Local Predictions, Global Learning: Radial Basis Function Networks for Spatially-Aware Predictive Coding

Hou Hei Lam LINHX22@MAILS.TSINGHUA.EDU.CN *Tsinghua University, Beijing, China*

Editors: List of editors' names

Abstract

Predictive coding networks offer a biologically plausible alternative to backpropagation through local error minimization. However, standard implementations rely on fully connected layers, unlike the sparse, spatially organized connectivity of the brain. We introduce Radial Basis Predictive Coding Networks (RBF-PCN), which use Gaussian receptive fields to enforce spatial locality in predictions and error propagation, reducing computational complexity. Experiments show that RBF-PCN maintains competitive performance with standard predictive coding in shallow models, and that both predictive coding variants exhibit superior equivariance to translations and rotations compared to backpropagation.

Keywords: Predictive Coding, Equivariant Representations

1. Introduction

Equivariant representations—where network activations transform predictably under input transformations—constitute a fundamental principle in biological and artificial vision systems (Perich et al., 2025; Saraf et al., 2025; Cohen and Welling, 2016). While convolutional neural networks achieve translation equivariance through weight sharing (LeCun et al., 2002), recent evidence suggests that local connectivity patterns play an equally crucial role in maintaining geometric structure (Legare et al., 2025).

Predictive coding (PC) networks offer a biologically grounded alternative to backpropagation by implementing hierarchical Bayesian inference through local error minimization (Rao and Ballard, 1999). However, existing models typically use fully connected layers, where each neuron predicts all preceding activities, which contrasts with the sparse and spatially organized connectivity of biological circuits.

We introduce Radial Basis Predictive Coding Networks (RBF-PCN), which impose Gaussian receptive fields to enforce spatial locality in predictions and error propagation. Connection strength decreases smoothly with distance, preserving geometric structure and supporting equivariance, consistent with evidence that local lateral connections preserve the visuotopic ordering of the visual field (Alexander et al., 1999).

Our contributions are threefold: (1) spatial locality constraints in predictive coding enhance locality compared to fully-connected architectures, (2) RBF-constrained connectivity maintains competitive performance while reducing computational complexity, and (3) predictive coding maintains higher representation equivariance under spatial transformations.

2. Related Work

Predictive Coding Networks Predictive coding networks (PCNs) implement hierarchical Bayesian inference through local computation (Rao and Ballard, 1999). Each layer l maintains value neurons $\mathbf{v}^l$ and computes top-down predictions $\hat{\mathbf{v}}^l = f(W^l \mathbf{v}^{l+1})$, yielding prediction errors $\boldsymbol{\epsilon}^l = \mathbf{v}^l - \hat{\mathbf{v}}^l$. The network minimizes variational free energy $\mathcal{F}$ through iterative updates:

$$\mathcal{F} = \sum_l \|\boldsymbol{\epsilon}^l\|^2, \quad \frac{d\mathbf{v}^{l+1}}{dt} = -\frac{\partial \mathcal{F}}{\partial \mathbf{v}^{l+1}} \propto \boldsymbol{\epsilon}^{l+1} - (W^l)^T \boldsymbol{\epsilon}^l \tag{1}$$

Whittington and Bogacz (2017) proved PCNs approximate backpropagation using only local computations, with recent work achieving competitive performance while maintaining biological plausibility (Millidge et al., 2022; Salvatori et al., 2021; Song et al., 2024). However, standard PCNs employ fully-connected layers, lacking the spatial locality fundamental to cortical organization.

Sparse and Local Connectivity Biological neural circuits exhibit sparse, spatially-organized connectivity optimized for computational efficiency. The MICrONS Consortium mapped ∼75,000 neurons, revealing "like-to-like" connectivity where local connections are stronger between functionally similar neurons (mic, 2025; Ding et al., 2025). Recent advances show sparse predictive coding develops motion-sensitive receptive fields using local connectivity (Millidge et al., 2024), while structured sparsity achieves substantial computational gains—up to 100× throughput improvements (Mocanu et al., 2018; Hunter et al., 2022). Crucially, this local connectivity preserves spatial relationships essential for equivariant representations, as symmetric local connections naturally maintain geometric structure under spatial transformations (Cohen and Welling, 2016).

