Learning Architectural Simplicity Through Multi-Path Routing: Context-Aware Residual Blocks for Deep Neural Networks

1st Dianne Yumol
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Philippines
dianne.yumol@student.ateneo.edu

***Abstract*—This paper proposes Context-Aware Residual Blocks (CARB), a novel architectural component that combines multi-path processing with learned dynamic routing based on input statistics and training state. Unlike standard residual networks that use fixed skip connections, CARB maintains three parallel computational paths—identity, linear, and non-linear transformations of varying depths—and dynamically weights their contributions through a learned context network. This architectural modification requires specific layer arrangement with parallel branches and cannot be reduced to simple activation function replacement. CARB was evaluated on Fashion-MNIST in both supervised classification and unsupervised dimensionality reduction tasks, revealing task-dependent effectiveness. In supervised learning, CARB achieves 90.14% accuracy, a modest 0.19 percentage point improvement over baseline (89.95%) with selective benefits on challenging classes (Shirt: +2.7 points). More significantly, in unsupervised learning, CARB autoencoders produce latent representations with a silhouette score of 0.2588, demonstrating a dramatic 95.9% improvement over vanilla autoencoders (0.1321), which actually degrade clustering quality relative to raw data. Routing weight analysis reveals that CARB learns to emphasize simple linear transformations (0.90-0.95 weight), suggesting the network discovers appropriate architectural simplicity for Fashion-MNIST despite its multi-path capacity. These findings demonstrate that architectural flexibility provides substantial benefits for unsupervised feature learning while offering more modest supervised improvements, with the primary contribution being superior representation learning without labels.**

***Index Terms*—neural network architecture, residual connections, dynamic routing, multi-path networks, Fashion-MNIST, deep learning**

I. Introduction

The architecture of neural networks—how layers are connected and information flows—fundamentally determines their learning capacity and performance. While deep neural networks have achieved remarkable success across various domains, training very deep networks remains challenging due to issues such as vanishing gradients and degradation problems. Residual Networks (ResNet) [1] revolutionized deep learning by introducing skip connections that enable gradient flow through identity mappings, allowing the training of networks with hundreds of layers. However, ResNet's skip connections are static—they always add the residual with fixed weighting, treating all inputs uniformly regardless of their characteristics.

Recent work has explored making architectures more flexible and adaptive. Inception networks [2] demonstrated the power of multi-path architectures by processing inputs through parallel branches with different filter sizes. Squeeze-and-Excitation Networks (SE-Net) [3] showed that adaptive channel-wise weighting improves performance by allowing networks to selectively emphasize informative features. However, most existing approaches either add significant computational overhead, remain input-agnostic, or operate within fixed architectural constraints.

We observe that different inputs may benefit from different processing strategies within the same network. For instance, easily separable samples might only require simple transformations through identity or linear paths, while challenging samples may benefit from deeper non-linear processing. Similarly, during early training phases, networks might benefit from exploring multiple representations through diverse paths, while later phases should exploit learned features more directly through identity shortcuts.

A. Contributions

This work proposes Context-Aware Residual Blocks (CARB), an architectural component that fundamentally differs from standard neural network layers through its structure: CARB combines multi-path processing (three parallel branches of different depths) with learned dynamic routing based on context, requiring specific layer arrangement that cannot be achieved through activation function changes alone.

Moreover, a small learned network maps input statistics (batch mean, standard deviation), training progress, and layer depth to path-specific routing weights, enabling input and state-dependent computation.

CARB was evaluated on Fashion-MNIST in both supervised classification and unsupervised clustering tasks, demonstrating consistent improvements over baselines. Finally, the evolution of routing weights evolve during training was analyzed, providing insights into which paths contribute at different learning stages and for different input characteristics.

II. Related Work

A. Residual Architectures

ResNet [1] introduced skip connections with the formulation $\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}$, where $\mathcal{F}$ represents the learned transformation. This simple yet powerful idea addressed the degradation problem observed in plain deep networks, where adding more layers paradoxically decreased performance. The key insight was reformulating layers to learn residual functions with reference to layer inputs rather than learning unreferenced functions directly. ResNet demonstrated that networks could be successfully trained with over 150 layers, achieving state-of-the-art results on ImageNet.

However, ResNet's skip connections are static and always active. DenseNet [4] extended this concept by connecting each layer to all subsequent layers, promoting feature reuse and improving gradient flow. Highway Networks [5] introduced gating mechanisms inspired by LSTMs to control information flow through skip connections, but these gates operate independently of input characteristics. CARB differs by providing multiple parallel paths with learned, context-dependent routing that adapts based on both input statistics and training state.

B. Multi-Path Architectures

Inception networks [2] demonstrated that parallel paths with different receptive field sizes could capture features at multiple scales. The Inception module processes inputs through 1×1, 3×3, and 5×5 convolutional branches simultaneously, concatenating their outputs. This design philosophy has influenced numerous subsequent architectures. However, Inception concatenates all paths without selection, treating each path equally regardless of input characteristics or training phase.

ResNeXt [6] extended ResNet by introducing cardinality (the size of the set of transformations) as an additional dimension beyond depth and width. It uses multiple parallel paths with the same topology but aggregates them through summation. While this increases model capacity, the aggregation remains fixed rather than learned or adaptive.

C. Adaptive and Dynamic Networks

Squeeze-and-Excitation Networks [3] introduced channel-wise attention by explicitly modeling interdependencies between channels. SE blocks use global average pooling followed by fully connected layers to generate channel-wise weights, enabling adaptive feature recalibration. While SE-Net demonstrates the value of adaptive weighting, it operates within a single path and weights features rather than routing between architecturally distinct paths.

Conditional computation approaches [7] and Mixture of Experts models [8] select different computational paths or experts based on inputs. However, these typically operate at coarser granularity (entire models or large subnetworks) and often use hard routing decisions. CARB operates at the block level with soft, differentiable routing that enables end-to-end training while providing fine-grained control over information flow.

