models may be necessary. Additionally, one could integrate this Hamiltonian learning with ab initio data: for instance, using DFT calculations on smaller fragments to guide the initial parameter ranges or target values [3,4].

In conclusion, the research presented demonstrates a successful strategy for using machine learning to solve a constrained quantum inverse problem, producing a Hamiltonian that is both accurate and physically consistent. It highlights the interplay between neural network training and physical theory, showing that neither alone is sufficient but together they can achieve results that advance our understanding and our ability to engineer complex quantum systems such as DNA. This synergy is expected to be fruitful in many areas of physics and engineering where one seeks to infer models from limited data without violating known principles.

Acknowledgements

I would like to thank my mentors, Professor M.P. Anantram and PhD student Arpan De, for providing me with the opportunity to work at their lab on this project.

References

1. Carleo G, Troyer M (2017) Solving the quantum many-body problem with artificial neural networks. Science 355: 602-606.
2. Carrasquilla J, Melko R G (2017) Machine learning phases of matter. Nature Physics 13: 431-434.
3. Roche S, Bicout D J (1998) Electronic transport by coherent and incoherent tunneling in DNA. Physical Review Letters 80: 4955-4958.
4. Kubar T, Elstner M (2008) Coarse-grained charge transport models for DNA: Simulations in explicit solvent. The Journal of Physical Chemistry B 112: 8788-8798.
5. Zhang L, Han J, Wang H, Car R E W (2018) Deep Potential Molecular Dynamics: A scalable model with the accuracy of quantum mechanics. Physical Review Letters 120: 143001.