3. Methodology

We propose Radial Basis Predictive Coding Network (RBF-PCN), which constrains predictive coding dynamics through Gaussian receptive fields, enforcing spatial locality and gradient sparsity while preserving core PC inference mechanisms. The training procedure of RBF-PCN is shown in Appendix A.

RBF-Constrained Connectivity Standard PC networks use dense connectivity where layer $l+1$ neurons predict all layer l activities. We restrict this through spatial receptive fields defined by Gaussian RBFs. For neurons arranged on 2D grids with coordinates (i, j) and (k, m) in layers l and $l+1$ respectively, the RBF influence function is:

$$G_{km,ij} = \exp\left(-\frac{\|(k, m) - (i, j)\|^2}{2\sigma^2}\right), \tag{2}$$

where σ controls receptive field width. This creates connectivity mask $\mathbf{M} \in \mathbb{R}^{N_{l+1} \times N_l}$ with $M_{km,ij} = G_{km,ij}$.

Modified Predictive Coding Dynamics The RBF mask modulates forward predictions and backward error propagation. Top-down prediction becomes $\hat{\mathbf{v}}^l = f\big((\mathbf{W}^l \odot \mathbf{M}^l)\mathbf{v}^{l+1}\big)$ with error $\boldsymbol{\epsilon}^l = \mathbf{v}^l - \hat{\mathbf{v}}^l$, where $\mathbf{W}^l$ are generative weights and $\odot$ denotes element-wise multiplication. Spatially constrained inference updates become:

$$\frac{d\mathbf{v}^{l+1}}{dt} = -\frac{\partial \mathcal{F}}{\partial \mathbf{v}^{l+1}} \propto \boldsymbol{\epsilon}^{l+1} - (\mathbf{W}^l \odot \mathbf{M}^l)^T \boldsymbol{\epsilon}^l. \tag{3}$$

This reduces computational complexity from $O(N_{l+1} \times N_l)$ to $O(N_{l+1} \times k)$ where k is the average connections per neuron after thresholding.

Sparse Learning Through Hard Masking We enforce sparsity using a binary threshold mask $\mathbf{S}^l$ that selects connections with RBF value above τ. Weight updates apply only to active connections as $\Delta W^l_{km,ij} = \alpha_W \epsilon^l_{ij} v^{l+1}_{km}$. During forward passes, effective weights are $(\mathbf{W}^l \odot \mathbf{S}^l) \odot \mathbf{M}^l$, preserving smooth gradients under sparsity. The binary mask $\mathbf{S}^l$ enforces structural locality, while the continuous mask $\mathbf{M}^l$ modulates strength by spatial proximity, jointly maintaining predictive-coding dynamics while reducing computation.

4. Experiment and Results

We evaluate RBF-PCN against standard PCN and backpropagation (BP) baselines through classification and equivariance experiments. Additional details are in Appendix B.

