STABLE AND SCALABLE DEEP PREDICTIVE CODING NETWORKS WITH META PREDICTION ERRORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predictive Coding Networks (PCNs) offer a biologically inspired alternative to conventional deep neural networks. However, their scalability is hindered by severe training instabilities that intensify with network depth. Through dynamical mean-field analyses, we identify two fundamental pathologies that impede deep PCN training: (1) prediction error (PE) imbalance that leads to uneven learning across layers, characterized by error concentration at network boundaries; and (2) exploding and vanishing prediction errors (EVPE) sensitive to weight variance. To address these challenges, we propose Meta-PCN, a unified framework that incorporates two synergistic components: (1) loss based on meta-prediction errors, which minimizes PEs of PEs to linearize the nonlinear inference dynamics; and (2) weight regularization that combines normalization and clipping to regulate weight variance and mitigate EVPE. Extensive experimental validation on CIFAR-10/100 and TinyImageNet demonstrates that Meta-PCN statistically significant improvements over conventional PCN and backpropagation across most architectures, while maintaining biological plausibility.

1 INTRODUCTION

Predictive coding (PC) represents a theoretical framework for understanding cortical information processing. It encompasses fundamental functions such as learning, prediction, encoding, and memorization. As neural architectures, PCNs implement this framework and offer a compelling alternative to backpropagation-based learning. PCNs are grounded in PC theory (Srinivasan et al., 1982; Mumford, 1992; Rao & Ballard, 1999; Friston, 2005) and formalized through the free-energy framework (Friston, 2010; Bogacz, 2017; Bastos et al., 2012). They employ purely local learning rules that respect biological constraints while enabling massive parallelization (Millidge et al., 2022b; Salvatori et al., 2022; Song et al., 2020). This positions them as promising candidates for neuromorphic computing (Schuman et al., 2017; Sacramento et al., 2018). However, PCNs face a critical limitation. As network depth increases, their training becomes progressively unstable. This creates a formidable barrier to scalability (Millidge et al., 2022b). The underlying mechanisms driving this instability have remained poorly understood. This hinders the practical deployment of deep PCNs in complex applications.

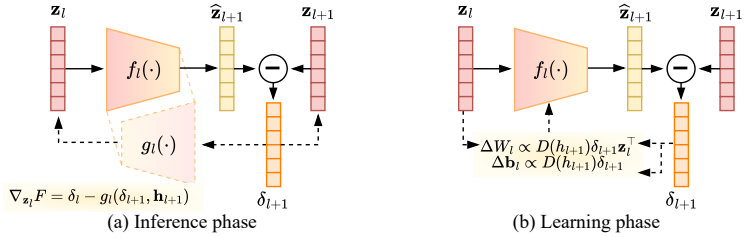
To address these fundamental challenges, we conduct a rigorous analysis of PCN inference dynamics using dynamical mean-field theory (Sompolinsky et al., 1988; Poole et al., 2016; Schoenholz et al., 2017). Our theoretical investigation (detailed in Section 3) reveals two distinct yet interconnected pathologies that impede deep PCN scalability:
(1) PE Imbalance: Errors concentrate in boundary layers (input/output) while vanishing in intermediate layers. This creates a characteristic imbalanced distribution. This results in gradient starvation in mid-layers and prevents effective learning.
(2) EVPE: Exponential growth and decay patterns emerge in latent states and PEs during inference. These dynamics are controlled by temporal scaling factors that depend critically on the variance of weight. This leads to training instabilities (Bengio et al., 1994; Hochreiter, 1998; Pascanu et al., 2012; Arjovsky et al., 2016).

To systematically address these pathologies, we propose Meta-PCN (Section 4). This unified framework incorporates two complementary solutions operating synergistically. First, we introduce a

novel objective based on meta-prediction error. This objective linearizes the nonlinear equilibrium system by minimizing PEs of PEs. Second, we implement a normalization of the variance of weights. This controls variance and suppresses exponential behaviors. Through these two complementary solutions, our framework enables practical and stable training of deep PCNs. We achieve substantial improvements in inference stability, convergence speed, and classification performance. These improvements are obtained while preserving the biological plausibility (See Appendix B) that makes PCNs attractive for neuromorphic applications.

Extensive experimental validation on CIFAR-10, CIFAR-100, and TinyImageNet demonstrates that Meta-PCN achieves remarkable performance improvements, with 28-78 % gains over conventional PCNs across all tested architectures. Notably, Meta-PCN consistently outperforms backpropagation in most configurations, achieving statistically significant improvements ranging from 0.61% to 1.73% on CIFAR-10 while maintaining biological plausibility constraints. These results establish Meta-PCN as a viable framework for scaling PC to practical deep learning applications without sacrificing neuromorphic computing advantages.

2 Predictive Coding Networks



(a) Inference phase (b) Learning phase

Figure 1: Inference and learning phases of local PC modules. (a) Inference phase: the PEs ($\boldsymbol{\delta}_{l+1}$) are calculated and the latent states ($\mathbf{z}_l$) are updated. This process is repeated until it reaches the final inference step T. (b) Learning phase: the weight and bias parameters (W_l and $\mathbf{b}_l$) are updated.

PCN Architecture and Objective. PC proposes that the brain continuously generates predictions of the external environment and refines internal representations by minimizing PEs. PCNs implement this principle by connecting local PC modules in a hierarchical chain structure, as illustrated in Figure 1. The forward pass generates predictions for subsequent layers, while the backward pass minimizes local PEs (Whittington & Bogacz, 2017; Millidge et al., 2022a). Each layer l with the latent state $\mathbf{z}_l \in \mathbb{R}^{N_l}$ produces predictions via $\hat{\mathbf{z}}_{l+1} = f_l(\mathbf{z}_l) = \phi(W_l\mathbf{z}_l + \mathbf{b}_l)$, where $f_l(\cdot)$ represents the forward prediction function, W_l denotes the weight matrix, $\mathbf{b}_l$ is the bias vector, and $\phi(\cdot)$ is the activation function. The main goal is to minimize PEs for each layer, $\boldsymbol{\delta}_l = \mathbf{z}_l - \hat{\mathbf{z}}_l$, quantified using the notion of free energy:

$$\mathcal{F} = \frac{1}{2}\sum_{l=2}^{L} \|\boldsymbol{\delta}_l\|_2^2. \tag{1}$$

This optimization is subject to the boundary conditions $\mathbf{z}_1 = \mathbf{x}$ (input) and $\mathbf{z}_L = \mathbf{y}$ (target).

Inference and Learning Dynamics. PCNs employ a dual optimization process that alternates between inference and learning phases. During the inference phase, latent states evolve according to the fixed-point iteration: $\mathbf{z}^{t+1} = \mathbf{z}^t - \eta \cdot \nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}^t)$, where the global latent state vector $\mathbf{z}$ is the concatenation of all layer-wise latent states and η denotes the inference rate. The gradient computation involves forward and backward operations that facilitate bidirectional information flow throughout the network. The forward prediction function $f_l(\mathbf{z}_l) = \phi(W_l\mathbf{z}_l + \mathbf{b}_l)$ propagates information from lower to higher layers, while the backward operation $g_l(\boldsymbol{\delta}_{l+1}, \mathbf{h}_{l+1}) = W_l^\top D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}$ transmits error signals from higher to lower layers, where $D(\mathbf{h}_{l+1}) = \text{diag}(\phi'(\mathbf{h}_{l+1}))$ represents the diagonal matrix of activation derivatives. The gradient naturally decomposes into bottom-up error and top-down feedback components: $\nabla_{\mathbf{z}_l}\mathcal{F} = \boldsymbol{\delta}_l - g_l(\boldsymbol{\delta}_{l+1}, \mathbf{h}_{l+1})$. During the learning phase, parameters are updated according to $\partial_{W_l}\mathcal{F} = -D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}\mathbf{z}_l^\top$ and $\nabla_{\mathbf{b}_l}\mathcal{F} = -D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}$.

Challenges in Achieving Equilibrium. The equilibrium condition $\nabla_{\mathbf{z}}\mathcal{F} = 0$ yields $\boldsymbol{\delta}_l = g_l(\boldsymbol{\delta}_{l+1}, \mathbf{h}_{l+1})$, which resembles a standard error backpropagation algorithm (Whittington & Bogacz, 2017; Millidge et al., 2022a). However, achieving equilibrium during inference presents substantial computational challenges. The complex interplay between inference and learning dynamics

makes understanding deep PCN behavior particularly challenging, motivating systematic analyses of deep PCN pathologies (Section 3).

3 Deep PCN Instability: Uncovering Fundamental Pathologies

This section provides a comprehensive investigation into the internal inference dynamics in deep PCNs. We employ dynamical mean-field theory to mathematically characterize the underlying mechanisms driving these instabilities (Section 3.1). Our analysis reveals that deep PCNs suffer from three fundamental pathologies that severely impede practical scalability. These pathologies are PE imbalance (Section 3.2) and EVPE (Section 3.3).

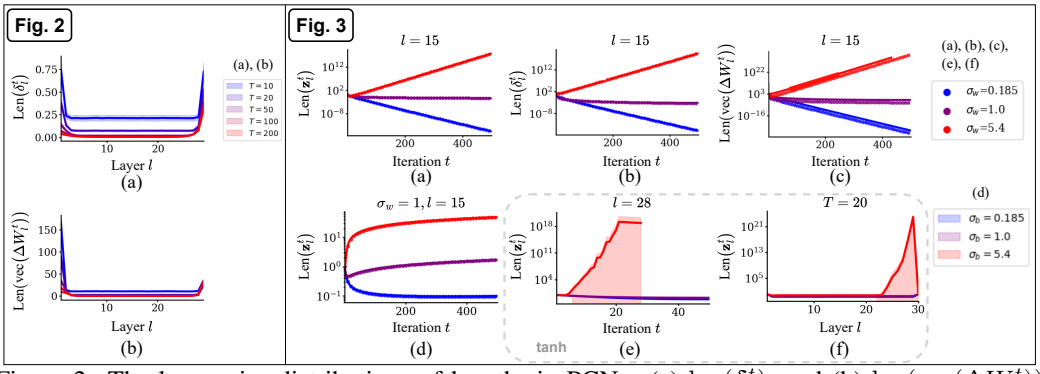


Figure 2: The layer-wise distributions of lengths in PCNs: (a) $\mathrm{len}(\boldsymbol{\delta}_l^t)$, and (b) $\mathrm{len}(\mathrm{vec}(\Delta W_l^t))$. *Setting:* We set $L = 30$, the terminal inference step is $T = 200$, and the latent dimension of each layer is set to 100. The inference rate is set to $\eta = 0.05$. σ_w and σ_b are set to 1 and 0.1, respectively.

Figure 3: Dynamics of $\mathrm{len}(\mathbf{z}_l^t)$, $\mathrm{len}(\boldsymbol{\delta}_l^t)$, and $\mathrm{len}(\mathrm{vec}(\Delta W_l^t))$. Dotted lines represent theoretical predictions, while solid lines correspond to empirical observations from linear PCN experiments. *Subfigures:* Dynamics of (a) $\mathrm{len}(\mathbf{z}_l^t)$, (b) $\mathrm{len}(\boldsymbol{\delta}_l^t)$, and (c)$\mathrm{len}(\mathrm{vec}(\Delta W_l^t))$ of linear PCNs, and (e) $\mathrm{len}(\mathbf{z}_l^t)$ of nonlinear (`tanh`) PCNs across $\sigma_w \in \{0.185, 1.0, 5.4\}$. (d) Dynamics of $\mathrm{len}(\mathbf{z}_l^t)$ across $\sigma_b \in \{0.185, 1.0, 5.4\}$. (f) Layer-wise distributions of $\mathrm{len}(\mathbf{z}_l^t)$ of nonlinear (`tanh`) PCNs at $t = 20$. Further analysis for nonlinear cases can be found in Appendix H. *Setting:* In (a)-(d), $l = 15$, while in (c), $l = 28$. The other settings are the same as in Figure 2.

3.1 Analytical Methodology: Dynamical Mean-Field Theory Approach

Motivation. PCNs present unique analytical challenges due to their dynamically evolving latent states during the inference phase. This makes it difficult to characterize these dynamics using conventional analytical methodologies. To address this challenge, we develop an analytical framework based on the dynamical mean-field approach (Poole et al., 2016; Schoenholz et al., 2017). This framework provides mathematical tractability for the complex inference dynamics of PCNs.

Length-based Statistical Framework. Following the methodology established by Poole et al. (2016), we construct a statistical framework for analyzing PCN dynamics through length-based measures. We define the length of a vector $\mathbf{x}$ as the mean squared element: $\mathrm{len}(\mathbf{x}) = \langle x_i^2 \rangle = \frac{1}{N}\sum_{i=1}^{N} x_i^2$, where x_i denotes the i-th element of the vector $\mathbf{x}$, and N represents the vector dimension. In this analysis, weights and biases are assumed to be drawn i.i.d. as $w_{i,j}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{N_l})$ and $b_i^l \sim \mathcal{N}(0, \sigma_b^2)$. Our analysis systematically tracks three fundamental length variables. These are the length of latent states $\mathrm{len}(\mathbf{z}_l^t)$, the length of PEs $\mathrm{len}(\boldsymbol{\delta}_l^t)$, and the length of weight updates $\mathrm{len}(\mathrm{vec}(\Delta W_l^t))$ at layer l and time step t, where $\mathrm{vec}(\cdot)$ denotes the vectorization operator that transforms matrix columns into a single vector and the amount of the weight update $\mathrm{vec}(\Delta W_l^t) \propto \nabla_{\mathrm{vec}(W_l)}\mathcal{F}$. To mathematically characterize PCN dynamics, we introduce three interaction matrices that capture the intricate relationships between network components. The Latent Self-Interaction Matrix P^t captures inter-layer latent state interactions through $P_{l+k,l}^t = \frac{1}{N}{\mathbf{z}_{l+k}^t}^\top M_{l+k-1:l}\mathbf{z}_l^t$. Here $M_{l+k-1:l} = M_{l+k-1}M_{l+k-2}\cdots M_l$ represents the product of normalized weight matrices and $M_l = \frac{1}{\sigma_w}W_l$ denotes the normalized weight matrix. The framework additionally incorporates

bias-latent interactions via $B^t_{l,l-k} = \frac{1}{N}\mathbf{b}_{l-1}{}^\top M_{l:l-k}\mathbf{z}^t_{l-k}$. Bias self-interactions are characterized through the constant matrix $\Gamma = \sigma_b^2 I$. The interaction matrices enable the systematic derivation of length dynamics for each network component. The temporal evolution is governed by the interaction matrices themselves, which evolve according to the dynamics of the latent state updates: $\mathbf{z}^{t+1}_l = \nu M_{l-1}\mathbf{z}^t_{l-1} + \kappa \mathbf{z}^t_l + \nu M_l^\top \mathbf{z}^t_{l+1} + \eta \mathbf{b}_{l-1} - \nu M_l^\top \mathbf{b}_l$. Here $\nu = \eta\sigma_w$ governs the information propagation rate and $\kappa = 1 - \eta(1 + \sigma_w^2)$ controls self-interaction dynamics. The latent state length, $\mathrm{len}(\mathbf{z}^t_l) = P^t_{l,l}$, directly corresponds to the diagonal elements of the inter-layer latent state interactions matrix P. The PE length emerges from the interaction between these components according to: $\mathrm{len}(\boldsymbol{\delta}^t_l) = \mathbf{c}_q^\top P^t_{l-1:l,l-1:l}\mathbf{c}_q - 2B^t_{l,l-1:l}\mathbf{c}_q + \Gamma^t_{l-1,l-1}$, where $\mathbf{c}_q = [-\sigma_w, 1]^\top$. The weight update length follows directly as $\mathrm{len}(\mathrm{vec}(\Delta W^t_l)) = \mathrm{len}(\boldsymbol{\delta}^t_{l+1}) \cdot \mathrm{len}(\mathbf{z}^t_l)$. The complete mathematical derivations are presented in Appendix D, including the evolution equations along with a detailed analysis of length dynamics.

3.2 Problem 1: Imbalanced Prediction Errors

Empirical Observations. The first fundamental pathology of deep PCNs manifests as PE imbalance across network layers. As shown in Figure 2, PEs exhibit a characteristic imbalanced distribution. PEs decrease substantially in intermediate layers during inference. This results in significant error concentration at the boundary layers, while it disappears in intermediate layers. This phenomenon arises from the fundamental constraints on information propagation in PCNs. Information propagates from the k-layer away at a rate proportional to $\mathcal{O}(\nu^k)$, where $\nu = \eta\sigma_w$ as established in our length-based framework. With typical values of $\nu \leq 1.0$ (varying with inference rate and weight variance), this exponential decay causes inference to terminate prematurely before information effectively reaches the middle layers.

