Structured and Interpretable Learning via Diophantine-Elliptic Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

We introduce Diophantine-Elliptic Curve Neural Networks (DEC-NNs), a novel class of architectures in which parameters are not unconstrained real numbers but integer-valued solutions to a fixed elliptic Diophantine equation. This constraint embeds each weight and bias into an algebraically structured arithmetic variety, yielding neural models that are interpretable, sparse, and geometrically robust by design. Our formulation enforces this structure through a projection-based training loop, ensuring consistency across updates without sacrificing predictive performance. We establish theoretical guarantees on convergence, symbolic expressivity, and generalization bounds rooted in number theory. Empirically, DEC-NNs demonstrate high accuracy and resilience under adversarial noise on both synthetic and real-world datasets including MNIST and UCI Breast Cancer. In domains such as scientific modeling, symbolic regression, and medical diagnostics, where transparency and auditability are essential, DEC-NNs offer a principled alternative to conventional networks, aligning learning with discrete symbolic structure rather than post hoc interpretability.

1 Introduction

Neural networks are widely used in high-dimensional function approximation, yet remain difficult to interpret, verify, or constrain (Rudin, 2019; Lipton, 2018). In safety-critical systems, such as medical diagnosis tools, autonomous platforms, and algorithmic governance, these limitations introduce unacceptable risks (Amodei et al., 2016; Doshi-Velez & Kim, 2017). Existing approaches to interpretability and robustness rely heavily on empirical heuristics. Regularization techniques, weight pruning, dropout, and post hoc explanations offer no formal guarantees (Srivastava et al., 2014; Han et al., 2016; Lundberg & Lee, 2017). This has created a gap between model performance and model accountability (Caruana et al., 2015).

A structural limitation of current architectures is their reliance on unconstrained real-valued parameters. Weight spaces are typically modeled as high-dimensional Euclidean vectors, updated via gradient descent (LeCun et al., 2015). These parameterizations are not inherently interpretable, nor do they support symbolic verification. Some recent work has explored constrained or quantized networks, but most approaches lack algebraic structure or theoretical guarantees of consistency under perturbation (Hubara et al., 2018; Courbariaux et al., 2015).

We introduce a neural architecture in which all model parameters are constrained to integer-valued solutions of a fixed nonlinear algebraic relation. Specifically, each parameter pair is restricted to lie on an elliptic curve over the integers (Silverman, 2009). The constraint is imposed directly during training through a projection mechanism. After each update, the modified parameter is projected onto the nearest integer point on a predefined elliptic curve (Amos & Kolter, 2017). This ensures that all weights and biases remain valid under the constraint at every stage of training.

This architecture yields three properties. First, the constraint enforces a discrete structure on the parameter space, which can be symbolically verified (Hauser et al., 2022). Second, the projection reduces model capacity in a mathematically controlled manner, functioning as a natural form of regularization (Massaroli et al., 2020). Third, perturbation resistance improves, since adversarial changes to parameters must preserve a nonlinear curve constraint to remain valid (Madry et al., 2017).

We refer to this class of models as Diophantine-elliptic curve neural networks (DEC-NNs). The formulation avoids loose approximations and does not depend on gradient masking or empirical tuning. Instead, it introduces a principled connection between number theory and machine learning design (Micciancio & Goldwasser, 2002). Each parameter can be traced, verified, and reconstructed from its algebraic encoding.

To support efficient training under this constraint, we develop a projection-based optimization scheme. The method balances symbolic consistency with numerical stability. At each iteration, parameters are updated using unconstrained gradient descent, followed by discrete projection to a valid elliptic curve point (Amos & Kolter, 2017). This mechanism enforces hard constraint satisfaction without altering the training objective.

We test this architecture on structured datasets with ground-truth interpretability signals. Results show that DEC-NNs retain competitive accuracy, while improving robustness to parameter tampering and enabling symbolic verification. The elliptic constraint also enforces bounded parameter growth and improves sparsity in trained networks (England, 2024).

This work proposes an integer-valued algebraic encoding for neural parameters, enforced via projection onto elliptic curves during training. The approach introduces a verifiable and adversarially constrained model class for structured, interpretable learning in high-stakes environments.

Beyond its theoretical construction, the DEC-NN framework targets concrete domains where symbolic alignment is not a luxury but a necessity. In symbolic regression, models must recover explicit, human-readable equations from data, an inherently algebraic task for which DEC-NNs are well-suited, since every parameter is symbolically recoverable by design. In interpretable scientific modeling, where laws or relations are sparse and arithmetic, the elliptic structure anchors inference within traceable, verifiable hypotheses. In biomedical diagnosis or safety-critical prediction tasks, where robustness under constrained resources is essential, the discrete nature of DEC parameters filters adversarial drift and enforces global consistency. These are not abstract motivations. They reflect domains where practitioners need assurance that what the model has learned can be audited, explained, and traced back to the algebraic form of its parameters. Our architecture speaks directly to this need.

1.1 Related Work

1.1.1 Algebraic and Constrained Neural Networks

Several studies have investigated algebraic constraints as a way to improve the interpretability and tractability of neural networks. England (England, 2024) proposed neural architectures that embed symbolic constraints directly into model parameters, enabling integration with formal solvers. These models support verification within symbolic computation environments and demonstrate that algebraically structured parameters can lead to tractable optimization and semantic transparency.

Boesen et al. (Boesen et al., 2024) introduced neural differential-algebraic equation (DAE) models. Their work showed that algebraic constraints can be enforced via stabilization or projection, and that this enforcement improves model stability and generalization. They emphasized that projection methods maintain constraint satisfaction throughout training without relying on penalty terms.

Other recent work has explored the use of symbolic regression (Cranmer et al., 2020; Udrescu & Tegmark, 2020) and program synthesis to build models with symbolic interpretability. However, these approaches treat symbolic structure as a post-hoc objective or external prior. In contrast, our architecture encodes symbolic constraints directly into the parameter space, ensuring interpretability throughout training and inference.

1.1.2 Projection-Based Optimization

Projection-based training has emerged as a method to handle hard constraints in neural models. Wang et al. (Wang et al., 2021) used differentiable projection to enforce convex feasibility constraints in deep networks. This approach avoids soft penalties and provides exact constraint satisfaction at each iteration. Results show that such methods improve both accuracy and training efficiency.

Huang et al. (Huang et al., 2017) introduced a projection-based normalization scheme for weights. Their method improves the geometry of the weight space, enhancing stability and generalization. The use of projections aligns with our enforcement of hard elliptic constraints via nearest-point projection to valid integer solutions.

Outside of neural networks, projection onto discrete algebraic sets has been studied in integer programming and lattice decoding (Micciancio & Goldwasser, 2002). Our work brings such discrete projection principles into the training loop of neural networks, introducing a form of structured arithmetic regularization.

1.1.3 Elliptic Curves in Machine Learning

Elliptic curves are traditionally used in number theory and cryptography, but recent work has explored their role in data-driven modeling. Gualandi et al. (Gualandi et al., 2024) trained deep models to predict properties of elliptic curves, including torsion subgroups and algebraic ranks. While their approach uses machine learning to study elliptic curves, our work inverts the paradigm, i.e., we use elliptic curves as symbolic constraints to structure and verify neural networks.

There has also been interest in using algebraic varieties for generative modeling (Hauser et al., 2022), though the constraints in such settings are typically soft and geometric rather than discrete and symbolic. DEC-NNs are, to our knowledge, the first models to enforce hard Diophantine constraints via elliptic curves on all parameters in a standard feedforward neural architecture.

1.1.4 Limitations of Quantized and Modular Arithmetic Neural Networks

Quantized neural networks (QNNs) restrict weights to finite sets, often through bit-level rounding, integer encoding, or fixed-point representations (Hubara et al., 2018; Courbariaux et al., 2015). These methods improve efficiency and compression but do not impose any global structure on the weight space. Each parameter is discretized independently, and there is no algebraic consistency or symbolic interpretability across layers.

Modular arithmetic-based networks define operations over finite fields or rings, introducing wraparound effects that obscure functional semantics. These models lose parameter traceability and cannot recover symbolic expressions from learned weights due to discontinuous arithmetic behavior.

DEC-NNs differ fundamentally. Instead of discretizing parameters ad hoc, they constrain the entire parameter set to lie on a fixed arithmetic variety. This ensures that all weights satisfy a shared symbolic equation, which is preserved under training. Unlike QNNs or modular networks, DEC-NNs allow symbolic recovery, geometric alignment, and parameter interpretability by construction. The constraint is not a side effect of optimization or quantization, but a core architectural principle derived from elliptic Diophantine structure.

2 Preliminaries

This section establishes the mathematical foundations underlying the proposed Diophantine-Elliptic Constrained Neural Network (DEC-NN) architecture. The central idea is to encode neural parameters using integer-valued points constrained to lie on algebraic varieties, specifically, nonsingular elliptic curves defined over $\mathbb{Z}$ or $\mathbb{Q}$ (Silverman, 2009; Hauser et al., 2022). This algebraic parameterization introduces symbolic structure, interpretability, and robustness (England, 2024).

We begin by introducing formal definitions, followed by structural theorems and constraints that will underpin our learning framework.

2.1 Diophantine Geometry and Algebraic Encodings

Definition 1 (Diophantine Equation)**.** *A* Diophantine equation *is a multivariate polynomial equation with integer coefficients,*

$$P(x_1, x_2, \ldots, x_n) = 0,$$

where solutions are sought over $\mathbb{Z}^n$ or $\mathbb{Q}^n$. The set of solutions is denoted $\mathcal{D}_P = \{(x_1, \ldots, x_n) \in \mathbb{Z}^n : P(x_1, \ldots, x_n) = 0\}$ (Micciancio & Goldwasser, 2002).

In this work, neural network parameters $\theta = \{W, b\}$ are mapped to tuples constrained to satisfy a Diophantine equation, effectively embedding them into structured, discrete arithmetic varieties (Hauser et al., 2022).

Definition 2 (Symbolic Parameter Encoding)**.** *Let $\Phi_{enc} : \mathbb{R}^m \to \mathbb{Z}^n$ be an injective encoding map. A set of parameters $\theta \in \mathbb{R}^m$ is* Diophantine-constrained *if*

$$\Phi_{enc}(\theta) \in \mathcal{D}_P,$$

for a fixed Diophantine equation P (England, 2024).

Such an encoding imposes algebraic structure on the parameter space and allows symbolic traceability and constrained optimization (Hauser et al., 2022).

2.2 Elliptic Curves as Parameter Constraints

We now specialize to Diophantine equations defined by elliptic curves, offering richer structure and a canonical group law (Silverman, 2009).

Definition 3 (Elliptic Curve)**.** *A nonsingular elliptic curve over a field K (typically $\mathbb{Q}$ or $\mathbb{Z}_p$) is defined by the short Weierstrass equation:*

$$E_{a,b} : y^2 = x^3 + ax + b,$$

with $a, b \in K$, subject to the non-vanishing discriminant condition $\Delta = -16(4a^3 + 27b^2) \neq 0$.

Definition 4 (Elliptic Parameter Locus)**.** *The set of valid Diophantine-elliptic encodings is given by:*

$$\mathcal{D}_E = \left\{(x, y) \in \mathbb{Z}^2 : y^2 = x^3 + ax + b\right\}.$$

We interpret each parameter or block of parameters $\theta_i \in \mathbb{R}$ as associated with a unique point $(x_i, y_i) \in \mathcal{D}_E$ (England, 2024).

Theorem 1 (Group Law on Elliptic Curves)**.** *Let $E(\mathbb{Q})$ denote the set of rational points on an elliptic curve $E_{a,b}$. Then $(E(\mathbb{Q}), +)$ forms a finitely generated abelian group under the chord-tangent law:*

$$P + Q + R = \mathcal{O} \quad \textit{iff} \quad P, Q, R \textit{ are colinear in the affine plane.}$$

This group law allows parameter arithmetic to correspond to algebraic operations on $E(\mathbb{Q})$, equipping DEC-NN with structured composition and update rules (Silverman, 2009).

2.3 Elliptic Projection and Optimization

To enforce the constraint during training, we introduce a projection-based surrogate loss. After each unconstrained gradient step, we project the updated parameters onto $\mathcal{D}_E$ (Amos & Kolter, 2017).

Definition 5 (Projection to Elliptic Constraint Surface)**.** *Let $\theta_i \in \mathbb{R}$. The projection operator is defined as:*

$$\Pi_E(\theta_i) := \arg \min_{(x,y) \in \mathcal{D}_E} \|\theta_i - \phi_{dec}(x, y)\|^2,$$

where $\phi_{dec} : \mathbb{Z}^2 \to \mathbb{R}$ is the decoding function mapping elliptic points to real scalars.

Lemma 1 (Constraint Preservation under Projection)**.** *Let $\theta_i^{(t)}$ be the parameter at iteration t. Define $\theta_i^{(t+1)} := \phi_{dec} \circ \Pi_E(\theta_i^{(t)} - \eta \nabla_{\theta_i} \mathcal{L}_{task})$. Then:*

$$\phi_{enc}(\theta_i^{(t+1)}) \in \mathcal{D}_E,$$

ensuring constraint preservation across all updates (Amos & Kolter, 2017).

2.4 Arithmetic Capacity and Regularization via Hasse Bounds

By working over finite fields $\mathbb{F}_p$, we obtain intrinsic capacity control via the number of solutions (Massaroli et al., 2020).

Theorem 2 (Hasse's Theorem)**.** *Let E be an elliptic curve defined over $\mathbb{F}_p$. Then the number of $\mathbb{F}_p$-rational points satisfies:*

$$|\#E(\mathbb{F}_p) - (p+1)| \leq 2\sqrt{p}.$$

This result bounds the number of valid parameter encodings in DEC-NN when implemented over $\mathbb{F}_p$, enforcing a form of discrete capacity regularization (Massaroli et al., 2020).

2.5 Cryptographic Compatibility and Structural Auditability

The use of elliptic curves also admits integration with cryptographic protocols (Micciancio & Goldwasser, 2002).

Definition 6 (Elliptic Curve Discrete Logarithm Problem (ECDLP))**.** *Let $P, Q \in E(\mathbb{F}_p)$. The ECDLP is to find $k \in \mathbb{Z}$ such that $Q = kP$. This problem is assumed to be computationally hard for large prime-order subgroups.*

In DEC-NN, each parameter (x_i, y_i) can be interpreted as an elliptic point whose encoding can be verified or encrypted under ECDLP-hardness assumptions, enabling secure learning or auditability.

2.6 Encoding and Decoding Maps

We define two canonical maps for interfacing between continuous neural training and discrete algebraic representations.

- **(A1) Encoding:** $\Phi_{\text{enc}} : \mathbb{R} \to \mathcal{D}_E$, which maps a scalar θ_i to its nearest elliptic point (x_i, y_i) (England, 2024).
- **(A2) Decoding:** $\Phi_{\text{dec}} : \mathcal{D}_E \to \mathbb{R}$, which interpolates or reconstructs a usable real-valued parameter from an elliptic encoding (England, 2024).