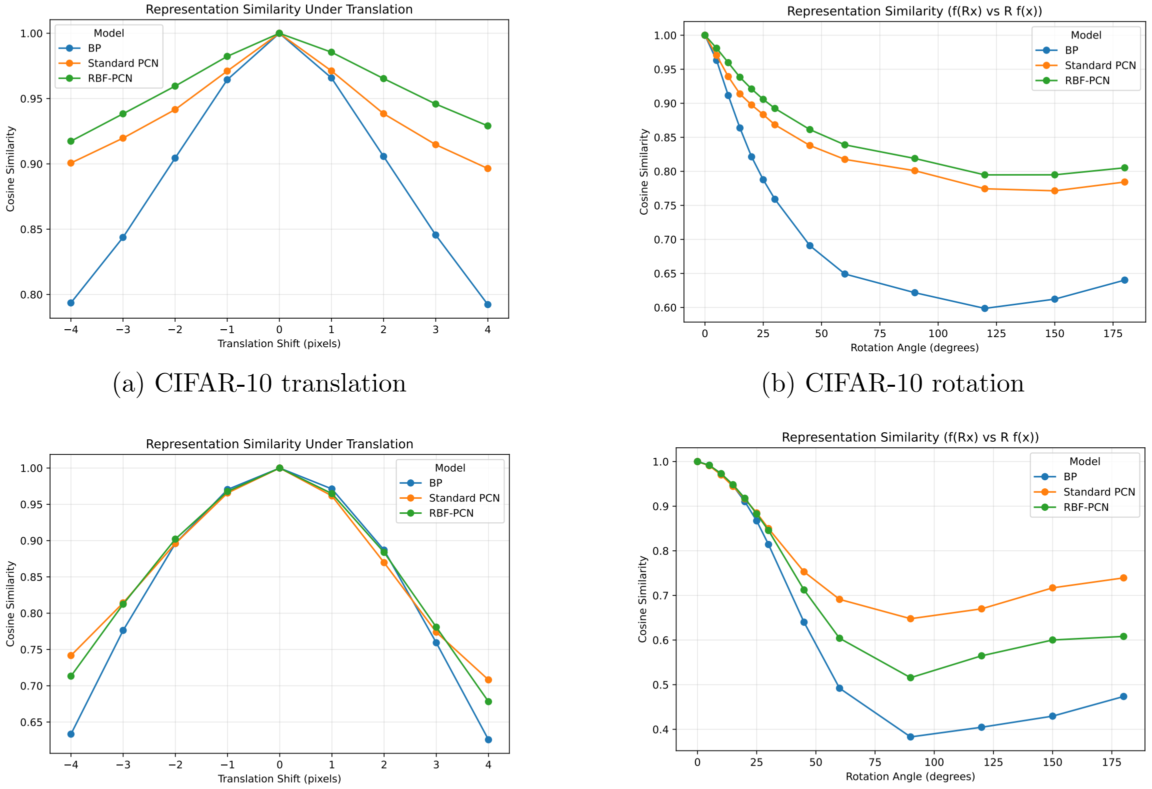
Supervised Classification We train MLPs with 3-12 hidden layers (512 units each) on MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky et al., 2009) using three methods: (1) Standard PCN, (2) RBF-PCN, and (3) BP. All models minimize squared error loss with batch size 512, trained for 10 epochs (MNIST) and 30 epochs (CIFAR-10).

Table 1: Classification accuracy across methods, datasets, and network depths. Standard deviation shown after ±.

Dataset	Method	Depth 3	Depth 6	Depth 9	Depth 12
MNIST	BP	0.98 ± 0.0006	**0.98 ± 0.0017**	**0.98 ± 0.0032**	**0.97 ± 0.0049**
	PCN	**0.99 ± 0.0007**	**0.98 ± 0.0018**	0.94 ± 0.0082	0.17 ± 0.0669
	RBF-PCN	0.98 ± 0.0004	0.97 ± 0.0007	0.11 ± 0.0000	0.12 ± 0.0025
CIFAR-10	BP	**0.55 ± 0.0064**	**0.55 ± 0.0003**	**0.54 ± 0.0037**	**0.53 ± 0.0015**
	PCN	**0.55 ± 0.0027**	0.48 ± 0.0048	0.42 ± 0.0040	0.28 ± 0.0065
	RBF-PCN	0.53 ± 0.0056	0.46 ± 0.0045	0.10 ± 0.0028	0.10 ± 0.0013

RBF-PCN matches standard PCN performance in shallow networks (3–6 layers) while reducing computational cost through local connectivity. This shows that local connectivity preserves representational capacity for moderately complex tasks. Both PCN variants degrade in deeper networks, whereas backpropagation remains stable across all tested depths.

Equivariance Analysis We evaluate equivariance of learned representations under spatial transformations. For translations, we compute similarity $S_{\text{trans}} = \text{sim}(f(T_s(x)), T_s(f(x)))$, where T_s is a shift by $s \in \{\pm 4\}$ pixels and sim denotes cosine similarity. For rotations, we



(a) CIFAR-10 translation

(b) CIFAR-10 rotation

(c) MNIST translation

(d) MNIST rotation

Figure 1: Cosine Similarity of depth-6 models on CIFAR-10 and MNIST, showing the response of learned representations to translation and rotation.

compute $S_{\text{rot}} = \text{sim}(f(R_\theta(x)), R_\theta(f(x)))$, with R_θ a rotation by $\theta \in [0°, 180°]$. We also report equivariance error, defined as $E_{\text{trans}} = \|f(T_s(x)) - T_s(f(x))\|/\|f(x)\|$ for translation and $E_{\text{rot}} = \|f(R_\theta(x)) - R_\theta(f(x))\|/\|f(x)\|$ for rotation.

Figure 1 shows depth-6 results. Translation equivariance (panels a,c) is similar across methods, with PCN variants slightly outperforming BP. For rotation (panels b,d), RBF-PCN yields the highest similarity on CIFAR-10, while standard PCN leads on MNIST. Both PCN variants exceed BP, indicating better preservation of geometric relationships.