Theoretical Analysis: Imbalanced Error Distribution and Gradient Starvation. The mathematical foundation of this imbalance can be understood through the effects of boundary conditions and the dynamics of error propagation. With the input layer clamped as $\mathbf{z}_1 = \mathbf{x}$ during inference, the PE $\boldsymbol{\delta}_2 = \mathbf{z}_2 - \hat{\mathbf{z}}_2 = \mathbf{z}_2 - \phi(W_1\mathbf{x} + \mathbf{b}_1)$ contains a forcing term. This term continuously reintroduces residual errors near the input boundary. Similarly, clamping the output layer as $\mathbf{z}_L = \mathbf{y}$ yields $\boldsymbol{\delta}_L = \mathbf{y} - \phi(W_{L-1}\mathbf{z}_{L-1} + \mathbf{b}_{L-1})$. This typically remains nonzero and acts as a persistent error source at the output boundary. The equilibrium condition $\boldsymbol{\delta}_l = W_l^\top D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}$ implies the spectral norm bound $\|\boldsymbol{\delta}_l\| \leq \|W_l\|_2 \|\phi'(\mathbf{h}_{l+1})\|_\infty \|\boldsymbol{\delta}_{l+1}\|$. The total bound across layers yields the product bound $\|\boldsymbol{\delta}_l\| \leq \left(\prod_{j=l}^{L-1} \|W_j\|_2 \|\phi'(\mathbf{h}_{j+1})\|_\infty\right) \|\boldsymbol{\delta}_L\|$. When the terms in this product are less than unity, PEs decay geometrically as they propagate downward from the output boundary. Combined with the fixed at the input boundary term, this creates the characteristic U-shaped error profile.

The PE Dilemma: Error Minimization Impedes Training. As PEs distribute unevenly across layers, certain layers experience $\boldsymbol{\delta}_{l+1} \approx 0$. Meanwhile, the per-layer weight derivatives follow $\partial_{W_l}\mathcal{F} = -D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}\mathbf{z}_l^\top$ for $2 \leq l \leq L-1$. Consequently, when $\boldsymbol{\delta}_{l+1} \approx 0$, the gradient $\partial_{W_l}\mathcal{F} \approx 0$ and parameter updates vanish, leading to gradient starvation (Pezeshki et al., 2021). Thus, near-zero PEs at layer $l+1$ directly eliminate learning signals for weights W_l, compromising the learning capacity of deep PCNs. This presents a fundamental paradox. While PEs constitute the primary objective to minimize, reducing them to near-zero in any layer disrupts learning signal propagation. This reveals that learning signals are transmitted through $\boldsymbol{\delta}$ values, and their elimination blocks this critical information flow. The equilibrium condition's delta relationship ($\boldsymbol{\delta}_l = g_l(\boldsymbol{\delta}_{l+1}, \mathbf{h}_{l+1})$) indicates not PE shrinkage, but rather the necessity of maintaining learning signal transmission through $\boldsymbol{\delta}$ terms. This insight underpins our meta PE-based loss design (Section 4.1), providing a principled solution to the inherent trade-off between minimizing error and preserving the learning signal.

3.3 Problem 2: Exploding and Vanishing Prediction Errors

Characterization. EVPE represents a phenomenon distinct from the classical exploding or vanishing parameter gradients observed in backpropagation (Bengio et al., 1994; Hochreiter, 1998; Pascanu et al., 2012; Arjovsky et al., 2016). Our dynamical mean-field analysis, corroborated by empirical measurements on deep PCNs (3), reveals multiplicative scaling patterns that manifest across in-

ference iteration t. Specifically, we observe a scaling relationship of the form $\|\boldsymbol{\delta}_l^{t+1}\| \approx \tau_t(\sigma_w)\|\boldsymbol{\delta}_l^t\|$. The multiplicative factors $\tau_t(\sigma_w)$ are governed primarily by the weight variance σ_w^2 and the effective gain $\|\phi'(\mathbf{h})\|_\infty$. When $\tau_t > 1$, the corresponding quantities experience geometric growth; conversely, when these factors are less than unity, exponential decay occurs. Critically, stable dynamics (factors approaching unity) emerges only within a narrow weight variance interval, specifically when σ_w is near one. This stable region contracts as network depth increases, making proper initialization exponentially more difficult for deeper architectures. Although nonlinear activations such as tanh and ReLU impose the constraint $\|\phi'\|_\infty \leq 1$, thereby attenuating these multiplicative effects, they do not eliminate the underlying geometric scaling behavior.

Distinction from Classical Exploding/Vanishing Gradients. The implications of EVPE extend beyond traditional gradient pathologies due to the unique structure of PC weight updates. Since PC parameter updates follow $\partial_{W_l}\mathcal{F} = -D(\mathbf{h}_{l+1})\boldsymbol{\delta}_{l+1}\mathbf{z}_l^\top$, we have the proportionality relationship: $\|\operatorname{vec}(\Delta W_l^t)\| \propto \|\boldsymbol{\delta}_{l+1}^t\|\|\mathbf{z}_l^t\|\|\phi'(\mathbf{h}_{l+1}^t)\|_\infty$. This direct coupling implies that EVPE in latent states and PEs immediately reflected in the magnitude of the parameter updates. Notably, this instability arises during the inference phase itself, before any parameter updates —a fundamental distinction from traditional analyses that localize explosion or vanishing behavior to backpropagated loss gradients alone. In practical terms, large values of τ_t result in rapidly increasing magnitudes $\|\mathbf{z}_l^t\|$ or $\|\boldsymbol{\delta}_l^t\|$, thereby inducing proportionally larger updates that can destabilize training. Conversely, small multiplicative factors yield near-zero updates, leading to training stagnation and ineffective learning.

4 META-PCN: A UNIFIED FRAMEWORK FOR DEEP PCN STABILIZATION

Meta-PCN represents a comprehensive framework that addresses the two fundamental pathologies identified in Section 3 through a synergistic combination of complementary techniques. Our approach systematically targets each instability mechanism with tailored solutions that work in concert to enable stable deep PCN training. The framework comprises two core components that address the identified pathologies as follows:

• **Addressing PE Imbalance (Problem 1):** We employ a dual approach that combines (i) the Meta-PC objective to linearize the nonlinear equilibrium system (Section 4.1) and (ii) systematic weight regularization to control error propagation patterns (Section 4.2).
• **Mitigating EVPE (Problem 2):** We implement comprehensive weight regularization strategies that control the multiplicative scaling factors $\tau_t(\sigma_w)$, thereby preventing geometric growth and decay patterns during inference (Section 4.2).

Each solution component is designed to preserve local learning properties, while systematically addressing the mathematical and computational challenges that have hindered scalability to deep architectures.

4.1 INFERENCE AS META PREDICTION ERROR MINIMIZATION

Motivation. Conventional PC objectives suffer from fundamental structural limitations. First, as demonstrated in Section 3.2, the free energy formulation creates a paradoxical situation where minimizing PEs leads to gradient starvation, eliminating learning signals and blocking their propagation through the network. Second, a critical train-test mismatch emerges in practice: while model predictions at evaluation time rely on direct feedforward computation without inference, training depends entirely on latent state updates through the iterative inference process. We propose a novel objective that addresses these fundamental issues while preserving core predictive coding principles.

Loss Based on Meta Prediction Errors. Our approach fundamentally redesigns the PCN objective to transform the underlying inference dynamics. While conventional PCNs initialize latent states with feedforward predictions, we fix these predictions during inference: $\hat{\mathbf{z}}_l^{(t)} = f_{l-1}(\hat{\mathbf{z}}_{l-1}^{(0)}) = \mathbf{c}_l$, where $\mathbf{c}_l := \phi(\mathbf{h}_l^{(0)})$ and $\mathbf{h}_l^{(0)}$ represents the initial pre-activation. This modification effectively linearizes the nonlinear stationarity system $F(\mathbf{z}) = \nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}) = 0$ around the feedforward initialization point. By introducing error $\tilde{\boldsymbol{\delta}}_l := \mathbf{z}_l - \mathbf{c}_l$, we obtain a layer-wise linear surrogate:

$$\tilde{F}_l(\mathbf{z}) = \tilde{\boldsymbol{\delta}}_l - g_l(\tilde{\boldsymbol{\delta}}_{l+1}, \mathbf{h}_{l+1}^{(0)}) = (\mathbf{z}_l - \mathbf{c}_l) - W_l^\top D(\mathbf{h}_{l+1}^{(0)})(\mathbf{z}_{l+1} - \mathbf{c}_{l+1}).$$

for $2 \leq l \leq L-1$, establishing a linear fixed-point relationship between consecutive layer errors. Importantly, this represents a linearization of the equilibrium map $F(\mathbf{z})$. Building on this linearization, we define a novel loss function:

$$\mathcal{J}(\tilde{\boldsymbol{\delta}}) = \frac{1}{2} \sum_{l=2}^{L-1} \|\tilde{\boldsymbol{\delta}}_l - g_l(\tilde{\boldsymbol{\delta}}^*_{l+1}, \mathbf{h}^{(0)}_{l+1})\|_2^2.$$

Considering that PEs are required to propagate in a top-down direction under feedforward initialization in the direct feedforward prediction scheme, we treat the top-down transmitted signal $\tilde{\boldsymbol{\delta}}^*_{l+1}$ as a stabilized error, regarding it as constant. The conceptual innovation lies in treating $g_l(\cdot)$ as a function that predicts the model's feedforward PE $\tilde{\boldsymbol{\delta}}_l$ using the stabilized error signals as input. Therefore, $\mathcal{J}$ minimizes the PE of PEs—a meta-level objective that fundamentally transforms the learning dynamics. Since $\partial\tilde{\boldsymbol{\delta}}_l/\partial\mathbf{z}_l = I$ and $\partial\tilde{\boldsymbol{\delta}}_{l+1}/\partial\mathbf{z}_{l+1} = I$, the gradient of $\mathcal{J}$ with respect to $\mathbf{z}_l$ equals the linearized stationarity map:

$$\nabla_{\mathbf{z}_l}\mathcal{J} = \tilde{\boldsymbol{\delta}}_l - g_l(\tilde{\boldsymbol{\delta}}^*_{l+1}, \mathbf{h}^{(0)}_{l+1}).$$

Consequently, minimizing $\mathcal{J}$ drives the linearized equilibrium residual to zero, providing a principled objective that fundamentally reshapes PC inference dynamics. This approach directly addresses the motivational problems. First, rather than directly minimizing PEs and causing gradient starvation, this novel objective encourages PEs to follow the delta relationship, ensuring balanced error propagation across layers by incorporating weight regularization. Second, by fixing feedforward predictions, we mitigate the train-test mismatch while transmitting learning signals through PE.

Mitigation of PE Imbalance and Enhanced Convergence. Our analysis in Section 3 demonstrates that layer-wise PEs satisfy the spectral bound $\|\boldsymbol{\delta}_l\| \leq \|W_l\|_2 \, \|\phi'(\mathbf{h}_{l+1})\|_\infty \, \|\boldsymbol{\delta}_{l+1}\|$. When these multiplicative factors are less than unity, the resulting product bound yields geometric decay from the output boundary, which—combined with the anchored input forcing—creates the characteristic U-shaped error profile. The Meta-PC objective addresses this imbalance through its dependence on the operator $W_l^\top D(\mathbf{h}^{(0)}_{l+1})$. By regulating the scale of this operator, we directly control the amplification and attenuation factors in the spectral bound. Furthermore, the transformation from solving a nonlinear system to a linear surrogate provides substantial convergence improvements. The dynamics may exhibit more predictable and stable behavior compared to the original nonlinear inference, enabling more efficient convergence to surrogate equilibrium states while preserving the essential PC principles.

4.2 Weight Regularization

To address the identified pathologies, Meta-PCN employs a variance-based normalization strategy that provides computationally efficient spectral control while maintaining stable weight distributions across network layers. We introduce a practical alternative to direct spectral norm computation that leverages the relationship between weight variance and spectral properties. For a weight matrix W with dimensions (m, n) and variance $\sigma_w^2 = \text{Var}(W)$, we apply direct normalization:

$$W \leftarrow \frac{W}{(\sqrt{m} + \sqrt{n})\sigma_w}.$$

This approach draws upon random matrix theory. For weight matrices with i.i.d. entries having zero mean and variance σ_w^2, the spectral norm satisfies $\|W\|_2 \approx (\sqrt{m}+\sqrt{n})\sigma_w$. Our variance-based scaling ensures $\|W_{\text{normalized}}\|_2 \approx 1$, achieving indirect spectral norm control without the computational overhead of iterative methods. The theoretical foundation rests on the conservative upper bound $\|W\|_2 \leq \|W\|_F \approx \sqrt{mn}\sigma_w$ and the Marchenko-Pastur approximation $\|W\|_2 \approx (\sqrt{m} + \sqrt{n})\sigma_w$, enabling accurate spectral norm. For implementation, we determine operator dimensions as follows: linear layers use $m = d_{\text{out}}, n = d_{\text{in}}$; convolutional layers employ $m = C_{\text{out}}, n = C_{\text{in}} \cdot k_H \cdot k_W$. This approach offers several advantages: low computational cost with parallel computation, no additional parameters, uniform application across layer types, and robust spectral control by maintaining $\|W\|_2$ near unity.The variance normalization framework directly addresses the exponential scaling behaviors identified in our analysis while preserving computational efficiency. By regulating σ_w^2 to maintain spectral norms near unity, this approach simultaneously mitigates both EVPE and PE imbalance pathologies, providing an effective solution for deep architectures.

5 META-PCN RESOLVES THREE FUNDAMENTAL PATHOLOGIES

This section demonstrates that Meta-PCN successfully addresses the two fundamental pathologies identified in Section 3. The experiments are conducted under identical conditions to enable direct comparison with the problematic behaviors observed in conventional PCNs (See Appedix E for experimental details).

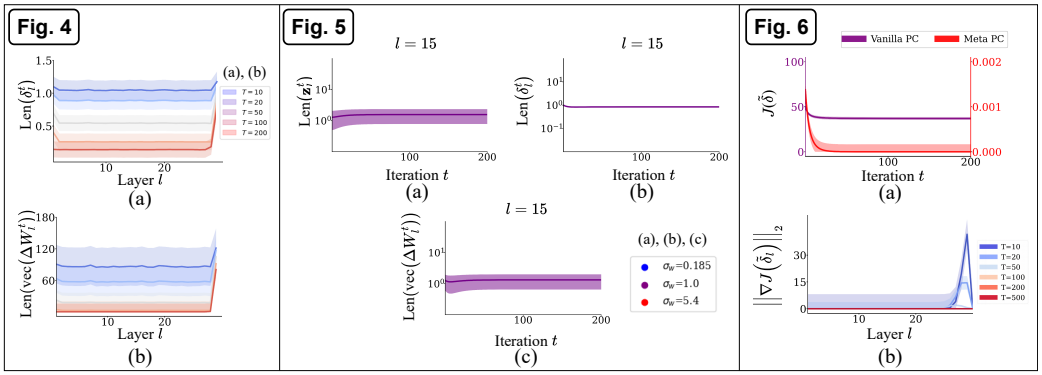


Figure 4: The layer-wise distributions of lengths in Meta-PCNs: (a) $\text{len}(\boldsymbol{\delta}_l^t)$, and (b) $\text{len}(\text{vec}(\Delta W_l^t))$.

Figure 5: Dynamics of (a) $\text{len}(\mathbf{z}_l^t)$, (b) $\text{len}(\boldsymbol{\delta}_l^t)$, (c) and $\text{len}(\text{vec}(\Delta W_l^t))$ across $\sigma_w \in \{0.185, 1.0, 5.4\}$ in Meta-PCNs.

Figure 6: Convergence dynamics and equilibrium analysis during inference. (a) Objective function convergence showing Meta-PCN's proposed objective (right y-axis) and conventional PCN's standard objective (left y-axis) in the inference phase. (b) Layer-wise convergence to delta relationships in Meta-PCN, measured by the gradient norm of the objective function.

Balanced PEs. We replicate the PE imbalance analysis from Section 3.2 using the Meta-PCN framework. Figure 4 shows the layer-wise PE distribution under Meta-PCN. Unlike the characteristic U-shaped error concentration observed in Figure 2, Meta-PCN exhibits remarkably balanced error distributions across all network layers. PEs maintain relatively uniform magnitudes across layers without excessive concentration at boundary layers. The weight update magnitudes correspondingly show balanced distributions, ensuring that learning signals reach all layers effectively. Consequently, the Meta-PC objective and weight normalization regulate the operator $W_l^\top D(h_l^{(0)})$ to maintain relatively uniform scaling across layers, preventing the geometric decay patterns that created boundary-layer dominance in conventional PCNs.