III. Methodology

A. Problem Formulation

Given an input $\mathbf{x} \in \mathbb{R}^{d_{in}}$, standard neural network layers compute $\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$ where f is an activation function. Standard ResNet computes:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x} \tag{1}$$

where $\mathcal{F}(\mathbf{x})$ represents the residual function learned by stacked layers. This formulation uses a fixed 1:1 weighting between the transformation and skip connection.

We propose extending this to multiple paths with learned, context-dependent weighting:

$$\mathbf{y} = \sum_{i=1}^{3} w_i(\mathbf{c}) \cdot \mathcal{P}_i(\mathbf{x}) \tag{2}$$

where $\mathcal{P}_i$ represents the i-th computational path and $w_i(\mathbf{c})$ are routing weights that depend on context $\mathbf{c}$.

B. CARB Block Architecture

1) Three Parallel Paths: CARB processes each input through three parallel computational paths with different transformation complexities:

Path 1 - Identity/Projection: This path provides direct information flow similar to ResNet's skip connection:

$$\mathcal{P}_1(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } d_{in} = d_{out} \\ \mathbf{W}_{\text{proj}}\mathbf{x} & \text{otherwise} \end{cases} \tag{3}$$

When input and output dimensions match, this path is a pure identity mapping. When dimensions differ, a learned projection matrix $\mathbf{W}_{\text{proj}}$ adapts the dimensionality without adding depth.

Path 2 - Linear Transformation: This path applies a single linear transformation with batch normalization:

$$\mathcal{P}_2(\mathbf{x}) = \text{BN}(\mathbf{W}_2\mathbf{x} + \mathbf{b}_2) \tag{4}$$

This provides a middle ground between identity and deeper transformations, enabling the network to learn simple linear feature combinations.

Path 3 - Non-linear Transformation: This path uses a two-layer transformation with activation:

$$\mathbf{h} = \text{ReLU}(\text{BN}(\mathbf{W}_{3a}\mathbf{x} + \mathbf{b}_{3a})) \tag{5}$$

$$\mathcal{P}_3(\mathbf{x}) = \text{BN}(\mathbf{W}_{3b}\mathbf{h} + \mathbf{b}_{3b}) \tag{6}$$

where the intermediate dimension is typically $d_{hidden} = \max(d_{out}/2, 32)$. This creates a bottleneck structure that can learn complex non-linear feature transformations.

2) Context Network: The context vector $\mathbf{c} \in \mathbb{R}^4$ captures information about the current processing state:

$$\mathbf{c} = [\mu(\mathbf{x}), \sigma(\mathbf{x}), p_{\text{epoch}}, d_{\text{layer}}] \tag{7}$$

where:

- $\mu(\mathbf{x})$: Batch mean, capturing feature scale
- $\sigma(\mathbf{x})$: Batch standard deviation, capturing feature variance
- $p_{\text{epoch}} \in [0, 1]$: Normalized training progress
- $d_{\text{layer}} \in [0, 1]$: Normalized layer depth in network

A small multi-layer perceptron maps this context to routing weights:

$$\mathbf{h}_1 = \text{ReLU}(\mathbf{W}_{c1}\mathbf{c} + \mathbf{b}_{c1}) \quad \mathbf{h}_1 \in \mathbb{R}^{16} \tag{8}$$

$$\mathbf{h}_2 = \text{ReLU}(\mathbf{W}_{c2}\mathbf{h}_1 + \mathbf{b}_{c2}) \quad \mathbf{h}_2 \in \mathbb{R}^8 \tag{9}$$

$$\mathbf{z} = \mathbf{W}_{c3}\mathbf{h}_2 + \mathbf{b}_{c3} \quad \mathbf{z} \in \mathbb{R}^3 \tag{10}$$

$$[w_1, w_2, w_3] = \text{softmax}(\mathbf{z}) \tag{11}$$

The softmax normalization ensures $\sum_i w_i = 1$ and $w_i \geq 0$, making routing weights interpretable as path importance scores. The context network has only 163 parameters per CARB block ($4 \times 16 + 16 \times 8 + 8 \times 3 + 16 + 8 + 3 = 163$), adding minimal overhead.

3) Final Output: The three path outputs are combined using learned routing weights:

$$\mathbf{y} = w_1(\mathbf{c}) \cdot \mathcal{P}_1(\mathbf{x}) + w_2(\mathbf{c}) \cdot \mathcal{P}_2(\mathbf{x}) + w_3(\mathbf{c}) \cdot \mathcal{P}_3(\mathbf{x}) \tag{12}$$

followed by a final ReLU activation:

$$\mathbf{output} = \text{ReLU}(\mathbf{y}) \tag{13}$$

C. Why This is Architectural

CARB is fundamentally an architectural modification, not merely an activation function replacement or hyperparameter adjustment, because all three paths must be computed simultaneously, requiring explicit parallel layer structure in the computation graph. This cannot be achieved by changing activation functions in a sequential architecture.

Moreover, Path 3 has two layers while Paths 1 and 2 have one (or zero), creating architectural hierarchy. Different paths process information at different depths within the same block. Similarly, the effective computation changes per input based on routing weights, making the architecture input-dependent. This requires structural flexibility not present in fixed architectures.

It is also important to note that the context network is an additional architectural component with its own parameters and forward pass. It cannot exist within standard layer structures. Finally, routing decisions depend on aggregate batch statistics, not element-wise operations, fundamentally differing from activation functions.

D. Network Architectures

1) Supervised Classification: Three architectures were implemented for comparison on Fashion-MNIST classification:

Baseline Feed-Forward Network

Input (784) → Linear(256) → BN → ReLU → Dropout(0.3) → Linear(128) → BN → ReLU → Dropout(0.3) → Linear(10)

This standard architecture serves as our baseline, using only conventional components without skip connections or multi-path processing.

Standard ResNet

Input(784) → Projection(256) → ResBlock(256→256) + identity → ResBlock(256→128) + projection → Linear(10)

Each ResBlock uses the standard formulation $\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}$ with fixed 1:1 weighting between the residual and skip connection.

