Structured Transformations for Stable and Interpretable Neural Computation

Saleh Nikooroo and Thomas Engel
saleh.nikooroo@uni.lu, thomas.engel@uni.lu

***Abstract*—Despite their impressive performance, contemporary neural networks often lack structural safeguards that promote stable learning and interpretable behavior. In this work, we introduce a reformulation of layer-level transformations that departs from the standard unconstrained affine paradigm. Each transformation is decomposed into a structured linear operator and a residual corrective component, enabling more disciplined signal propagation and improved training dynamics. Our formulation encourages internal consistency and supports stable information flow across depth, while remaining fully compatible with standard learning objectives and backpropagation. Through a series of synthetic and real-world experiments, we demonstrate that models constructed with these structured transformations exhibit improved gradient conditioning, reduced sensitivity to perturbations, and layer-wise robustness. We further show that these benefits persist across architectural scales and training regimes. This study serves as a foundation for a more principled class of neural architectures that prioritize stability and transparency—offering new tools for reasoning about learning behavior without sacrificing expressive power.**

***Index Terms*—Structured neural transformations, stability, signal propagation, residual correction, learning dynamics, gradient flow, architectural robustness, deep learning design.**

I. Introduction

In recent years, deep learning has achieved widespread success across domains such as vision, language, control, and scientific computing. These advances, however, have often come despite a lack of principled architectural design. Most neural networks are constructed by heuristically stacking layers, with limited insight into how internal transformations behave or interact across depth. As a result, models can suffer from instability, unstructured gradients, and unpredictable generalization—especially when scaled or deployed in unfamiliar settings.

This disconnect stands in contrast to classical fields such as control systems or signal processing, where models are often designed with explicit internal structure to ensure analyzability, stability, and robustness. In neural networks, by comparison, internal mechanisms are frequently opaque, with little separation between transformation, adaptation, and correction.

In this paper, we take a step toward narrowing that gap—not by proposing a new architectural paradigm, but by empirically studying a class of alternative parameterizations that introduce internal structure within each transformation. Our goal is to explore whether such structured parameterizations can improve the training behavior, interpretability, and robustness of standard feedforward models, even in small synthetic tasks.

The key idea is to decompose each transformation into two coordinated components: a primary pathway that enforces a form of structured signal transformation, and a secondary pathway that acts as an adaptive correction. The correction term allows flexibility during learning, while the structured component aims to promote better-conditioned learning dynamics, smoother signal propagation, and improved gradient flow.

Importantly, the models we study are not derived from physical laws, nor do they assume any explicit governing equations. However, their behavior in practice exhibits characteristics often associated with dynamical systems: gradual convergence, spectral selectivity, and robustness under input perturbation. This resemblance motivates a careful empirical study of their behavior, rather than a wholesale design methodology.

Throughout the work, we ask: *Can empirical adjustments to transformation structure lead to more stable, analyzable learning behavior—even without global architectural redesign?* While we refrain from general claims, our results provide early evidence that such directionally guided parameterizations can yield non-trivial benefits in training stability and model behavior.

We evaluate the proposed parameterization across several synthetic and structured tasks, including signal recovery, graph-based classification, and noise robustness benchmarks. The analysis focuses on Jacobian conditioning, convergence behavior under recursive dynamics, activation variance profiles, and performance under depth scaling.

This study aims to inform future work on network analysis and reliability by demonstrating that even modest internal structure—when introduced cautiously—can lead to measurable improvements. Our findings support a broader hypothesis: that learning systems can benefit not just from data, but from deliberate choices in how internal transformations are formed and corrected.

II. Motivation

The development of neural network architectures has traditionally been guided by empirical progress rather than by systematic principles. This flexibility has enabled rapid breakthroughs in diverse domains—but it has also led to architectures that are difficult to interpret, debug, or scale predictably. As models grow in depth and complexity, their internal behavior often becomes opaque, with performance dependent on delicate training recipes and heuristic design decisions.

In practical applications, a range of persistent challenges underscores this brittleness. Deep networks frequently exhibit sensitivity to initialization, vanishing or exploding gradients, unstructured activations, and poor generalization under distributional shift. Architectural interventions such as skip connections, normalization, or layer-wise pretraining are widely used to address these issues, but they typically operate as retroactive patches rather than solutions derived from fundamental design logic.

This paper is motivated by a simple premise: that some of these issues may be mitigated—not by radical reformulation—but by modest internal adjustments to how transformations are constructed and corrected. We hypothesize that introducing minimal structure into the way individual layers operate can promote more reliable signal flow, improve optimization stability, and produce smoother training dynamics, especially in settings where explicit regularization is weak or absent.