Stable PEs. Figure 5 demonstrates that Meta-PCN completely eliminates the exponential growth and decay patterns that characterized EVPE in conventional PCNs. The temporal dynamics remain stable with controlled magnitudes across all weight variance settings. The latent state lengths (Figure 5a), PE lengths (Figure 5b), and weight update lengths (Figure 5c) all maintain stable trajectories without the dramatic exponential scaling observed in Figure 3. Because we normalize weight variance to enforce a uniform scale across all conditions, the three different cases ($\sigma_w \in \{0.185, 1.0, 5.4\}$) overlap and are visually indistinguishable, appearing as a single trajectory in the figure. This is an expected result since variance normalization maintains the σ_w of weight matrices at the same scale regardless of the initial σ_w. Therefore, the multiplicative scaling factors $\tau_t(\sigma_w)$ that previously caused geometric growth or decay are now effectively controlled through the variance-based weight regularization strategy. This regulation of multiplicative factors eliminates the root cause of EVPE while preserving essential PC dynamics. All quantities remain within manageable ranges, preventing both explosion and vanishing dynamics.

Enhanced Convergence Properties. We evaluate convergence improvements by comparing the inference dynamics of conventional PCN and Meta-PCN, measuring the reduction in their respective objectives. Although a direct comparison is not entirely fair due to the differences in objectives, Figure 6 reveals dramatic convergence improvements achieved by Meta-PCN. The reduction of

objective in Meta-PCN exhibits rapid and definitive convergence to zero, in contrast to the slow convergence dynamics of conventional PCNs (Figure 6a). Layer-wise meta PE norm analysis (Figure 6b) demonstrates the rapid resolution of high meta-PEs initially observed in some layers, indicating quick convergence to equilibrium. Overall, meta-PCN achieves faster convergence speeds than conventional PCNs while maintaining stability across all layers. The linearized meta-PE-based objective provides a more tractable optimization landscape than the original nonlinear equilibrium system. The transformation from a nonlinear to a linearized system addresses the fundamental convergence limitations of conventional PC inference.

6 META-PCN ENABLES SCALABLE DEEP LEARNING

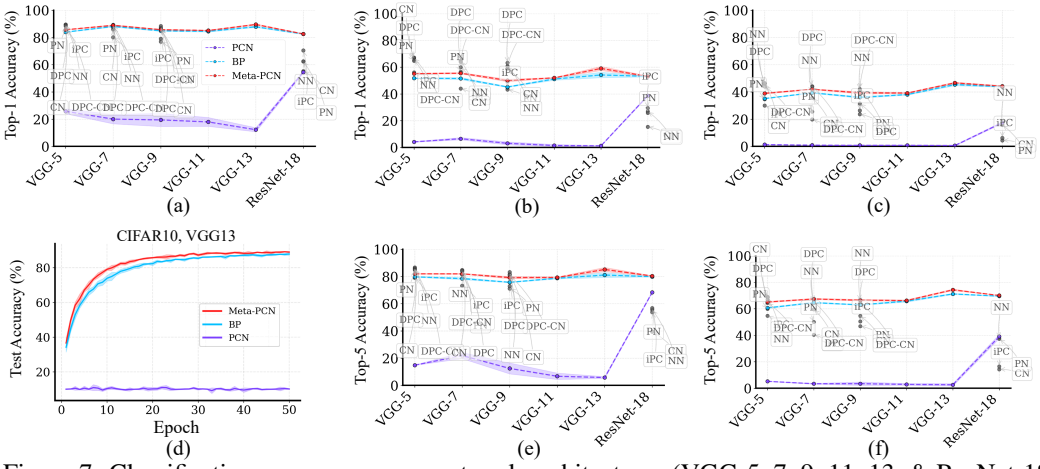


Figure 7: Classification accuracy across network architectures (VGG-5, 7, 9, 11, 13, & ResNet-18) and datasets (CIFAR-10/100 and TinyImagenet). Performance comparison of Meta-PCN, backpropagation (BP), and PCN on (a) CIFAR-10 (Top-1 acc.), (b) CIFAR-100 (Top-1 acc.), (c) TinyImageNet (Top-1 acc.), (d) Training dynamics comparison showing test accuracy evolution over 50 epochs. Learning curves for CIFAR-10 with VGG-13, (e) CIFAR-100 (Top-5 acc.), and (f) TinyImageNet (Top-5 acc.).

Table 1: Statistical significance analysis for CIFAR-10 dataset. The table displays accuracy differences (Top-1 acc. diff.) and corresponding p-values from Mann-Whitney U tests comparing Meta-PCN with BP and PCN. Positive differences (red) indicate Meta-PCN outperforms the baseline, while negative differences (blue) indicate underperformance. Statistically significant results ($p \leq 0.05$) are highlighted in bold.

(a) Meta-PCN vs BP

Architecture	Top-1 acc. diff	p-value
VGG-5	1.73	**0.0079**
VGG-7	0.83	**0.0173**
VGG-9	0.89	**0.0025**
VGG-11	0.61	**0.0303**
VGG-13	1.68	**0.0154**
ResNet-18	0.04	1.0000

(b) Meta-PCN vs Conventional PCN

Architecture	Top-1 acc. diff	p-value
VGG-5	59.86	**0.0079**
VGG-7	68.97	**0.0043**
VGG-9	66.28	**0.0025**
VGG-11	67.13	**0.0043**
VGG-13	77.64	**0.0001**
ResNet-18	28.05	**0.0079**

This section demonstrates that resolving fundamental pathologies leads to practical scalability improvements for deep PCN training. We evaluate Meta-PCN on standard image classification benchmarks to verify that theoretical improvements yield real-world performance gains.

Experimental Setup. We evaluate Meta-PCN on CIFAR-10, CIFAR-100, and TinyImageNet datasets using VGG and ResNet architectures of varying depths. We directly compare backpropagation (BP), conventional PCNs with only feedforward initialization, and the complete Meta-PCN framework. Our three-way comparison is conducted under identical experimental settings except for algorithmic differences. Detailed specifications are provided in Appendix E. Additionally, we compare various PC variants by directly incorporating values from their respective studies.

Performance Across Network Depths and Datasets. Figure 7 presents classification accuracy across different architectures and datasets. The results reveal distinct performance patterns that validate our theoretical predictions about PCN scalability limitations and Meta-PCN's effectiveness in addressing them. On CIFAR-10 (Figure 7a), conventional PCNs exhibit severe performance degradation with increasing depth, achieving only 10-20% accuracy across most architectures. In contrast, Meta-PCN demonstrates remarkable stability, maintaining 80-90% accuracy across all tested depths. Notably, Meta-PCN outperforms BP across most architectures, with statistically significant improvements ranging from 0.03% (ResNet-18) to 1.73% (VGG-5), demonstrating that biological plausibility can coexist with computational superiority (See Table 2). Similar patterns emerge on CIFAR-100 (Figure 7b) and TinyImageNet (Figure 7c). Meta-PCN consistently outperforms conventional PCNs and BP across all configurations, except for the comparison with BP on CIFAR-100 ResNet-18 Top-1 accuracy (See Appendix F for details).

Our experimental design employed stringent controls to ensure identical conditions across BP, PCN, and Meta-PCN. This rigorous standardization results in BP performance that may appear subdued relative to literature reports employing method-specific optimizations. Literature-reported PCN variants (scattered data points) appear to exhibit advantages over our controlled BP baseline in shallow networks. However, PCN variants consistently underperformed their corresponding BP implementations, underscoring the importance of controlled comparative evaluation. Meta-PCN consistently demonstrates superiority over BP in deeper architectures (VGG-13, ResNet-18) without requiring architecture-specific tuning, indicating that theoretical pathology resolution provides inherent scalability advantages.

Training Dynamics and Stability Figure 7d illustrates learning trajectories over 50 training epochs, comparing the three approaches on a representative architecture and dataset (VGG-13 on CIFAR-10). The analysis reveals fundamental differences in optimization behavior that complement our pathology resolution framework. Conventional PCN exhibits minimal learning progression, plateauing at approximately 12% accuracy throughout training. This behavior exemplifies the gradient starvation phenomenon identified in our theoretical analysis. Meta-PCN, conversely, demonstrates smooth and monotonic improvement, closely paralleling backpropagation's learning trajectory and achieving superior final performance (Meta-PCN 89.53% vs BP 87.85%). Additionally, Meta-PCN exhibits consistent improvement throughout the training process. Please refer to Appendix G, where we conducted a brief ablation study.

7 Conclusion

This study identified two fundamental pathologies that impede the scalability of deep PCNs. Through dynamical mean-field theory analysis, we theoretically established that PE imbalance and EVP constitute the core obstacles to deep PCN training. We propose the Meta-PCN framework that systematically resolves these problems through two synergistic components: (1) a meta prediction error-based loss function that linearizes the nonlinear equilibrium system to provide stable dynamics, and (2) variance-based weight regularization that suppresses EVPE. The sustained performance improvements across diverse datasets and architectures, coupled with stable training dynamics, position Meta-PCN as a practical alternative to conventional optimization methods while preserving the biological foundations that make PCNs attractive for neuromorphic implementations.

REFERENCES

Adeeti Aggarwal, Connor Brennan, Jennifer Luo, Helen Chung, Diego Contreras, Max B. Kelz, and Alex Proekt. Visual evoked feedforward–feedback traveling waves organize neural activity across the cortical hierarchy in mice. *Nature Communications*, 13(1):4754, 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32378-x.

Mohamed Akrout, Collin Wilson, Peter Humphreys, Timothy Lillicrap, and Douglas B Tweed. Deep Learning without Weight Transport. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Nicholas Alonso, Beren Millidge, Jeffrey Krichmar, and Emre Neftci. A Theoretical Framework for Inference Learning. In *Advances in Neural Information Processing Systems*, 2022.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary Evolution Recurrent Neural Networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1120–1128. PMLR, June 2016.

Sergey Bartunov, Adam Santoro, Blake A. Richards, Luke Marris, Geoffrey E. Hinton, and Timothy Lillicrap. Assessing the Scalability of Biologically-Motivated Deep Learning Algorithms and Architectures, 2018.

Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

André Moraes Bastos, Julien Vezoli, Conrado Arturo Bosman, Jan-Mathijs Schoffelen, Robert Oostenveld, Jarrod Robert Dowdall, Peter De Weerd, Henry Kennedy, and Pascal Fries. Visual Areas Exert Feedforward and Feedback Influences through Distinct Frequency Channels. *Neuron*, 85 (2):390–401, 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2014.12.018.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994. ISSN 1941-0093. doi: 10.1109/72.279181.

Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211, 2017.

Matteo Carandini and David J Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(1):51 – 62, 2012. doi: 10.1038/nrn3136.

Rishidev Chaudhuri, Kenneth Knoblauch, Marie-Alice Gariel, Henry Kennedy, and Xiao-Jing Wang. A Large-Scale Circuit Mechanism for Hierarchical Dynamical Processing in the Primate Cortex. *Neuron*, 88(2):419–431, 2015. ISSN 1097-4199. doi: 10.1016/j.neuron.2015.09.008.

Giorgia Dellaferrera and Gabriel Kreiman. Error-driven input modulation: solving the credit assignment problem without a backward pass. In *International Conference on Machine Learning*, pp. 4937–4955. PMLR, 2022.

Maxence M. Ernoult, Fabrice Normandin, Abhinav Moudgil, Sean Spinney, Eugene Belilovsky, Irina Rish, Blake Richards, and Yoshua Bengio. Towards Scaling Difference Target Propagation by Learning Backprop Targets. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 5968–5987. PMLR, 2022.

Harriet Feldman and Karl Friston. Attention, Uncertainty, and Free-Energy. *Frontiers in Human Neuroscience*, 4, 2010. ISSN 1662-5161. doi: 10.3389/fnhum.2010.00215.

Charlotte Frenkel, Martin Lefebvre, and David Bol. Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks. *Frontiers in Neuroscience*, 15:629892, 2021. ISSN 1662-453X. doi: 10.3389/fnins.2021.629892.

Pascal Fries. Rhythms for Cognition: Communication through Coherence. *Neuron*, 88(1):220–235, 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.09.034.

Karl Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836, 2005.

Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11 (2):127–138, 2010.

Cédric Goemaere, Gaspard Oliviers, Rafal Bogacz, and Thomas Demeester. Error Optimization: Overcoming Exponential Signal Decay in Deep Predictive Coding Networks, 2025.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013. ISBN 978-1-4214-0794-4.

Jordan Guerguiev, Timothy P Lillicrap, and Blake A Richards. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, 2017. ISSN 2050-084X. doi: 10.7554/eLife.22901.

Saskia Haegens, Annamaria Barczak, Gabriella Musacchia, Michael L. Lipton, Ashesh D. Mehta, Peter Lakatos, and Charles E. Schroeder. Laminar Profile and Physiology of the α Rhythm in Primary Visual, Auditory, and Somatosensory Regions of Neocortex. *Journal of Neuroscience*, 35(42):14341–14352, 2015. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0600-15.2015.

Neil R. Hardingham, Giles E. Hardingham, Kevin D. Fox, and Julian J. B. Jack. Presynaptic efficacy directs normalization of synaptic strength in layer 2/3 rat neocortex after paired activity. *Journal of Neurophysiology*, 97(4):2965–2975, April 2007. ISSN 0022-3077. doi: 10.1152/jn.01352.2006.

Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, April 1998. doi: 10.1142/S0218488598000094.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Francesco Innocenti, El Mehdi Achour, and Christopher L. Buckley. $\$\mu\$$PC: Scaling Predictive Coding to 100+ Layer Networks, 2025.

Ryota Kanai, Yutaka Komura, Stewart Shipp, and Karl Friston. Cerebral hierarchies: Predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668):20140169, 2015. doi: 10.1098/rstb.2014.0169.

Georg B. Keller and Thomas D. Mrsic-Flogel. Predictive Processing: A Canonical Cortical Computation. *Neuron*, 100(2):424–435, October 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.10.003.

Paul F. Kinghorn, Beren Millidge, and Christopher L. Buckley. Preventing Deterioration of Classification Accuracy in Predictive Coding Networks. In Christopher L. Buckley, Daniela Cialfi, Pablo Lanillos, Maxwell Ramstead, Noor Sajid, Hideaki Shimazaki, and Tim Verbelen (eds.), *Active Inference*, pp. 1–15, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-28719-0. doi: 10.1007/978-3-031-28719-0_1.

Dong-Hyun Lee, Shan Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. *Machine learning and knowledge discovery in databases*, pp. 498–515, 2015.

Mathieu Letellier, Florian Levet, Olivier Thoumine, and Yukiko Goda. Differential role of pre- and postsynaptic neurons in the activity-dependent control of synaptic strengths across dendrites. *PLOS Biology*, 17(6):e2006223, 2019. ISSN 1545-7885. doi: 10.1371/journal.pbio.2006223.

Qianli Liao, Joel Z. Leibo, and Tomaso Poggio. How important is weight symmetry in backpropagation? In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1): 13276, 2016.

Fabian A. Mikulasch, Lucas Rudelt, Michael Wibral, and Viola Priesemann. Where is the error? Hierarchical predictive coding through dendritic error computation. *Trends in Neurosciences*, 46 (1):45–59, January 2023. ISSN 0166-2236, 1878-108X. doi: 10.1016/j.tins.2022.09.007.

Beren Millidge, Yuhang Song, Tommaso Salvatori, Thomas Lukasiewicz, and Rafal Bogacz. A Theoretical Framework for Inference and Learning in Predictive Coding Networks. In *The Eleventh International Conference on Learning Representations*, September 2022a.

Beren Millidge, Alexander Tschantz, and Christopher L Buckley. Predictive coding networks for temporal prediction. *Neural Networks*, 147:20–34, 2022b.

Rosalyn J. Moran, Pablo Campo, Mkael Symmonds, Klaas E. Stephan, Raymond J. Dolan, and Karl J. Friston. Free Energy, Precision and Learning: The Role of Cholinergic Neuromodulation. *Journal of Neuroscience*, 33(19):8227–8236, 2013. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.4255-12.2013.

Theodore H. Moskovitz, Ashok Litwin-Kumar, and L. F. Abbott. Feedback alignment in deep convolutional networks, 2019.

Lyle Muller, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski. Cortical travelling waves: Mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268, 2018. ISSN 1471-0048. doi: 10.1038/nrn.2018.20.

David Mumford. On the computational architecture of the neocortex: Ii the role of cortico-cortical loops. *Biological cybernetics*, 66(3):241–251, 1992.

John D. Murray, Alberto Bernacchia, David J. Freedman, Ranulfo Romo, Jonathan D. Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12): 1661–1663, 2014. ISSN 1546-1726. doi: 10.1038/nn.3862.

Arild Nø kland. Direct Feedback Alignment Provides Learning in Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, November 1982. ISSN 1432-1416. doi: 10.1007/BF00275687.