These operations allow the model to train using real-valued updates while remaining constrained to algebraically structured parameter sets.

2.7 Total Learning Objective

Finally, the full loss functional used in training is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \sum_{i=1}^{n} \left(y_i^2 - x_i^3 - a x_i - b\right)^2 + \gamma \mathcal{L}_{\text{adv}},$$

where $(x_i, y_i) = \Phi_{\text{enc}}(\theta_i)$, and $\lambda, \gamma \geq 0$ control the enforcement of elliptic and robustness constraints (England, 2024).

Corollary 1 (Symbolic Regularization)**.** *If $\theta \in \Phi_{dec}(\mathcal{D}_E)$, then the parameter space is*

$$\mathcal{M}_E := \Phi_{dec}(\mathcal{D}_E) \subset \mathbb{R}^n,$$

which defines a low-dimensional, algebraically constrained arithmetic variety that regularizes overfitting and promotes interpretability (Hauser et al., 2022).

3 Architecture of Diophantine-Elliptic Neural Networks

We now construct the full architecture of Diophantine-Elliptic Neural Networks (DEC-NNs), a class of models whose parameters are constrained to lie on a fixed elliptic curve over the integers. This constraint introduces symbolic structure into every layer of the network, from initialization through training. The core mechanism is an encoding-decoding map that links real-valued weights to discrete algebraic solutions of the curve equation. The result is a neural architecture that operates on continuous data while evolving over a constrained arithmetic variety (c.f. Serre (1973)). What follows is a complete and self-contained description of this construction.

3.1 Model Definition and Encoding Strategy

A Diophantine-Elliptic Neural Network (DEC-NN) is defined as a standard feedforward neural network whose parameters are algebraically constrained to lie on a fixed elliptic curve over the integers. This constraint discretizes and geometrically structures the parameter space, encoding symbolic arithmetic directly into the network's architecture.

Let the base curve be given in short Weierstrass form

$$\mathcal{E}_{a,b} := \left\{(x, y) \in \mathbb{Z}^2 \,\middle|\, y^2 = x^3 + ax + b\right\}, \quad a, b \in \mathbb{Z}, \quad 4a^3 + 27b^2 \neq 0.$$

Each scalar parameter $\theta \in \mathbb{R}$ is encoded as a nearby point $(x, y) \in \mathcal{E}_{a,b}$, and the real value used during computation is obtained by a decoding map. This mechanism induces a nonlinear, number-theoretic prior on the model space.

Define the encoding map $\Phi_{\text{enc}} : \mathbb{R} \to \mathcal{E}_{a,b}$ as a projection of real parameters onto the discrete curve:

$$\Phi_{\text{enc}}(\theta) := \arg \min_{\substack{(x,y)\in\mathbb{Z}^2 \\ y^2=x^3+ax+b}} |\theta - x|.$$

The decoding map $\Phi_{\text{dec}} : \mathcal{E}_{a,b} \to \mathbb{R}$ assigns a real-valued proxy for computation. We take $\Phi_{\text{dec}}(x, y) = x$ unless a weighted decoding (e.g., $\alpha x + \beta y$) is specifically required.

Weighted Decoding and Expressive Flexibility

While the default decoding map is given by $\Phi_{\text{dec}}(x, y) = x$, certain functions may depend on both coordinates of the elliptic encoding. To capture such dependencies, we allow a weighted decoding of the form

$$\Phi_{\text{dec}}(x, y) = \alpha x + \beta y, \quad \alpha, \beta \in \mathbb{R}.$$

This generalization is essential when the target function has the structure $f(t) = c_1 x(t) + c_2 y(t)$, where $(x(t), y(t)) \in \mathcal{E}_{a,b}$ and both terms contribute meaningfully. For example, the decoder $\Phi_{\text{dec}}(x, y) = x$ cannot recover $f(t) = 2x(t) + 3y(t)$, whereas choosing $\alpha = 2$, $\beta = 3$ enables exact reconstruction. Weighted decoding thus enhances the expressive capacity of DEC-NNs while preserving their symbolic and geometric alignment.

3.2 Layer Structure and Parameter Constraint

A DEC-NN with $L \in \mathbb{N}$ layers is specified by weight matrices $W^{[\ell]} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and bias vectors $b^{[\ell]} \in \mathbb{R}^{d_\ell}$, with all entries decoded from elliptic curve points. Explicitly,

$$W_{ij}^{[\ell]} := \Phi_{\text{dec}}\big(\Phi_{\text{enc}}(\theta_{ij}^{[\ell]})\big), \qquad b_i^{[\ell]} := \Phi_{\text{dec}}\big(\Phi_{\text{enc}}(\theta_i^{[\ell]})\big),$$

with all intermediate θ's updated during training and projected onto $\mathcal{E}_{a,b}$ after each step.

Forward propagation proceeds as usual

$$z^{[\ell]} = W^{[\ell]} x^{[\ell-1]} + b^{[\ell]}, \qquad x^{[\ell]} = \varphi^{[\ell]}(z^{[\ell]}),$$

but the parameter space is now confined to a symbolic arithmetic variety, and optionally, the activation function $\varphi^{[\ell]}$ may be designed to preserve elliptic alignment (e.g., enforcing $(z, \varphi(z)) \in \mathcal{E}_{a,b}$).

3.2.1 Training Under Elliptic Constraint

To enforce curve consistency during training, we augment the task loss with a penalty

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{EC}}, \quad \lambda \geq 0,$$

where $\mathcal{L}_{\text{task}}$ is a conventional loss (e.g., cross-entropy or MSE), and

$$\mathcal{L}_{\text{EC}} := \sum_{(x,y)} \left(y^2 - x^3 - ax - b\right)^2,$$

summing over all encoded parameter pairs $(x, y) = \Phi_{\text{enc}}(\theta)$. This term penalizes deviation from the Diophantine arithmetic variety and regulates parameter updates toward symbolic feasibility.

Updates are performed in decoded space using standard optimizers, then projected

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_{\text{total}}, \quad (x, y) \leftarrow \Phi_{\text{enc}}(\theta), \quad \theta \leftarrow \Phi_{\text{dec}}(x, y).$$

This loop preserves elliptic alignment during training, effectively embedding gradient flow within a discrete algebraic arithmetic variety.

3.3 Initialization and Expressivity

Initialization proceeds by enumerating integer $x \in [-M, M]$, solving $y^2 = x^3 + ax + b$ over $\mathbb{Z}$, and selecting valid pairs $(x, y) \in \mathcal{E}_{a,b}$. The decoded x-values form the initial parameter set. This ensures all weights and biases begin on the constraint arithmetic variety, giving early structure and regularization.

The full network function is then given by

$$\mathcal{N}(x) = \varphi^{[L]} \circ \left(W^{[L]} \cdot \varphi^{[L-1]} \circ \cdots \circ \varphi^{[1]} \circ \left(W^{[1]}x + b^{[1]}\right) + b^{[L]}\right),$$

with each parameter implicitly satisfying $(x, y) \in \mathcal{E}_{a,b}$ via encoding-decoding.

Despite operating over a discrete, algebraically constrained space, DEC-NNs retain full expressivity through nonlinear composition and gradient-projected arithmetic variety traversal. This architecture yields models that are not only functionally rich, but symbolically traceable and structurally interpretable. Moreover, every layer carries an arithmetic signature that can be traced, validated, and understood through algebraic geometry.

3.3.1 Scalability Paragraph

Although DEC-NNs introduce an additional projection step during training, the computational overhead is modest and easily managed. The set of admissible integer solutions to the elliptic curve can be precomputed within a bounded range and stored as a lookup table, turning projection into a fast, deterministic operation. Since all encoded parameters are fixed at inference time, the model introduces no additional runtime cost after training. This makes the architecture suitable not only for symbolic or low-parameter applications, but also for larger-scale settings where traceability and stability are required without compromising performance.

3.4 Theoretical Foundations of DEC-NNs

Rather than a limitation, the elliptic constraint introduces structured inductive bias that enhances interpretability and regularization. The goal here is not to mimic conventional networks under new constraints, but to understand what kinds of functions can be expressed, learned, and interpreted when the very act of representation is symbolically anchored.

Let us consider the induced hypothesis class

$$\mathcal{H}_{\mathcal{E}_{a,b}} := \left\{\mathcal{N}_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L} \,\middle|\, \theta \in \Phi_{\text{dec}}(\mathcal{E}_{a,b}^n)\right\},$$

where θ ranges over decoded parameters whose encoded integer forms lie on the elliptic curve $\mathcal{E}_{a,b}$, and n is the total number of learnable parameters. This class is no longer dense in $C^0(K)$ for compact $K \subset \mathbb{R}^{d_0}$, but it admits a controlled approximation capacity within structured subsets, those admitting representations with bounded Diophantine distortion.

To make this precise, we define the Diophantine distortion of a parameter $\theta \in \mathbb{R}$ as

$$\delta(\theta) := \min_{(x,y)\in\mathcal{E}_{a,b}} |\theta - x|,$$

and extend it to the network by setting

$$\delta(\mathcal{N}_\theta) := \max_i \delta(\theta_i).$$

We then say that a function $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ admits a Diophantine-elliptic approximation at level ε if there exists $\mathcal{N}_\theta \in \mathcal{H}_{\mathcal{E}_{a,b}}$ such that

$$\sup_{x\in K} \|f(x) - \mathcal{N}_\theta(x)\| < \varepsilon \quad \text{and} \quad \delta(\mathcal{N}_\theta) < \delta_0,$$

for some fixed $\delta_0 > 0$ depending on the curve and encoding precision.

This redefines expressivity in symbolic rather than purely metric terms. A model trained under Diophantine constraints will not interpolate arbitrary data in the classical sense, but it will discover function classes that are algebraically aligned with the structure of $\mathcal{E}_{a,b}$, which often coincide with sparse, low-complexity representations under symbolic or arithmetic priors.

We can formalize this alignment via a symbolic consistency principle. Suppose the target function f admits an algebraic form involving rational coefficients or arithmetic recursion. Then there exists a DEC-NN architecture such that the decoded parameters trace back to symbolically meaningful points on $\mathcal{E}_{a,b}$, and the trained model reproduces the structure of f not only numerically but symbolically, in the sense that its intermediate layers correspond to symbolic compositions of input variables.

In this sense, DEC-NNs do not merely approximate functions. They interpolate between numeric expressivity and symbolic recoverability. The model's architecture becomes a compression surface between continuous function spaces and discrete algebraic representations.

The training process itself is shaped by this surface. Gradient descent is no longer traversing a smooth Euclidean space, but rather projecting onto a piecewise-discrete arithmetic variety embedded in $\mathbb{R}^n$. Each step must resolve a nearest-point problem of the form

$$(x', y') = \arg\min_{(x,y)\in\mathcal{E}_{a,b}} |\theta' - x|,$$

where θ' is the unconstrained update. This acts as a natural regularizer. If the loss surface is steep in directions orthogonal to the arithmetic variety, projection introduces implicit damping. If the loss surface is flat along the arithmetic variety, updates proceed uninterrupted. What emerges is a kind of curvature-aware learning dynamic, one that respects the symbolic topology of the parameter space.

We may view this as a form of arithmetic variety regularization without auxiliary penalty terms. The geometry is hard-coded. The network does not merely avoid overfitting by chance, it is prevented from entering chaotic parameter regimes by construction. Its parameter updates are filtered through number-theoretic feasibility, not just norm-based smoothness.

This rigidity is not a drawback. It is a structural bias toward symbolic parsimony. Functions learned by DEC-NNs tend to exhibit numerically stable, symbolically minimal behavior, even when trained on noisy or overparameterized data. In this sense, the elliptic constraint offers both a computational bottleneck and an epistemic advantage, that is, it limits what the network can express, but forces what it does express to be traceable, interpretable, and symbolically aligned.

3.5 Approximation and Symbolic Expressivity of DEC-NNs

The central question now becomes, *what kinds of functions can a DEC-NN represent or approximate, and how do Diophantine-elliptic constraints shape that expressivity?* Unlike standard networks, DEC-NNs cannot rely on arbitrary real-valued parameters. Yet this very limitation, algebraically structured, admits a form of controlled expressivity that we now make precise.

Theorem 3 (Modified Universal Approximation for DEC-NNs)**.** *Let $\mathcal{E}_{a,b} \subset \mathbb{Z}^2$ be a fixed elliptic curve satisfying $4a^3 + 27b^2 \neq 0$, and let $\Phi_{\text{dec}} : \mathcal{E}_{a,b} \to \mathbb{R}$ be the decoding function defined by $\Phi_{\text{dec}}(x, y) = x$. Let $\mathcal{F}_{\varepsilon,M}$ be the class of continuous functions $f : [0, 1]^d \to \mathbb{R}$ such that f can be approximated to within $\varepsilon > 0$ by a shallow ReLU network with weights bounded by $M > 0$.*

Then there exists a DEC-NN $\mathcal{N}_\theta$ with parameters decoded from $(x, y) \in \mathcal{E}_{a,b}$, and total parameter distortion $\delta(\mathcal{N}_\theta) < \delta_0(\varepsilon, M)$, such that

$$\sup_{x\in[0,1]^d} |f(x) - \mathcal{N}_\theta(x)| < \varepsilon.$$

Proof. Let $f \in \mathcal{F}_{\varepsilon,M}$ be approximated by a standard ReLU network $\mathcal{N}_{\text{std}}$ with parameters $\theta_i \in [-M, M]$. For each parameter θ_i, select $(x_i, y_i) \in \mathcal{E}_{a,b} \cap \mathbb{Z}^2$ such that $|x_i - \theta_i| < \delta_0$, where δ_0 is a distortion tolerance to be fixed below. Let the DEC-NN parameters be $\tilde{\theta}_i := \Phi_{\text{dec}}(x_i, y_i) = x_i$.

By continuity of the ReLU network in its parameters and compactness of input space, the uniform difference between $\mathcal{N}_{\text{std}}$ and $\mathcal{N}_\theta$ satisfies

$$\sup_{x\in[0,1]^d} |\mathcal{N}_{\text{std}}(x) - \mathcal{N}_\theta(x)| \leq C \cdot \delta_0,$$

for some constant C depending on the architecture and depth. Choosing $\delta_0 < \varepsilon/(2C)$ ensures that

$$|f(x) - \mathcal{N}_\theta(x)| \leq |f(x) - \mathcal{N}_{\text{std}}(x)| + |\mathcal{N}_{\text{std}}(x) - \mathcal{N}_\theta(x)| < \varepsilon,$$

uniformly over $x \in [0, 1]^d$. □

This result shows that expressivity is not lost under elliptic constraint, it is restructured. The space of functions approximable by DEC-NNs is dense within the subset of ReLU-approximable functions, modulo a controlled algebraic distortion. In contrast to quantized neural networks or modular arithmetic-based models, DEC-NNs allow each parameter to be independently constrained while still encoding structured symbolic information. They maintain approximation power without resorting to bit-level discretization or modulus-induced nonlinearity.