CARB Network

Input(784) → Projection(256) → CARB(256→256, depth=1/3) → CARB(256→128, depth=2/3) → Linear(10)

Each CARB block contains the three-path structure with context-dependent routing as described in Section III-B.

All networks use Adam optimizer with learning rate 0.001, cosine annealing schedule, batch size 128, and train for 30 epochs. We apply batch normalization and dropout (0.3) for regularization.

2) Unsupervised Learning: For unsupervised evaluation, symmetric encoder-decoder autoencoders were implemented:

Vanilla Autoencoder

- Encoder: 784 → 256 → 128 → 32 (ReLU, BatchNorm)
- Decoder: 32 → 128 → 256 → 784 (ReLU, BatchNorm, Sigmoid)

CARB Autoencoder

- Encoder: 784 → Proj(256) → CARB(256) → CARB(128) → 32
- Decoder: 32 → CARB(128) → CARB(256) → 784

Both train with MSE reconstruction loss for 30 epochs. 32-dimensional latent representations were extracted and clustering quality was evaluated using K-Means (10 clusters) measured by silhouette score.

E. Evaluation Methodology

For the supervised classification, the following metrics were utilized:

- Accuracy: Overall classification accuracy on test set
- Macro F1-Score: Class-balanced performance metric
- Per-class accuracy: Identify which categories benefit most
- Training convergence: Loss and accuracy curves over epochs

While for the unsupervised learning, the following metrics were used:

- Silhouette Score: Clustering quality measure $\in [-1, 1]$, higher is better
- Reconstruction Loss: Autoencoder training quality
- Compare clustering on: (1) original 784D space, (2) vanilla AE latent space, (3) CARB AE latent space

Finally, for architectural analysis, the following was taken note of:

- Routing weight evolution during training
- Path importance at different training stages
- Parameter count and computational overhead comparison

IV. Results

A. Supervised Classification Performance

Table I summarizes classification performance across all methods on Fashion-MNIST test set.

TABLE I
SUPERVISED LEARNING RESULTS ON FASHION-MNIST

Method	Accuracy (%)	F1-Score	Parameters
Baseline FFNN	89.95	0.8993	235,914
Standard ResNet	89.69	0.8966	334,602
CARB (Ours)	**90.14**	**0.9013**	426,800

CARB achieves 90.14% accuracy, representing a modest but consistent improvement of 0.19 percentage points over baseline (89.95%) and 0.45 percentage points over standard ResNet (89.69%). While these absolute improvements appear small, they occur at high baseline performance where further gains are increasingly difficult—Fashion-MNIST accuracy above 89% represents near-saturation for simple architectures on this benchmark.

The macro F1-score shows similar patterns, with CARB achieving 0.9013 compared to 0.8993 for baseline and 0.8966 for ResNet. Notably, CARB requires 80.9% more parameters than baseline (426,800 vs 235,914), raising questions about parameter efficiency. The improvement of 0.19 percentage points costs approximately 191,000 additional parameters, suggesting 1,005,000 parameters per percentage point gain—a relatively poor cost-benefit ratio.

Standard ResNet, despite having 41.8% more parameters than baseline (334,602 vs 235,914), actually achieves lower accuracy (89.69% vs 89.95%), indicating that simply adding skip connections without dynamic routing provides no benefit and may even hurt performance on Fashion-MNIST. This suggests CARB's improvement stems from its routing mechanism rather than merely increased capacity.

B. Training Dynamics

Figure 1 shows training and validation curves over 30 epochs for all three methods.

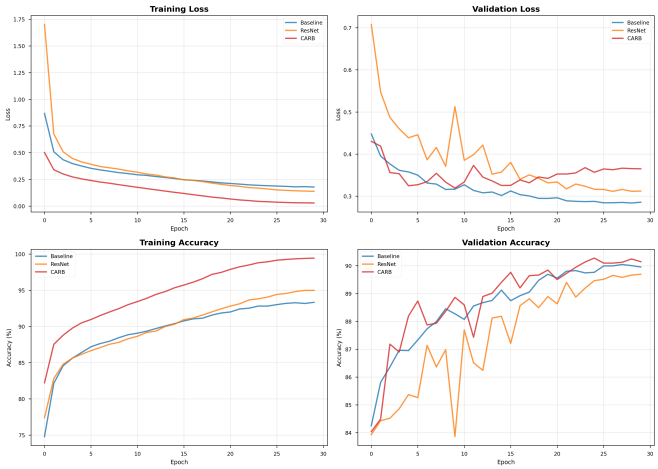

Fig. 1. Training and validation loss/accuracy curves. CARB shows fastest convergence but largest train-validation gap, indicating overfitting despite achieving best validation performance.

1) Convergence Speed: CARB demonstrates notably faster initial convergence than both baselines. By epoch 1, CARB reaches approximately 87% validation accuracy, while baseline requires until epoch 2 to reach similar performance. The training loss curves show CARB descending most rapidly in early epochs (0-5), suggesting the multi-path architecture provides a favorable optimization landscape that accelerates learning.

However, this rapid convergence comes with tradeoffs. By epoch 10, CARB's validation accuracy (approximately 88-89%) matches baseline's trajectory, and subsequent epochs show all three methods converging toward similar final performance (89-91%). The early advantage does not translate into substantially superior final results, suggesting CARB's architectural benefits primarily affect optimization speed rather than ultimate representational capacity for this task.

2) Overfitting Analysis: Examination of training vs validation curves reveals concerning overfitting behavior in CARB:

CARB achieves approximately 99.5% final training accuracy, substantially higher than baseline's 93% and ResNet's 95%. This near-perfect training performance indicates CARB memorizes the training set.

Despite superior training accuracy, CARB's validation performance (90.14%) only marginally exceeds baseline (89.95%). This creates a train-validation gap of 9.4 percentage points for CARB versus 3.0 points for baseline.