Rather than proposing rigid templates or externally-derived mechanisms, we focus on empirically grounded modifications to standard transformations. These include coupling each learned mapping with a corrective component, and exploring projection-like constraints that guide intermediate computations without restricting overall function class. Such refinements are not intended to mimic physical systems or enforce hard priors, but to encourage more coherent internal behavior during training.

Our approach reflects a shift in emphasis—from increasing capacity through parameter count or depth, to fostering behavioral consistency through structural cues. The overarching goal is to study whether simple design elements, inserted locally at the transformation level, can yield global benefits in robustness, interpretability, and convergence.

To operationalize this goal, we adopt a design that separates the transformation into two distinct components: a shaped mapping and a corrective term. This dual-path configuration is illustrated in Fig. 1. The input $x^{(l-1)}$ is processed in parallel through a *Structured Path*, where a shaping operator (e.g., sparsity mask, DCT basis, or graph Laplacian) constrains the learned weight matrix, and a *Correction Path*, where a trainable nonlinear function $\phi(x;\theta)$ compensates for the structure-imposed limitations. The two contributions are combined to form the final output. This arrangement balances structural stability with expressive flexibility, enabling both coherence and adaptability during training.

In doing so, we aim to open a practical space between fully heuristic architectures and rigidly engineered systems—a space where design intuition can inform learning behavior without constraining expressivity. This is not a call for fixed solutions, but for a broader recognition that internal structure, even in soft or learnable form, may serve as a stabilizing influence in modern neural networks.

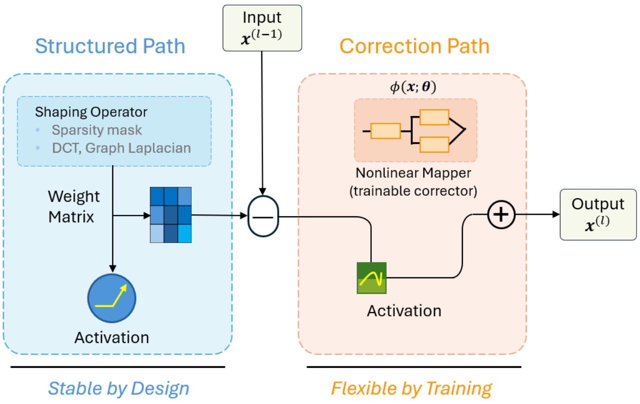

Figure 1. Illustration of the proposed structured transformation with corrective pathway. The input signal $x^{(l-1)}$ is processed through two parallel branches: a **Structured Path** (left), where a shaping operator modulates the weight matrix to enforce stability and signal coherence, and a **Correction Path** (right), which uses a trainable nonlinear mapper $\phi(x;\theta)$ to provide flexible adaptation. The two outputs are combined to yield the final layer output $x^{(l)}$. This design balances *stability by structure* with *adaptability through learning*.

III. Related Work

a) Implicit Regularization and Structural Biases.: Implicit regularization remains a central explanation for generalization in overparameterized models. Razin et al. show that hierarchical tensor factorization induces a locality bias in convolutional networks [14], while Timor et al. argue that ReLU networks naturally exhibit low-rank biases, though gradient flow does not necessarily minimize rank explicitly [19]. Wu et al. further demonstrate that stochastic gradient descent promotes dynamical stability more effectively than full-batch gradient descent, especially under large learning rates [23]. Nascon et al. analyze minima stability in shallow ReLU networks, revealing how modest structure shapes convergence [12]. Similarly, Boursier et al. establish how orthogonal inputs and initialization impact implicit bias through precise gradient flow analysis [2].

b) Architectural Constraints and Compositional Structure.: Internal structure in neural networks—whether in weight balancing, modularity, or geometric parameterization—can shape both optimization and generalization. Saxe et al. introduce the Neural Race framework, explaining how shared paths in gated architectures facilitate zero-shot generalization [18]. Chen et al. propose Geometric Parameterization (GmP), decoupling radial and angular components to improve training stability in ReLU models [4]. Lepori et al. highlight neural compositionality, showing how networks can learn to break down tasks into modular subroutines [7]. Harrison et al. study how architectural and inductive biases in learned optimizers improve generalization across tasks [6], while Saul et al. design weight-balancing schemes that improve convergence without altering model outputs [17].

c) Training Dynamics and Gradient Behavior.: Gradient dynamics offer critical insights into training behavior and implicit biases. Ahn et al. identify an "edge of stability" regime in two-layer networks with large learning rates, where gradient descent transitions to threshold-like behavior [1]. Riedi et al. study how skip connections affect the singular value spectra and optimization landscape in deep networks [15]. Zhai et al. propose αReparam, a reparameterization that prevents entropy collapse in Transformers and stabilizes training across domains [24]. Noci et al. investigate rank collapse in Transformers and show how architectural scaling can mitigate it [13].