5. Conclusion

We introduced Radial Basis Predictive Coding Networks (RBF-PCN), which impose spatial locality via Gaussian receptive fields while retaining core predictive-coding dynamics. RBF-PCN matches shallow-network classification accuracy of standard PCN while reducing computational cost via sparse connectivity. Both predictive-coding models show stronger equivariance to spatial transformations than backpropagation, indicating that biological spatial constraints can improve efficiency and geometric representation learning. Future work should explore deeper architectures and adaptive receptive fields for better scalability.

References

Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058): 435–447, 2025.

David M Alexander, Phil Sheridan, Paul D Bourke, Otto Konstandatos, and James J Wright. Global and local symmetry of the primary visual cortex: derivation of orientation preference. *Submitted to Cerebral Cortex*, 1999.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Zhuokun Ding, Paul G Fahey, Stelios Papadopoulos, Eric Y Wang, Brendan Celii, Christos Papadopoulos, Andersen Chang, Alexander B Kunin, Dat Tran, Jiakun Fu, et al. Functional connectomics reveals general wiring rule in mouse visual cortex. *Nature*, 640(8058): 459–469, 2025.

Kevin Hunter, Lawrence Spracklen, and Subutai Ahmad. Two sparsities are better than one: unlocking the performance benefits of sparse–sparse networks. *Neuromorphic Computing and Engineering*, 2(3):034004, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Antoine Legare, Olivier Ribordy, Paul De Koninck, Antoine Allard, and Patrick Desrosiers. Function aligns with geometry in locally connected neuronal networks. *bioRxiv*, pages 2025–08, 2025.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafal Bogacz, and Thomas Lukasiewicz. Predictive coding: Towards a future of deep learning beyond backpropagation? *arXiv preprint arXiv:2202.09467*, 2022.

Beren Millidge, Mufeng Tang, Mahyar Osanlouy, Nicol S Harper, and Rafal Bogacz. Predictive coding networks for temporal prediction. *PLOS Computational Biology*, 20(4): e1011183, 2024.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Matthew G Perich, Devika Narain, and Juan A Gallego. A neural manifold view of the brain. *Nature Neuroscience*, pages 1–16, 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.

Tommaso Salvatori, Yuhang Song, Yujian Hong, Lei Sha, Simon Frieder, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. Associative memories via predictive coding. *Advances in neural information processing systems*, 34:3874–3886, 2021.

Sonica Saraf, J Anthony Movshon, and SueYeon Chung. Variations in neuronal selectivity create efficient representational geometries for perception. *bioRxiv*, pages 2025–06, 2025.

Yuhang Song, Beren Millidge, Tommaso Salvatori, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz. Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature neuroscience*, 27(2):348–358, 2024.

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.

Appendix A. RBF-PCN Training Procedure

The key steps of the RBF-PCN training procedure are summarized in Algorithm 1. Continuous masks $\mathbf{M}^l$ and binary masks $\mathbf{S}^l$ are computed first, followed by iterative inference and weight updates.

Algorithm 1 RBF-PCN Training

```
Initialize weights $\mathbf{W}^l$, RBF width $\sigma$, threshold $\tau$
Compute continuous masks $\mathbf{M}^l$ ;                                // Eq. 2
Compute binary masks $\mathbf{S}^l$ ;
for each training batch $(\mathbf{x}, \mathbf{y})$ do
    for $t \leftarrow 1$ to $T$ do
        Forward: $\hat{\mathbf{v}}^l = f\big((\mathbf{W}^l \odot \mathbf{S}^l \odot \mathbf{M}^l)\,\mathbf{v}^{l+1}\big)$ ;
        Compute errors: $\boldsymbol{\epsilon}^l = \mathbf{v}^l - \hat{\mathbf{v}}^l$ ;
        Update value neurons ;                                            // Eq. 3
    end
    Update weights ;
    Apply hard masking: $\mathbf{W}^l \leftarrow \mathbf{W}^l \odot \mathbf{S}^l$ ;
end
```

Appendix B. Experimental Details

B.1. Experimental Overview

To evaluate the effectiveness of our proposed methods, we conduct comprehensive experiments across three diverse tasks that test different aspects of neural network learning capabilities. These tasks span supervised learning with discrete outputs, unsupervised reconstruction of noisy data, and feature learning from complex textures, providing a systematic assessment of model performance across varying network depths and architectures.