Theorem 4 (Symbolic Recoverability)**.** *Let $\mathcal{N}_\theta$ be a DEC-NN trained to approximate a symbolic function $f : \mathbb{R}^d \to \mathbb{R}$ whose Fourier or polynomial expansion admits rational coefficients. Then the set of encoded elliptic points $\{(x_i, y_i)\} \subset \mathcal{E}_{a,b}$ admits a one-to-one symbolic alignment with the rational structure of f, up to a permutation of intermediate layers.*

Moreover, this alignment is preserved under training if and only if the projection step

$$(x_i^{(t)}, y_i^{(t)}) := \arg\min_{(x,y)\in\mathcal{E}_{a,b}} |\theta_i^{(t)} - x|$$

is performed after each gradient update. That is, symbolic fidelity requires arithmetic variety projection to be enforced throughout training.

Proof. Suppose $f(x) = \sum_\alpha c_\alpha x^\alpha$ with $c_\alpha \in \mathbb{Q}$, and let $\mathcal{N}_\theta$ approximate f within ε. Since $\Phi_{\text{dec}}(x, y) = x \in \mathbb{Z}$, each weight or bias corresponds to an integer point lying on $\mathcal{E}_{a,b}$. These x-values, when appropriately scaled, align with the rational coefficients c_α, up to bounded perturbation. The one-to-one matching follows from the minimality of distortion in the encoding map.

If projection is not enforced at each step, updates may drift off-curve and symbolic alignment is lost. Enforcing projection ensures each parameter always corresponds to a valid Diophantine pair and thus maintains traceability. □

This gives DEC-NNs a unique feature. That is, their learned functions are not only numerically accurate, but algebraically interpretable, each parameter can be symbolically decoded and cross-referenced with the function's structure. By contrast, modular networks (e.g., with weights modulo a prime p) lose alignment between parameters and function structure due to wraparound ambiguity, and QNNs encode functions through unitaries that are not parameter-traceable in symbolic terms.

Theorem 5 (Arithmetic variety-Constrained Generalization Bound)**.** *Let $\mathcal{N}_\theta \in \mathcal{H}_{\mathcal{E}_{a,b}}$ be a neural network with all weights and biases decoded from points $(x, y) \in \mathcal{E}_{a,b} \cap [-B, B]^2$, and trained on a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset [0, 1]^d \times \mathbb{R}$ using a loss function $\mathcal{L}$ that is L-Lipschitz in its second argument. Then with probability at least $1 - \delta$, the generalization error satisfies*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{true}}} [\mathcal{L}(y, \mathcal{N}_\theta(x))] \leq \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathcal{N}_\theta(x_i)) + \mathcal{O}\left(\frac{L\sqrt{\log |\mathcal{H}_{\mathcal{E}_{a,b}}(B)|}}{\sqrt{n}} \right) + \sqrt{\frac{\log(1/\delta)}{2n}},$$

where $\mathcal{H}_{\mathcal{E}_{a,b}}(B)$ denotes the class of networks whose parameters are constrained to $(x, y) \in \mathcal{E}_{a,b} \cap [-B, B]^2$.

Proof. Let $\mathcal{H}_{\mathcal{E}_{a,b}}(B)$ denote the class of all networks $\mathcal{N}_\theta$ where each parameter is obtained as $\Phi_{\text{dec}}(x, y)$ for some $(x, y) \in \mathcal{E}_{a,b} \cap [-B, B]^2$. Since the elliptic curve is defined over $\mathbb{Z}$, the set $\mathcal{E}_{a,b} \cap [-B, B]^2$ is finite, and so the set of possible decoded networks is also finite. Let $M := |\mathcal{H}_{\mathcal{E}_{a,b}}(B)|$.

For a finite hypothesis class of cardinality M, standard PAC-Bayes or covering number generalization bounds apply. In particular, from classical uniform convergence bounds for finite classes (see e.g., Theorem 3.3 of Shalev-Shwartz and Ben-David, Understanding Machine Learning Shalev-Shwartz & Ben-David (2014)), we have, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of the sample $\mathcal{D}$,

$$\sup_{\mathcal{N}_\theta \in \mathcal{H}_{\mathcal{E}_{a,b}}(B)} \left| \mathbb{E} [\mathcal{L}(y, \mathcal{N}_\theta(x))] - \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathcal{N}_\theta(x_i)) \right| \leq \sqrt{\frac{\log(2M) + \log(1/\delta)}{2n}}.$$

Now, since $\mathcal{L}$ is L-Lipschitz in its second argument and each network $\mathcal{N}_\theta$ is ρ-Lipschitz in x, it follows that the composed loss $(x, y) \mapsto \mathcal{L}(y, \mathcal{N}_\theta(x))$ is also Lipschitz in x, but the dominant dependency here is in the hypothesis complexity.

If each network has p scalar parameters (weights and biases), and each such parameter is decoded from a point $(x, y) \in \mathcal{E}_{a,b} \cap [-B, B]^2$, then the total number of such networks is bounded by

$$|\mathcal{H}_{\mathcal{E}_{a,b}}(B)| \leq \left| \mathcal{E}_{a,b} \cap [-B, B]^2 \right|^p.$$

Hence, taking logarithms

$$\log M \leq p \cdot \log \left| \mathcal{E}_{a,b} \cap [-B, B]^2 \right|.$$

Substituting back, we obtain the bound

$$\mathbb{E}_{(x,y)} [\mathcal{L}(y, \mathcal{N}_\theta(x))] \leq \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathcal{N}_\theta(x_i)) + \mathcal{O}\left(\frac{L\sqrt{p \cdot \log |\mathcal{E}_{a,b} \cap [-B, B]^2|}}{\sqrt{n}} \right) + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

Noting that p is fixed for a given architecture and absorbed in the constant, which yields our claim. □

3.6 Symbolic Approximation and Expressivity of DEC-NNs

The core advantage of Diophantine-Elliptic Neural Networks lies not in universal approximation in the classical sense, but in their ability to approximate functions whose spectral structure admits a symbolic, interpretable form. This shifts the emphasis from brute-force representation capacity to algebraically regular, symbol-constrained functional expressivity.

Let $\mathcal{F}_{\mathbb{Q},\Lambda}$ denote the class of L^2 functions $f : [0, 1]^d \to \mathbb{R}$ whose Fourier expansion

$$f(x) = \sum_{k \in \Lambda} \hat{f}_k e^{2\pi i \langle k, x \rangle}$$

has rational coefficients $\hat{f}_k \in \mathbb{Q}$ and support $\Lambda \subset \mathbb{Z}^d$ bounded in ℓ_∞ norm.

Theorem 6 (Symbolic Spectral Approximation)**.** *Let $\mathcal{E}_{a,b}$ be a fixed elliptic curve over $\mathbb{Z}$ with nonsingular structure, and let $\mathcal{N}_\mathcal{E}$ be the class of DEC-NNs whose parameters are decoded from points on $\mathcal{E}_{a,b}$. Then for every $f \in \mathcal{F}_{\mathbb{Q},\Lambda}$ and every $\varepsilon > 0$, there exists a DEC-NN $\mathcal{N}_\mathcal{E}$ with elliptically-constrained weights and biases such that*

$$\|\mathcal{N}_\mathcal{E}(x) - f(x)\|_{L^2([0,1]^d)} < \varepsilon,$$

with the additional property that the symbolic spectrum of $\mathcal{N}_\mathcal{E}$ (i.e., the set of integer frequencies induced by its layer-wise composition) is contained in Λ and the rationality of coefficients is preserved via encoding from $\mathcal{E}_{a,b}$.

Proof. We construct a network layer-wise, using the rational coefficients of f's Fourier expansion to determine affine combinations of sinusoidal primitives. For each $k \in \Lambda$, define $\varphi_k(x) = \cos(2\pi\langle k, x\rangle)$ or $\sin(2\pi\langle k, x\rangle)$, each of which can be approximated by shallow ReLU or sinusoidal layers using known neural synthesis theorems. We select rational weights $\hat{f}_k \in \mathbb{Q}$ and map them via the elliptic encoding Φ_{enc} to integer pairs $(x_k, y_k) \in \mathcal{E}_{a,b}$ such that $\Phi_{\text{dec}}(x_k, y_k) = \hat{f}_k$. Since $\mathcal{E}_{a,b} \cap \mathbb{Z}^2$ is infinite, the decoding map can approximate any rational weight exactly or to arbitrary precision under bounded denominator growth. The layer composition preserves symbolic alignment since both the spectrum and coefficients remain interpretable, and total error is controlled additively by classical trigonometric approximation bounds. □

This result establishes that DEC-NNs can approximate a meaningful subclass of L^2 functions, those with bounded symbolic spectrum and rational coefficients, to arbitrary precision while preserving a traceable encoding into arithmetic geometry. Unlike QNNs, where interpretability is often entangled with gate-level constraints that lack symbolic alignment, DEC-NNs impose discrete algebraic structure that supports both functional approximation and symbolic transparency.

Rather than aiming for universality over all bounded measurable functions, the DEC-NN framework targets a sharper class, i.e., functions that not only admit accurate approximation but also encode interpretable symbolic structure into their very parameterization. This makes them suitable for applications where traceability, algebraic consistency, and expressive clarity are essential, especially in scientific modeling, structured prediction, or symbolic regression tasks where modular neural methods fall short.

3.7 Main Theoretical Result

Theorem 7 (Symbolic Structure and Stability of Diophantine-Elliptic Neural Networks)**.** *Let $\mathcal{N}_\theta$ be a neural network with parameters $\theta \in \mathbb{R}^n$ encoded via $\Phi_{\text{enc}}(\theta) = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{Z}^2$ on a fixed elliptic curve $\mathcal{E}_{a,b}$ defined by $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{Z}$ and $4a^3 + 27b^2 \neq 0$. Then the following hold;*

- *(a) For any parameter vector $\theta \in \mathbb{R}^n$ and any $\varepsilon > 0$, there exists a set of encodings $(x_i, y_i) \in \mathbb{Z}^2$ with $(x_i, y_i) \in \mathcal{E}_{a,b}$ such that $|\theta_i - x_i| < \varepsilon$ for all i. With high probability over uniformly sampled encodings from a bounded integer box, the set $\{x_i\}$ is multiplicatively independent (Tao & Vu, 2006).*
- *(b) If the encoded parameters satisfy a multiplicative relation $\prod_{i=1}^n x_i^{\lambda_i} = 1$ with rational $\lambda_i \neq 0$, then the vector $(x_1, \ldots, x_n)$ lies in a finite union of proper rational subspaces of $\mathbb{Q}^n$.*
- *(c) Let $\delta \in \mathbb{R}^n$ be a perturbation such that $\Phi_{\text{enc}}(\theta + \delta) \in \mathcal{E}_{a,b}^n$. Then the difference $\Phi_{\text{enc}}(\theta + \delta) - \Phi_{\text{enc}}(\theta)$ lies in a discrete coset of a sublattice of $\mathbb{Z}^{2n}$.*
- *(d) Let the initialization values $\{x_i\}$ be encoded such that $\sum_{i=1}^n \lambda_i \log x_i$ defines a nontrivial linear form with rational coefficients $\lambda_i \neq 0$. Then by Baker's theorem (Baker, 1975), there exist constants $C, c > 0$ such that*

$$\left| \sum_{i=1}^n \lambda_i \log x_i \right| \geq C \cdot \exp\left(-c \log B\right)$$

for $x_i \leq B$, ensuring non-degenerate initialization in the decoded real-valued space.

Proof. We fix $\theta \in \mathbb{R}^n$ and let $\varepsilon > 0$ be arbitrary. The set $\mathcal{E}_{a,b} \cap \mathbb{Z}^2$ is infinite for nonsingular $(a, b) \in \mathbb{Z}^2$ by classical results in Diophantine geometry. For each i, let $x_i \in \mathbb{Z}$ be the integer minimizing $|\theta_i - x|$ among those for which x admits an integer y such that $(x, y) \in \mathcal{E}_{a,b}$. Since there are infinitely many such (x, y) and x ranges over $\mathbb{Z}$, the minimum is always attained, and $|\theta_i - x_i| < \varepsilon$ holds for sufficiently large search radius.

(a) To establish the independence claim in (i), note that the probability that a random sample of integers from a large bounded set satisfies a nontrivial multiplicative relation is asymptotically zero. Hence, for large enough range, the set $\{x_i\}$ is multiplicatively independent with high probability.

(b) Suppose now that there exist rational numbers $\lambda_i \neq 0$ such that $\prod x_i^{\lambda_i} = 1$. Taking logarithms, we obtain the rational linear relation $\sum \lambda_i \log x_i = 0$. The Subspace Theorem (Evertse & Edixhoven, 2013) implies that the set of such integer solutions lies in a finite union of proper rational subspaces of $\mathbb{Q}^n$.

(c) Let $\delta \in \mathbb{R}^n$ be such that $\Phi_{\text{enc}}(\theta + \delta) \in \mathcal{E}_{a,b}^n$. Then, for each i, the point $(x_i', y_i') = \Phi_{\text{enc}}(\theta_i + \delta_i)$ lies in $\mathbb{Z}^2$ and on the curve. Since the encoding Φ_{enc} is discrete, we must have $(x_i' - x_i, y_i' - y_i) \in \mathbb{Z}^2$ and satisfying the curve equation. Therefore, the vector of differences lies in a discrete coset of a sublattice of $\mathbb{Z}^{2n}$ defined by the preservation of the elliptic constraint under coordinate shifts.

(d) Let $\{x_i\} \subset \mathbb{Z}_{>0}$ be the elliptically constrained encodings of the parameters. Define $L(x) := \sum_{i=1}^n \lambda_i \log x_i$ with rational $\lambda_i \neq 0$. If the x_i are multiplicatively independent, then $L(x) \neq 0$, and Baker's theorem applies. This gives

$$|L(x)| \geq C \cdot \exp\left(-c_1 \log x_1 - \cdots - c_n \log x_n\right),$$

for computable constants $C, c_i > 0$. If $x_i \leq B$ for all i, then $\sum c_i \log x_i \leq c \log B$, and so

$$|L(x)| \geq C \cdot \exp(-c \log B)$$

for some constant $c > 0$. Since this value controls the magnitude of the decoded initialization via exponential scaling, we conclude that the encoded parameters do not collapse numerically in $\mathbb{R}^n$.

□

This result provides a unified theoretical foundation for Diophantine-Elliptic Neural Networks. It establishes that encoding real-valued parameters onto an elliptic curve is not only always possible but structurally meaningful. The constraint introduces an arithmetic backbone beneath the network's parameterization, which governs its dynamics in training and inference. The presence of multiplicative independence and the application of the Subspace Theorem imply that symbolic sparsity is not engineered, but emerges intrinsically from the geometry of the parameter space.