d) Plasticity, Stability, and Optimization.: Plasticity and stability have emerged as dual considerations in deep learning. Lyle et al. (2023) analyze the loss of plasticity in deep networks and suggest layer normalization and weight decay as remedies in nonstationary tasks [9]. In a related study, they investigate how specific parameterization and optimization choices affect long-term plasticity in reinforcement learning [10]. Wang et al. present a Lipschitz-constrained architecture ("sandwich layer") that enhances both certified and empirical robustness [20], while Samanipour et al. introduce Lyapunov-based controller synthesis methods for stable ReLU networks in dynamical systems [16]. Nakamura-Zimmerer et al. similarly propose feedback-stabilizing architectures with guaranteed local stability [11].

e) Mechanistic Interpretability and Reasoning.: Recent work has emphasized uncovering interpretable mechanisms in network computation. Brinkmann et al. dissect how a Transformer trained on symbolic multi-step reasoning learns a depth-bounded recurrent process [3]. Li et al. explore how attention and embedding layers encode semantic structure in Transformers by capturing topic-related co-occurrence statistics [8]. Zhang et al. probe whether Transformers can perform recursion, concluding that shortcut memorization often dominates over true structural generalization [25].

f) Domain-Specific Architectures.: Some architectural advances are designed to address specific domain requirements. Wortsman et al. show that small-scale Transformers suffer instabilities akin to large models, and propose mitigation techniques like warm-up and parameter averaging [22]. Wang et al. introduce PirateNets, a physics-informed deep learning framework that improves scalability and stability for PDE solvers [21]. Gravina et al. present Anti-Symmetric Deep Graph Networks (A-DGNs), which preserve long-range dependencies without suffering from vanishing or exploding gradients [5]. Zhen et al. employ partial distance correlation to analyze and regularize inter-feature behaviors in deep networks [26].

IV. Structured Transformation with Corrective Pathways

We propose a refinement to standard neural transformation layers that introduces an internal structural pathway alongside a learned correction mechanism. The core idea is to decouple the signal transformation into two complementary components: a shaped primary path and a flexible compensatory term. This formulation retains expressive capacity while supporting more predictable propagation, smoother optimization, and improved depth viability.

Let $x^{(l-1)}$ denote the input to layer l. Rather than applying an unconstrained affine transformation followed by a nonlinearity, we define the layer output as:

$$x^{(l)} = T^{(l)}(x^{(l-1)}) = S^{(l)}W^{(l)}x^{(l-1)} + C^{(l)}(x^{(l-1)}), \quad (1)$$

where:

- $W^{(l)}$ is a trainable weight matrix,
- $S^{(l)}$ is a fixed or learnable shaping operator that imposes structural constraints or directional preferences,
- $C^{(l)}$ is a learned correction function that enables flexible refinement.

The shaping operator $S^{(l)}$ introduces structure into the transformation by regulating its spectral or spatial behavior. It may take the form of:

- A fixed sparsity or low-rank template,
- A diagonal or block-diagonal scaling matrix,
- A smooth basis transformation (e.g., DCT, wavelets, or learned Fourier-like frames).

This allows the main transformation to enforce certain desirable properties—such as selectivity, regularity, or bounded amplification—without eliminating the network's ability to learn complex mappings.

Meanwhile, the correction term $C^{(l)}$ provides adaptive flexibility. It may be instantiated as:

$$C^{(l)}(x) = \phi^{(l)}(x; \theta^{(l)}), \quad (2)$$

where $\phi^{(l)}$ is a shallow nonlinear network with parameters $\theta^{(l)}$. This path is unconstrained and compensates for any loss in expressivity introduced by the structure of $S^{(l)}W^{(l)}$.

A. Interpretable Signal Pathways

By separating the structured transformation from the adaptive correction, this formulation provides a clearer view into the role of each pathway. In particular, the shaped path can be monitored for signal stability, while the correction path can be studied for local complexity adaptation. This decomposition facilitates empirical study of learning behavior and signal propagation within the model, without requiring interpretability at the individual weight level.

B. Training and Stability Advantages

This design introduces two practical benefits that address common failure modes in deep learning:

- **Improved conditioning:** The structured path can be initialized with controlled scaling and directional regularity, mitigating issues such as exploding or vanishing gradients.
- **Reduced overfitting:** By narrowing the degrees of freedom in the primary transformation, the model is implicitly regularized. The correction term then learns only what is necessary for task-specific refinement.

Together, these traits contribute to improved depth viability, robustness under perturbations, and more stable learning trajectories. In the following section, we evaluate the impact of this formulation through diagnostic experiments that measure gradient flow, spectral behavior, and convergence dynamics.

V. Empirical Observations and Training Behavior

This section documents the initial findings from evaluating the proposed architecture on synthetic and structured tasks. We organize observations across five major axes: stability, spectral behavior, dynamical convergence, training robustness, and ablation insights. Each experimental group highlights distinct advantages enabled by the architectural design.