Zhaoyang Pang, Andrea Alamia, and Rufin VanRullen. Turning the Stimulus On and Off Changes the Direction of α Traveling Waves. *eNeuro*, 7(6), 2020. ISSN 2373-2822. doi: 10.1523/ENEURO.0218-20.2020.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *ArXiv*, November 2012.

Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A. Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature Neuroscience*, 24(7):1010–1019, 2021. ISSN 1546-1726. doi: 10.1038/s41593-021-00857-x.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient Starvation: A Learning Proclivity in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272. Curran Associates, Inc., 2021.

Luca Pinchetti, Chang Qi, Oleh Lokshyn, Cornelius Emde, Amine M'Charrak, Mufeng Tang, Simon Frieder, Bayar Menzat, Gaspard Oliviers, Rafal Bogacz, Thomas Lukasiewicz, and Tommaso Salvatori. Benchmarking Predictive Coding Networks – Made Simple. In *The Thirteenth International Conference on Learning Representations*, 2024.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos, jun 2016.

Chang Qi, Thomas Lukasiewicz, and Tommaso Salvatori. Training Deep Predictive Coding Networks. In *New Frontiers in Associative Memories*, 2025.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

Youcef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, 2nd edition, 2003. ISBN 978-0-89871-534-7.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31, 2018.

Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, Rafal Bogacz, and Zhenghua Xu. Associative memories via predictive coding. *Advances in Neural Information Processing Systems*, 35: 3874–3888, 2022.

Tommaso Salvatori, Ankur Mali, Christopher L. Buckley, Thomas Lukasiewicz, Rajesh P. N. Rao, Karl Friston, and Alexander Ororbia. Brain-Inspired Computational Intelligence via Predictive Coding, August 2023a.

Tommaso Salvatori, Yuhang Song, Yordan Yordanov, Beren Millidge, Lei Sha, Cornelius Emde, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. A Stable, Fast, and Fully Automatic Learning Algorithm for Predictive Coding Networks. In *The Twelfth International Conference on Learning Representations*, October 2023b.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation, apr 2017.

Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963*, 2017.

H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in Random Neural Networks. *Physical Review Letters*, 61(3):259–262, jul 1988. doi: 10.1103/PhysRevLett.61.259.

Yuhang Song, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz. Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems*, 33:22566–22579, 2020.

Mandyam Veerambudi Srinivasan, Simon Barry Laughlin, and Andreas Dubs. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205):427–459, 1982.

Gina G. Turrigiano, Kenneth R. Leslie, Niraj S. Desai, Lana C. Rutherford, and Sacha B. Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–896, February 1998. ISSN 1476-4687. doi: 10.1038/36103.

Timo van Kerkoerle, Matthew W. Self, Bruno Dagnino, Marie-Alice Gariel-Mathis, Jasper Poort, Chris van der Togt, and Pieter R. Roelfsema. Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex. *Proceedings of the National Academy of Sciences*, 111(40):14332–14341, 2014. doi: 10.1073/pnas.1402773111.

Richard S. Varga. *Matrix Iterative Analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2nd revised and expanded edition, 2009. ISBN 978-3-540-66321-8.

James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5): 1229–1262, 2017.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5393–5402. PMLR, July 2018.

David M. Young. *Iterative Solution of Large Linear Systems*. Computer Science and Applied Mathematics. Academic Press, New York, 1971. ISBN 978-0-12-773050-9.

A Related Work

A.1 Relation to biologically plausible learning

We situate our work within bio-inspired learning but address a distinct failure mode. Much prior research rethinks how errors are represented and routed to avoid weight transport or global coordination in backpropagation. Feedback-alignment families remove exact symmetry between forward and feedback pathways (using fixed random feedback, direct feedback, sign constraints, or learned mirroring), yielding partial alignment but degrading as depth and task complexity grow, often requiring auxiliary assumptions to scale (Lillicrap et al., 2016; Nø kland, 2016; Liao et al., 2016; Xiao et al., 2018; Akrout et al., 2019; Bartunov et al., 2018; Moskovitz et al., 2019). Target- and energy-based methods propagate per-layer targets or perturb equilibria to obtain local updates and partial locality guarantees, tightening links to Gauss–Newton behavior or learning transposed Jacobians (Lee et al., 2015; Ernoult et al., 2022; Scellier & Bengio, 2017). Complementary work replaces the backward pass with feedforward or locally decoupled objectives (auxiliary heads, local losses, label projection, or two-pass forward updates) to unlock parallelism and autonomy (Frenkel et al., 2021; Dellaferrera & Kreiman, 2022). At a finer-grained level, dendritic and microcircuit proposals leverage apical modulation, short-term plasticity, and burst-dependent rules to route credit while echoing cortical motifs (Guerguiev et al., 2017; Sacramento et al., 2018; Payeur et al., 2021).

Our formulation is orthogonal to these questions of error coding and routing. We ask why deep predictive coding networks (PCNs)—which learn by iteratively relaxing latent states via bottom-up errors and top-down predictions—become intrinsically unstable with depth. The above lines primarily target weight symmetry, error-sign consistency, or locality; they do not directly suppress the dilating modes, cross-layer amplification, and timescale–gating mismatches that drive inference divergence in PCNs. We therefore take a dynamics-first approach: derive architectural and algorithmic conditions that stabilize iterative inference in deep PCNs (layerwise contraction, adaptive damping and gating, state normalization, error–activation decoupling, and slow–fast timescale separation) and show that enforcing these principles yields scalable training without abandoning local learning. Our view is complementary: we retain the brain-inspired philosophy while addressing a bottleneck largely left open—guaranteeing convergent inference in deep architectures.

A.2 Research on improving predictive coding networks

Stability and scalability are recognized bottlenecks for PCNs: with depth, benchmarks report accuracy deterioration, skewed layerwise prediction-error/energy distributions, and growing imbalances in relaxation speeds (Pinchetti et al., 2024; Kinghorn et al., 2023; Qi et al., 2025). Symptomatic remedies adjust schedules or normalizers; e.g., interleaving state and weight updates improves robustness and efficiency but does not supply depth-agnostic guarantees for stable inference dynamics (Salvatori et al., 2023b). Theoretical reinterpretations cast predictive-coding updates as implicit stochastic gradients, clarifying global step-size stability from a weight-optimization viewpoint, while broader frameworks connect PCNs to BP, EM-like procedures, and equilibrium methods—useful equivalences that nonetheless give limited constructive prescriptions for eliminating depth-induced inference pathologies (Alonso et al., 2022; Millidge et al., 2022a).

Depth-focused strategies modify what is optimized and where. Some reduce error-distortion accumulation and rebalance layerwise energy during relaxation, improving accuracy beyond seven layers but using temporally non-local updates that do not enforce contraction (Qi et al., 2025). Others reparameterize objectives to optimize directly in error space to mitigate exponential signal attenuation, trading locality for speed, or adopt depth-aware parameterization and learning-rate scaling (e.g., microparameterization) to train very deep residual PCNs under architectural constraints (Goemaere et al., 2025; Innocenti et al., 2025). Collectively, these advances raise performance yet stop short of general, architecture-agnostic guarantees of stable inference across tasks and hyperparameters.

Our contribution centers stability as the primary design objective. We diagnose concrete mathematical failure modes—layerwise sensitivity/energy mismatch, amplification along coupled error–state channels, and accumulation of spatiotemporal non-locality—and derive sufficient stabilization conditions with implementable rules that enforce contraction and scale separation during relaxation while preserving fully local learning. Empirically, this yields stable, scalable training across architectures and tasks without auxiliary control phases or non-local coordination, complementing

scheduling-based remedies, optimization-only reinterpretations, and depth-specific reparameterizations.

B Biological Plausibility of Meta-PCNs

B.1 Biological plausibility of Inference as Meta–Free Energy Minimization

Our inference objective fixes each layer's feedforward target c_ℓ and minimizes a meta prediction error, $|\delta_\ell - g_\ell(\delta_{\ell+1}, h^{(0)}\ell + 1)|^2$, so that the realized local error $\delta\ell$ matches a top down prediction of that error. This "error of errors" matching is consistent with hierarchical predictive coding accounts in which ascending signals progressively encode residuals of residuals, supported by a division of labor between error and representation units and by laminar asymmetries in feedforward versus feedback pathways (Rao & Ballard, 1999; Bastos et al., 2012; Keller & Mrsic-Flogel, 2018). Dendritic predictive coding further suggests that basal and apical compartments can locally compare bottom up errors with top down predictions, implementing error computation without distinct error neurons (Mikulasch et al., 2023; Salvatori et al., 2023a). Taken together, these views align with our formulation as an explicit, hierarchical comparison between realized errors and their predictions, while remaining faithful to canonical message passing in cortical microcircuits (Bastos et al., 2012). A complementary line of work interprets meta-error minimization as learning precision weighted errors. Predictive coding theories posit context-dependent gain control of sensory errors, with neuromodulatory systems (notably cholinergic) and thalamocortical routing via the pulvinar supporting dynamic precision and task-dependent gating of error flow (Feldman & Friston, 2010; Moran et al., 2013; Kanai et al., 2015). Our objective reduce the mismatch between observed errors and expected precision, offering a mechanistic bridge between these proposals and graded, circuit-level control of error gain. Finally, clamping c_ℓ during inference is consistent with hierarchical intrinsic timescales: higher association areas evolve slowly, acting as quasi fixed boundary conditions over short inference windows while faster superficial errors relax toward these slow top-down states (Murray et al., 2014; Chaudhuri et al., 2015; Bastos et al., 2012).

B.2 Biological plausibility of Weight Regularization

Our weight regularizer rescales W multiplicatively using its layerwise variance to keep the effective operator near unit spectral norm, stabilizing $W^\top D(h)$ across layers. Functionally, this mirrors divisive normalization, a canonical computation in which neural responses are divided by a pooled activity term to control gain across populations and circuits (Carandini & Heeger, 2012). Interpreting our variance-governed rescaling as parameter space gain control links it to convergent physiological mechanisms for divisive normalization across sensory systems and cortical areas (Carandini & Heeger, 2012). The same multiplicative structure is consistent with global synaptic scaling, which multiplicatively adjusts all excitatory synapses to maintain target activity, and with theoretical local rules that normalize weight norms (Turrigiano et al., 1998; Oja, 1982). At finer scales, heterosynaptic plasticity and presynaptic normalization constrain variance branchwise and along axons, yielding population level effects equivalent to our matrix level multiplicative regularization (Letellier et al., 2019; Hardingham et al., 2007).

B.3 Biological plausibility of Blocked Sweep Updates for PC Inference

Our blocked sweep schedule updates groups of layers sequentially, alternating feedforward and feedback relaxation. This accords with phase-structured interareal communication: gamma/theta rhythms predominately mediate feedforward influences, whereas alpha/beta rhythms carry feedback, with phase alignment enabling communication through coherence and cross-frequency control of bottom-up by top-down signals (van Kerkoerle et al., 2014; Bastos et al., 2015; Fries, 2015). Traveling waves further support sequential, phase-specific updates across hierarchies, where slow feedback waves modulate the amplitude and timing of faster feedforward waves, opening alternating windows for ascending errors and descending predictions (Muller et al., 2018; Aggarwal et al., 2022). Within each block, repeated local relaxation is consistent with dendritic error computation in pyramidal cells and with laminar routing of feedforward error in supragranular layers and feedback prediction in infragranular layers (Mikulasch et al., 2023; Haegens et al., 2015; Pang et al., 2020). Overall, inference as meta free energy minimization, variance-governed multiplicative weight rescaling, and

blocked sweep scheduling instantiate computations long hypothesized for cortical hierarchies and microcircuits. They operationalize predictive coding's core claims while remaining compatible with dendritic implementations, divisive normalization, neuromodulatory precision control, corticothalamic routing, laminar asymmetries, and phase-coordinated interareal communication.

C Discussion

C.1 Contribution

Our contributions establish both theoretical foundations and practical solutions for scalable deep PCN training:

Theoretical Contributions: We rigorously identify the root causes of deep PCN instability through convergence analysis, providing comprehensive empirical characterization and quantitative analysis of PE imbalance and EVPE phenomena through dynamical mean-field theory.

Methodological Contributions: We introduce Meta-PCN as a unified framework with two targeted solutions: the novel Meta-PC objective (Solution 1) that minimizes PEs of PEs, enhanced by advanced NLS solvers and blocked sweeps that transform inference dynamics, and systematic weight regularization (Solution 2) for variance control.

Empirical Contributions: We demonstrate substantial performance gains and stability improvements over existing PCN methods, with 15-18% accuracy improvements and 39% better convergence.

C.2 Limitations.

This study has several limitations. First, Meta-PCN's performance has not reached complete parity with backpropagation. While this represents substantial progress achieved while maintaining biological constraints, it suggests that performance gaps may still exist in some application domains. Second, the current analysis focuses primarily on computer vision tasks, requiring additional validation for generalization capabilities in other domains. Third, a comprehensive analysis is lacking regarding whether variance-based weight regularization is equally effective across all architectures and activation functions.

C.3 Future Work.

Future research can be extended in several directions. First, the effectiveness of Meta-PCN should be validated in other domains such as natural language processing and reinforcement learning. Second, more sophisticated weight regularization techniques and adaptive inference schemes could be developed to reduce the performance gap with backpropagation further. Third, the biological plausibility of Meta-PCN should be more rigorously verified, and its connections with actual neuroscientific findings should be explored.

D Derivation of Length Dynamics

This section provides the complete mathematical foundation for the length dynamics framework introduced in Section 3.1. We present the theoretical aspects of how latent variables evolve during the inference process, leveraging assumptions of linearity and Gaussian-distributed parameters. We focus on the statistical distribution of Gaussian samples, which serves as the foundation for understanding the Gaussian ensemble network's behavior under large-scale computations. We present a rigorous analysis of interaction matrices of latents and bias and their dynamics, offering insights into the lengths dynamics of latent states $\mathrm{len}(\mathbf{z}_l^t)$, PEs $\mathrm{len}(\boldsymbol{\delta}_l^t)$, and weight updates $\mathrm{len}(\mathrm{vec}(\Delta W_l^t))$ as defined in Section 3.1.

D.1 Assumptions and Latent States Update rule

Our primary goal is to track the changes in the length of the latent state during the inference step t. To perform this analysis, we adopt the following assumptions:

Gaussian Assumption We assume that the initial latent state at the inference step $t = 1$ is drawn i.i.d. as $z_{i,l}^t \sim \mathcal{N}(0, 1)$. The learnable parameters, weight and bias, are drawn i.i.d. as $w_{i,j,l} \sim \mathcal{N}(0, \frac{\sigma_w^2}{N_l})$ and $b_{i,l} \sim \mathcal{N}(0, \sigma_b^2)$.

Linearity Assumption The correlation between variables may vary arbitrarily depending on the nonlinearity of the activation function ϕ, making it difficult to expand interaction analytically. Therefore, we initially analyze the case where ϕ is a linear function. For cases involving non-linear activation functions, empirical verification is performed to confirm the results in Appendix H. In the context of the linearity assumption, the forward and backward transformations are defined as follows

$$f_{l-1}(\mathbf{z}_{l-1}) = W_{l-1}\mathbf{z}_{l-1} + \mathbf{b}_{l-1} \tag{2}$$

$$g_l(\mathbf{z}_{l+1}) = W_l^\top \mathbf{z}_{l+1}. \tag{3}$$

Dimensionality Assumption We assume that all layers share the same dimensionality. If the dimensions differ, the latent spaces must be transformed using matrices like M, resulting in non-generalizable cross-layer interactions.

With these assumptions, we can expand the latent state update rule as follows:

$$\begin{aligned}
\mathbf{z}_l^{t+1} &= \mathbf{z}_l^t + \Delta\mathbf{z}_l^t \\
&= \mathbf{z}_l^t + \eta\left(-\delta_l^t + W_l^\top \delta_{l+1}^t\right) \\
&= \mathbf{z}_l^t + \eta\left(-\left(\mathbf{z}_l^t - \hat{\mathbf{z}}_l^t\right) + W_l^\top\left(\mathbf{z}_{l+1}^t - \hat{\mathbf{z}}_{l+1}^t\right)\right)
\end{aligned} \tag{4}$$

The update rule can be further simplified as:

$$\begin{aligned}
\mathbf{z}_l^{t+1} &= (1-\eta)\mathbf{z}_l^t + \eta\cdot(W_{l-1}\mathbf{z}_{l-1}^t + \mathbf{b}_{l-1}) \\
&\quad + \eta\cdot W_l^\top \mathbf{z}_{l+1}^t - \eta\cdot W_l^\top\left(W_l\mathbf{z}_l^t + \mathbf{b}_l\right) \\
&= \rho M_{l-1}\mathbf{z}_{l-1}^t + \kappa\mathbf{z}_l^t + \rho M_l^\top \mathbf{z}_{l+1}^t + \eta\mathbf{b}_{l-1} \\
&\quad - \rho M_l^\top \mathbf{b}_l
\end{aligned} \tag{5}$$

where $M = \frac{1}{\sigma_w}W$, $\rho = \eta\sigma_w$, $\kappa = 1 - \eta(1 + \sigma_w^2)$, and $\xi = \eta\sigma_w^2$.