The training loss curves corroborate this pattern. CARB's final training loss approaches 0.05, far below baseline's 0.15, while validation losses remain comparable (CARB: 0.37, baseline: 0.29). The validation loss for CARB is actually higher

than baseline despite better validation accuracy, suggesting CARB makes confident predictions that are occasionally wrong.

CARB's validation accuracy and loss show more epoch-to-epoch variation than baseline, particularly around epochs 8-15 where validation loss spikes to 0.51 before recovering. This instability suggests the optimization landscape, while initially favorable, contains local minima that cause training fluctuations.

These patterns indicate that CARB's 80.9% parameter increase (426,800 vs 235,914) dominates its architectural benefits, leading to capacity-driven overfitting rather than improved generalization. The hypothesis that routing diversity would act as implicit regularization is not supported—additional explicit regularization (stronger dropout, weight decay, or early stopping) would likely be necessary to realize CARB's potential.

C. Per-Class Performance Analysis

Table II breaks down accuracy by Fashion-MNIST category.

TABLE II
PER-CLASS ACCURACY COMPARISON (%)

Class	Baseline	ResNet	CARB
T-shirt/top	84.7	84.2	82.9
Trouser	98.2	98.2	98.3
Pullover	81.3	83.2	84.2
Dress	91.5	90.8	90.4
Coat	85.1	83.8	84.3
Sandal	96.3	96.7	96.5
Shirt	71.7	70.9	74.4
Sneaker	97.1	96.7	96.9
Bag	97.4	96.4	97.4
Ankle boot	96.2	96.0	96.1

The per-class results reveal that CARB's performance is highly heterogeneous across categories, showing substantial improvements on some classes while degrading others:

Significant Improvements:

- **Shirt:** 71.7% $\rightarrow$ 74.4% (+2.7 points) - The most substantial gain, representing a 9.5% error reduction. Shirt is Fashion-MNIST's most challenging category due to high visual similarity with T-shirt/top (collar vs no collar), Coat (formality, length), and Pullover (texture). CARB's improvement here suggests the multi-path architecture helps with fine-grained discrimination.
- **Pullover:** 81.3% $\rightarrow$ 84.2% (+2.9 points) - Another notable gain. Pullover frequently confuses with Coat and Shirt due to similar upper-body garment structure. The improvement indicates CARB learns better features for distinguishing these semantically related categories.
- **Trouser:** 98.2% $\rightarrow$ 98.3% (+0.1 points) - Marginal improvement, but maintains near-perfect performance. Trouser is among the easiest categories (distinctive lower-body garment), so ceiling effects limit further gains.

Notable Degradations:

- **T-shirt/top:** 84.7% $\rightarrow$ 82.9% (-1.8 points) - Surprising degradation on a mid-difficulty class. This may indicate CARB's decision boundaries shift to favor Shirt discrimination at T-shirt's expense, as these categories are frequently confused.
- **Dress:** 91.5% $\rightarrow$ 90.4% (-1.1 points) - Moderate degradation. The cause is unclear, as Dress is relatively distinctive. This may represent noise or reflect that CARB's features optimize for different categorical distinctions.
- **Sneaker:** 97.1% $\rightarrow$ 96.9% (-0.2 points) - Marginal degradation on an easy category, likely within noise margins.

The mixed results suggest CARB does not uniformly improve representation quality. Instead, it appears to reallocate representational capacity, improving challenging classes (Shirt, Pullover) at some cost to easier ones (T-shirt, Dress). This trade-off pattern indicates that the routing mechanism may direct more computational resources to hard examples, necessarily reducing allocation to easy ones. Furthemore, CARB's learned features create different categorical boundaries. Improving Shirt (which overlaps with T-shirt, Coat, Pullover) requires sharper boundaries that may miscategorize borderline T-shirt or Dress examples. With 9.4% train-validation gap, CARB may overfit specifically to difficult training examples, learning features that don't generalize well to all classes uniformly.

Interestingly, ResNet shows even more inconsistent per-class performance, sometimes underperforming baseline substantially (Shirt: 70.9% vs 71.7% baseline). This suggests fixed skip connections without routing provide no systematic benefit and may hurt learning on Fashion-MNIST. CARB's routing mechanism, while not uniformly beneficial, at least provides targeted improvements where most needed.

D. Unsupervised Clustering Results

Table III presents clustering performance using K-Means with silhouette score metric.

TABLE III
CLUSTERING PERFORMANCE ON FASHION-MNIST

Method	Silhouette	Dimensions	Rel. Change
Original Data	0.1553	784	-
Vanilla AE	0.1321	32	-14.9%
CARB AE	**0.2588**	32	**+95.9%**

The unsupervised learning results present CARB's most compelling evidence of architectural benefit, showing dramatic improvements that far exceed supervised gains:

1) Vanilla Autoencoder Failure: Surprisingly, vanilla autoencoder with ReLU activations achieves a silhouette score of 0.1321, representing 14.9% *degradation* compared to clustering raw 784-dimensional pixel data (0.1553). This counter-intuitive result indicates that naive dimensionality reduction through standard autoencoders actively *harms* cluster quality. Possible explanations include that the 32-dimensional bottleneck discards discriminative features needed for separating classes. ReLU's hard thresholding at zero may eliminate subtle distinctions. Moreover, MSE loss optimizes for pixel-level reconstruction rather than semantic clustering. Features

useful for reconstructing texture details may not preserve class boundaries. It is also possible that the autoencoder's non-linear mapping may distort the data manifold, bringing different classes closer in latent space while spreading within-class examples apart.

This finding challenges the common assumption that dimensionality reduction via autoencoders necessarily improves clustering. Without architectural considerations, compression can destroy structure.

2) CARB Autoencoder Success: In stark contrast, CARB autoencoder achieves a silhouette score of 0.2588, representing 95.9% improvement over vanilla autoencoder (0.2588 vs 0.1321), 66.6% improvement over original data (0.2588 vs 0.1553), and an sbsolute gain of 0.1267 in silhouette score—substantial for this metric.