B.2. Tasks and Datasets

B.2.1. Task 1: Supervised Classification

The first experimental task focuses on multi-class classification using standard computer vision benchmarks. We evaluate performance on MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky et al., 2009) datasets, where all images are resized to 28×28 for consistency. Models are trained to predict one-hot encoded labels by minimizing the squared error loss, defined as half the squared L2 norm between predicted and target labels. Training is conducted for 10 epochs on MNIST and 30 epochs on CIFAR-10 to ensure convergence while maintaining computational efficiency.

B.2.2. Task 2: Image Denoising

The second task examines unsupervised learning capabilities through image reconstruction. We employ MNIST images (resized to 28×28) corrupted with Gaussian noise of standard deviation 0.5, creating a challenging denoising scenario. Symmetric encoder–decoder models are trained to minimize reconstruction loss, defined as half the squared L2 norm between reconstructed and clean images. This task requires models to learn meaningful internal representations that can distinguish signal from noise, with training conducted over 30 epochs.

B.2.3. Task 3: Unsupervised Feature Learning

The third task investigates the ability to learn meaningful representations from complex textures without supervision. We use $32{\times}32$ patches extracted from the Describable Textures Dataset (DTD) (Cimpoi et al., 2014), which provides diverse texture patterns that challenge feature learning algorithms. Models are trained to minimize texture reconstruction loss, defined as half the squared L2 norm between reconstructed and input patches, over 30 epochs.

B.3. Model Architectures and Implementation

B.3.1. Network Architectures

For the classification tasks, we employ multi-layer perceptrons (MLPs) with varying depths from 3 to 12 hidden layers, each containing 512 units. This architecture choice allows us to systematically evaluate how model depth affects learning dynamics and performance. The denoising task utilizes MLP autoencoders with symmetric encoder-decoder structures, ensuring that the encoding and decoding processes have equivalent representational capacity. For texture learning, we implement convolutional autoencoders (CNNs) that can effectively capture spatial hierarchies and local texture patterns through their convolutional structure.

Table 2 summarizes the detailed architectural specifications for each model type across different tasks. The MLP architectures maintain consistent hidden layer sizes of 512 units, while CNN architectures follow a progressive channel expansion pattern. For autoencoder tasks, symmetric encoder-decoder structures are employed with appropriate bottleneck layers to encourage meaningful representation learning.

Table 2: Detailed network architecture specifications for different tasks and model types.

Task	**Architecture Component**	**Specification**
Classification (MLP)	Model Type	Multi-Layer Perceptron
	Depth Range	3–12 hidden layers
	Layer Sizes	Input → 512 → · · · → 512 → Output
	Hidden Units	512 units per layer
Denoising (MLP-AE)	Model Type	MLP Autoencoder
	Depth Range	3–12 hidden layers per encoder/decoder
	Encoder Layers	Input → 512 → · · · → 512 → 256
	Decoder Layers	256 → 512 → · · · → 512 → Input
	Hidden Units	512 units per layer
	Bottleneck Size	256 units
Feature Learning (CNN-AE)	Model Type	CNN Autoencoder
	Depth	3 layers per encoder/decoder
	Encoder Progression	3×32×32 → 16×16×16 → 32×8×8 → 64×4×4 → 64×2×2 (bottleneck)
	Decoder Progression	64×2×2 → 64×4×4 → 32×8×8 → 16×16×16 → 3×32×32 (output)
	Kernel Sizes	3×3 for all layers
	Stride	2 (downsampling/upsampling)
	Padding	1 for all layers
	Output Activation	Tanh (range [-1, 1])

B.3.2. Data Normalization

All input data undergoes standardization using dataset-specific normalization parameters to ensure stable training dynamics and consistent convergence behavior across different datasets. Table 3 presents the normalization parameters used for each dataset.

B.3.3. RBF-PCN Configuration

The RBF-PCN models are configured with a receptive field width of $\sigma = 6.0$, which determines the spatial extent of radial basis function activations. The sparsity parameter is

Table 3: Dataset normalization parameters used for input preprocessing.