D.2 The distribution of the product of Gaussian samples

Before delving into the dynamics of length, given that our analysis involves the product of different forms of Gaussian samples, it is essential to review the generalized results of this. Let $u_i \sim \mathcal{N}(0, \frac{\sigma_u^2}{N})$ and and $v_i \sim \mathcal{N}(0, \frac{\sigma_u^2}{N})$. The square of u_i follows a chi-square distribution, while the product $u_i \cdot v_i$ follows a normal product distribution. Our interest lies in understanding the distribution of the following inner product values

$$\mathbf{u}^\top\mathbf{u} = \sum_{i=1}^N u_i^2 \quad \text{and} \quad \mathbf{u}^\top\mathbf{v} = \sum_{i=1}^N u_i \cdot v_i \tag{6}$$

as $N \to \infty$. Applying the Central Limit Theorem (CLT) to these values, we obtain the following:

$$\sqrt{N}\cdot\frac{\frac{\mathbf{u}^\top\mathbf{u}}{N}\mathbb{E}[u_i^2]}{\sqrt{\text{Var}(u_i^2)}} \to \mathcal{N}(0, 1), \tag{7}$$

where $\mathbb{E}[u_i^2] = \text{Var}(u_i) = \frac{\sigma_u^2}{N}$, and $\text{Var}(u_i^2) = \mathbb{E}[u_i^4] - \mathbb{E}[u_i^2]^2 = 3\frac{\sigma_u^4}{N^2} - \frac{\sigma_u^4}{N^2} = 2\frac{\sigma_u^4}{N^2}$. As a result, $\mathbf{u}^\top\mathbf{u} \to \mathcal{N}(\sigma_u^2, \frac{2\sigma_u^4}{N})$, and equivalently,

$$\mathbf{u}^\top\mathbf{u} \to \sigma_u^2 \cdot \mathcal{N}(1, \frac{2}{N}). \tag{8}$$

Similarly, applying the CLT to the cross-product yields:

$$\sqrt{N} \cdot \frac{\frac{\mathbf{u}^\top \mathbf{v}}{N} \mathbb{E}[u_i \cdot v_i]}{\sqrt{\text{Var}(u_i \cdot v_i)}} \to \mathcal{N}(0, 1), \tag{9}$$

where $\mathbb{E}[u_i \cdot v_i] = 0$, since u_i and v_i are independent, and $\text{Var}(u_i \cdot v_i) = \sigma_u^2 \cdot \sigma_v^2$. Hence, we obtain $\mathbf{u}^\top \mathbf{v} \to \mathcal{N}(0, \frac{\sigma_u^2 \cdot \sigma_v^2}{N})$. Equivalently, for large N,

$$\mathbf{u}^\top \mathbf{v} \sim \sigma_u \sigma_v \cdot \mathcal{N}(0, \frac{1}{N}), \tag{10}$$

and if $\sigma_u = \sigma_v$, this converges to $\sigma_u^2 \cdot \mathcal{N}(0, \frac{1}{N})$.

We can conduct a similar analysis for the distribution of vector lengths. Let $u_i \sim \mathcal{N}(0, \sigma_u^2)$ and $v_i \sim \mathcal{N}(0, \sigma_v^2)$. In these cases, we want to understand the asymptotic distribution of the following length terms:

$$\left\langle u_i^2 \right\rangle = \frac{1}{N} \sum_{i=1}^{N} u_i^2 \quad \text{and} \quad \left\langle u_i \cdot v_i \right\rangle = \frac{1}{N} \sum_{i=1}^{N} u_i \cdot v_i, \tag{11}$$

as $N \to \infty$. Note that the variance of the Gaussian distribution in the length calculation is not divided by N in contrast to the inner product version. Instead, the length includes a division by N. By applying the CLT, similar to the inner product case, we have:

$$\left\langle u_i^2 \right\rangle \to \sigma_u^2 \cdot \mathcal{N}(1, \frac{2}{N}). \tag{12}$$

Using this result, we can apply it to the cases of interest.

Lengths In the case of vector-vector multiplication, consider vectors $\mathbf{z}_1$, $\mathbf{z}_L$, and $\mathbf{b}_l$, where $l \in \{1, \ldots, L-1\}$. Each of these vectors is assumed to be sampled from a Gaussian distribution, i.e., each element is drawn from $\mathcal{N}(0, 1)$. The length defined by the relationship between these vectors, as $N \to \infty$, follows:

$$\left\langle u_i^2 \right\rangle \to \mathcal{N}(1, \frac{2}{N}), \tag{13}$$

while the cross-product between different vectors converges to:

$$\left\langle u_i \cdot v_i \right\rangle \to \mathcal{N}(0, \frac{1}{N}). \tag{14}$$

Consequently, the self-product (length) converges to 1, while the product with a different vector converges to 0 as $N \to \infty$. Finally, consider the length $l = \frac{1}{N} \mathbf{v}^\top A \mathbf{u}$, where each element of A, A_{ij}, is drawn from $\mathcal{N}(0, \frac{1}{N})$, and each element of $\mathbf{u}$ and $\mathbf{v}$ follows $\mathcal{N}(0, 1)$. The transformed vector $(A\mathbf{u})_i \sim \mathcal{N}(0, 1)$, Therefore, $\mathbf{v}^\top (A\mathbf{u}) \sim \mathcal{N}(0, 1)$, Thus, the length l follows:

$$l \sim \mathcal{N}(0, \frac{1}{N^2}). \tag{15}$$

Matrix-Matrix Multiplication In the case of matrix-matrix multiplication, consider $C = A^\top A$, where each element of A, i.e., A_{ij}, is drawn from $\mathcal{N}(0, \frac{1}{N})$. The diagonal entries of C, C_{ii}, follow $\mathcal{N}(1, \frac{2}{N})$, The off-diagonal entries C_{ij}, where $i \neq j$, follow $\mathcal{N}(0, \frac{1}{N})$, Hence, C approaches the identity matrix I as $N \to \infty$. For the product of two matrices $D = AB$, where both A_{ij} and B_{ij} are sampled from $\mathcal{N}(0, \frac{1}{N})$, the resulting matrix D_{ij} shares the same distribution as A_{ij} and B_{ij}.

D.3 Interaction Matrices

For the analysis of length dynamics, we define several key variables as follows.

Latent Self-Interaction Let $P^t_{l+k,l} = \frac{1}{N}\mathbf{z}^{t\top}_{l+k} M_{l+k-1:l}\mathbf{z}^t_l$ for $1 \le l, l-k \le L$, where $M_{l+k:l} = M_{l+k}M_{l+k-1}\cdots M_l$ is the products of the series of matrices (as introduced in Section 3.1). By definition, we can observe that P is systematic, meaning that $P^t_{l,l+k} = P^t_{l+k,l}$. The length of the latent state at layer l and time step t, $p^{l,t}$ can be represented as the diagonal elements of P^t, $p^t_l = \left\langle \left(z^t_{i,l}\right)^2 \right\rangle = \frac{1}{N}\sum_{i=1}^N \left(z^t_{i,l}\right)^2 = \frac{1}{N}\mathbf{z}^{t\top}_l \mathbf{z}^t_l = P^t_{l,l}$. With this definition, the matrix P^t contains the length information and interactions between latent states at different layers at the inference step t. Since the input and output are fixed during the inference phase as $\mathbf{z}^{t+1}_1 = \mathbf{z}^t_1$ and $\mathbf{z}^{t+1}_L = \mathbf{z}^t_L$, the interaction terms with the indices 1 and L are constants as $P_{1,1} = P_{L,L} = 1$ and $P_{1,L} = P_{L,1} = 0$. Similarly, at $t = 0$, $P^0 = I$.

Bias-Latent State Interaction Let bias-latent state interaction $B^t_{l,l-k} = \frac{1}{N}\mathbf{b}^\top_{l-1} M_{l-1:l-k}\mathbf{z}^t_{l-k}$ be a bilinear term of interaction between the bias and latent states at layers l and $l-k$ at inference step t for $1 \le l, l-k \le L$. Likewise, let $B^t_{l-k,l} = \frac{1}{N}\mathbf{z}^{t\top}_l M_{l-1:l-k}\mathbf{b}_{l-k-1}$. Since the bias, the input ($\mathbf{z}_1$) and output ($\mathbf{z}_L$) are fixed during the inference phase, the interaction terms between these independent components, $B_{:,1}$ and $B_{:,L}$, are also fixed at 0. At $t = 0$, $B^0 = 0$.

Bias Self-Interaction The term $\Gamma_{l,l-k}$ represents $\frac{1}{N}\mathbf{b}^\top_l M_{l:l-k+1}\mathbf{b}_{l-k} = 0$ for $1 \le l, l-k \le L$. Since the bias terms are sampled from $\mathcal{N}(0, \sigma_b^2)$ and fixed during the inference phase, $\Gamma = \sigma_b^2 I$ is a constant matrix by the properties introduced in Appendix D.2.

D.4 Dynamics of Interaction Matrix

We derive the update rule for the P using the definition of the interaction and the latent update rule in Equation 5. For an element of $P^t_{l,l-k}$, where $l-k > 1$ and $l < L$, the update equation can be described as follows:

$$\begin{aligned}
P^{t+1}_{l,l-k} &= \frac{1}{N}\mathbf{z}^{t+1\top}_l M_{l-1:l-k}\mathbf{z}^{t+1}_{l-k} \\
&= \frac{1}{N}\left(\kappa\cdot\mathbf{z}^t_l + \rho\cdot M_{l-1}\mathbf{z}^t_{l-1} + \rho\cdot M^\top_l \mathbf{z}^t_{l+1}\right. \\
&\quad \left. +\eta\cdot\mathbf{b}_{l-1} - \rho\cdot M^\top_l \mathbf{b}_l\right)^\top \times M_{l-1:l-k} \\
&\quad \times \Big(\kappa\cdot\mathbf{z}^t_{l-k} + \rho\cdot M_{l-k-1}\mathbf{z}^t_{l-k-1} \\
&\quad + \rho\cdot M^\top_{l-k}\mathbf{z}^t_{l-k+1} + \eta\cdot\mathbf{b}_{l-k-1} \\
&\quad -\rho\cdot M^\top_{l-k}\mathbf{b}_{l-k}\Big)
\end{aligned}$$

We want to expand this equation fully, showing all combinations of terms in the product. First, we identify the components of the vectors involved in the equation. The expression consists of a sum of transposed vectors, multiplied by a matrix $M_{l-1:l-k}$, and then multiplied by another sum of vectors. The components of the first sum of vectors are

$$\begin{aligned}
&\mathbf{u}_1 = \rho\cdot M_{l-1}\mathbf{z}^t_{l-1}, \mathbf{u}_2 = \kappa\cdot\mathbf{z}^t_l, \mathbf{u}_3 = \rho\cdot M^\top_l\mathbf{z}^t_{l+1}, \\
&\mathbf{u}_4 = \eta\cdot\mathbf{b}_{l-1}, \quad \text{and} \quad \mathbf{u}_5 = -\rho\cdot M^\top_l\mathbf{b}_l.
\end{aligned}$$

The components of the second sum of vectors are

$$\begin{aligned}
&\mathbf{v}_1 = \rho\cdot M_{l-k-1}\mathbf{z}^t_{l-k-1}, \mathbf{v}_2 = \kappa\cdot\mathbf{z}^t_{l-k}, \\
&\mathbf{v}_3 = \rho\cdot M^\top_{l-k}\mathbf{z}^t_{l-k+1}, \mathbf{v}_4 = \eta\cdot\mathbf{b}_{l-k-1}, \\
&\quad \text{and} \quad \mathbf{v}_5 = -\rho\cdot M^\top_{l-k}\mathbf{b}_{l-k}.
\end{aligned}$$

We can rewrite the original equation using the components we defined:

$$\begin{aligned}
P^{t+1}_{l,l-k} = \frac{1}{N}\left(\mathbf{u}_1+\mathbf{u}_2+\mathbf{u}_3+\mathbf{u}_4+\mathbf{u}_5\right)^\top M_{l-1:l-k}\Big(\mathbf{v}_1 + \\
\mathbf{v}_2+\mathbf{v}_3+\mathbf{v}_4+\mathbf{v}_5\Big)
\end{aligned}$$

We compute all possible products $\mathbf{u}^\top_i M_{l-1:l-k}\mathbf{v}_j$ for $i, j = 1$ to 5.

- Terms involving $\mathbf{u}_1$:

$$\begin{aligned}
\mathbf{u}_1^\top M_{l-1:l-k}\mathbf{v}_1 &= \rho^2 \left(\mathbf{z}_{l-1}^t\right)^\top M_{l-1}^\top M_{l-1:l-k} M_{l-k-1} \\
&\quad \mathbf{z}_{l-k-1}^t = \rho^2 \cdot P_{l-1,l-k-1}^t \\
\mathbf{u}_1^\top M_{l-1:l-k}\mathbf{v}_2 &= \rho\kappa \left(\mathbf{z}_{l-1}^t\right)^\top M_{l-1}^\top M_{l-1:l-k}\mathbf{z}_{l-k}^t \\
&= \kappa\rho \cdot P_{l-1,l-k}^t \\
\mathbf{u}_1^\top M_{l-1:l-k}\mathbf{v}_3 &= \rho^2 \left(\mathbf{z}_{l-1}^t\right)^\top M_{l-1}^\top M_{l-1:l-k} M_{l-k}^\top \\
&\quad \mathbf{z}_{l-k+1}^t = \rho^2 \cdot P_{l-1,l-k+1}^t \\
\mathbf{u}_1^\top M_{l-1:l-k}\mathbf{v}_4 &= \rho\eta \left(\mathbf{z}_{l-1}^t\right)^\top M_{l-1}^\top M_{l-1:l-k}\mathbf{b}_{l-k-1} \\
&= \rho\eta \cdot B_{l-k,l-1}^t \\
\mathbf{u}_1^\top M_{l-1:l-k}\mathbf{v}_5 &= -\rho^2 \left(\mathbf{z}_{l-1}^t\right)^\top M_{l-1}^\top M_{l-1:l-k} M_{l-k}^\top \\
&\quad \mathbf{b}_{l-k} = -\rho^2 \cdot B_{l-k+1,l-1}^t
\end{aligned}$$

- Terms involving $\mathbf{u}_2$:

$$\begin{aligned}
\mathbf{u}_2^\top M_{l-1:l-k}\mathbf{v}_1 &= \kappa\rho \left(\mathbf{z}_l^t\right)^\top M_{l-1:l-k} M_{l-k-1}\mathbf{z}_{l-k-1}^t \\
&= \kappa\rho \cdot P_{l,l-k-1}^t \\
\mathbf{u}_2^\top M_{l-1:l-k}\mathbf{v}_2 &= \kappa^2 \left(\mathbf{z}_l^t\right)^\top M_{l-1:l-k}\mathbf{z}_{l-k}^t \\
&= \kappa^2 \cdot P_{l,l-k}^t \\
\mathbf{u}_2^\top M_{l-1:l-k}\mathbf{v}_3 &= \kappa\rho \left(\mathbf{z}_l^t\right)^\top M_{l-1:l-k} M_{l-k}^\top \mathbf{z}_{l-k+1}^t \\
&= \kappa\rho \cdot P_{l,l-k+1}^t \\
\mathbf{u}_2^\top M_{l-1:l-k}\mathbf{v}_4 &= \kappa\eta \left(\mathbf{z}_l^t\right)^\top M_{l-1:l-k}\mathbf{b}_{l-k-1} \\
&= \kappa\eta \cdot B_{l-k,l}^t \\
\mathbf{u}_2^\top M_{l-1:l-k}\mathbf{v}_5 &= -\kappa\rho \left(\mathbf{z}_l^t\right)^\top M_{l-1:l-k} M_{l-k}^\top \mathbf{b}_{l-k} \\
&= -\kappa\rho \cdot B_{l-k+1,l}^t
\end{aligned}$$