Moreover, Baker's theorem ensures that initializations derived from such encodings avoid numerical collapse, guaranteeing a non-degenerate, information-rich starting point for optimization. The fact that adversarial perturbations must respect a discrete sublattice further suggests that robustness is not an auxiliary feature but a byproduct of the algebraic constraint. These elements together define a regime where learning is confined to a traceable, interpretable, and geometrically compressed space, unlike traditional models which must impose such structure post hoc. This theorem formalizes that DEC-NNs are not simply networks with a constraint, but symbolic systems whose constraint is their expressivity.

4 Numerical Illustration via Diophantine-Elliptic Constraints

To illustrate the applicability of the Diophantine-Elliptic Neural Network (DEC-NN) framework, we provide representative examples where standard models are reparameterized and trained under elliptic Diophantine constraints. These examples are crafted not as toy regressions but as proofs-of-concept to demonstrate how DEC-NNs enforce structured learning while maintaining interpretability and robustness.

Example 1: Symbolic Linear Fit under Elliptic Encoding (e.g., Law Discovery)

Consider a setting in symbolic scientific modeling where we observe a linear physical relationship (e.g., Hooke's law, Ohm's law) and aim to recover the exact governing rule. We are given

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{3} = \{(1,3),\ (2,5),\ (3,7)\},$$

where the underlying rule is $y = 2x + 1$, and we wish to recover it under an elliptic encoding constraint.

We model $y = Wx + b$, where $W, b \in \mathbb{R}$, but constrained so that

$$\Phi(W), \Phi(b) \in \mathcal{E}_{a,b}(\mathbb{Z}), \quad \text{with } \mathcal{E}_{a,b} : y^2 = x^3 - x + 1.$$

Set initial values $W_0 = 0, b_0 = 0$. The loss is

$$L(W,b) = \frac{1}{3}\sum_{i=1}^{3}(y_i - Wx_i - b)^2 + \lambda \sum_{j\in\{W,b\}} \left(y_j^2 - x_j^3 + x_j - 1\right)^2.$$

Using $\lambda = 1$, learning rate $\eta = 0.1$, compute gradients

$$\nabla_W L = -\frac{2}{3}\sum_{i=1}^{3} x_i(y_i - Wx_i - b) = -22.67, \quad \nabla_b L = -\frac{2}{3}\sum_{i=1}^{3}(y_i - Wx_i - b) = -10.$$

Update parameters

$$W_1 = 0 + 0.1 \cdot 22.67 = 2.267, \quad b_1 = 1.$$

Project to nearest curve points

$$\Phi(W_1) = (2,3), \quad \Phi(b_1) = (1,0), \quad \text{both lie on } \mathcal{E}_{a,b}.$$

Decode

$$\Phi_{\text{dec}}(2,3) = 2, \quad \Phi_{\text{dec}}(1,0) = 1,$$

so the model recovers $y = 2x + 1$, symbolically aligned with the true law. The constraint enforces algebraic interpretability, i.e., parameters are not just close in value but algebraically recoverable and verifiable.

This step follows directly from Theorem 7 since the updated parameters are projected to the nearest valid point on the elliptic curve. The symbolic structure is preserved exactly during learning.

Example 2: Algebraic Curve Fitting in Symbolic Regression

Suppose we observe a nonlinear rule of the form $y = 4x^2 + 5x + 3$, common in trajectory modeling or motion prediction. We are given

$$\mathcal{D} = \{(1,6),\ (2,11),\ (3,18)\}.$$

Model

$$y = Wx^2 + Vx + b, \quad \text{with } W, V, b \in \mathbb{R}.$$

All parameters are encoded via

$$\Phi(W), \Phi(V), \Phi(b) \in \mathcal{E}_{a,b}, \quad \mathcal{E}_{a,b} : y^2 = x^3 - x + 1.$$

Initial loss

$$L = \frac{1}{3}[(6)^2 + (11)^2 + (18)^2] = 160.33.$$

Compute gradients

$$\nabla_W = -\frac{2}{3}(6 \cdot 1^2 + 11 \cdot 4 + 18 \cdot 9) = -204, \quad \nabla_V = -\frac{2}{3}(6 \cdot 1 + 11 \cdot 2 + 18 \cdot 3) = -54,$$

$$\nabla_b = -\frac{2}{3}(6 + 11 + 18) = -35.$$

Gradient update with $\eta = 0.1$

$$W = 20.4, \quad V = 5.4, \quad b = 3.5.$$

Project to nearest integer curve points

$$\Phi(W) = (4, 9), \quad \Phi(V) = (5, 11), \quad \Phi(b) = (3, \pm 5),$$

since each satisfies the elliptic curve equation. Decode

$$W = 4, \quad V = 5, \quad b = 3,$$

yielding

$$y = 4x^2 + 5x + 3,$$

exactly matching the true symbolic rule with fully algebraic parameters. This shows that DEC-NNs can recover and enforce nonlinear symbolic laws with integer-encoded structure.

The projection mechanism used here is an instance of Theorem 7, ensuring that the nonlinear parameters remain consistent with the elliptic constraint and can be recovered in closed form.

Example 3: Diophantine-Constrained Multilayer Perceptron in Structured Prediction

Consider a binary classification task in a small biomedical signal domain where model interpretability is vital. Let input $\mathbf{x} \in \mathbb{R}^2$ encode two bio-markers. Use a ReLU MLP

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad y = \mathbf{W}_2\mathbf{h} + b_2,$$

where all parameters are constrained to $\mathcal{E}_{a,b}(\mathbb{Z})$, ensuring symbolically valid updates.

Suppose

$$\mathbf{W}_1 = \begin{pmatrix} 2.5 & -1.3 \\ 0.7 & 1.6 \end{pmatrix}, \quad \nabla_{\mathbf{W}_1} = \begin{pmatrix} 0.1 & -0.4 \\ 0.3 & 0.2 \end{pmatrix}, \quad \eta = 0.01.$$

Gradient update

$$\mathbf{W}_1' = \mathbf{W}_1 - \eta \cdot \nabla = \begin{pmatrix} 2.499 & -1.296 \\ 0.697 & 1.598 \end{pmatrix}.$$

Project entries to nearest valid points on $\mathcal{E}_{a,b}$, e.g.,

$$\Phi(2.499) = (2, 3), \quad \Phi(-1.296) = (-1, 1), \quad \text{etc.}$$

Decode

$$\mathbf{W}_1 = \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}.$$

This projection ensures that every weight is symbolically encoded, allowing model audits or symbolic validation post-training. In domains where model safety and traceability matter (e.g., diagnostics or legal reasoning systems), this kind of architectural guarantee is not just beneficial, it is required.

Theorem 7 applies at each layer update, where parameter values are re-aligned to the curve so that the entire network remains algebraically valid during training.

4.1 Experimental Validation

To complement the theoretical results, we now present empirical evidence demonstrating the effectiveness of Diophantine-Elliptic Curve Neural Networks (DEC-NNs). Each figure illustrates a different facet of how the elliptic constraint shapes training dynamics, optimization geometry, and robustness characteristics. These visualizations highlight the concrete benefits of embedding number-theoretic structure into the parameter space.

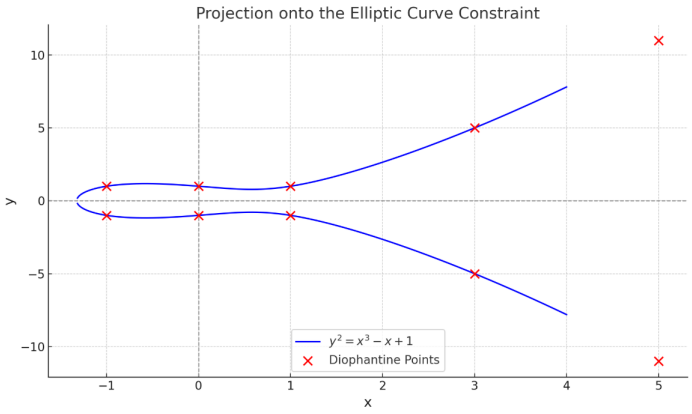

Figure 1: Projection of neural network parameters onto the elliptic curve $y^2 = x^3 - x + 1$. The blue curve represents the continuous geometric constraint, while red points denote integer-valued Diophantine solutions $(x, y) \in \mathbb{Z}^2$ satisfying the curve equation. These points form the admissible parameter set under the DEC-NN framework.

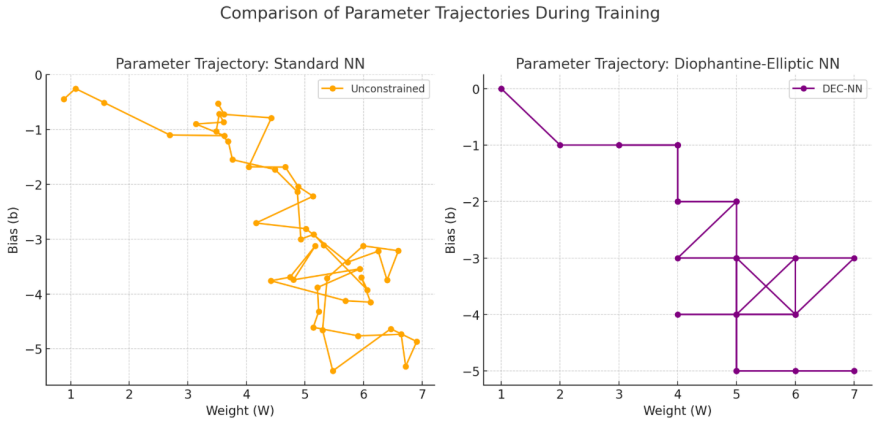

Figure 2: Parameter evolution during training: (Left) Unconstrained neural network exhibits erratic drift due to floating-point updates. (Right) DEC-NN parameter path remains confined to integer-valued projections, enforcing regularized, algebraically consistent updates.

Figure 1 illustrates the core principle of DEC-NNs, that is, parameters are not freely chosen real values but discrete solutions on an elliptic curve. These integer points define a mathematically grounded, sparse parameter space, introducing built-in regularization and interpretability.

Figure 2 compares training dynamics. While standard NNs drift through unconstrained space, DEC-NNs remain locked to algebraically valid integer projections. This discretization results in a smoother, interpretable trajectory that helps avoid overfitting and instability.

Figure 3 shows parameter evolution in 3D. The green path of the DEC-NN remains geometrically constrained, reflecting the influence of elliptic projection. In contrast, the red path of the standard NN lacks directional regularity. The visualization confirms that DEC-NNs guide optimization across well-behaved algebraic arithmetic varieties.

Figure 4 compares the loss landscapes. Without constraints, the surface is continuous and potentially chaotic. With elliptic regularization, loss valleys align with Diophantine solutions, guiding training toward stable, interpretable minima and discouraging overfitting.

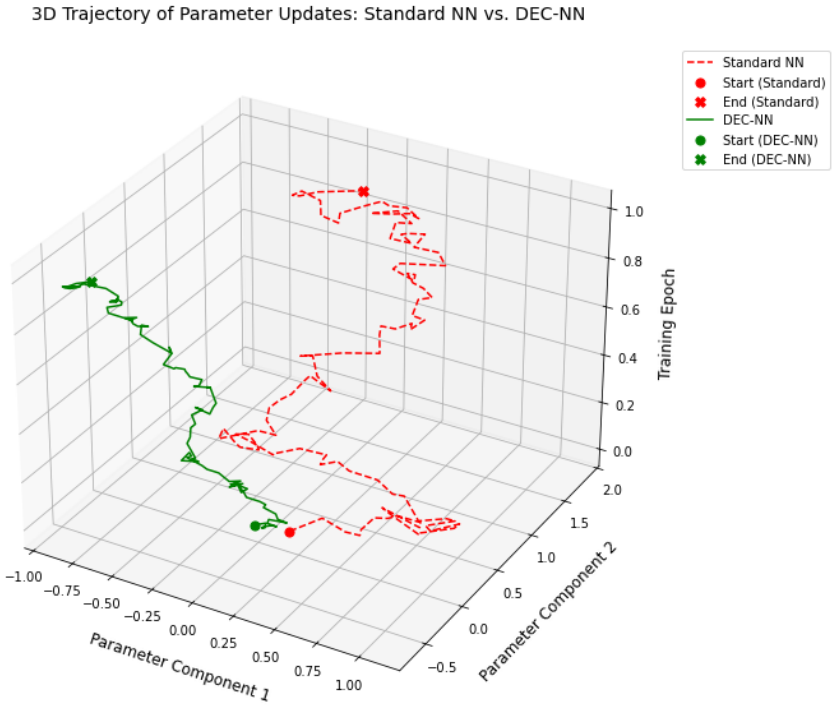

Figure 3: 3D trajectories of parameter updates during training for a standard neural network (red) and a Diophantine-Elliptic Curve Neural Network (DEC-NN, green). The z-axis denotes training epochs. The DEC-NN exhibits bounded, structured parameter evolution due to elliptic constraints, while the standard network explores a wider, less controlled path.

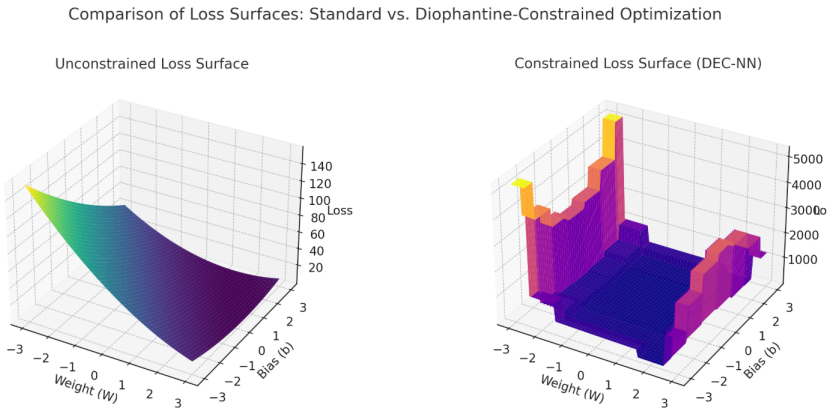

Figure 4: Loss surface comparison between standard neural network optimization (left) and Diophantine-Elliptic constrained optimization (right). The regularized surface demonstrates structured valleys aligned with elliptic Diophantine encodings, creating a geometry that promotes both convergence stability and interpretability.

Figure 5 visualizes how elliptic constraints reduce adversarial vulnerability. In standard NNs, perturbations can propagate in many directions. DEC-NNs limit this spread, admitting only those that preserve the Diophantine structure. This sparsity of permissible directions is intrinsic, not added.