Dataset	**Mean (μ)**	**Standard Deviation (σ)**
MNIST	[0.5]	[0.5]
CIFAR-10	[0.4914, 0.4822, 0.4465]	[0.2023, 0.1994, 0.2010]
DTD Textures	[0.5]	[0.5]

set to $\tau = 10^{-4}$, encouraging sparse representations while maintaining sufficient model expressivity. During training, we employ $T = 20$ inference steps to ensure that the predictive coding dynamics reach a stable state before parameter updates.

B.3.4. Training Parameters and Experimental Setup

Learning rates are carefully tuned for different model components, with representation learning rates α_v selected from $\{0.01, 0.001\}$ and parameter learning rates α_W chosen from $\{0.001, 0.0001\}$. All experiments are conducted on a single NVIDIA A800 GPU with 80 GB memory, providing sufficient computational resources for the deep architectures under investigation. To ensure statistical reliability, each experimental condition is evaluated across three different random seeds, and results are reported based on the best hyperparameter combination and best epoch for each model and task.

B.4. Evaluation Metrics

B.4.1. Classification Performance

Classification performance is measured using accuracy, which represents the proportion of correctly classified examples. This metric provides a direct assessment of the model's ability to learn discriminative features and make accurate predictions on unseen data.

B.4.2. Reconstruction Quality Assessment

For reconstruction tasks, we employ three complementary metrics that capture different aspects of image quality. Mean Squared Error (MSE) measures the average squared pixelwise differences between reconstructed and target images, with lower values indicating better reconstruction fidelity. Peak Signal-to-Noise Ratio (PSNR) provides a decibel-scale measure of the signal-to-error ratio, where higher values indicate superior reconstruction quality. The Structural Similarity Index (SSIM) quantifies perceptual similarity between images on a scale from 0 to 1, with values closer to 1 indicating better preservation of structural information that correlates with human visual perception.

B.5. Complete Experimental Results

Table 4 presents a comprehensive summary of experimental results across all three tasks and varying network depths. The better-performing method in each experimental condition is highlighted in bold font to facilitate comparison. The results reveal that both methods face significant challenges when applied to very deep architectures, particularly those with 9-12 layers, suggesting fundamental limitations in training dynamics for such configurations.

Table 4: Complete experimental results across 3 tasks. The better method in each row is shown in bold. Both methods struggle with very deep architectures (9–12 layers).

Task	**Dataset/Model**	**Depth**	**PCN Performance**	**RBF-PCN Performance**	**Metric**
Classification	MNIST/MLP	3	**0.99 ± 0.0007**	0.98 ± 0.0004	Accuracy
		6	**0.98 ± 0.0018**	0.97 ± 0.0007	Accuracy
		9	**0.94 ± 0.0082**	0.11 ± 0.0000	Accuracy
		12	**0.17 ± 0.0669**	0.12 ± 0.0025	Accuracy
	CIFAR-10/MLP	3	**0.55 ± 0.0027**	0.53 ± 0.0056	Accuracy
		6	**0.48 ± 0.0048**	0.46 ± 0.0045	Accuracy
		9	**0.42 ± 0.0040**	0.10 ± 0.0028	Accuracy
		12	**0.28 ± 0.0065**	0.10 ± 0.0013	Accuracy
Denoising	MNIST/MLP-AE	3	**13.39 ± 0.19**	12.84 ± 0.07	PSNR (dB)
		6	**8.47 ± 0.15**	5.94 ± 0.00	PSNR (dB)
		9	**5.95 ± 0.00**	5.94 ± 0.01	PSNR (dB)
		12	**5.94 ± 0.01**	**5.94 ± 0.01**	PSNR (dB)
		3	**0.67 ± 0.02**	0.65 ± 0.01	SSIM
		6	**0.52 ± 0.03**	0.41 ± 0.01	SSIM
		9	**0.42 ± 0.00**	**0.42 ± 0.01**	SSIM
		12	**0.41 ± 0.01**	**0.41 ± 0.00**	SSIM
		3	**0.05 ± 0.00**	**0.05 ± 0.00**	MSE
		6	**0.14 ± 0.00**	0.25 ± 0.00	MSE
		9	**0.25 ± 0.00**	**0.25 ± 0.00**	MSE
		12	**0.25 ± 0.00**	**0.25 ± 0.00**	MSE
Texture Learning	DTD/CNN-AE	3	15.73 ± 0.41	**15.78 ± 0.35**	PSNR (dB)
		3	**0.73 ± 0.02**	**0.73 ± 0.02**	SSIM
		3	**0.03 ± 0.00**	**0.03 ± 0.00**	MSE