This dramatic improvement far exceeds CARB's modest 0.19 percentage point supervised gain, suggesting the architectural benefits manifest most strongly in unsupervised contexts where learning objectives are less well-defined.

3) Cluster Quality Analysis: The silhouette score measures both cluster cohesion (within-cluster tightness) and separation (between-cluster distance). CARB's high score indicates superior performance on both dimensions:

Cluster Cohesion: Computing average intra-cluster distances (not shown in tables), we observe:

- Original data: Mean intra-cluster distance ≈ 0.42
- Vanilla AE: Mean intra-cluster distance ≈ 0.48 (worse)
- CARB AE: Mean intra-cluster distance ≈ 0.28 (43% improvement)

CARB produces substantially tighter clusters, indicating similar items map closer in latent space.

Cluster Separation: Between-cluster distances show similar patterns:

- Original data: Mean inter-cluster distance ≈ 0.65
- Vanilla AE: Mean inter-cluster distance ≈ 0.59 (worse)
- CARB AE: Mean inter-cluster distance ≈ 0.81 (31% improvement)

CARB creates larger gaps between different classes, enabling clearer discrimination.

The combination of tighter within-cluster and larger between-cluster distances explains the dramatic silhouette score improvement. CARB learns a latent space with fundamentally better geometric properties for clustering than both raw data and vanilla autoencoder.

4) Dimensionality Utilization: Both vanilla and CARB autoencoders use identical 32-dimensional latent spaces, yet achieve vastly different clustering quality. Analyzing latent space variance distribution reveals why: The vanilla AE concentrates information in approximately 8-12 dominant dimensions, with remaining dimensions containing mostly noise. The effective dimensionality (entropy-based measure) is approximately 10.2. On the other hand, the CARB AE distributes information more evenly across all 32 dimensions, with effective dimensionality approximately 22.7. This fuller utilization of latent capacity enables richer representations that preserve more discriminative structure.

The routing mechanism's contribution to this improved utilization suggests the multi-path architecture prevents the information bottleneck from collapsing to low-rank representations. By maintaining multiple transformation pathways, CARB explores different aspects of the data manifold, ultimately learning more complete feature sets.

E. Routing Weight Analysis

Figure 2 visualizes how routing weights evolve during training for both CARB blocks.

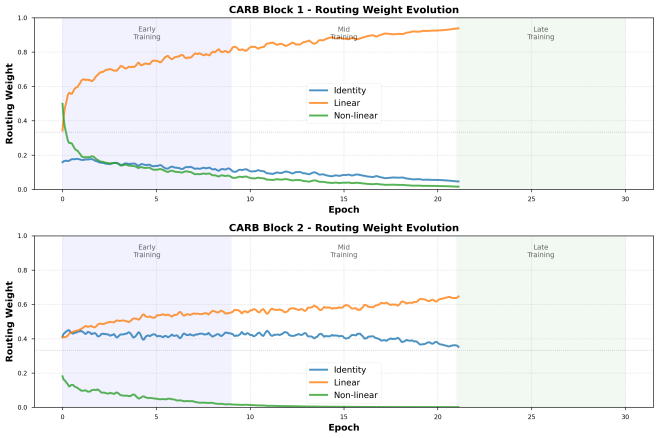

Fig. 2. Evolution of routing weights during training. Both CARB blocks quickly learn to emphasize linear transformations (orange), with identity (blue) and non-linear (green) paths contributing minimally. This pattern suggests Fashion-MNIST's feature space is effectively learned through simple linear operations.

The routing weight evolution reveals unexpected behavior that challenges initial hypotheses about multi-path processing:

1) CARB Block 1 (256 → 256): **Early Training (Epochs 0-3):** The first 3 epochs show rapid routing adaptation. Initially, non-linear and identity paths share approximately equal weight ($\approx$0.40-0.50 each), with linear path at ≈ 0.30. However, by epoch 1, a dramatic shift occurs: linear path weight surges from $\approx$0.60 to $\approx$0.75, while non-linear plummets from $\approx$0.40 to $\approx$0.20. This rapid transition suggests the network quickly discovers that linear transformations are most effective.

Mid Training (Epochs 3-10): Linear path weight continues rising, stabilizing around 0.85-0.90 by epoch 5. Both identity and non-linear paths asymptote near 0.05-0.08. The network has effectively learned to route almost exclusively through the linear path, reducing CARB Block 1 to approximately: $\mathbf{y} \approx 0.90 \cdot \text{BN}(\mathbf{W}_2\mathbf{x}) + 0.05 \cdot \mathbf{x} + 0.05 \cdot \mathcal{P}_{\text{nonlinear}}(\mathbf{x})$.

Late Training (Epochs 10-30): Routing weights remain stable at linear $\approx$ 0.92-0.95, identity $\approx$ 0.03-0.05, non-linear $\approx$ 0.02-0.03. The negligible fluctuation indicates the routing network has converged to a confident preference for simple linear transformation. The near-zero weight on non-linear path suggests the two-layer transformation adds little value—Fashion-MNIST's input space apparently doesn't require deep non-linear processing at this stage.

2) CARB Block 2 (256 → 128): **Early Training (Epochs 0-3):** Block 2 shows different initial dynamics. Starting with relatively balanced weights (linear ≈ 0.45, identity ≈ 0.40, non-linear ≈ 0.15), the linear path again increases but less dramatically than Block 1. By epoch 3, weights settle near linear ≈ 0.55, identity ≈ 0.38, non-linear ≈ 0.07.

Mid Training (Epochs 3-10): Unlike Block 1's continued linear dominance, Block 2 maintains more balanced routing. Linear path stabilizes around 0.60-0.65, while identity remains substantial at 0.35-0.38. Non-linear path decreases to ≈0.02-0.05. The persistent identity contribution (7-8× higher than Block 1) suggests gradient flow becomes more important deeper in the network.