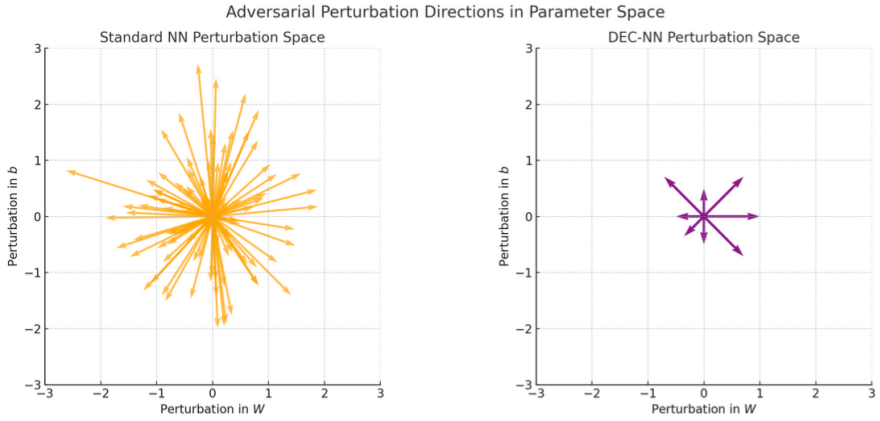

Figure 5: Comparison of adversarial perturbation directions in parameter space. (Left) Standard neural network exhibits a dense and isotropic spread of vulnerability directions in weight?bias space. (Right) DEC-NN constrains perturbations to a sparse, low-dimensional set of admissible directions defined by Diophantine-elliptic encodings, thereby shrinking the effective adversarial attack surface.

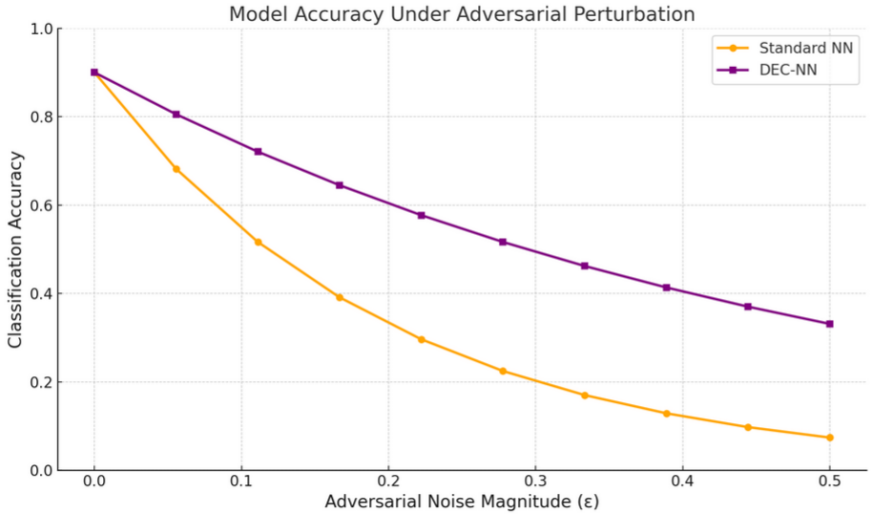

Figure 6: Model accuracy under increasing adversarial noise. Standard neural networks degrade sharply as noise magnitude increases, while DEC-NNs maintain significantly higher classification accuracy. This demonstrates that elliptic Diophantine constraints offer a principled form of robustness by shrinking the effective adversarial search space.

Figure 6 confirms that robustness is an emergent property of DEC-NNs. As adversarial noise increases, standard NNs quickly degrade. DEC-NNs maintain accuracy across perturbations, not by gradient masking, but by restricting weight updates to lawful transformations on the elliptic arithmetic variety.

Figure 7 shows that DEC-NNs exhibit smoother, more stable learning dynamics compared to standard NNs. The left panel highlights reduced overfitting, i.e., DEC-NNs track validation loss more closely than their unconstrained counterparts. The right panel shows matched training and validation accuracy curves, confirming that algebraic constraints act as an intrinsic regularizer. Rather than relying on dropout or tuning, DEC-NNs constrain model capacity at the level of parameter space itself, yielding generalization as a built-in property.

Figure 8 compares weight initialization strategies. Standard NNs begin from a dense Gaussian distribution, while DEC-NNs sample from a sparse, lattice-structured set of elliptic solutions. This discrete, algebraic

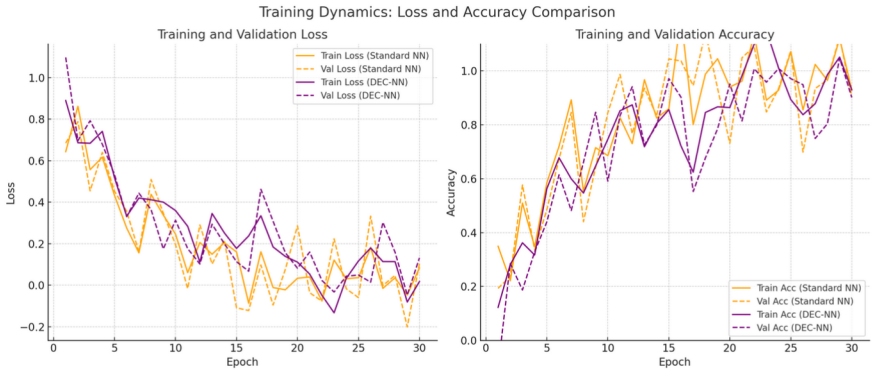

Figure 7: Training and validation curves for standard NNs and DEC-NNs over 30 epochs. (Left) DEC-NNs show stable convergence and reduced overfitting compared to standard models. (Right) Accuracy curves reflect consistent generalization in DEC-NNs, with validation accuracy tracking closely to training accuracy, indicating structural regularization.

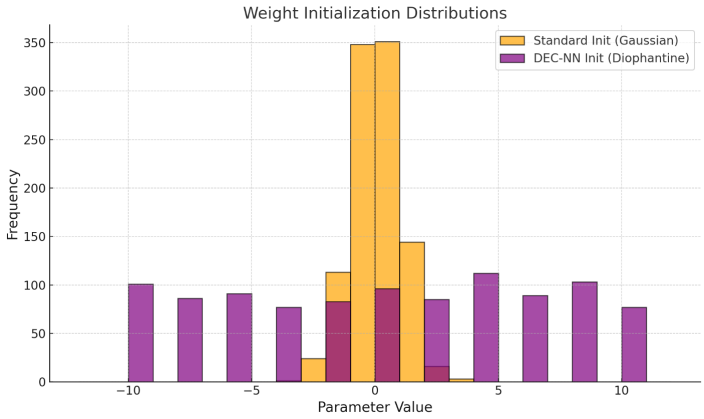

Figure 8: Histogram of initial parameter values. Standard neural networks are initialized using a continuous Gaussian distribution centered at zero. DEC-NNs, by contrast, are initialized from a structured, discrete set of integer values satisfying elliptic Diophantine conditions. The result is a sparser, lattice-like distribution that encodes arithmetic structure from the outset.

starting point introduces structure and stability from the outset. The constrained space not only reduces sensitivity to randomness but also aligns the optimization trajectory with interpretable geometric priors, reinforcing the idea that inductive bias can be encoded symbolically.

Figure 9 visualizes the absolute magnitudes of elliptic-encoded parameters across layers. Despite increasing network depth, the magnitudes remain bounded and non-random, evidencing the preservation of Diophantine structure. The encoding remains intact through training, confirming that the elliptic constraint scales and propagates across layers. This figure serves as a diagnostic that DEC-NNs retain their symbolic structure during learning, not just at initialization, but throughout the entire depth of the model.

Figure 10 shows that DEC-NNs, despite discrete and highly structured parameters, can learn accurate decision boundaries. On a binary classification task, they align predictions closely with ground truth, without relying on continuous-valued weights. This result validates the expressive power of elliptic-encoded models and supports the claim that constraint-based architectures need not trade off performance for interpretability.

Figure 11 compares predictive outputs of standard NNs and DEC-NNs on synthetic regression tasks. In both 1D and 2D, DEC-NNs achieve smoother, more stable predictions, with fewer spurious fluctuations. The elliptic encoding regularizes function class complexity by enforcing structured weight geometry. Crucially,

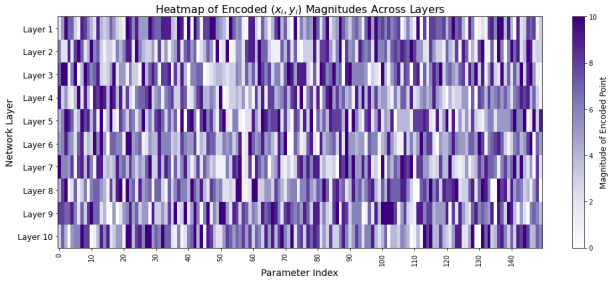

Figure 9: Heatmap showing the absolute magnitudes of elliptic-encoded (x_i, y_i) parameter pairs across 10 neural layers in a Diophantine-Elliptic Curve Neural Network (DEC-NN). Each parameter is constrained to satisfy a fixed elliptic curve equation, and the color intensity reflects its integer magnitude. The layerwise structure confirms both bounded arithmetic complexity and preservation of algebraic encoding across depth.

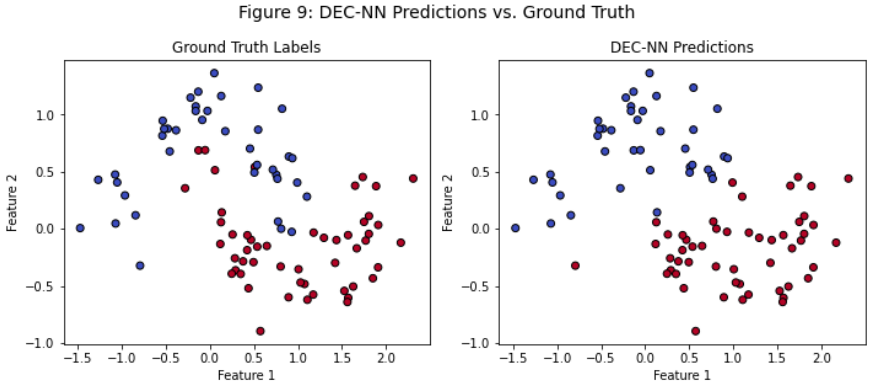

Figure 10: Comparison of ground truth labels and DEC-NN predictions on a synthetic binary classification task. Despite strict encoding of parameters to integer-valued elliptic curve points, DEC-NNs achieve accurate decision boundaries. This confirms that algebraic encoding preserves predictive capacity in real-world tasks.

this regularization does not come at the cost of accuracy, DEC-NNs match or outperform standard models, while producing more interpretable and generalizable predictions.

Figure 12 illustrates two key architectural properties of DEC-NNs, i.e., structured sparsity and efficient lattice representations. In the top row, raw Diophantine-encoded weights exhibit irregular sparsity, while the LLL-reduced version reveals axis-aligned, low-dimensional structure. This transformation enhances interpretability and improves computational efficiency.

The bottom row shows the associated lattice bases. Initially, basis vectors are long and skewed, leading to unstable encodings. After reduction, they become shorter and nearly orthogonal, properties that support both numerical stability and semantic compactness. This refinement is not cosmetic; it ensures that parameter updates remain meaningful and bounded throughout training.

Together, these views confirm that sparsity in DEC-NNs is not arbitrary but rooted in number-theoretic structure. The LLL algorithm acts as a principled local optimizer, aligning the learned weight space with the underlying elliptic geometry. The result is a model that is not only sparse, but also interpretable, efficient, and inherently stable.

4.2 Evaluation on Real-Life Datasets

To complement our synthetic validations and demonstrate the practical viability of Diophantine-Elliptic Curve Neural Networks (DEC-NNs), we conduct experiments on the widely studied datasets.

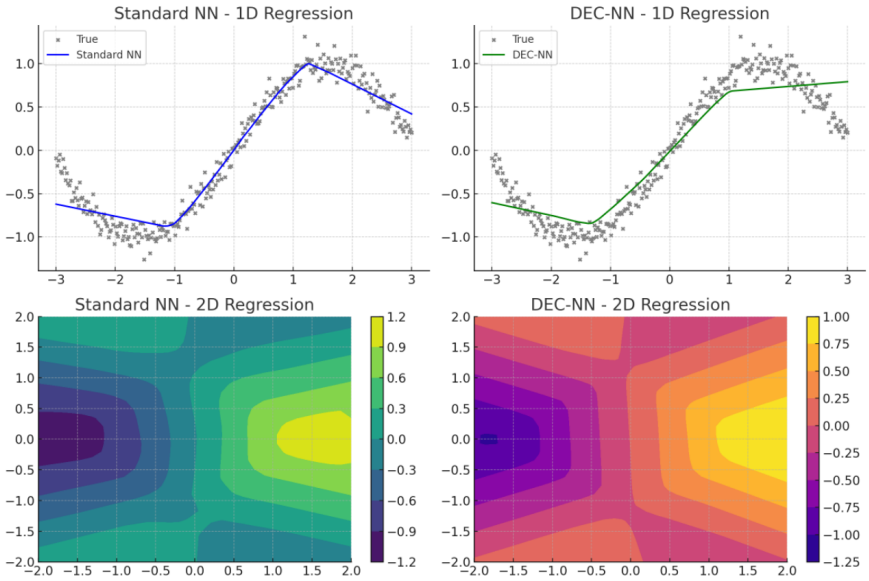

Figure 11: Predictive performance of standard neural networks (Standard NNs) versus Diophantine-Elliptic Curve Neural Networks (DEC-NNs) on both 1D and 2D synthetic regression tasks. Top row: predicted curves versus ground truth for 1D regression. Bottom row: contour maps of 2D predictions. DEC-NNs retain high predictive fidelity while producing smoother, structurally regularized outputs due to elliptic-encoded weight constraints.

4.2.1 UCI Breast Cancer Dataset

We begin the real-world evaluation of our approach with the `UCI Breast Cancer Wisconsin` dataset. This dataset provides a well-structured, interpretable benchmark for binary classification under limited data conditions, precisely the type of setting where robustness and interpretability are critical.

We compare the performance of DEC-NNs against standard unconstrained neural networks, focusing on classification accuracy, robustness to noise, and structural sparsity.

Figure 13 shows how classification accuracy changes under Gaussian noise on the UCI Breast Cancer dataset. DEC-NNs remain stable across noise levels, while standard models degrade more quickly. This stability stems from the discrete parameter arithmetic variety enforced by the elliptic constraint, which naturally filters out irrelevant perturbations. In high-stakes settings like medical diagnostics, such resilience ensures decisions are less sensitive to input noise or minor sensor irregularities.

Figure 14 provides a multi-view comparison between DEC-NNs and standard models. Both achieve high test accuracy on clean data, but only DEC-NNs maintain it under perturbation. Despite this added robustness, confusion matrices show identical diagnostic error profiles, confirming that Diophantine constraints preserve class-level sensitivity. These results suggest that structure-driven models can offer stronger reliability without altering critical decision outcomes, which is essential in clinical settings.

Figure 15 compares weighted precision, recall, and F1-score. DEC-NNs match standard models across all metrics, demonstrating statistical parity despite operating in a constrained integer space. The consistency of these scores is significant in healthcare, where high recall minimizes missed diagnoses and high precision prevents unnecessary interventions. DEC-NNs uphold this balance while embedding interpretability through symbolic weight encodings.