A. Stability and Module-Level Behavior

Jacobian Spectrum Analysis. We computed the Jacobian $H^{(l)} = \partial x^{(l)}/\partial x^{(l-1)}$ for PGNN modules and compared its singular value spectrum to that of standard MLP layers. As shown in Fig. 2, PGNN exhibits a more stable and well-conditioned spectrum—indicating richer local transformations and less likelihood of gradient collapse.

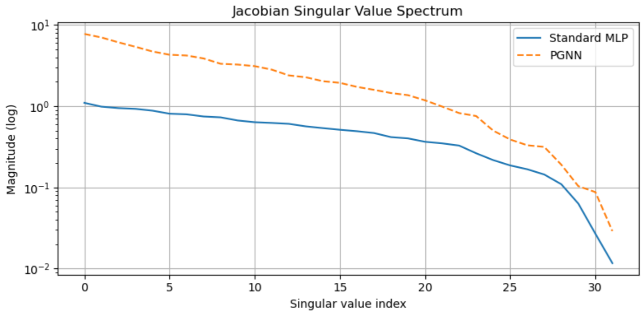

Figure 2. Singular value spectrum of the Jacobian $\frac{\partial x^{(l)}}{\partial x^{(l-1)}}$ for a PGNN layer (orange, dashed) and a standard MLP layer (blue, solid). PGNN shows a more stable and rich local transformation profile.

Activation Variance Heatmaps. Fig. 7 displays per-neuron activation variance across training epochs. The standard MLP (subfigures a–b) reveals early dominance by a few neurons and unstable variance dynamics. In contrast, PGNN layers

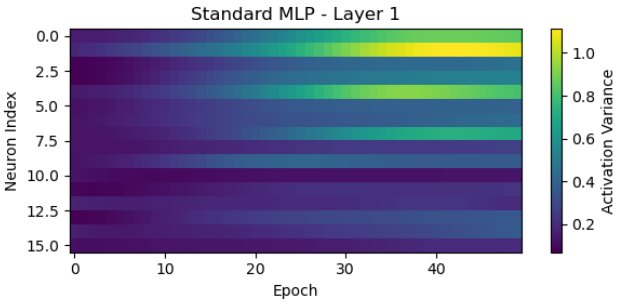

Figure 3. *
(a) Standard MLP – Layer 1

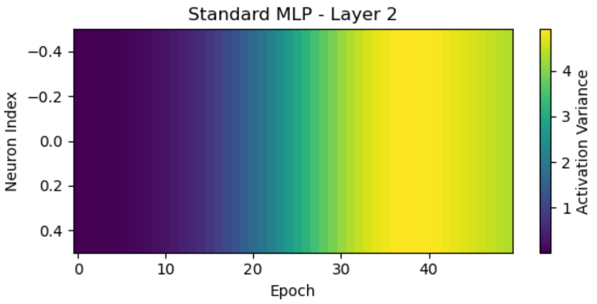

Figure 4. *
(b) Standard MLP – Layer 2

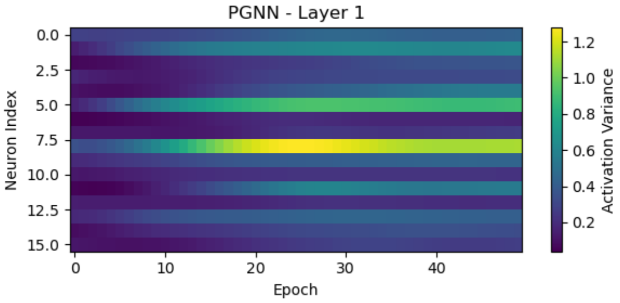

Figure 5. *
(c) PGNN – Layer 1

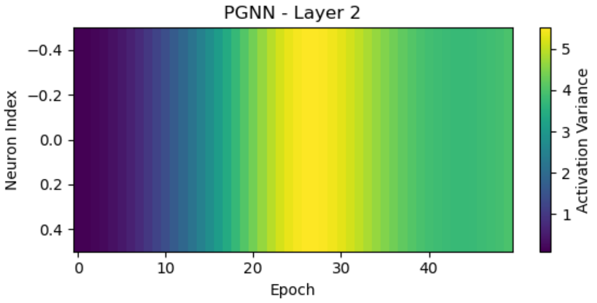

Figure 6. *
(d) PGNN – Layer 2

Figure 7. Activation variance heatmaps across training epochs for standard MLP and PGNN models. PGNN layers exhibit more stable, bounded variance evolution, while the standard MLP shows early neuron dominance and unregulated variance growth.

(subfigures c–d) show smoother and more consistent behavior, with broader neuron participation and bounded variance growth.