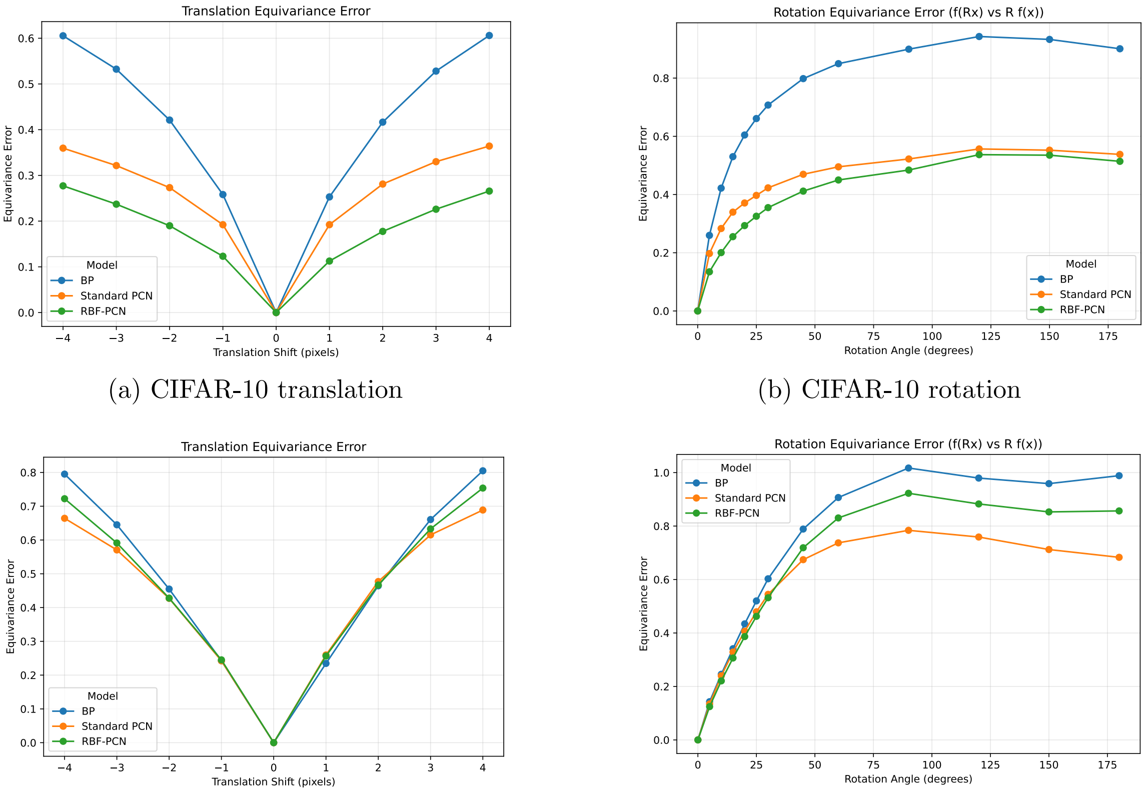
B.6. Equivariance Analysis

To assess the robustness and geometric properties of learned representations, we conduct a detailed analysis of equivariance under spatial transformations for the classification task. This analysis examines how well the learned features preserve their response patterns when input images undergo translation and rotation transformations, which is crucial for understanding the geometric understanding encoded in the network representations.

Figure 2 presents the equivariance error measurements for depth-6 models evaluated on both CIFAR-10 and MNIST datasets. The analysis encompasses both translation and rotation transformations, providing insights into how the different methods handle spatial invariances. These results complement and align with the equivariance analysis presented in the main text, offering additional granular details about the spatial transformation properties of the learned representations.

B.7. Reconstruction Results

Here in Figure 3 and 4, we show the denoising and reconstruction results of depth-3 models for both Task 2 (Denoising) and Task 3 (Texture Learning), comparing PCN and RBF-PCN reconstructions with the original inputs.