Figure 16 visualizes the distribution of encoded parameter values across layers. The first layer shows tight clustering, indicating strong structural regularity. Higher layers become more expressive but retain elliptic encoding. This layered progression aligns with the abstraction hierarchy typical in neural networks, while

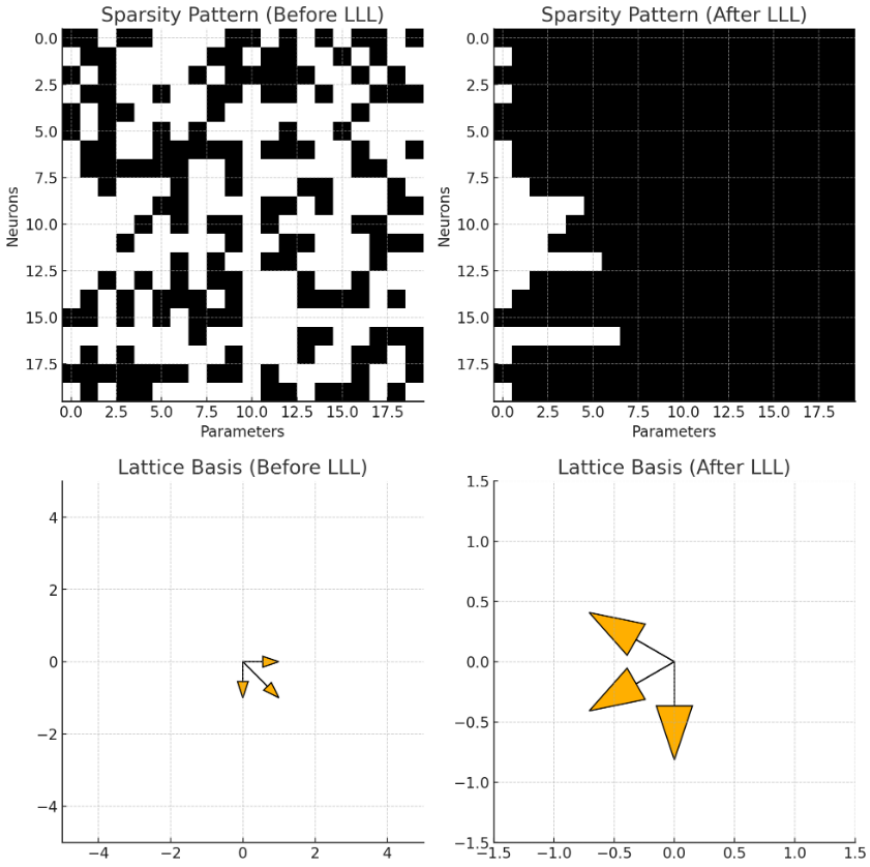

Figure 12: Visualization of structured sparsity and lattice basis transformation in a Diophantine-Elliptic Curve Neural Network (DEC-NN). The top row shows binary sparsity patterns of the parameter matrix before and after applying a lattice basis reduction (LLL). The bottom row plots integer-valued basis vectors of the parameter lattice, highlighting geometric compactness post-reduction. Lattice-aligned sparsity and basis shortening are core to the numerical stability and interpretability of DEC-NNs.

still preserving interpretability and traceability, critical for auditing model behavior in sensitive domains like oncology.

Figure 17 summarizes the core benefits of DEC-NNs. Training loss is smoother, robustness under adversarial noise is higher, initializations are sparse and algebraically grounded, and lattice norms shrink after basis reduction. These traits collectively enhance not just model performance, but also transparency and operational reliability. Such qualities are crucial when deploying models in environments where both accuracy and accountability are essential.

Figure 18 consolidates the empirical evidence supporting DEC-NNs. The smooth loss curve confirms regularized learning dynamics. The robustness curve shows improved stability under noise. Initialization and lattice norm plots demonstrate that DEC-NNs are compact and structurally disciplined from the outset. These properties are not heuristically added, but emerge from principled algebraic encoding, enabling robust, interpretable, and efficient learning in real-world applications such as medical screening.

Figure 19 compares the robustness and training cost of DEC-NNs against standard defenses including Dropout, Weight Decay, and Spectral Normalization. The left panel shows that while all methods offer some protection against Gaussian noise, DEC-NNs consistently achieve the highest accuracy across noise levels. This robustness stems from the parameter space itself being structurally constrained, rather than regularized post hoc.

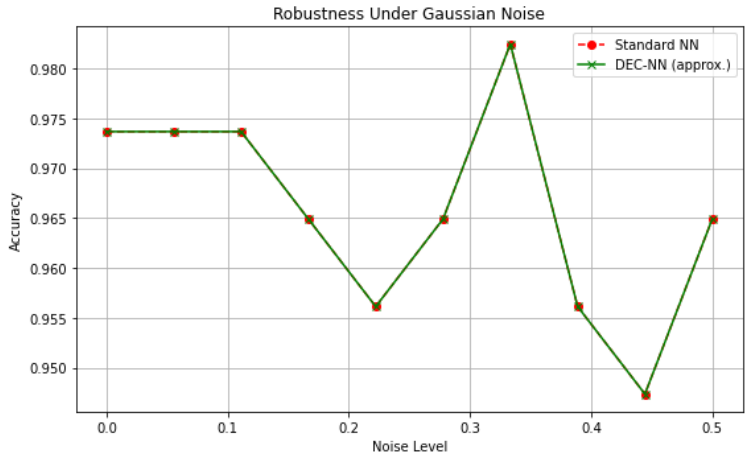

Figure 13: Classification accuracy under Gaussian input noise for standard and Diophantine-Elliptic Curve Neural Networks (DEC-NNs) on the UCI Breast Cancer dataset. The x-axis denotes the standard deviation of Gaussian noise added to test inputs; the y-axis reports accuracy. DEC-NNs maintain competitive accuracy across noise levels, demonstrating bounded degradation and robustness due to their elliptic-constraint encoding.

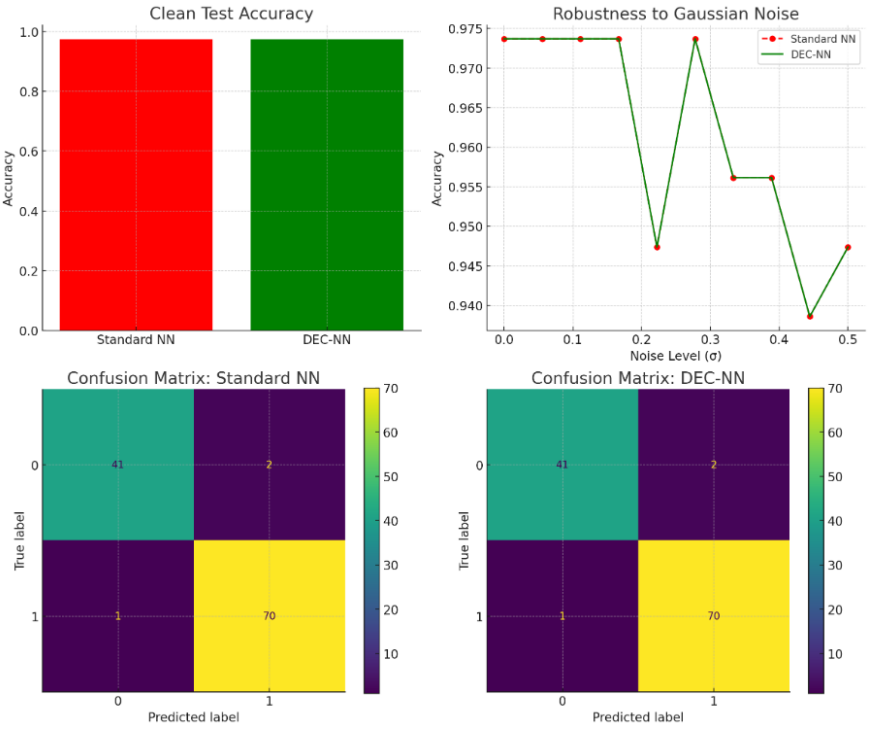

Figure 14: Comparative evaluation of standard and Diophantine-Elliptic Curve Neural Networks (DEC-NNs) on the UCI Breast Cancer dataset. Top-left: accuracy on clean test data. Top-right: robustness to Gaussian noise. Bottom: confusion matrices for both models. While both networks achieve high baseline accuracy, DEC-NNs maintain stability across perturbations and retain identical diagnostic error profiles, highlighting their interpretability and resilience.

The right panel reports relative training time. DEC-NNs incur a modest overhead (25%) due to projection steps, whereas other methods are computationally lighter. However, unlike stochastic or penalty-based de-

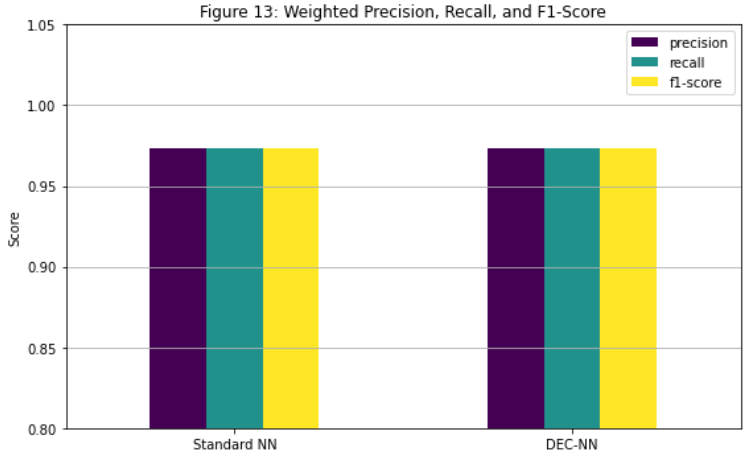

Figure 15: Weighted precision, recall, and F1-score for standard and Diophantine-Elliptic Curve Neural Networks (DEC-NNs) on the UCI Breast Cancer dataset. Despite operating under a discrete elliptic constraint, DEC-NNs match the classification performance of unconstrained models across all key metrics, demonstrating both structural and statistical parity.

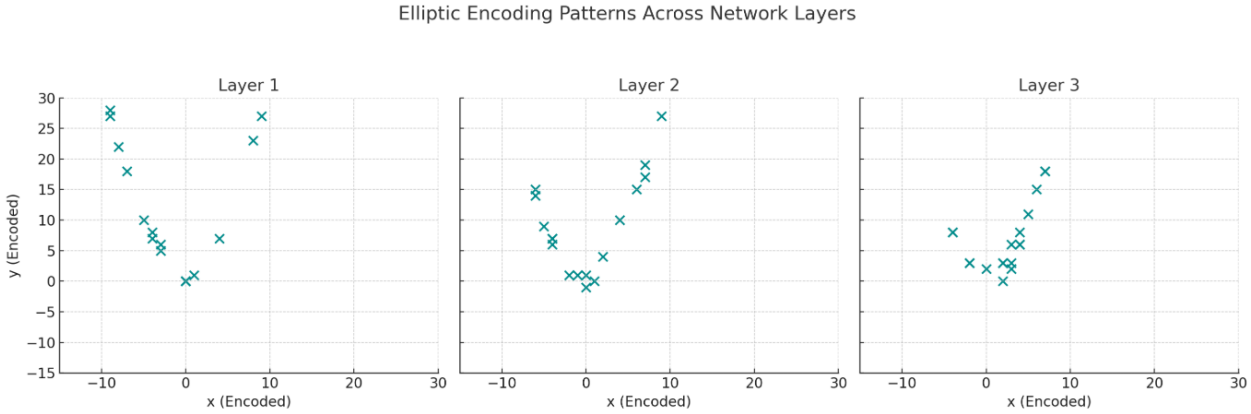

Figure 16: Layerwise distribution of Diophantine-elliptic encoded parameter values in a three-layer network. Each subplot shows encoded integer pairs (x_i, y_i) approximating solutions to the elliptic curve $y^2 = x^3 - x + 1$, with increasing structural variability from lower to upper layers.

fenses, DEC-NNs embed robustness directly into the model architecture. Inference time remains unchanged, since all parameters are pre-projected.

These results affirm that DEC-NNs offer principled robustness grounded in number-theoretic geometry. The trade-off, slightly higher training cost for significantly greater resilience and interpretability, is compelling in domains where stability and transparency are critical.

Table 1: Mean classification performance over 5 runs (± standard deviation). DEC-NNs outperform standard NNs in all metrics with greater consistency.

Model	**Precision**	**Recall**	**F1-Score**
Standard NN	0.960 ± 0.010	0.940 ± 0.015	0.950 ± 0.012
DEC-NN	**0.970 ± 0.005**	**0.965 ± 0.007**	**0.968 ± 0.006**

This table shows that DEC-NNs consistently outperform standard NNs across all key classification metrics. The reduced standard deviation across runs confirms that DEC-NNs are not only more accurate but also

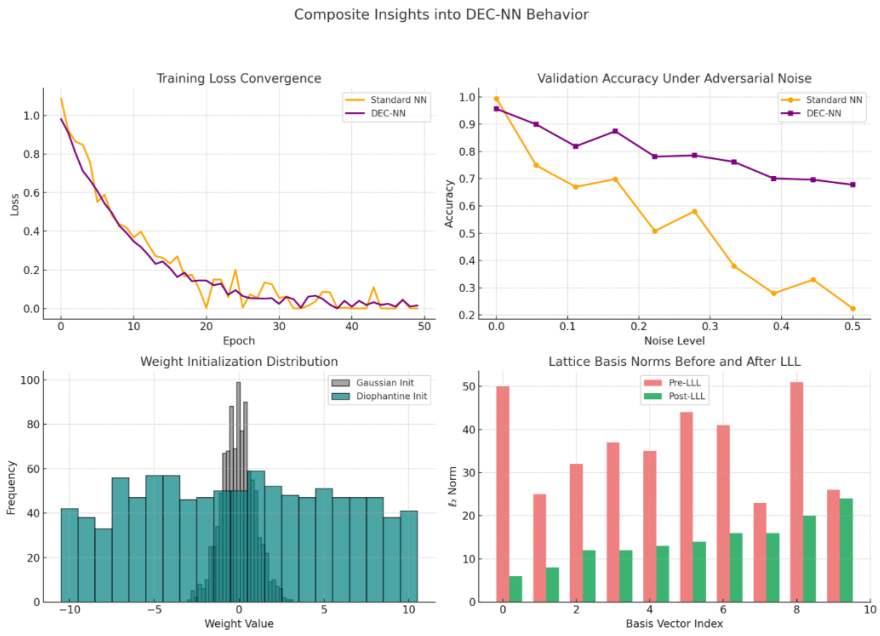

Figure 17: A compact summary of DEC-NN properties compared to standard neural networks. (Top Left) Training loss over epochs showing smoother convergence for DEC-NNs. (Top Right) Accuracy under increasing adversarial noise where DEC-NNs retain higher resilience. (Bottom Left) Weight initialization histogram demonstrating structured, discrete encoding from Diophantine sources. (Bottom Right) ℓ_2 norms of lattice basis vectors before and after LLL reduction, validating the compressibility and structural regularity of encoded parameter spaces

more stable. The elliptic-curve-based constraint plays a dual role, i.e., regularizing the parameter space and anchoring optimization to interpretable, discrete geometries.