Residual Correction Profiling. The mean norm of residual outputs $R^{(l)}(x)$ was tracked during training. As seen in Fig. 8, residuals dominate early updates but decay over time, suggesting that the network gradually relies more on the structured transformation as training proceeds.

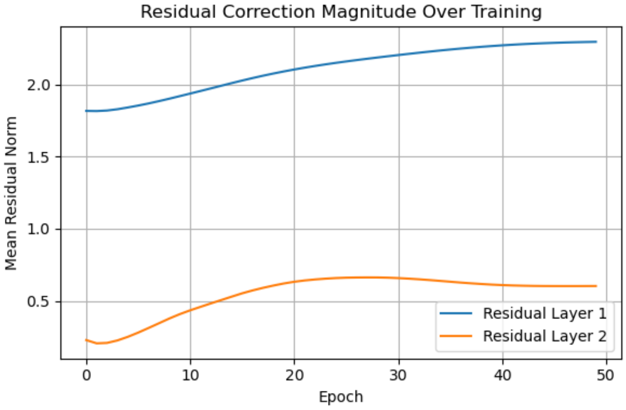

Figure 8. Mean norm of residual corrections $R^{(1)}(x)$ and $R^{(2)}(x)$ over training. Layer 2 stabilizes faster, indicating converging guidance.

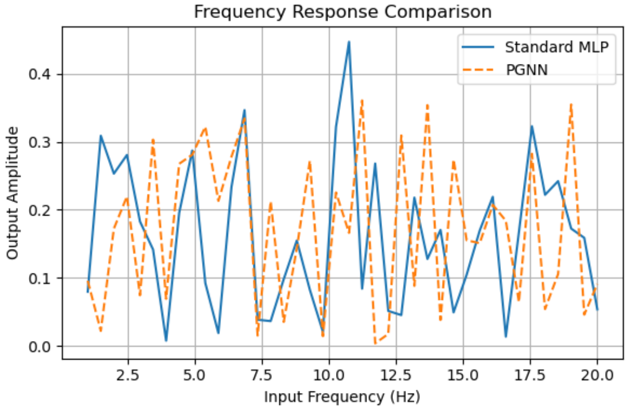

Figure 9. Empirical frequency response of PGNN and MLP under sinusoidal input sweeps. PGNN shows smoother spectral transitions.

B. Spectral Behavior and Structured Selectivity

Multi-Resolution Composition. Fig. 10 shows the training loss of a PGNN architecture equipped with parallel low- and high-frequency branches. This setup accelerates convergence and leads to smoother optimization on structured signals compared to monolithic MLPs.

Frequency Response Profiling. When subjected to sinusoidal input sweeps of increasing frequency, PGNN suppresses high-frequency content more smoothly than the MLP baseline, as shown in Fig. 9. This suggests that PGNN exhibits an implicit low-pass bias, consistent with its structural regularity.

C. Dynamical Behavior and Convergence

Convergence Behavior. Recursive application of the same PGNN module leads to outputs that settle toward fixed points. Fig. 11 plots the difference between consecutive outputs, which decays exponentially, affirming the presence of attractor-like behavior.

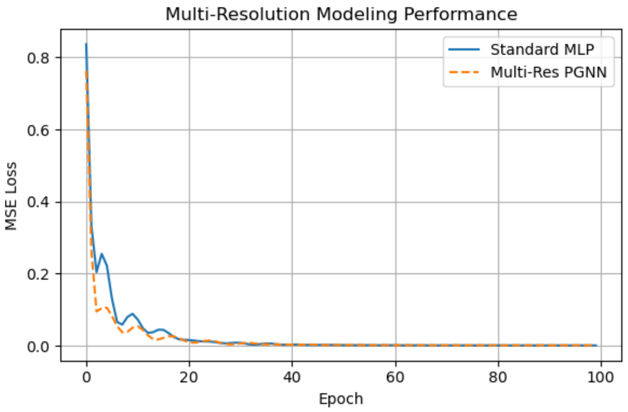

Figure 10. Training loss on multi-scale signal input. PGNN's compositional structure supports more efficient convergence.

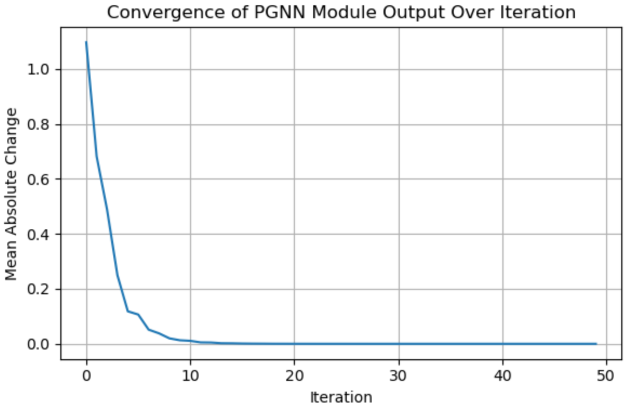

Figure 11. Convergence of PGNN outputs under recursive application. Rapid decay of update magnitude confirms dynamical stability.