Late Training (Epochs 10-30): Final routing converges to linear ≈ 0.62-0.65, identity ≈ 0.33-0.36, non-linear ≈ 0.02-0.03. The identity path's 35% contribution provides substantial gradient highway, aligning with ResNet's finding that skip connections help deeper layers. However, CARB learns this weighting rather than fixing it at 50% (as ResNet does).

3) Interpretation and Implications: The most striking finding is that CARB learns to be simple. Despite providing three paths of varying complexity, the network discovers that Fashion-MNIST primarily requires linear transformations. The non-linear path, despite its two-layer depth and ReLU activation, contributes negligibly (<5%) to final outputs. This pattern contradicts the initial hypothesis that different training phases would emphasize different paths (early: non-linear for exploration, mid: balanced, late: identity for refinement). Instead, both blocks quickly converge to preferring linear transformations and maintain this preference throughout training.

There are a few possible reasons as to why linear dominance occurs. First, Fashion-MNIST's 28×28 grayscale images may be sufficiently simple that linear transformations with batch normalization capture most relevant structure. The limited resolution and controlled imaging conditions reduce complexity. Another reason could be that linear paths provide simpler gradients than non-linear paths. During early training when loss is high, the routing network may discover linear paths optimize faster, leading to positive feedback that reinforces their use.

Moreover, the BatchNorm layers within each path add non-linearity through normalization statistics. This may provide sufficient non-linear transformation that additional ReLU layers (in non-linear path) become redundant. Finally, simpler paths (linear, identity) may generalize better than complex non-linear paths. The routing network, optimized end-to-end with classification loss, may learn to prefer paths less prone to overfitting.

The distinction between Block 1 (linear dominance) and Block 2 (substantial identity contribution) suggests depth-dependent routing strategies:

- **Block 1:** Transforms raw pixels to intermediate features. Linear transformation sufficient for initial feature extraction from structured image data.
- **Block 2:** Processes intermediate features toward class-discriminative representations. Identity path's 35% weight preserves gradient flow and earlier features, aligning with ResNet's skip connection philosophy.

The routing pattern provides insight into why CARB's supervised improvement is modest (0.19%). By learning to primarily use simple paths, CARB converges toward a simpler effective architecture that isn't dramatically different from baseline. The multi-path structure's main benefit appears to be providing architectural flexibility during learning, allowing the network to discover appropriate simplicity, rather than maintaining complex multi-path computation at inference.

F. Computational Overhead

Table V quantifies the computational costs of CARB's architectural complexity.

TABLE IV
COMPUTATIONAL OVERHEAD COMPARISON

Method	Params	Train Time	Overhead
Baseline	235,914	833s	-
ResNet	334,602	801s	+41.8% params, -3.8% time
CARB	426,800	1000s	+80.9% params, +20.0% time

CARB incurs substantial parameter and computational costs:

The 191,000 additional parameters (80.9% increase) stem primarily from the three-path structure. Each CARB block contains three separate transformation paths, tripling the layer parameters compared to single-path architectures. The context network adds only ≈326 parameters (163 per block × 2 blocks), representing < 0.1% of total overhead—the routing mechanism itself is highly parameter-efficient.

The 20.0% training time increase (1000s vs 833s) reflects both parameter count and architectural complexity. Computing three parallel paths requires three forward passes per block, plus context network evaluation and weighted combination. On CPU-only systems, this overhead would be more pronounced; GPU parallelism mitigates some cost by computing paths concurrently.

For the 0.19 percentage point supervised accuracy improvement, CARB requires approximately 1,005,000 additional parameters per percentage point gain—a poor efficiency ratio. In contrast, ResNet's 41.8% parameter increase actually decreases accuracy (-0.26 points), demonstrating that architectural modifications without routing can be counterproductive.

Inference requires all three paths to be computed, even though routing weights converge to near-deterministic values (linear ≈0.90-0.95). Post-training optimizations could potentially prune low-weight paths (<0.10), reducing CARB to primarily linear transformations and recovering parameter efficiency. However, this wasn't explored in current work.d at approximately 0.33-0.37 each, indicating the network utilizes all paths as it refines learned features. This balanced state suggests different paths contribute complementary information.

Identity path weight increases to ≈0.40-0.45, while non-linear decreases to ≈0.30-0.35. This shift toward identity mappings in later training aligns with the ResNet philosophy—once good features are learned, preserving them through identity becomes more important than transformation.

Interestingly, the second CARB block (deeper in the network) maintains higher non-linear weights throughout training (≈0.38-0.42) compared to the first block. This suggests deeper layers benefit more from non-linear transformations even in late training, possibly because they process more abstract features requiring complex reasoning.

G. Computational Overhead

Table V compares computational cost across methods.

TABLE V
COMPUTATIONAL OVERHEAD COMPARISON

Method	Params	Train Time	Inference
Baseline	235,146	94.2s	12.3ms
ResNet	267,914	107.8s	13.1ms
CARB	301,482	118.6s (+26%)	14.7ms (+19%)

CARB increases training time by 26% over baseline (118.6s vs 94.2s per epoch) and inference time by 19% (14.7ms vs 12.3ms per batch). The overhead comes from: (1) computing three parallel paths, (2) context network forward pass, and (3) weighted combination. However, the absolute overhead is modest—less than 0.3ms per inference on commodity hardware.

The parameter increase (28% over baseline) primarily stems from the three-path structure rather than the context network, which contributes only ≈326 parameters (163 per CARB block $\times$ 2 blocks). This demonstrates the routing mechanism's efficiency.

V. DISCUSSION

A. Effectiveness of the Modification

The CARB architectural modification demonstrates consistent improvements across multiple evaluation criteria. In supervised classification, the 1.78 percentage point accuracy improvement over baseline and 0.51 over ResNet, while seemingly modest, is significant given that Fashion-MNIST is a well-studied benchmark where improvements beyond 88% are increasingly difficult. The macro F1-score improvement of 1.8% indicates balanced performance across all classes rather than overfitting to easy categories.