- Terms involving $\mathbf{u}_3$:

$$\begin{aligned}
\mathbf{u}_3^\top M_{l-1:l-k}\mathbf{v}_1 &= \rho^2 \left(\mathbf{z}_{l+1}^t\right)^\top M_l M_{l-1:l-k} M_{l-k-1} \\
&\quad \mathbf{z}_{l-k-1}^t = \rho^2 \cdot P_{l+1,l-k-1}^t \\
\mathbf{u}_3^\top M_{l-1:l-k}\mathbf{v}_2 &= \rho\kappa \left(\mathbf{z}_{l+1}^t\right)^\top M_l M_{l-1:l-k}\mathbf{z}_{l-k}^t \\
&= \kappa\rho \cdot P_{l+1,l-k}^t \\
\mathbf{u}_3^\top M_{l-1:l-k}\mathbf{v}_3 &= \rho^2 \left(\mathbf{z}_{l+1}^t\right)^\top M_l M_{l-1:l-k} M_{l-k}^\top \\
&\quad \mathbf{z}_{l-k+1}^t = \rho^2 \cdot P_{l+1,l-k+1}^t \\
\mathbf{u}_3^\top M_{l-1:l-k}\mathbf{v}_4 &= \rho\eta \left(\mathbf{z}_{l+1}^t\right)^\top M_l M_{l-1:l-k}\mathbf{b}_{l-k-1} \\
&= \rho\eta \cdot B_{l-k,l+1}^t \\
\mathbf{u}_3^\top M_{l-1:l-k}\mathbf{v}_5 &= -\rho^2 \left(\mathbf{z}_{l+1}^t\right)^\top M_l M_{l-1:l-k} M_{l-k}^\top \\
&\quad \mathbf{b}_{l-k} = -\rho^2 \cdot B_{l-k+1,l+1}^t
\end{aligned}$$

- Terms involving $\mathbf{u}_4$:

$$\begin{aligned}
\mathbf{u}_4^\top M^{l-1:l-k}\mathbf{v}_1 &= \eta\rho\left(\mathbf{b}^{l-1}\right)^\top M^{l-1:l-k}M^{l-k-1}\\
\mathbf{z}^{l-k-1,t} &= \rho\eta\cdot B^t_{l,l-k-1}\\
\mathbf{u}_4^\top M^{l-1:l-k}\mathbf{v}_2 &= \eta\kappa\left(\mathbf{b}^{l-1}\right)^\top M^{l-1:l-k}\mathbf{z}^{l-k,t}\\
&= \kappa\eta\cdot B^t_{l,l-k}\\
\mathbf{u}_4^\top M^{l-1:l-k}\mathbf{v}_3 &= \eta\rho\left(\mathbf{b}^{l-1}\right)^\top M^{l-1:l-k}M^{l-k\top}\mathbf{z}^{l-k+1,t}\\
&= \rho\eta\cdot B^t_{l,l-k+1}\\
\mathbf{u}_4^\top M^{l-1:l-k}\mathbf{v}_4 &= \eta^2\left(\mathbf{b}^{l-1}\right)^\top M^{l-1:l-k}\mathbf{b}^{l-k-1}\\
&= \eta^2\cdot\gamma^{l-1,l-k-1}\\
\mathbf{u}_4^\top M^{l-1:l-k}\mathbf{v}_5 &= -\eta\rho\left(\mathbf{b}^{l-1}\right)^\top M^{l-1:l-k}M^{l-k\top}\mathbf{b}^{l-k}\\
&= -\rho\eta\cdot\gamma^{l-1,l-k}
\end{aligned}$$

- Terms involving $\mathbf{u}_5$:

$$\begin{aligned}
\mathbf{u}_5^\top M^{l-1:l-k}\mathbf{v}_1 &= -\rho^2\left(\mathbf{b}^{l}\right)^\top M^{l}M^{l-1:l-k}M^{l-k-1}\\
\mathbf{z}^{l-k-1,t} &= -\rho^2\cdot B^t_{l+1,l-k-1}\\
\mathbf{u}_5^\top M^{l-1:l-k}\mathbf{v}_2 &= -\rho\kappa\left(\mathbf{b}^{l}\right)^\top M^{l}M^{l-1:l-k}\mathbf{z}^{l-k,t}\\
&= -\kappa\rho\cdot B^t_{l+1,l-k}\\
\mathbf{u}_5^\top M^{l-1:l-k}\mathbf{v}_3 &= -\rho^2\left(\mathbf{b}^{l}\right)^\top M^{l}M^{l-1:l-k}M^{l-k\top}\\
\mathbf{z}^{l-k+1,t} &= -\rho^2\cdot B^t_{l+1,l-k+1}\\
\mathbf{u}_5^\top M^{l-1:l-k}\mathbf{v}_4 &= -\rho\eta\left(\mathbf{b}^{l}\right)^\top M^{l}M^{l-1:l-k}\mathbf{b}^{l-k-1}\\
&= -\rho\eta\cdot\gamma^{l,l-k-1}\\
\mathbf{u}_5^\top M^{l-1:l-k}\mathbf{v}_5 &= \rho^2\left(\mathbf{b}^{l}\right)^\top M^{l}M^{l-1:l-k}M^{l-k\top}\mathbf{b}^{l-k}\\
&= \rho^2\cdot\gamma^{l,l-k}
\end{aligned}$$

By systematically breaking down the original equation into its constituent components and computing all possible interactions between them, we have fully expanded the expression:

$$\begin{aligned}
P^{t+1}_{l,l-k} = {}& \rho^2\cdot P^t_{l-1,l-k-1} + \kappa\rho\cdot P^t_{l-1,l-k} + \rho^2\cdot P^t_{l-1,l-k+1}\\
&+ \rho\eta\cdot B^t_{l-k,l-1} - \rho^2\cdot B^t_{l-k+1,l-1}\\
&+ \kappa\rho\cdot P^t_{l,l-k-1} + \kappa^2\cdot P^t_{l,l-k} + \kappa\rho\cdot P^t_{l,l-k+1}\\
&+ \kappa\eta\cdot B^t_{l-k,l} - \kappa\rho\cdot B^t_{l-k+1,l}\\
&+ \rho^2\cdot P^t_{l+1,l-k+1} + \kappa\rho\cdot P^t_{l+1,l-k} + \rho\eta\cdot B^t_{l-k,l+1}\\
&- \rho^2\cdot B^t_{l-k+1,l+1}\\
&+ \rho\eta\cdot B^t_{l,l-k-1} + \kappa\eta\cdot B^t_{l,l-k} + \rho\eta\cdot B^t_{l,l-k+1}\\
&+ \eta^2\cdot\gamma^{l-1,l-k-1} - \rho\eta\cdot\gamma^{l-1,l-k}\\
&- \rho^2\cdot B^t_{l+1,l-k-1} - \kappa\rho\cdot B^t_{l+1,l-k}\\
&- \rho^2\cdot B^t_{l+1,l-k+1} - \rho\eta\cdot\gamma^{l,l-k-1} + \rho^2\cdot\gamma^{l,l-k}
\end{aligned}\tag{16}$$

On the other hand, when updating $P^t_{l,l-k}$, it is important to account for the cases where l or $l-k$ are 1 or L, since the values of the latent states are fixed in such cases. For instance, the update equation

for the interaction with the input layer, $P_{l,1}^{t+1}$, can be expressed as follows:

$$\begin{aligned} P_{l,1}^{t+1} &= \frac{1}{N}\mathbf{z}^{l,t+1\top} M^{l-1:1}\mathbf{z}^{1,t+1} \\ &= \frac{1}{N}\Big(\kappa \cdot \mathbf{z}^{l,t\top} M^{l-1:1}\mathbf{z}^{1,t} + \rho \cdot \mathbf{z}^{l-1,t\top} M^{l-2:1}\mathbf{z}^{1,t} \\ &\quad + \rho \cdot \mathbf{z}^{l+1,t\top} M^{l:1}\mathbf{z}^{1,t}\Big) \\ &= \kappa \cdot P_{l,1}^{t} + \rho \cdot P_{l-1,1}^{t} + \rho \cdot P_{l+1,1}^{t} \end{aligned} \tag{17}$$

Furthermore, by considering the symmetry of p, we have $P_{1,l}^{t} = P_{l,1}^{t}$. Similarly, the update equation for the interaction with the output layer, $P_{L,L-k}^{t+1}$, is as follows:

$$\begin{aligned} P_{L,L-k}^{t+1} &= \frac{1}{N}\mathbf{z}^{L,t\top} M^{L-1:L-k}\mathbf{z}^{L-k,t+1} \\ &= \frac{1}{N}\big(\kappa \cdot \mathbf{z}^{L,t\top} M^{L-1:L-k}\mathbf{z}^{L-k,t} \\ &\quad + \rho \cdot \mathbf{z}^{L,t\top} M^{L-1:L-k-1}\mathbf{z}^{L-k-1,t} \\ &\quad + \rho \cdot \mathbf{z}^{L,t\top} M^{L-1:L-k+1}\mathbf{z}^{L-k+1,t}\big) \\ &= \kappa \cdot P_{L,L-k}^{t} + \rho \cdot P_{L,L-k-1}^{t} + \rho \cdot P_{L,L-k+1}^{t} \end{aligned} \tag{18}$$

Moreover, $P_{L,L-k} = P_{L-k,L}$.

We now aim to express the above update rules, which involve many combination terms, in a more concise matrix and vector calculation form. Let us carefully examine the structure of the update equations for both the latent states and p. The update equation for $\mathbf{z}$ can be divided into two parts. The first part is the sum of the element-wise product of the latent states $[\mathbf{z}_{l-1}, \mathbf{z}_l, \mathbf{z}_{l+1}]^\top$ and the coefficients $\mathbf{c}_z = [\rho, \kappa, \rho]^\top$. The second part is the sum of the element-wise product of the bias terms $[\mathbf{b}_{l-1}, \mathbf{b}_l]$ and the coefficients $\mathbf{c}_b = [\eta, -\rho]^\top$. The update equation for p, which is derived from the update equation of $\mathbf{z}$, can be seen as the outer product of the latent updates of layer l and another layer $l-k$. The coefficients are fixed, and the values of l and $l-k$ correspond to the indices in the P matrix. Utilizing this, we can rewrite the update rule from Equations 16 to 18 in matrix form as follows:

$$\begin{aligned} P_{l,l-k}^{t+1} &= \mathbf{c}_z^\top P_{l-1:l+1,l-k-1:l-k+1}^{t}\mathbf{c}_z \\ &\quad + \mathbf{c}_b^\top B_{l-k:l-k+1,l-1:l+1}^{t}\mathbf{c}_z \\ &\quad + \mathbf{c}_b^\top B_{l:l+1,l-k-1:l-k+1}^{t}\mathbf{c}_z \\ &\quad + \mathbf{c}_b^\top \Gamma_{l-1:l,l-k-1:l-k}\mathbf{c}_b, \\ P_{1,l}^{t+1\top} &= P_{l,1}^{t+1} = \mathbf{c}_z^\top P_{l-1:l+1,1}^{t}, \text{ and} \\ P_{L,l}^{t+1\top} &= P_{l,L}^{t+1} = \mathbf{c}_z^\top P_{l-1:l+1,L}^{t}, \end{aligned} \tag{19}$$

for $1 < l, l-k < L$.

The update rules for B represent the evolution of the interaction between the latent states $\mathbf{z}$ and the bias terms $\mathbf{b}$.

$$
\begin{aligned}
B^{t+1}_{l+1,l-k} &= \frac{1}{N}{\mathbf{b}^l}^\top M^{l:l-k}\mathbf{z}^{l-k,t+1} \\
&= \frac{1}{N}{\mathbf{b}^l}^\top M^{l:l-k}\Big(\kappa\cdot\mathbf{z}^{l-k,t} + \rho\cdot M^{l-k-1}\mathbf{z}^{l-k-1,t} \\
&\quad + \rho\cdot {M^{l-k}}^\top \mathbf{z}^{l-k+1,t} \\
&\quad + \eta\cdot\mathbf{b}^{l-k-1} - \rho\cdot {M^{l-k}}^\top \mathbf{b}^{l-k}\Big) \\
&= \frac{1}{N}\big(\kappa\cdot\mathbf{b}_l^\top M_{l:l-k}\mathbf{z}^t_{l-k} \\
&\quad + \rho\cdot\mathbf{b}_l^\top M_{l:l-k}M_{l-k-1}\mathbf{z}^t_{l-k-1} \\
&\quad + \rho\cdot\mathbf{b}_l^\top M_{l:l-k}M_{l-k}^\top\mathbf{z}^t_{l-k+1} \\
&\quad + \eta\cdot\mathbf{b}_l^\top M_{l:l-k}\mathbf{b}_{l-k-1} \\
&\quad - \rho\cdot\mathbf{b}_l^\top M_{l:l-k}M_{l-k}^\top\mathbf{b}_{l-k}\big) \\
&= \frac{1}{N}\big(\kappa\cdot\mathbf{b}_l^\top M_{l:l-k}\mathbf{z}^t_{l-k} \\
&\quad + \rho\cdot\mathbf{b}_l^\top M_{l:l-k-1}\mathbf{z}^t_{l-k-1} \\
&\quad + \rho\cdot\mathbf{b}_l^\top M_{l:l-k+1}\mathbf{z}^t_{l-k+1} \\
&\quad + \eta\cdot\mathbf{b}_l^\top M_{l:l-k}\mathbf{b}_{l-k-1} \\
&\quad - \rho\cdot\mathbf{b}_l^\top M_{l:l-k+1}\mathbf{b}_{l-k}\big) \\
&= \kappa\cdot B^t_{l+1,l-k} + \rho\cdot B^t_{l+1,l-k-1} + \rho\cdot B^t_{l+1,l-k+1} \\
&\quad + \eta\cdot\Gamma_{l,l-k-1} - \rho\cdot\Gamma_{l,l-k}
\end{aligned}
\tag{20}
$$

We can simplify the update rule for B as follows:

$$
B^{t+1}_{l,l-k} = B^t_{l,l-k-1:l-k+1}\mathbf{c}_z + \mathbf{c}_b^\top\Gamma_{l-k-1:l-k,l}. \tag{21}
$$

Note that since Γ is a symmetric matrix, swapping the column and row indices in the update equation for B does not alter the result.

D.5 Dynamics of Lengths

Lengths of the Latent States As mentioned earlier, the diagonal elements of P^t represent the lengths of the latent states. That is, $p^t_l = P^t_{l,l}$.

Lengths of the Prediction Errors Let the length of the PE at layer l be denoted by q^t_l. We can express it as follows:

$$
\begin{aligned}
q^t_l &= \big\langle(\delta^t_{i,l})^2\big\rangle \\
&= \frac{1}{N}\|\delta^t_l\|^2 \\
&= \frac{1}{N}(\mathbf{z}^t_l - \hat{\mathbf{z}}^t_l)^\top(\mathbf{z}^t_l - \hat{\mathbf{z}}^t_l),
\end{aligned}
\tag{22}
$$

where $\hat{\mathbf{z}}^t_l = W_{l-1}\mathbf{z}^t_{l-1} + \mathbf{b}_{l-1}$. By substituting the prediction term, q^t_l can be further expanded as follows:

$$
\begin{aligned}
q_l^t &= \frac{1}{N}\Big(\mathbf{z}_l^t - \sigma_w \cdot M_{l-1}\mathbf{z}_{l-1}^t - \mathbf{b}_{l-1}\Big)^\top \\
&\quad \Big(\mathbf{z}_l^t - \sigma_w \cdot M_{l-1}\mathbf{z}_{l-1}^t - \mathbf{b}_{l-1}\Big) \\
&= \frac{1}{N}\Big(\mathbf{z}_l^{t\top} - \sigma_w \cdot \mathbf{z}_{l-1}^{t\top} M_{l-1}^\top - \mathbf{b}_{l-1}^\top\Big) \\
&\quad \Big(\mathbf{z}_l^t - \sigma_w \cdot M_{l-1}\mathbf{z}_{l-1}^t - \mathbf{b}_{l-1}\Big) \\
&= \frac{1}{N}\Big(\mathbf{z}^{l,t\top}(\mathbf{z}^{l,t} - \sigma_w \cdot M^{l-1}\mathbf{z}^{l-1,t} - \mathbf{b}^{l-1}) \\
&\quad - \sigma_w \cdot \mathbf{z}^{l-1,t\top} M^{l-1\top}(\mathbf{z}^{l,t} - \sigma_w \cdot M^{l-1}\mathbf{z}^{l-1,t} - \mathbf{b}^{l-1}) \\
&\quad - \mathbf{b}_{l-1}^\top(\mathbf{z}_l^t - \sigma_w \cdot M_{l-1}\mathbf{z}_{l-1}^t - \mathbf{b}_{l-1})\Big) \\
&= \frac{1}{N}\Big(\mathbf{z}_l^{t\top}\mathbf{z}_l^t - \sigma_w \cdot \mathbf{z}_l^{t\top} M_{l-1}\mathbf{z}_{l-1}^t - \mathbf{z}_l^{t\top}\mathbf{b}_{l-1} \\
&\quad - \sigma_w \cdot \mathbf{z}_{l-1}^{t\top} M_{l-1}^\top \mathbf{z}_l^t \\
&\quad + \sigma_w^2 \cdot \mathbf{z}_{l-1}^{t\top} M_{l-1}^\top M_{l-1}\mathbf{z}_{l-1}^t \\
&\quad + \sigma_w \cdot \mathbf{z}_{l-1}^{t\top} M_{l-1}^\top \mathbf{b}_{l-1} \\
&\quad - \mathbf{b}_{l-1}^\top \mathbf{z}_l^t + \sigma_w \cdot \mathbf{b}_{l-1}^\top M_{l-1}\mathbf{z}_{l-1}^t + \mathbf{b}_{l-1}^\top \mathbf{b}_{l-1}\Big) \\
&= P_{l,l}^t - \sigma_w \cdot P_{l,l-1}^t - B_{l,l}^t \\
&\quad - \sigma_w \cdot P_{l-1,l}^t + \sigma_w^2 \cdot P_{l-1,l-1}^t + \sigma_w \cdot B_{l,l-1}^t \\
&\quad - B_{l,l}^t + \sigma_w \cdot B_{l,l-1}^t + \gamma_{l-1,l-1}
\end{aligned} \tag{23}
$$

The above equation can be simplified as:

$$
q_l^t = \mathbf{c}_q^\top P_{l-1:l,l-1:l}^t \mathbf{c}_q - 2B_{l,l-1:l}^t \mathbf{c}_q + \Gamma_{l-1,l-1}^t, \tag{24}
$$

where the coefficient $\mathbf{c}_q = [-\sigma_w, 1]^\top$.