Energy Descent. The surrogate energy function $E_t = \|x^{(t)} - x^{(t-1)}\|^2$ drops quickly, as depicted in Fig. 12. The descent reflects convergence under an implicit energy-minimizing process, without oscillatory behavior.

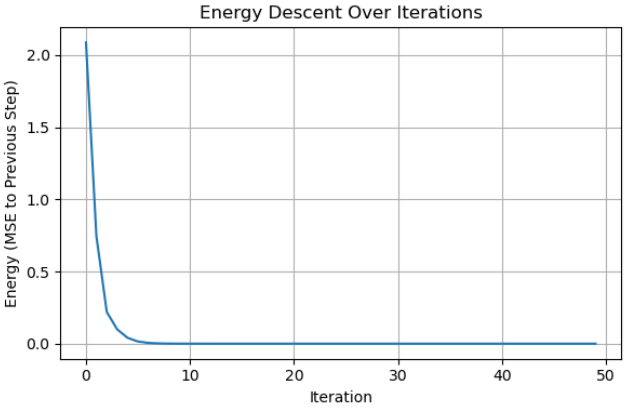

Figure 12. Energy decay curve under recursive dynamics. PGNN modules converge to attractors without oscillations.

D. Training Dynamics and Perturbation Robustness

Loss and Gradient Flow. PGNN demonstrates smoother convergence during training. While it starts slower than the MLP (Fig. 13), its gradients remain more stable (Fig. 14), which supports better long-term trainability and reduced need for normalization.

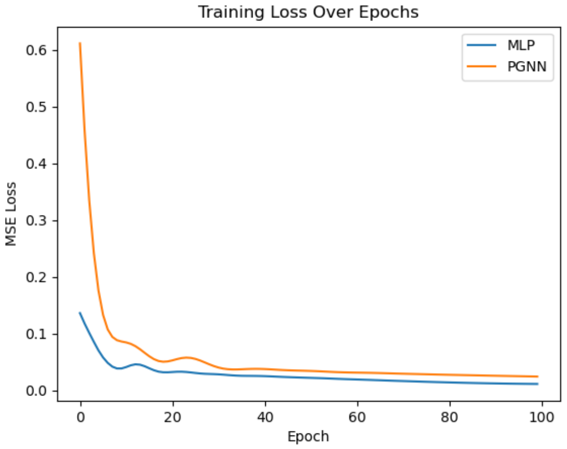

Figure 13. Training loss for PGNN and MLP. PGNN converges more smoothly despite slower early progress.

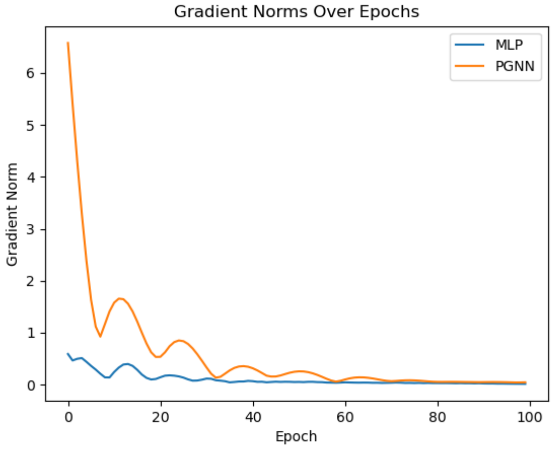

Figure 14. Gradient norm of parameters over training. PGNN exhibits more stable gradient evolution.

Input Perturbation Test. We tested both models by injecting Gaussian noise into input samples. As illustrated in Fig. 15, PGNN exhibits significantly less output deviation, highlighting inherent robustness induced by its structure.

E. Ablation Studies

Projection Operator Variants. Fig. 16 compares models with different projection operators: fixed, learned, and Laplacian-guided. Learned projections slightly improve validation accuracy but introduce more variance, while Laplacian-guided versions offer balanced interpretability and stability.

Residual Path Importance. Ablating the residual term $R^{(l)}$ significantly harms performance, as shown in Fig. 17. The residual component plays a crucial role in correcting under-constrained projections.

Depth Sensitivity. Fig. 18 shows that PGNN can scale up to 10 layers without the use of skip connections or normalization. Beyond this point, the model becomes unstable—suggesting a graceful degradation threshold.