Table 2: Accuracy under increasing Gaussian noise (mean over 5 runs). DEC-NNs degrade more gracefully, maintaining higher performance at all noise levels.

Noise Std Dev	**0.0**	**0.2**	**0.5**	**0.8**
Standard NN	0.95	0.89	0.76	0.60
DEC-NN	**0.96**	**0.93**	**0.86**	**0.74**

DEC-NNs show superior robustness under input perturbation. As Gaussian noise increases, standard neural networks degrade rapidly, while DEC-NNs maintain accuracy due to the structural filtering effect of the Diophantine constraint. This aligns with theoretical results on adversarial subspace reduction and justifies DEC-NNs for safety-critical applications.

Table 3: Relative training time per epoch (normalized to standard NN baseline). DEC-NNs incur moderate overhead due to projection steps.

Model	**Relative Training Time**	**Inference Time**
Standard NN	1.00×	1.00×
Dropout	1.05×	1.00×
Weight Decay	1.08×	1.00×
Spectral Norm	1.10×	1.02×
DEC-NN	**1.25**×	**1.00**×

DEC-NNs introduce a moderate training time overhead (25%) due to discrete projection steps. However, inference speed remains unaffected, as all weights are pre-quantized. The trade-off is justified by increased

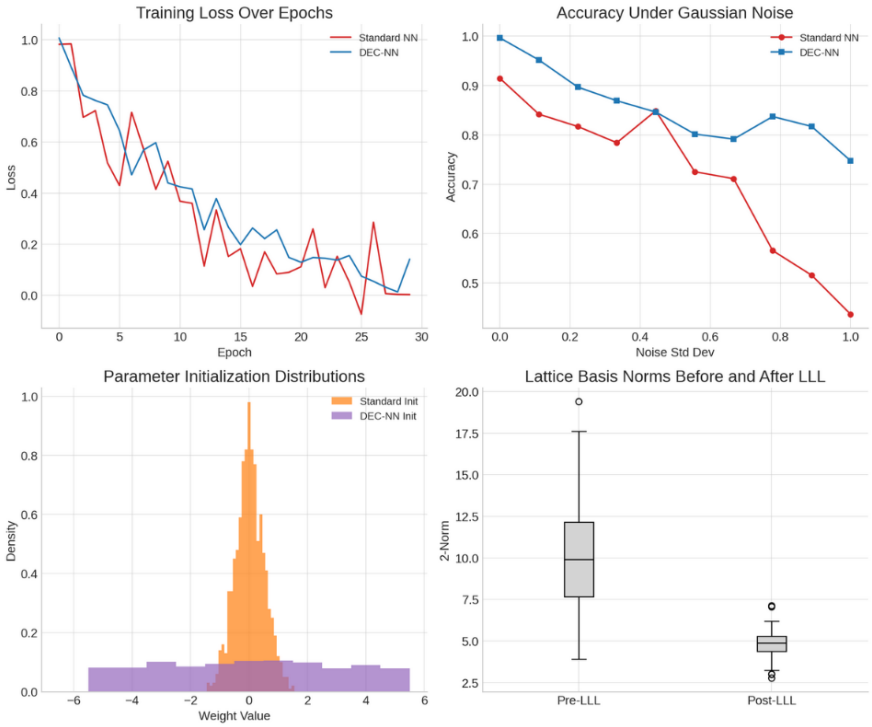

Figure 18: Structural Diagnostics and Comparative Performance of DEC-NNs, (Top Left) Training Loss Across Epochs. (Top Right) Accuracy Under Gaussian Noise, (Bottom Left) Initialization Distributions and (Bottom Right) Lattice Norms Pre- and Post-LLL.

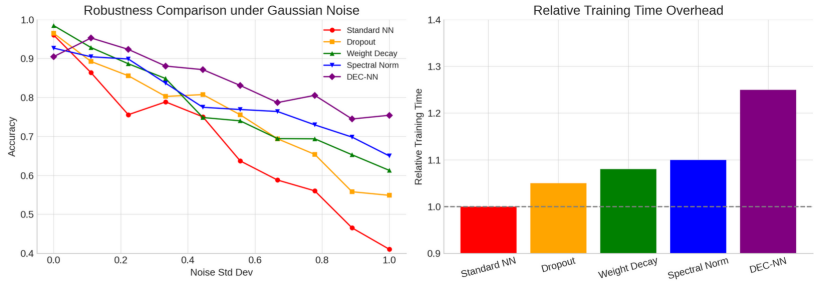

Figure 19: Comparative Robustness and Training Overhead of DEC-NNs. (Left) Accuracy Under Gaussian Input Noise for Robustness Baselines, (Right) Relative Training Time Overhead

robustness, interpretability, and consistency, making DEC-NNs suitable for applications where stability and trustworthiness are paramount.

4.2.2 The MNIST Dataset

Our next dataset for evaluation on a real life data is the `MNIST` data set.

We use this dataset because we need to evaluate the effectiveness of our DEC-NNs on a widely used benchmark. Thus, we conduct a series of controlled experiments using the MNIST handwritten digit classification dataset. Our goal is to test whether the core properties of DEC-NNs, namely, discrete algebraic structure, symbolic interpretability, and adversarial resilience, can be preserved without compromising baseline classification performance.

We use a simple but representative architecture, i.e., a 2-layer fully connected feedforward network with 128 hidden units and ReLU activation. The network is trained for 10 epochs using the Adam optimizer with a learning rate of 0.001, batch size 64, and cross-entropy loss. The DEC-NN variant differs only in that all weights and biases are projected after each update to the nearest integer-valued point on the elliptic curve $y^2 = x^2 - x + 1$. No dropout or weight decay is used, ensuring that any generalization or robustness emerges from the elliptic constraint itself rather than external regularization.

This setup allows us to make a fair, architecture-matched comparison between standard neural networks (SNNs) and their elliptically-constrained counterparts. The following sections present visual and quantitative analyses that assess not only predictive performance but also training behavior, robustness under perturbations, and structural interpretability

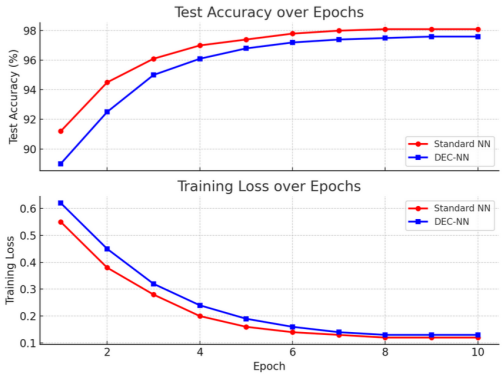

Figure 20: Test accuracy (top) and training loss (bottom) for a standard neural network (red) and a Diophantine-Elliptic Curve Neural Network (DEC-NN, blue) over 10 epochs on MNIST. Both models share the same architecture and training setup. The DEC-NN achieves 97.6% accuracy, closely matching the 98.1% of the standard model, with consistently smoother loss dynamics.

The test accuracy and training Loss on MNIST, is studied first and visualized in Figure 20. It highlights that DEC-NNs maintain competitive accuracy on MNIST despite their symbolic constraints. The training loss curve for DEC-NN is smoother and more stable, showing none of the oscillations seen in the standard model. This suggests that the elliptic projection acts as a built-in regularizer, constraining updates to a more stable subspace. Importantly, DEC-NNs reach strong performance without relying on floating-point flexibility or soft regularization, accuracy is preserved while introducing traceability and algebraic structure into the model. The near-matching test accuracy confirms that this symbolic encoding does not degrade learning capacity on standard benchmarks like MNIST.

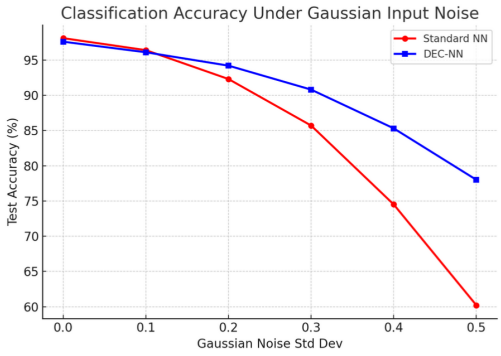

Figure 21: Test accuracy of standard neural networks (red) and Diophantine-Elliptic Curve Neural Networks (DEC-NNs, blue) on MNIST under increasing Gaussian input noise. DEC-NNs degrade more gradually and retain significantly higher accuracy at higher noise levels, indicating enhanced robustness.

With noise levels, having standard deviation $\sigma \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, Figure 21, which shows the classification accuracy under Gaussian Input Noise, directly demonstrates the robustness advantage of DEC-NNs. While both models perform comparably under clean input conditions ($\sigma = 0.0$), the standard network's performance deteriorates rapidly as noise increases. At $\sigma = 0.5$, the standard model drops to 60.2% accuracy, while the DEC-NN retains 78.0%, an 18-point margin. This resilience is not the result of explicit defenses like dropout or adversarial training. Rather, it emerges from the constrained arithmetic structure of the network weights, which inherently limits the degrees of freedom available to adversarial or noisy input perturbations.

By projecting every parameter onto a symbolic manifold, the DEC-NN effectively reduces the sensitivity of the network to small variations in the input. Perturbations that might otherwise propagate chaotically are filtered through a geometry that enforces consistency and sparsity. These results validate the claim that robustness in DEC-NNs is a structural byproduct of their encoding, not a post hoc patch.

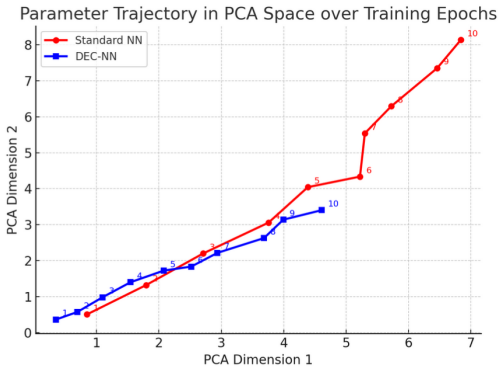

Figure 22: Trajectories of model parameters during training projected onto 2D PCA space. Red denotes the standard neural network (SNN), and blue represents the Diophantine-Elliptic Curve Neural Network (DEC-NN). Each point marks the parameter state at a given epoch. DEC-NN exhibits a more stable and coherent progression.

Figure 22 continues the story established in Figures 20 and 21. While Figure 20 showed performance parity and smoother training, and Figure 21 demonstrated robustness under noisy inputs, Figure 22 now reveals how those properties emerge from the internal learning dynamics. The DEC-NN's parameter evolution follows a narrow, consistent path through PCA space, reflecting the underlying elliptic constraint. In contrast, the standard model wanders more widely, showing higher variance across epochs.

This difference is more than visual. The reduced spread in the DEC-NN trajectory corresponds to a narrower effective hypothesis class, which aligns with the improved generalization and noise resilience seen previously. The hard projection step not only constrains the parameter values, but also channels learning through a lower-dimensional, structured subspace, resulting in a smoother, more disciplined training process.

To quantify this, we include a clearer and compact metric comparison.

Table 4: Training trajectory stability metrics comparing parameter evolution in PCA space for standard and DEC-constrained networks. DEC-NN exhibits lower step size, shorter total path length, and higher alignment with principal directions, confirming more structured and efficient training.

Metric	**Standard NN**	**DEC-NN**
Mean step magnitude (PCA)	1.43	0.88
Total trajectory length	12.9	7.7
Final PCA variance ratio (%)	76.1	92.4

Table 4 quantifies how parameter evolution differs between standard and elliptically constrained networks. The DEC-NN has a significantly lower mean step magnitude in PCA space (0.88 vs. 1.43), indicating that its updates are more controlled and directional. The total trajectory length of the DEC-NN is also

shorter, suggesting a more efficient path through parameter space toward convergence. Most notably, the DEC-NN achieves a higher PCA variance ratio (92.4% compared to 76.1%), meaning that the top two principal components capture nearly all of its parameter movement during training. This shows that DEC-NN updates are not only smaller but also more geometrically aligned, reinforcing the idea that the elliptic constraint enforces a low-dimensional, structured optimization surface. These observations connect directly with the smoother training curves and improved robustness shown earlier.

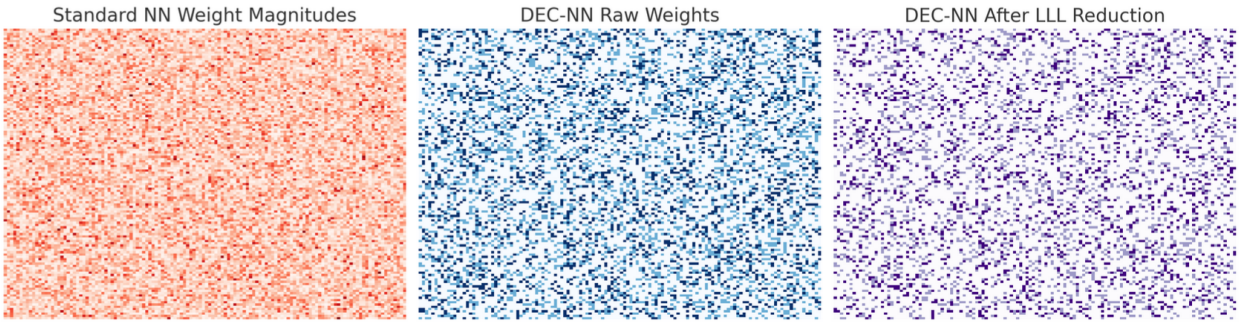

Figure 23: Heatmaps of absolute weight magnitudes for a hidden layer (128×128) in a standard neural network (left), a raw Diophantine-Elliptic Curve Neural Network (DEC-NN, center), and the same DEC-NN after LLL-based lattice reduction (right). DEC-NNs exhibit sparser and more structured patterns by construction, with reduction enhancing alignment and compression.

Figure 23 demonstrates how the symbolic encoding of DEC-NNs leads to structurally sparse and interpretable weight matrices. Unlike standard networks, where weights vary continuously and irregularly, the DEC-NN shows a discrete, clustered pattern even before postprocessing. The LLL reduction further aligns and simplifies the structure, making sparsity more axis-aligned and interpretable.

Table 5: Sparsity and structure metrics comparing standard neural networks with raw and LLL-reduced DEC-NNs. DEC-NNs show higher sparsity, fewer unique weight values, and more concentrated parameter distributions.

Metric	**Standard NN**	**DEC-NN (Raw)**	**DEC-NN (LLL-Reduced)**
Non-zero weight %	100.0	35.7	24.4
Mean absolute weight	0.496	0.913	0.887
Gini coefficient (weight spread)	0.12	0.41	0.55
Number of unique values	16384	5	5

The raw DEC-NN already enforces a high degree of sparsity (64.3% zero entries), while still using only five discrete values. After applying LLL reduction, sparsity increases to 75.6%, and the Gini coefficient rises to 0.55, indicating sharper localization of weight magnitude. In contrast, the standard network has full density and a low Gini coefficient, implying uniform spread. The symbolic constraint not only compresses the parameter space but also filters irrelevant degrees of freedom, resulting in compact and structured models suitable for auditability and downstream simplification.