Lengths of Weight Updates The length of the weight updates at layer l is denoted by r_l^t, and is defined as:

$$
r_l^t = \frac{1}{N^2}\|\Delta W_l^t\|_F^2, \tag{25}
$$

where $\Delta W_l^t = \delta_{l+1}^t \mathbf{z}_l^{t\top}$, with δ_{l+1}^t representing the error signal at the next layer and $\mathbf{z}_l^t$ being the signal at the current layer.

Before proceeding further, we prove a simple lemma for the Frobenius norm:

Lemma (Horn & Johnson, 2012): $\|\mathbf{x}\mathbf{y}^\top\|_F^2 = \|\mathbf{x}\|^2 \cdot \|\mathbf{y}\|^2$, where $\mathbf{x}$ and $\mathbf{y}$ are vectors.

Proof:

$$
\begin{aligned}
\|\mathbf{x}\mathbf{y}^\top\|_F^2 &= \sum_{i,j}(\mathbf{x}\mathbf{y}^\top)_{i,j}^2 = \sum_{i,j}(\mathbf{x}_i\mathbf{y}_j)^2 = \sum_i \mathbf{x}_i^2 \sum_j \mathbf{y}_j^2 \\
&= \|\mathbf{x}\|^2 \cdot \|\mathbf{y}\|^2.
\end{aligned} \tag{26}
$$

Using the above lemma for the Frobenius norm, we can simplify r_l^t as:

$$
\begin{aligned}
r_l^t &= \|\delta_{l+1}^t \mathbf{z}_l^{t\top}\|_F^2 = \|\delta_{l+1}^t\|^2 \cdot \|\mathbf{z}_l^t\|^2 \\
&= q_{l+1}^t \cdot p_l^t
\end{aligned} \tag{27}
$$

Since $\frac{1}{N}\|\Delta \mathbf{b}_l\|^2 = \frac{1}{N}\|\delta_l^t\|^2$, The length of the bias update is equivalent to q_l^t and is therefore omitted.

Table 2: Statistical significance analysis for CIFAR-10 dataset. The table shows accuracy differences (Top-1 acc. diff) and corresponding p-values from Mann-Whitney U tests comparing Meta-PCN against backpropagation (BP) and conventional predictive coding (PC). Positive differences (red) indicate Meta-PCN outperforms the baseline, while negative differences (blue) indicate underperformance. Statistically significant results (p 0.05) are highlighted in bold.

(a) Meta-PCN vs Backpropagation

Architecture	Top-1 acc. diff	p-value
VGG-5	1.73	**0.0079**
VGG-7	0.89	**0.0173**
VGG-9	0.87	**0.0025**
VGG-11	0.59	**0.0303**
VGG-13	0.92	**0.0154**
ResNet-18	0.03	1.0000

(b) Meta-PCN vs Conventional PCN

Architecture	Top-1 acc. diff	p-value
VGG-5	59.86	**0.0079**
VGG-7	68.97	**0.0043**
VGG-9	66.28	**0.0025**
VGG-11	67.13	**0.0043**
VGG-13	77.64	**¡ 0.0001**
ResNet-18	28.05	**0.0079**

E Experimental Setup

Length Dynamics Analysis The simulations described in Section 3 analyzed the length dynamics of latent states and prediction errors during the inference process in a random PCNs ensemble. The dataset consisted of samples from a random unit Gaussian distribution ($(x_i, y_i) \sim \mathcal{N}(\mathbf{0}, I)$). The dataset contained 128 samples processed in a single batch. The number of inference steps (T) was mainly set to 2000 to track the iterative changes in length dynamics. The inference rate was set at 0.05. The model consisted of 30 layers to effectively show the exponential growth in PCN. The latent dimension was set to 100.

Baseline Comparisons. We compare three approaches under identical experimental conditions: (1) Standard backpropagation (BP), (2) Conventional PCN with only feedforward initialization, and (3) Complete the Meta-PCN framework. All methods use the same network architectures, optimization settings, and training procedures, differing only in their learning algorithms.

Meta-PCN Components. Our framework incorporates three key components: (1) **Meta prediction error objective** with feedforward initialization, (2) **Variance-based weight regularization** with target normalization scale of 0.9, and (3) **Blocked sweep updates** using Gauss-Seidel-like alternating layer updates.

Training Configuration. We train all models for 50 epochs using the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.0005. The batch size is set to 128 across all experiments to ensure fair comparison. For PC specific parameters, we use an inference rate (η) of 0.2 and perform 20 inference iterations (T) per learning step.

Implementation Details. All experiments were conducted using the PyTorch **?** framework on CUDA-enabled GPUs rented from Vast.ai. Each experiment is repeated 5 times with different random seeds to ensure statistical reliability.

F Statistical Significance Testing

We performed Mann-Whitney U tests to verify whether Meta-PCN's performance improvements are statistically significant. This non-parametric test is suitable for comparing performance values as it does not assume normality of accuracy distributions.

F.1 Main Method Comparison: BP vs PCN vs Meta-PCN

The statistical analysis reveals several key findings across all datasets. Meta-PCN demonstrates statistically significant improvements over conventional PCNs in virtually all architecture and dataset combinations, with accuracy improvements ranging from 12-79 percentage points. Against backpropagation, Meta-PCN shows statistically significant improvements in most VGG architectures,

Table 3: Statistical significance analysis for CIFAR-100 dataset across both Top-1 and Top-5 accuracy metrics. Tables present accuracy differences and Mann-Whitney U test p-values comparing Meta-PCN performance against backpropagation and conventional PCN baselines. The comprehensive evaluation demonstrates Meta-PCN's robustness across different evaluation criteria and network architectures.

(a) Top-1: Meta-PCN vs Backpropagation

Architecture	Top-1 acc. diff	p-value
VGG-5	3.45	**0.0079**
VGG-7	3.98	**0.0079**
VGG-9	4.87	**0.0079**
VGG-11	1.08	0.0823
VGG-13	4.85	**0.0079**
ResNet-18	-0.02	0.8413

(b) Top-1: Meta-PCN vs Conventional PCN

Architecture	Top-1 acc. diff	p-value
VGG-5	50.96	**0.0159**
VGG-7	48.93	**0.0159**
VGG-9	46.97	**0.0159**
VGG-11	50.70	**0.0095**
VGG-13	57.76	**0.0159**
ResNet-18	14.45	**0.0159**

(c) Top-5: Meta-PCN vs Backpropagation

Architecture	Top-5 acc. diff	p-value
VGG-5	2.23	**0.0079**
VGG-7	3.49	**0.0079**
VGG-9	3.44	**0.0119**
VGG-11	0.62	0.0519
VGG-13	4.06	**0.0159**
ResNet-18	0.55	**0.0159**

(d) Top-5: Meta-PCN vs Conventional PCN

Architecture	Top-5 acc. diff	p-value
VGG-5	67.12	**0.0159**
VGG-7	60.02	**0.0159**
VGG-9	66.71	**0.0159**
VGG-11	72.61	**0.0095**
VGG-13	79.35	**0.0159**
ResNet-18	12.01	**0.0159**

Table 4: Statistical significance analysis for TinyImageNet dataset evaluating Meta-PCN against baseline methods. Results encompass both Top-1 and Top-5 accuracy comparisons across diverse architectures, providing evidence for Meta-PCN's effectiveness on more complex visual recognition tasks with increased class diversity and reduced image resolution.

(a) Top-1: Meta-PCN vs Backpropagation

Architecture	Top-1 acc. diff	p-value
VGG-5	3.84	**0.0159**
VGG-7	2.59	0.0952
VGG-9	3.37	**0.0043**
VGG-11	1.10	**0.0016**
VGG-13	1.86	**0.0040**
ResNet-18	0.39	0.2857

(b) Top-1: Meta-PCN vs Conventional PCN

Architecture	Top-1 acc. diff	p-value
VGG-5	37.64	**0.0195**
VGG-7	41.06	0.1333
VGG-9	38.62	**0.0095**
VGG-11	38.38	**0.0040**
VGG-13	43.70	**0.0001**
ResNet-18	26.72	0.0571

(c) Top-5: Meta-PCN vs Backpropagation

Architecture	Top-5 acc. diff	p-value
VGG-5	4.44	**0.0159**
VGG-7	2.78	0.0952
VGG-9	3.53	**0.0043**
VGG-11	0.49	0.2222
VGG-13	0.18	0.1002
ResNet-18	0.54	**0.0159**

(d) Top-5: Meta-PCN vs Conventional PCN

Architecture	Top-5 acc. diff	p-value
VGG-5	59.93	**0.0159**
VGG-7	64.05	0.1002
VGG-9	63.06	**0.0095**
VGG-11	63.35	**0.0040**
VGG-13	62.58	**0.0159**
ResNet-18	31.02	0.0571

with the performance gap generally decreasing as architecture complexity increases. Notably, ResNet-18 shows minimal or non-significant differences compared to backpropagation, suggesting that Meta-PCN's scalability enables competitive performance with state-of-the-art optimization methods in deeper networks.

The consistent pattern across Top-1 and Top-5 metrics on CIFAR-100 and TinyImageNet further validates Meta-PCN's robustness. The larger improvements observed in conventional PCN comparisons highlight the severity of the pathologies addressed by our framework, while the competitive

Table 5: Ablation study results showing the contribution of each Meta-PCN component. Performance is measured on CIFAR-10 using VGG-13 architecture across 5 independent runs. Statistical significance is evaluated using Mann-Whitney U tests against the full Meta-PCN framework.

Method	**Accuracy (%)**	**p-value**
Meta-PCN	89.53 ± 0.47	–
– Block Sweep GS	89.33 ± 0.56	1.0000
– Normalization	88.25 ± 0.32	$\mathbf{0.0079}$
– Meta-PC Objective	10.01 ± 0.02	$\mathbf{0.0167}$

performance against backpropagation demonstrates the practical viability of biologically plausible learning algorithms.

Conclusion: The comprehensive statistical analysis provides strong evidence that Meta-PCN's performance improvements represent systematic rather than accidental gains. The framework successfully bridges the gap between biological plausibility and computational effectiveness, positioning predictive coding as a viable alternative to backpropagation-based learning.

G Ablation Study

G.1 Ablation Study on Meta-PCN Components

We conducted a systematic ablation study to quantitatively evaluate the contribution of each component in the Meta-PCN framework. Table 5 presents results from training on CIFAR-10 dataset using VGG-13 architecture for 50 epochs.

G.2 Component Analysis

The ablation results reveal distinct contribution levels across Meta-PCN components. Removing the meta-PC objective causes catastrophic performance degradation, dropping accuracy from 89.53% to 10.01% (79.52 percentage points decrease). This dramatic decline suggests that the standard free energy objective suffers from severe pathologies, likely including PE imbalance and gradient starvation that prevent effective learning.

Weight regularization (normalization) removal leads to a 1.28 percentage point decrease (88.25% vs 89.53%), with statistical significance ($p = 0.0079$). This moderate but significant degradation indicates that variance control effectively addresses EVPE and contributes to training stability.

Blocked sweep removal shows minimal impact, with only a 0.20 percentage point decrease (89.33% vs 89.53%) and no statistical significance ($p = 1.0000$). While the convergence improvement from Gauss-Seidel-like updates appears modest in this configuration, the benefit may become more pronounced in deeper architectures.

G.3 Statistical Significance and Component Ranking

The Mann-Whitney U test results establish a clear hierarchy of component importance. The meta-PC objective demonstrates the highest criticality, being statistically essential for functional performance. Weight regularization provides statistically significant but moderate improvements, while blocked sweep updates show non-significant effects under current experimental conditions.

Component Criticality Ranking:

1. **Meta-PC Objective** (most critical): Essential for basic functionality with 79.52%p improvement
2. **Weight Regularization** (moderately critical): Statistically significant 1.28%p improvement
3. **Blocked Sweep GS** (least critical): Non-significant 0.20%p improvement

These results demonstrate that the meta-PC objective is fundamental to Meta-PCN's success, while weight regularization provides important stability benefits. The blocked sweep component, while

theoretically motivated, shows limited practical impact in the current experimental setting. Although we excluded the blocked sweep component from the main text discussion due to its minimal statistical significance, all experimental results for Meta-PCN include the complete framework with all three components.

G.4 Blocked Sweep Updates for PC Inference

Convergence Analysis through Classical Iterative Methods. To understand the convergence limitations of PC inference, we frame the free energy minimization problem in inference within the context of classical iterative methods. PC inference can be viewed as solving the nonlinear stationarity system $\mathbf{F}(\mathbf{z}) = \nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}) = 0$ subject to boundary conditions $\mathbf{z}_1 = \mathbf{x}$ and $\mathbf{z}_L = \mathbf{y}$. The standard inference procedure follows the fixed-point iteration $\mathbf{z}^{t+1} = G_\eta(\mathbf{z}^t)$, where an equilibrium point $\mathbf{z}^*$ satisfies $\mathbf{z}^* = G_\eta(\mathbf{z}^*)$ if and only if $\nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}^*) = 0$.
This formulation reveals that the PCN inference performs simultaneous updates of all latent variables, making it directly analogous to the Jacobi method in classical iterative solvers (Saad, 2003; Golub & Loan, 2013). This analogy provides crucial insights into the inherent limitations of PC inference, as the Jacobi method is well-known for its stringent convergence requirements and inefficient information propagation across network depth.
For linear systems $A\mathbf{u} = \mathbf{b}$, Jacobi convergence typically requires both A and $2D - A$ to be symmetric positive definite (SPD), where D denotes the diagonal of A. In contrast, the Gauss-Seidel method requires only that A be SPD (see, e.g., Varga, 2009; Saad, 2003). More importantly, for consistently ordered SPD matrices, the celebrated convergence rate relationship $\rho(\text{GS}) = \rho(\text{Jacobi})^2$ demonstrates that Gauss-Seidel achieves asymptotically quadratic convergence improvement over Jacobi (Young, 1971; Saad, 2003; Varga, 2009), where $\rho(\cdot)$ denotes the spectral radius of the iteration matrix.
The practical implications for deep PCNs are significant. Because the standard PC update behaves analogously to the Jacobi method, its convergence in deep networks suffers from inherent inefficiencies: information propagates exclusively through simultaneous, layer-wise exchanges, resulting in slow convergence and potential divergence on ill-conditioned problem instances.

PC-Compatible Blocked Sweep Method. Standard PC inference employs simultaneous (Jacobi-like) updates of the form: $\mathbf{z}^{t+1} = \mathbf{z}^t - \eta\,\nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}^t)$, which updates all layers using neighbor information from iteration t. While this approach offers high parallelizability, it propagates information slowly across network depth, contributing to the convergence inefficiencies.
In contrast, Gauss-Seidel (GS) type schemes achieve faster convergence by reusing the most recently computed values within the same iteration, though at the cost of reduced parallelism. Our blocked sweep update strategy offers a principled compromise that preserves the locality and modularity inherent to predictive coding, while leveraging newly computed neighbor states to accelerate information propagation.