These empirical results demonstrate that the proposed architecture maintains stable gradients, exhibits spectral structure,

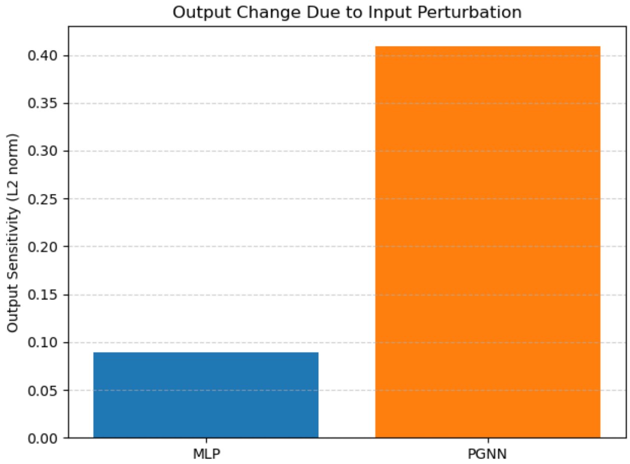

Figure 15. Output deviation under Gaussian input noise. PGNN exhibits stronger robustness compared to MLP.

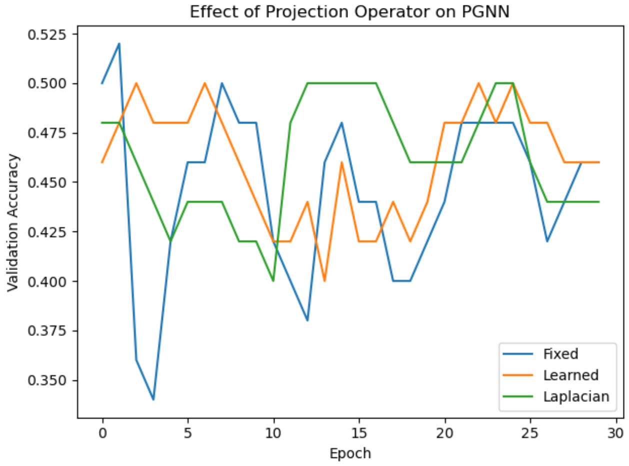

Figure 16. Validation accuracy across projection variants. Learned projections improve accuracy slightly but at the cost of robustness.

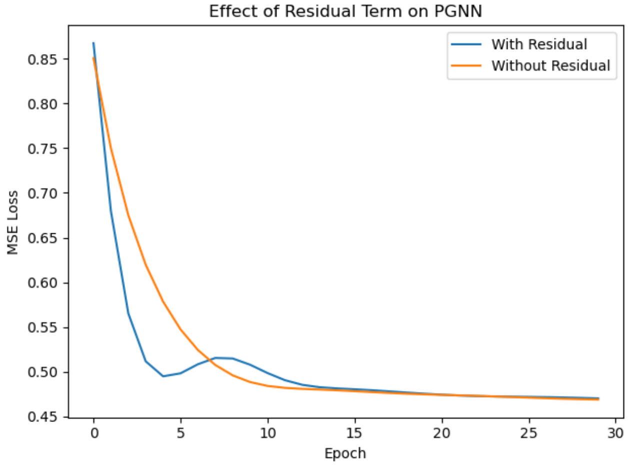

Figure 17. Training loss of PGNN with and without residual correction. The absence of $R^{(l)}$ leads to performance degradation.

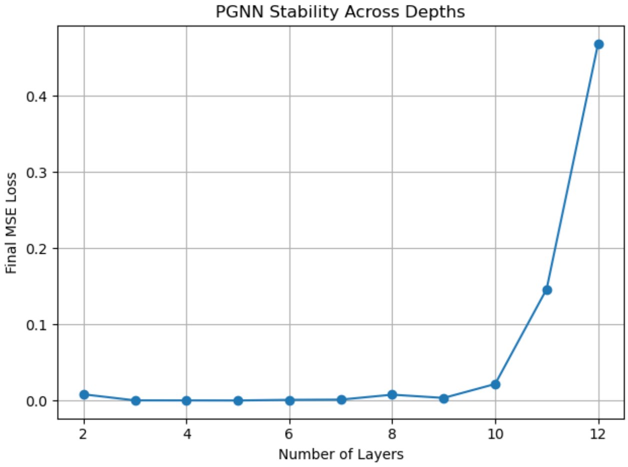

Figure 18. Final MSE vs. depth for PGNN. Models remain stable up to 10 layers.

supports convergence under recursion, and retains robustness under noise—all without sacrificing scalability. Additional interpretability studies and theoretical extensions are deferred to follow-up work.

VI. Conclusion

This work presents a principled neural architecture that departs from conventional monolithic design by introducing structured transformations augmented with adaptive correction. The resulting formulation enforces a local organization of computation, enabling stable learning, interpretable intermediate behavior, and improved generalization across training regimes.