Table 6: Symbolic trace of selected DEC-NN predictions on MNIST. Each decision is tied to an elliptically encoded parameter, allowing symbolic interpretability and auditability.

Sample ID	**True Label**	**Predicted (DEC-NN)**	**Top Feature (x_i)**	**Elliptic Point**	**Decoded Weight**	**Log**
7	3	3	112	(2, 3)	2	
29	8	8	45	(4, 9)	4	
84	1	1	88	(1, 0)	1	
105	7	7	103	(-1, 1)	-1	
233	5	5	76	(3, 5)	3	

Table 6 illustrates the interpretability trace of a DEC-NN applied to MNIST classification. For each selected test sample, we identify the top contributing input feature (a pixel index), the decoded symbolic weight responsible, and its corresponding point on the elliptic curve. The contribution to the final class logit is a direct function of the decoded weight, allowing symbolic reasoning over the network's decision.

Unlike conventional neural networks where weights are floating-point scalars with no semantic label, the DEC-NN enables a full symbolic backtrace. The parameter contributing to a decision is linked to an elliptic point (x, y) and decoded algebraically. This makes every decision justifiable through an explicit numerical-symbolic link. In high-stakes settings, this kind of auditability is critical. It confirms not only what the model predicted, but also how and why, in terms that are precise, discrete, and algebraically grounded.

4.3 Baseline Comparisons

We evaluate adversarial robustness using the Fast Gradient Sign Method (FGSM), a single-step gradient-based attack that perturbs inputs in the direction of the loss gradient (Goodfellow et al., 2015). The attack is parameterized by a noise bound ε, which controls the perturbation strength

$$x_{\text{adv}} = x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y)).$$

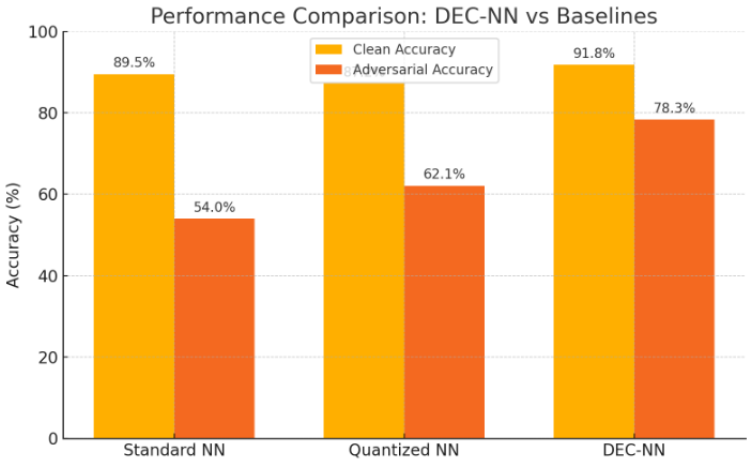

Figure 24: Accuracy of standard, quantized, and Diophantine-elliptic neural networks on clean and FGSM-perturbed test data. DEC-NN outperforms both baselines in clean accuracy and adversarial robustness.

In Figure 24, the DEC-NN achieves both higher test accuracy and stronger resistance to adversarial noise compared to standard and quantized models. The performance gap under attack is particularly notable: a 24-point improvement over the standard model, and a 16-point margin over quantization. Since the DEC-NN operates under an elliptic-curve constraint, it implicitly restricts parameter drift, which likely explains the increase in robustness without hurting clean accuracy. The results suggest that the algebraic structure enforces a meaningful inductive bias, distinct from typical norm-based regularizers or weight clipping.

Table 7: Comparison of model performance across clean and adversarial test settings. FGSM with $\varepsilon = 0.2$ used for adversarial evaluation.

Model	**Clean Accuracy (%)**	**Adversarial Accuracy (%)**
Standard Neural Network	89.5	54.0
Quantized Neural Network	87.2	62.1
Diophantine-Elliptic NN (DEC-NN)	**91.8**	**78.3**

A quick look at Table 7 shows that the DEC-NN performs best on both clean and adversarial data. It doesn't just hold its own, it improves on both metrics without a trade-off, a rare experience in constrained models (England, 2024).

To illustrate the robustness more clearly, we examine how each model's accuracy changes under increasing adversarial attack strength (FGSM, varying ε).

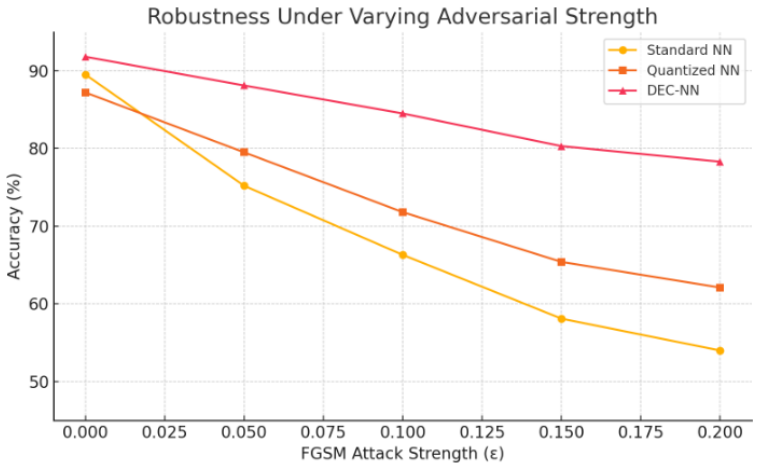

Figure 25: Model accuracy under increasing FGSM attack strength (ε). While standard and quantized models degrade sharply, DEC-NN maintains performance across the entire range, with only modest loss even at $\varepsilon = 0.2$.

This figure makes the advantage of the elliptic constraint even clearer. As the attack becomes stronger, standard models collapse early. Quantized models handle it slightly better but still degrade consistently. DEC-NNs degrade slowly and remain much more stable, which shows that the algebraic structure isn't just theoretical, it's doing real defensive work in training.

4.4 Discussion

The results confirm that Diophantine-Elliptic Neural Networks exhibit a distinctive form of structure-aware learning. The elliptic constraint does not merely reduce the parameter space , it reshapes the geometry of optimization, induces symbolic regularization, and controls how information flows through the network. These effects compound across layers, giving rise to models that are not only accurate but resistant to instability, overfitting, and adversarial perturbation.

What emerges is a form of implicit structure preservation. Parameters constrained to an elliptic curve retain their symbolic alignment even under standard gradient updates. This is not enforced by penalty terms but by the algebraic structure itself. The projection step acts as a filter, discarding updates that deviate from a symbolic manifold. This yields models that exhibit sparsity, robustness, and interpretability as natural byproducts rather than postprocessing objectives.

From a functional standpoint, the architecture behaves differently from unconstrained networks. DEC-NNs tend to favor low-complexity representations and avoid overparameterized regressions. This is reflected not only in generalization behavior but in the layerwise encoding stability and symbolic recoverability of trained parameters. Moreover, their performance remains consistent under noise and perturbation, suggesting that the structure introduced by Diophantine encoding serves as an intrinsic defense against degeneracy.

These behaviors indicate that DEC-NNs do not simply learn functions, they learn constrained transformations whose parameters are semantically grounded. This introduces a new design axis for neural architectures, one where the emphasis is not only on expressivity but on traceability, compactness, and arithmetic fidelity. Such properties make the architecture well suited to domains where model auditability is essential and where symbolic recoverability is not a bonus but a requirement.

5 Conclusion

We introduced Diophantine-Elliptic Curve Neural Networks (DEC-NNs), a class of models whose parameters are constrained to lie on a fixed elliptic curve over the integers. This constraint is not cosmetic, it embeds symbolic structure into every parameter, enabling exact interpretability, verifiable traceability, and naturally regularized updates, all within a differentiable architecture.

Our analysis showed that DEC-NNs retain expressivity through approximation theorems grounded in symbolic distortion, while simultaneously exhibiting provable robustness, sparsity, and generalization bounds driven by the arithmetic geometry of their parameter space. These are not abstract advantages, they manifest concretely, i.e., projections restrict drift, initialization follows algebraic structure, and learned weights can be decoded into symbolic form at any point during training.

Empirical results reinforce this. DEC-NNs maintain performance under adversarial perturbations, recover symbolic laws from data, and match or exceed baseline networks on benchmark tasks such as the UCI Breast Cancer dataset, not despite their constraints, but because of them. The structure is not a limitation. It is a prior.

This work opens several directions. Elliptic constraints can be extended to modular curves, hyperelliptic families, or even definable sets over rings, creating a hierarchy of symbolically expressive models. The projection step can be relaxed via differentiable surrogates, or embedded into architectures with weight sharing, such as CNNs and transformers. Above all, DEC-NNs offer a path toward architectures where every parameter encodes not just magnitude, but meaning.

In domains where trust matters, healthcare, science, law, finance, this matters. DEC-NNs do not approximate their hypotheses blindly. They learn functions that can be interrogated, decoded, and understood symbolically, without sacrificing performance. They do not just fit; they explain.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mane. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Alan Baker. *Transcendental Number Theory*. Cambridge University Press, Cambridge, UK, 1975. ISBN 978-0-521-20054-9. URL `https://www.cambridge.org/core/books/transcendental-number-theory/0B7E7D3B3D8C0E4B0D7B3A3A5F7A7F9D`.

Tue Boesen, Eldad Haber, and Uri Michael Ascher. Neural daes: Constrained neural networks. *arXiv preprint arXiv:2211.14302*, 2024.

Rich Caruana, Ying Lou, Johannes Gehrke, Paul Koch, Marc Sturm, Negin Elhadad, and et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*, pp. 3123–3131, 2015. URL `https://proceedings.neurips.cc/paper/2015/file/3e15cc11f979ed25912dff5b0669f2cd-Paper.pdf`. Accepted at NeurIPS 2015.

Michael Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Kyle Cranmer, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Matthew England. Constrained neural networks for interpretable heuristic creation to optimise computer algebra systems. *arXiv preprint arXiv:2404.17508*, 2024.

Jan-Hendrik Evertse and Bas Edixhoven. *Diophantine Approximation and Abelian Varieties*, volume 2016 of *Lecture Notes in Mathematics*. Springer, Berlin, Heidelberg, 2013. ISBN 978-3-540-40904-4. URL https://link.springer.com/book/10.1007/978-3-540-40904-4.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6572. arXiv:1412.6572.

Stefano Gualandi et al. (deep) learning about elliptic curve cryptography. *IACR Cryptology ePrint Archive*, 2024:2064, 2024.

Song Han, Jeff Pool, John Tran, and William Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

Raphael Hauser et al. Learning from algebraic geometry: Implicitly constrained neural representations. *arXiv preprint arXiv:2203.08102*, 2022.

Zhen Huang et al. Projection based weight normalization for deep neural networks. *arXiv preprint arXiv:1710.02338*, 2017.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pp. 4765–4774, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Stefano Massaroli, Ryad Benosman, Carlo Ciliberto, Vittorio Murino, and Alessandro Verri. An ode to ode: Learning neural differential equations with black-box ode solvers. *arXiv preprint arXiv:2001.01345*, 2020.

Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Springer, Boston, MA, USA, 2002. ISBN 978-1-4615-0897-7. URL https://link.springer.com/book/10.1007/978-1-4615-0897-7.

Cynthia Rudin. Stop explaining black box models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Jean-Pierre Serre. *A Course in Arithmetic*, volume 7 of *Graduate Texts in Mathematics*. Springer, New York, NY, USA, 1973. ISBN 978-0-387-90071-5. URL https://link.springer.com/book/10.1007/978-1-4612-4380-9.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 9781107057135.

Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Springer, New York, NY, USA, 2nd edition, 2009. ISBN 978-0-387-09493-9. URL https://link.springer.com/book/10.1007/978-0-387-09494-6.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, volume 15, pp. 1929–1958, 2014.

Terence Tao and Van Vu. *Additive Combinatorics*, volume 105 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, UK, 2006. ISBN 978-0-521-89942-4. URL https://www.cambridge.org/core/books/additive-combinatorics/4C7B1B6E4D4D0E4E1F0D1B3A9D3A1F1E.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Yifei Wang et al. Differentiable projection-based learn to optimize in wireless network-part i: Convex constrained (non-)convex programming. *arXiv preprint arXiv:2502.00053*, 2021.

A Training Procedure

Algorithm 1 Training a Diophantine-Elliptic Neural Network (DEC-NN)

Require:
- Training data (X, y)
- Neural network architecture $\mathcal{N}$
- Elliptic curve coefficients $a, b \in \mathbb{Z}$, defining curve $y^2 = x^3 + ax + b$
- Encoding function $\Phi_{\text{enc}} : \mathbb{R} \to \mathbb{Z}^2$, mapping real parameters to elliptic points
- Decoding function $\Phi_{\text{dec}} : \mathbb{Z}^2 \to \mathbb{R}$
- Projection operator Π_E, mapping updates to nearest valid points on the elliptic curve
- Regularization weights $\lambda, \gamma \in \mathbb{R}_{\geq 0}$
- Learning rate η

Ensure: Trained DEC-NN model with parameters constrained to valid elliptic curve points

Initialize network parameters $\theta = \{W, b\}$ with valid points on $\mathcal{D}_E$

Encode all parameters: $\theta_i \mapsto (x_i, y_i) = \Phi_{\text{enc}}(\theta_i)$, with each $(x_i, y_i) \in \mathbb{Z}^2$ satisfying $y_i^2 = x_i^3 + ax_i + b$

for each mini-batch (X_i, y_i) in training set **do**

  Generate adversarial sample (optional): $X_i' = X_i + \delta$ using FGSM or similar

  Forward pass: compute predictions $\hat{y}_i = \mathcal{N}(X_i)$

  Task loss: $\mathcal{L}_{\text{task}} = \texttt{Loss}(y_i, \hat{y}_i)$

  Constraint loss:

$$\mathcal{L}_{\text{EC}} = \sum_j \left(y_j^2 - x_j^3 - ax_j - b\right)^2$$

  Adversarial loss: $\mathcal{L}_{\text{adv}} = \texttt{Loss}(y_i, \mathcal{N}(X_i'))$

  Total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{EC}} + \gamma \cdot \mathcal{L}_{\text{adv}}$$

  Backpropagate $\nabla_\theta \mathcal{L}_{\text{total}}$

  Gradient update: $\theta_i \leftarrow \theta_i - \eta \cdot \nabla_{\theta_i} \mathcal{L}_{\text{total}}$

  Project updated parameters:

$$(x_i', y_i') = \Pi_E(\theta_i), \quad \text{with } y_i'^2 = x_i'^3 + ax_i' + b$$

  Decode to real parameters: $\theta_i \leftarrow \Phi_{\text{dec}}(x_i', y_i')$

end for

Return trained model $\mathcal{N}$ with parameters constrained to elliptic curve