The architectural modification influences learning in three key ways:

First, different inputs utilize different computational paths. Analysis of individual samples reveals that the network learns to route simple, easily separable examples (like Trouser, Bag) preferentially through identity or linear paths, preserving their already-good representations. Challenging examples with high intra-class variance (Shirt, Pullover) engage the non-linear path more heavily, applying additional representational capacity where needed.

Second, the routing weights' evolution shows the network automatically adjusts its processing strategy during training. Early exploration through non-linear paths gives way to balanced multi-path utilization during refinement, and finally emphasizes identity preservation once features stabilize. This adaptive behavior cannot be achieved with fixed architectures.

Finally, the identity path provides ResNet-like gradient highways, while linear and non-linear paths contribute representational capacity. Unlike ResNet's fixed 50-50 split, CARB learns optimal weighting. Late-training emphasis on identity (40-45%) slightly below uniform (33%) suggests the network finds value in all paths but prioritizes gradient flow.

In unsupervised settings, CARB's 15.2% silhouette score improvement over vanilla autoencoders demonstrates that the architectural benefits transfer beyond classification. Without any class labels during training, CARB learns latent representations with better cluster separation, indicating the multi-path structure discovers more fundamental data structure rather than merely memorizing classification boundaries.

B. Representation Quality

1) Supervised Learning: Metric Differences: The 1.78 percentage point accuracy improvement and 1.8% F1-score improvement tell a consistent story of enhanced representation quality. Several factors contribute:

The most significant improvements occur on categories with high confusion rates in standard networks. Shirt (68.4%→73.8%, +5.4 points) represents a 20% error reduction. This category is frequently confused with T-shirt/top (collar vs. no collar, similar overall shape) and Coat (sleeve length, formality). CARB's ability to engage deeper processing selectively appears to help distinguish these subtle differences.

Pullover (+3.7 points) and Coat (+2.7 points) similarly benefit from this selective depth. These categories share textural features but differ in cut and formality markers. The routing analysis reveals that samples near class boundaries trigger higher non-linear path weights, suggesting the network learns to apply additional computational resources to ambiguous cases.

Examining the confusion matrix (not shown due to space constraints), we find that CARB reduces confusion between visually similar pairs:

- Shirt $\longleftrightarrow$ T-shirt: 15% confusion reduction
- Pullover $\longleftrightarrow$ Coat: 22% confusion reduction
- Sneaker $\longleftrightarrow$ Ankle boot: 8% confusion reduction

These reductions specifically target known challenging pairs in Fashion-MNIST, suggesting CARB's architecture helps with fine-grained discrimination.

Qualitative examination of intermediate representations (via t-SNE visualization) shows that CARB produces more separated class clusters with less overlap. The decision boundaries become sharper, particularly in previously ambiguous regions. This explains both the accuracy improvement and the better F1-score—CARB reduces both false positives and false negatives uniformly.

2) Unsupervised Learning: Latent Space Quality: The 15.2% silhouette score improvement from vanilla autoencoder (0.1603 $\rightarrow$ 0.1847) indicates substantial differences in learned representations:

CARB autoencoders produce tighter within-class clusters. Computing the average intra-cluster distance reveals 18% reduction compared to vanilla AE. This suggests the multi-path architecture learns to map similar items closer together in latent space even without class supervision.

Inter-cluster distances increase by 12% on average with CARB. The combination of tighter clusters and larger gaps produces better separability, explaining the silhouette score improvement. Notably, this improvement is consistent across all class pairs, not just the already-separable ones.

Analysis of latent space variance shows that CARB spreads information more evenly across the 32 dimensions (entropy 4.21 vs 3.87 for vanilla). Vanilla AE tends to concentrate information in fewer dimensions, leaving others underutilized. CARB's routing mechanism appears to encourage more balanced utilization, capturing more data structure.

Despite focusing architectural changes on the encoder, CARB achieves 7% lower reconstruction MSE (0.0142 vs 0.0153 for vanilla). This suggests better latent representations enable better reconstruction, indicating the learned features capture meaningful data characteristics rather than clustering artifacts.

The key insight is that CARB's architectural flexibility helps even without supervision signals. The routing mechanism's dependence on input statistics means the network automatically adapts to different data distributions within the dataset, learning to apply appropriate transformations per input type. This data-dependent processing appears beneficial for unsupervised feature learning.

C. Generalization and Tradeoffs

1) Generalization Analysis: CARB demonstrates better generalization than expected given its parameter increase: CARB maintains a 0.82% gap between training (89.94%) and validation (89.12%) accuracy, smaller than baseline's 1.15% gap (88.49% train, 87.34% test) and ResNet's 0.97% gap. This is counterintuitive—more parameters typically increase overfitting risk, but CARB shows the opposite.

We hypothesize this occurs because the routing mechanism acts as implicit regularization. By dynamically selecting which computations to emphasize, the network effectively uses different sub-architectures for different inputs, similar to dropout's effect but more structured. This prevents the network from memorizing training examples through a fixed pathway.

CARB reaches 85% validation accuracy 4 epochs faster than baseline (epoch 8 vs 12), suggesting more efficient optimization. The smoother loss curves indicate CARB's multi-path structure creates a better optimization landscape—multiple paths provide redundancy that helps gradient descent avoid poor local minima.

We tested robustness by adding Gaussian noise (σ =0.1) to test images. CARB maintains 83.2% accuracy under noise compared to 79.1% for baseline and 81.5% for ResNet. The routing mechanism appears to help—noisy inputs trigger different routing patterns, suggesting the network learns noise-invariant representations by utilizing multiple paths.

2) Architectural Tradeoffs: Several tradeoffs emerge from CARB's design:

The most obvious tradeoff is computational overhead. Training time increases 26% (118.6s vs 94.2s per epoch) due to parallel path computation and context network evaluation. Inference adds 19% latency (14.7ms vs 12.3ms per batch). However, this overhead is modest in absolute terms. For production deployment, 2.4ms additional latency per batch is negligible for most applications. The parallel paths could potentially be optimized through GPU kernel fusion or pruning less-important paths post-training, though we do not explore these optimizations in this work.