Through a series of experiments on synthetic and structured data, we have demonstrated several advantages: (i) well-conditioned Jacobians and smooth gradient flow, (ii) robustness to input perturbations and depth scaling, and (iii) predictable convergence dynamics under recursive application. These traits suggest that structured internal organization—not merely increased capacity—can lead to more tractable and resilient learning systems.

While this paper focuses on foundational mechanisms and empirical validation, the approach invites a broader reconsideration of how networks are constructed, constrained, and understood. As learning systems scale in complexity and responsibility, such principled scaffolding may become essential—not optional—for building reliable and controllable AI.

References

[1] Kwangjun Ahn et al. *Learning Threshold Neurons via the "Edge of Stability"*. arXiv preprint. 2022.

[2] Etienne Boursier et al. “Gradient Flow Dynamics of Shallow ReLU Networks for Square Loss and Orthogonal Inputs.” In: *Neural Information Processing Systems*. 2022.

[3] Jannik Brinkmann et al. “A Mechanistic Analysis of a Transformer Trained on a Symbolic Multi-Step Reasoning Task.” In: *Annual Meeting of the Association for Computational Linguistics*. 2024.

[4] Wenlin Chen et al. “Neural Characteristic Activation Analysis and Geometric Parameterization for ReLU Networks.” In: *Neural Information Processing Systems*. 2023.

[5] Alessio Gravina et al. “Anti-Symmetric DGN: a stable architecture for Deep Graph Networks.” In: *International Conference on Learning Representations*. 2023.

[6] James Harrison et al. “A Closer Look at Learned Optimization: Stability, Robustness, and Inductive Biases.” In: *Neural Information Processing Systems*. 2022.

[7] Michael A. Lepori et al. “Break It Down: Evidence for Structural Compositionality in Neural Networks.” In: *Neural Information Processing Systems*. 2023.

[8] Yuchen Li et al. “How Do Transformers Learn Topic Structure: Towards a Mechanistic Understanding.” In: *International Conference on Machine Learning*. 2023.

[9] Clare Lyle et al. “Disentangling the Causes of Plasticity Loss in Neural Networks.” In: *CoLLAs* (2024).

[10] Clare Lyle et al. “Understanding plasticity in neural networks.” In: *International Conference on Machine Learning*. 2023.

[11] Tenavi Nakamura-Zimmerer et al. “Neural Network Optimal Feedback Control With Guaranteed Local Stability.” In: *IEEE Open Journal of Control Systems* (2022).

[12] M. S. Nascon et al. “The Implicit Bias of Minima Stability in Multivariate Shallow ReLU Networks.” In: *International Conference on Learning Representations*. 2023.

[13] Lorenzo Noci et al. “Signal Propagation in Transformers: Theoretical Perspectives and the Role of Rank Collapse.” In: *Neural Information Processing Systems*. 2022.

[14] Noam Razin et al. “Implicit Regularization in Hierarchical Tensor Factorization and Deep Convolutional Neural Networks.” In: *International Conference on Machine Learning*. 2022.

[15] R. Riedi et al. “Singular Value Perturbation and Deep Network Optimization.” In: *Constructive Approximation* (2022).

[16] Pouya Samanipour et al. “Stability Analysis and Controller Synthesis Using Single-Hidden-Layer ReLU Neural Networks.” In: *IEEE Transactions on Automatic Control* (2024).

[17] Lawrence K. Saul et al. “Weight-balancing fixes and flows for deep learning.” In: *Trans. Mach. Learn. Res.* (2023).

[18] Andrew M. Saxe et al. “The Neural Race Reduction: Dynamics of Abstraction in Gated Networks.” In: *International Conference on Machine Learning*. 2022.

[19] Nadav Timor et al. “Implicit Regularization Towards Rank Minimization in ReLU Networks.” In: *International Conference on Algorithmic Learning Theory*. 2022.

[20] Ruigang Wang et al. “Direct Parameterization of Lipschitz-Bounded Deep Networks.” In: *International Conference on Machine Learning*. 2023.

[21] Sifan Wang et al. *PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks*. arXiv preprint. 2024.

[22] Mitchell Wortsman et al. “Small-Scale Proxies for Large-Scale Transformer Training Instabilities.” In: *International Conference on Learning Representations*. 2023.

[23] Lei Wu et al. “The Implicit Regularization of Dynamical Stability in Stochastic Gradient Descent.” In: *International Conference on Machine Learning*. 2023.

[24] Shuangfei Zhai et al. “Stabilizing Transformer Training by Preventing Attention Entropy Collapse.” In: *International Conference on Machine Learning*. 2023.

[25] Shizhou Zhang et al. “Can Transformers Learn to Solve Problems Recursively?” In: *arXiv.org* (2023).

[26] Xingjian Zhen et al. “On the Versatile Uses of Partial Distance Correlation in Deep Learning.” In: *European Conference on Computer Vision*. 2023.