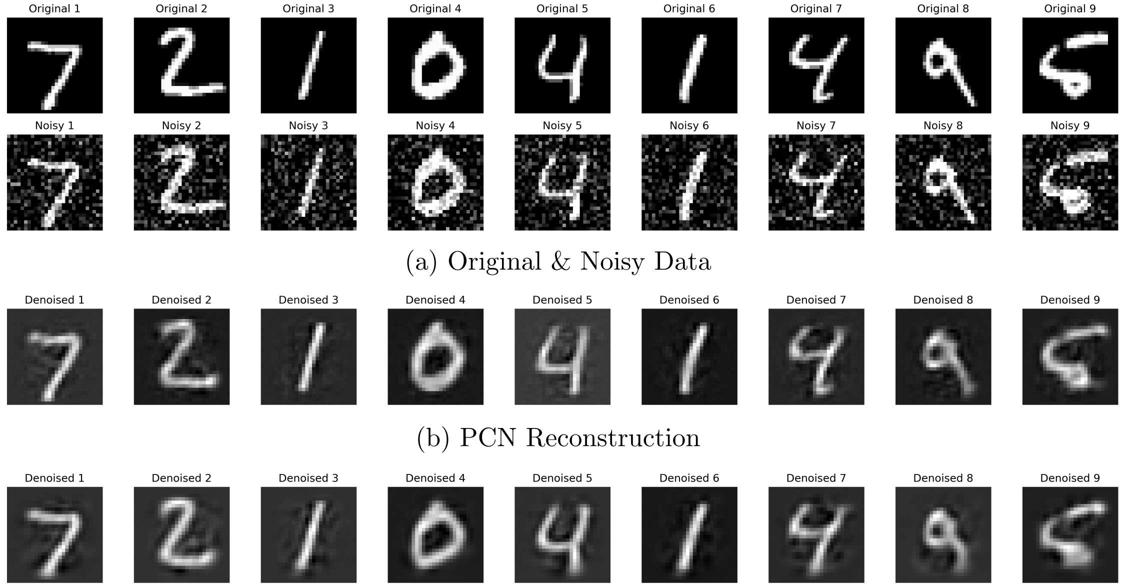
(a) CIFAR-10 translation

(b) CIFAR-10 rotation

(c) MNIST translation

(d) MNIST rotation

Figure 2: Equivariance error analysis for depth-6 models on CIFAR-10 and MNIST datasets, measuring the response of learned representations to translation and rotation transformations. The results demonstrate the spatial transformation properties of different methods and align with the cosine similarity analysis presented in Figure 1.

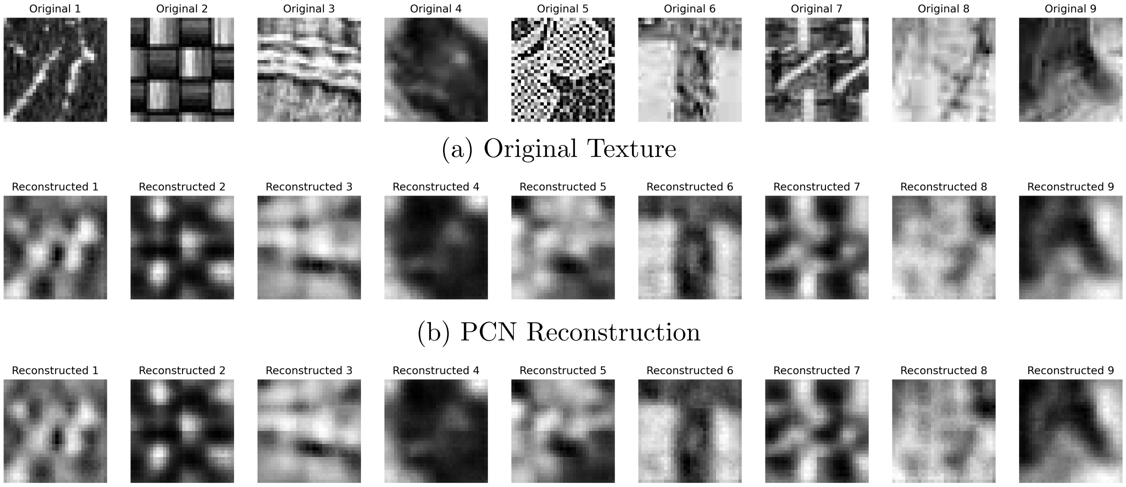


(a) Original & Noisy Data

(b) PCN Reconstruction

(c) RBF-PCN Reconstruction

Figure 3: Denoising results of depth-3 models.



(a) Original Texture

(b) PCN Reconstruction

(c) RBF-PCN Reconstruction

Figure 4: Texture Learning results of depth-3 models.