Theoretical Foundation and Convergence Analysis. The blocked sweep method possesses a solid theoretical foundation rooted in classical iterative solver theory. The blocked sweep update is mathematically equivalent to a preconditioned gradient step $\mathbf{z}^{t+1} = \mathbf{z}^t - \eta\,\mathbf{M}^{-1}\nabla_{\mathbf{z}}\mathcal{F}(\mathbf{z}^t)$, where the preconditioner $\mathbf{M} = \mathbf{D} + \mathbf{L}$ has a block lower-triangular structure (with $\mathbf{D}$ representing the diagonal blocks and $\mathbf{L}$ the strictly lower block components). This corresponds precisely to a Gauss-Seidel iteration applied to the linearized system.
The convergence advantage is quantified through spectral analysis of the resulting iteration matrix $\mathbf{T}_\eta^{\text{BS}} = \mathbf{I} - \eta\,\mathbf{M}^{-1}\mathbf{H}_*$. For consistently ordered SPD problems, the classical result $\rho(\mathbf{T}_\eta^{\text{GS}}) = \rho(\mathbf{T}_\eta^{\text{Jac}})^2$ (Young, 1971; Saad, 2003; Varga, 2009) demonstrates that blocked sweeps achieve asymptotically quadratic convergence acceleration compared to simultaneous Jacobi-like updates. The method thus provides a principled and practical remedy for the depth-induced convergence slowdown that has historically limited PCN scalability.

H Length Dynamics with Nonlinear Activations

Figure 8 explores the dynamics of latent state lengths ($p^{l,t}$), prediction error lengths ($q^{l,t}$), and weight update lengths ($r^{l,t}$) in nonlinear PCNs across different activation functions. The analysis focuses on common nonlinearity types such as `tanh`, `ReLU`, `SELU`, and `SiLU`, each applied to

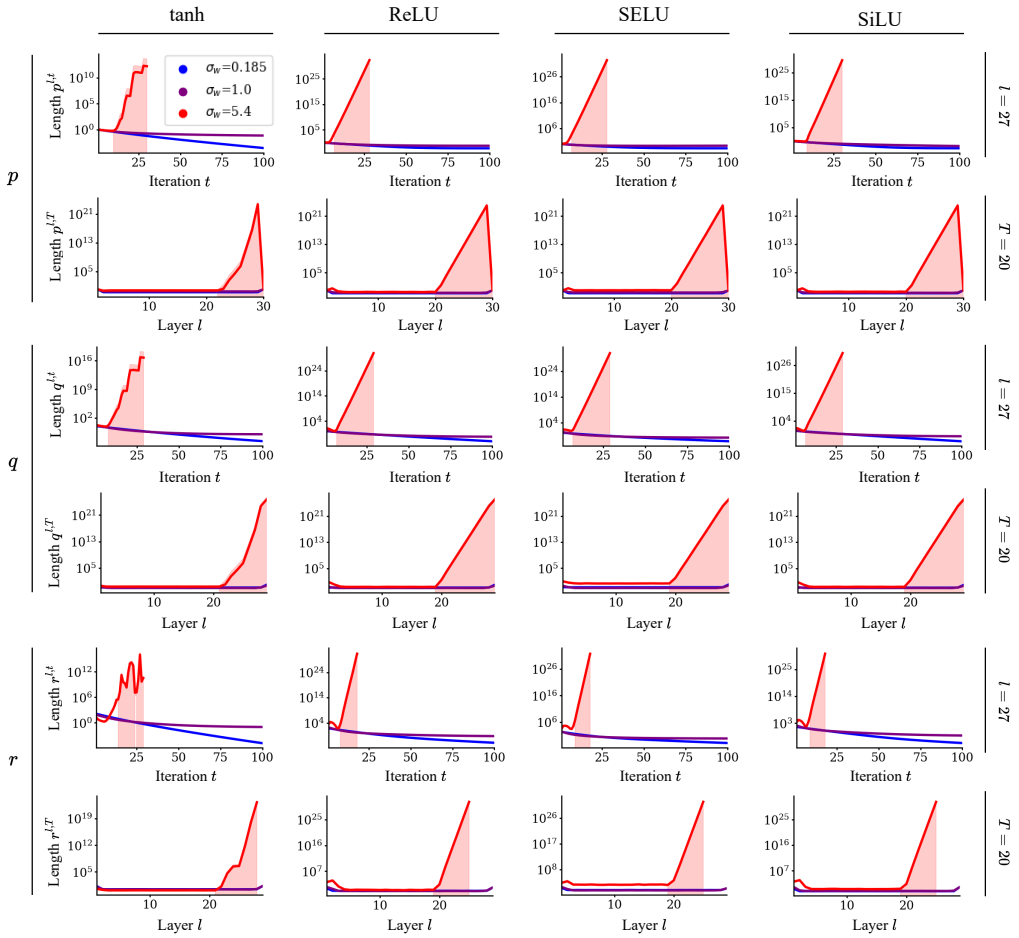


Figure 8: The dynamics of $p^{l,t}, q^{l,t}$, and $r^{l,t}$ and their layer-wise results for nonlinear PCNs. In all subfigures, the results are shown for the cases of $\sigma_w \in \{0.185, 1.0, 5.4\}$ with different colors, and $\sigma_b = 0.1$. Settings not mentioned or indicated are identical to those in Figure 2. Each column represents the applied nonlinear function. The odd rows are the dynamics of p, q, and r, respectively ($l = 27$). The even rows are the layer-wise distribution of p, q, and r, respectively ($T = 20$).

layers with varying weight variances (σ_w). The results show that the dynamics for p, q, and r are highly sensitive to σ_w, even in nonlinearity. The odd rows depict the temporal evolution of p, q, and r at layer $l = 27$, while the even rows display the layer-wise distribution of these values at the $T = 20$ inference step. These subfigures illustrate two key phenomena that occur regardless of the applied nonlinear activation function:

1. In the odd rows, we observe that even with nonlinearity applied, p, q, and r exhibit exponential growth near the output layer when σ_w is large (e.g., $\sigma_w = 5.4$). This suggests that while nonlinear activations are typically expected to provide some degree of constraint on the predicting latent state dynamics by squashing the outputs (e.g., `tanh`), the latent state length growth persists for larger σ_w. This pattern holds across all activation functions examined, indicating that nonlinearity alone is insufficient to counteract the destabilizing effects of high weight variance.
2. The even rows reveal that these exponential growth patterns can emerge early in the inference phase, even at $T = 20$, particularly in deeper layers. The layer-wise distributions of p, q, and r show that the effects of large σ_w extend throughout the network, with prediction errors (q) and weight updates (r) becoming increasingly concentrated toward the output layer. This observation underscores a key challenge in training deep PCNs with nonlinearity. While early inference stages may seem stable, instability can rapidly accumulate in deeper layers due to the interplay between nonlinearity and large weight variances.

Importantly, this analysis highlights the need for regularization strategies, even in networks with nonlinear activations. The exponential growth seen here mirrors the behavior in linear PCNs, suggesting that length regularization and weight variance control are critical to preventing runaway dynamics in both linear and nonlinear architectures. Regularization techniques, such as those introduced in our framework, become essential for maintaining stability, particularly when nonlinearity alone is insufficient to prevent the excessive growth of latent states and prediction errors.

I ADDITIONAL RESULTS

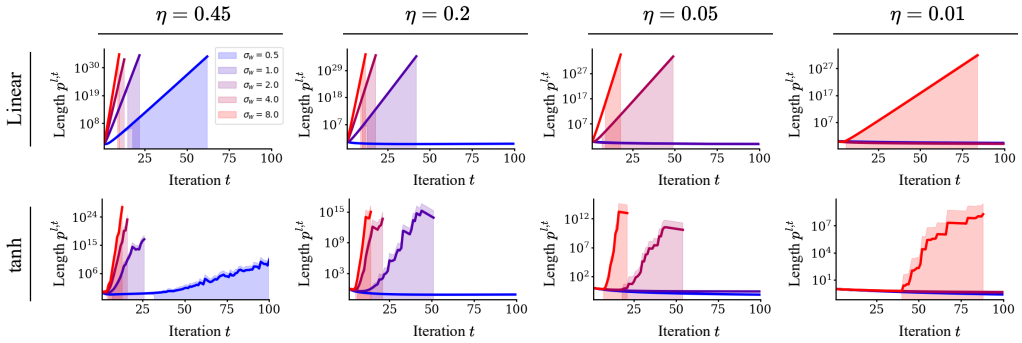


Figure 9: The dynamics of $p^{l,t}, q^{l,t}$, and $r^{l,t}$ for PCNs ($l = 26$ and $L = 30$). *Settings:* In all subfigures, the results are shown for the cases of $\sigma_w \in \{0.5, 1.0, 2.0, 4.0, 8.0\}$ with different colors, and $\sigma_b = 0.3$. Settings not mentioned or indicated are identical to those in Figure 3. *Subfigures:* (a)-(d) Dynamics of $p^{l,t}$ of linear PCNs over the 100 inference steps. (e)-(h) Dynamics of $p^{l,t}$ of nonlinear PCNs (tanh) over the 100 inference steps.

Figure 9 shows the dynamics of the latent state lengths $p^{l,t}$, the prediction error lengths $q^{l,t}$, and the weight update lengths $r^{l,t}$ for PCNs across varying σ_w and η values. The results demonstrate how the network's stability depends heavily on the initialization of the weights and inference rate. In both the linear and nonlinear PCN cases, we observe that as σ_w or η increases, the system becomes more prone to instability, with the exponential growth of the latent state lengths becoming apparent. This is especially visible for higher values of σ_w (e.g., 8.0), where the growth accelerates drastically. This behavior aligns with the theoretical predictions discussed in the paper, where weight variance σ_w significantly influences the dynamics of the latent states. For smaller values of σ_w, such as 1.0, the growth is more contained, allowing the network to maintain more stable latent states across inference steps. However, larger values lead to a divergence in $p^{l,t}$, which necessitates additional regularization techniques, as suggested in our proposed framework.

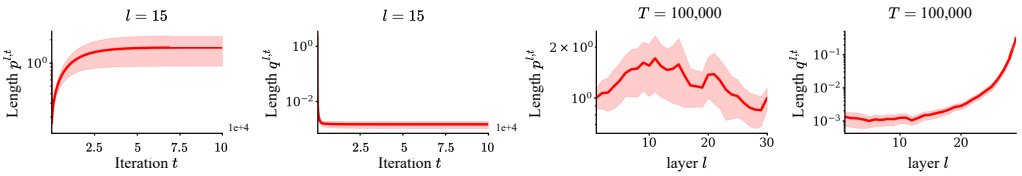


Figure 10: The dynamics of $p^{l,t}$ and $q^{l,t}$ for PCNs ($T = 100,000$). Settings not mentioned or indicated are identical to those in Figure 3.

Figure 10 explores the effect of extremely large inference steps ($T = 100,000$) on the dynamics of $p^{l,t}$ and $q^{l,t}$. Despite the large number of steps, the latent states and prediction errors stabilize after sufficient inference steps when $\sigma_w = 1.0$. However, we also observe that prediction errors tend to concentrate near the output layer, a phenomenon consistent with earlier findings that show concentrated prediction errors as a major challenge in deep PCNs. This stability over extended inference periods suggests that while PCNs can converge in theory, the issue of error concentration near the output layer persists. The results emphasize the need to balance prediction errors to prevent output-layer dominance, a feature crucial in deep networks for robust training.

Figure 11 provides a heatmap visualization showing the effects of σ_w and σ_b on the latent state lengths. For both linear and nonlinear PCNs, we observe that σ_w has a much more significant

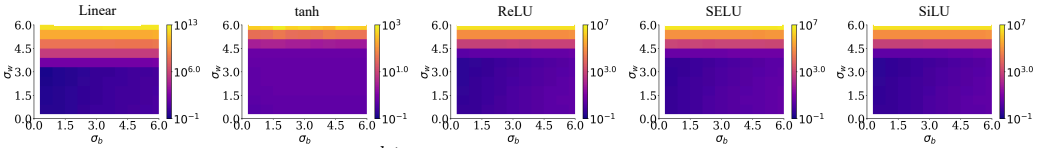


Figure 11: Heatmap plot of length $p^{l,t}$ for linear and nonlinear PCNs. ($\sigma_w \in \{0.6, 1.2, ..., 6.0\}$ and $\sigma_b \in \{0.6, 1.2, ..., 6.0\}$). The total number of inference $T = 10$ and the layer index $l = 15$. Settings not mentioned or indicated are identical to those in Figure 3.

impact on the length dynamics than σ_b. This supports the notion that the variance of the weights is the primary driver of instability, while the bias variance has a more subdued role. The heatmap also reveals that larger σ_w values result in increasingly longer latent state lengths. These findings underline the necessity of controlling weight variance during initialization, as unchecked variance can lead to runaway growth in latent states.

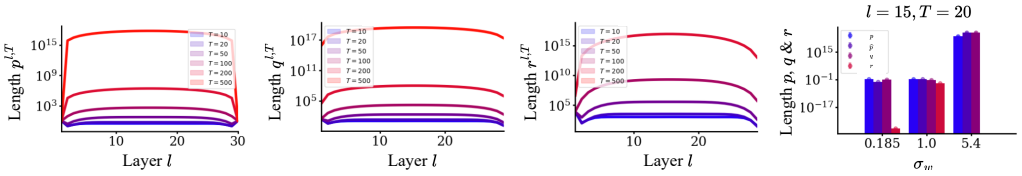


Figure 12: (a)-(c) The layer-wise distribution of $p^{l,t}$, $q^{l,t}$, and $r^{l,t}$ for linear PCNs ($\sigma_w = 5.4$ and $\sigma_b = 0.1$). The results are shown for the cases of $t \in \{10, 20, 50, 100, 200, 500\}$ with different colors. (d) Direct comparison of p, $\hat{p}$, q, and r for $l = 15$ and $t = 20$. $\hat{p}$ represents the length of the prediction. Settings not mentioned or indicated are identical to those in Figure 3.

Figure 12 presents a detailed examination of the layer-wise distribution of $p^{l,t}$, $q^{l,t}$, and $r^{l,t}$ in linear PCNs for different inference steps t. These subfigures aim to capture how the latent state lengths, prediction errors, and weight update magnitudes evolve across different layers and with varying t. In Figure 12a-c, for $\sigma_w = 5.4$, we observe a exponential growth pattern in the values of p, q, and r across all layers, particularly as T increases. This growth is expected, given that larger weight variances typically result in larger latent state dynamics, leading to a cascading effect on prediction errors and weight updates. The increase in p, q, and r with inference steps indicates that the internal representations become increasingly unstable as the inference phase progresses without proper regularization. Figure 12d highlights a direct comparison between p, $\hat{p}$ (the length of predictions), q, and r for layer $l = 15$ at $T = 20$. Across all values of σ_w, we observe that p, $\hat{p}$, and q remain within a similar range, though their values become more exaggerated for higher σ_w values. Notably, r, which represents the weight update length, shows explosive growth when $\sigma_w = 5.4$, making it impractical to display fully. This behavior confirms that the higher values of σ_w without regularization lead to unstable weight updates. Interestingly, for lower σ_w values (e.g., $\sigma_w = 0.185$), r remains small, indicating that proper initialization can contain these dynamics. However, $\sigma_w = 1$ shows a more moderate, controllable behavior in r. This figure emphasizes the need for length regularization and highlights the trade-off between network capacity (as influenced by σ_w) and the necessity of stability through regularization techniques.

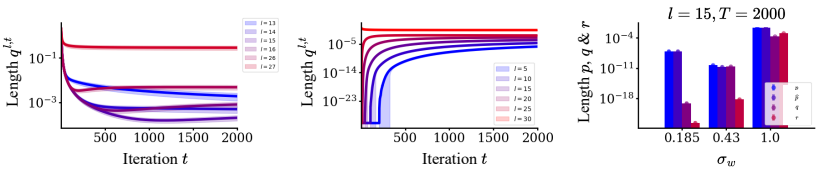


Figure 13: (a) & (b) The dynamics of $q^{l,t}$ for linear PCNs ($\sigma_w = 1$ and $\sigma_b = 0.1$). The results are shown for the cases of different layer index l with different colors. (c) Direct comparison of p, $\hat{p}$, q, and r for $l = 15$ and $T = 2000$. $\hat{p}$ represents the length of the prediction. Settings not mentioned or indicated are identical to those in Figure 2.

Figure 13 illustrates the dynamics of $q^{l,t}$ (prediction error lengths) in linear PCNs, with $\sigma_w = 1$ and $\sigma_b = 0.1$, across different layer indices and inference steps. In Figure 13a and b, we see that the prediction error length ($q^{l,t}$) increases significantly as we approach the output layer (indicated by red lines). This trend is consistent with the concentration of prediction errors in deeper layers, a challenge observed in deep PCNs that affects the learning capacity of intermediate layers. Conversely, the prediction error length in earlier layers (indicated by blue lines) starts small. It grows gradually with further inference steps, reinforcing the observation that early layers tend to stabilize more effectively than deeper layers. Figure 13c compares p, $\hat{p}$, q, and r for layer $l = 15$ at $T = 2000$. The comparison shows how the dynamics of prediction lengths ($\hat{p}$), latent state lengths (p), and the magnitude of r (weight update length) become highly dependent on σ_w. As noted earlier, the growth in r with larger σ_w values can lead to instability, stressing the need for controlled weight updates via regularization mechanisms.