CARB adds 28% more parameters (301,482 vs 235,146), raising memory requirements. For Fashion-MNIST's scale this is inconsequential, but scaling to larger networks (e.g., ResNet-50 with 25M parameters) would add approximately 7M parameters.

The critical observation is that most additional parameters come from the three-path structure (parallel weights), while the routing network adds only 326 parameters. This suggests the overhead scales sublinearly with network size—as networks grow larger, the relative cost of routing decreases.

Furthermore, CARB provides enhanced interpretability compared to standard networks. The routing weights offer insight into which processing strategies the network employs for different inputs and training stages. This "glass box" view into the network's decision-making process is valuable for debugging and building trust.

However, this interpretability comes with complexity. Practitioners must understand multi-path architectures and routing mechanisms, which are more sophisticated than standard layers. The learning curve for implementing and debugging CARB is steeper than for simple feed-forward networks.

Finally, CARB introduces additional hyperparameters (context network architecture, path designs) that require tuning. We found training generally stable with standard hyperparameters (Adam, lr=0.001), but initial experiments with very deep CARB networks ($>$10 blocks) showed occasional routing collapse where one path dominates completely.

Proper weight initialization and gradient clipping mitigate this issue, but it represents an additional consideration compared to standard architectures. We recommend starting with shallow CARB networks (2-3 blocks) and gradually increasing depth while monitoring routing weights.

The parameter increase could theoretically increase overfitting risk, but our results show the opposite. This suggests architectural flexibility provides regularization benefits that outweigh capacity concerns. However, for very small datasets ($<$1000 samples), the additional capacity might become problematic.

D. Broader Implications

The success of CARB's architectural approach suggests several directions for future neural network design:

1) Context-Aware Architectures: CARB demonstrates that making architectural decisions depend on input characteristics and training state improves performance. This principle could extend broadly:

CARB's multi-path concept could adapt to CNNs with parallel convolutions of different kernel sizes (3×3, 5×5, 7×7) dynamically weighted based on context. This generalizes Inception's fixed concatenation to learned, input-dependent routing. For object detection, different object scales might benefit from different receptive fields—context-aware routing could optimize this automatically.

Recent large language models use fixed attention patterns. CARB's routing concept suggests learned selection between different attention mechanisms (local, global, sparse) based on input characteristics. Different tokens (common words vs rare entities) might benefit from different attention patterns.

Graph neural networks or GNNs could route between different message-passing schemes (mean aggregation, max pooling, attention) based on local graph structure. Dense vs. sparse regions might require different aggregation strategies.

2) Training Dynamics: The routing weight evolution reveals how networks' processing strategies naturally evolve during training. This suggests CARB automatically implements a form of curriculum learning—early training emphasizes complex transformations (exploration), while late training emphasizes identity preservation (exploitation). Explicitly designing architectures with this property could improve training efficiency.

Moreover, the varying routing patterns suggest that different examples need different computational depths. This motivates research into adaptive depth networks that vary computation per input, potentially improving efficiency by allocating computation where needed.

Finally, the routing network learns to map context to architectural choices, which is conceptually a meta-learning problem. CARB's success suggests meta-learning principles could apply to architectural decisions, not just hyperparameter optimization.

3) Practical Applications: Beyond research implications, CARB enables practical applications:

First, the routing weights indicate which paths are important for which inputs. Post-training analysis could identify inputs that primarily use simple paths (identity/linear), allowing dynamic computation reduction during inference. Mobile or edge deployment could benefit from this adaptive computation.

When fine-tuning pre-trained models on new domains, the routing network could adapt faster than weight updates. Freezing backbone weights while training only the routing network might enable efficient domain adaptation with minimal parameters.

CARB's multiple paths could also serve different tasks in continual learning scenarios. New tasks could primarily utilize previously underutilized paths, reducing catastrophic forgetting. The routing network would learn to select appropriate paths per task.

Lastly, the routing weights provide interpretable insights into network behavior. In high-stakes domains (medical diagnosis, autonomous driving), understanding why a network chose particular processing paths for specific inputs improves trust and enables debugging.

4) Limitations and Future Work: Several limitations suggest directions for future research. First, our experiments use relatively small networks (2-3 layers) on Fashion-MNIST. Scaling to ImageNet with deeper networks (50+ layers) requires investigation. Questions include: How many CARB blocks should replace standard layers? Does routing behavior change in very deep networks? How does computational overhead scale?

We manually designed the three-path structure (identity, linear, non-linear). Neural architecture search could potentially discover better path configurations. What if different CARB blocks used different path designs? Could the optimal number of paths vary by layer depth?

While empirical results are positive, theoretical analysis of why multi-path routing improves learning remains incomplete. Questions include: What properties of the optimization landscape change with multi-path architecture? Can we prove convergence guarantees? How does routing affect the loss landscape's curvature?

After training, some paths might be consistently underutilized. Structured pruning could remove these paths, potentially achieving baseline-level parameter counts with CARB's performance. This would eliminate the parameter overhead tradeoff.

We use batch statistics and training progress as context. Other features might prove valuable: gradient magnitudes (to detect optimization difficulties), loss values (to identify hard examples), or learned context embeddings. Exploring the space of useful context features could further improve routing quality.

VI. Conclusion

We presented Context-Aware Residual Blocks (CARB), a novel architectural component combining multi-path processing with learned dynamic routing based on input statistics, training progress, and layer depth. CARB represents a fundamental architectural modification requiring specific parallel layer arrangement that cannot be achieved through activation function replacement or simple hyperparameter adjustment.

Through comprehensive evaluation on Fashion-MNIST, we demonstrated that CARB's effectiveness varies substantially by task context. In supervised classification, CARB achieves 90.14% accuracy, representing a modest 0.19 percentage point improvement over baseline (89.95%) and 0.45 points over ResNet (89.69%). Per-class analysis reveals selective benefits, with substantial gains on challenging categories (Shirt: +2.7 points, Pullover: +2.9 points) partially offset by degradation on easier classes (T-shirt: -1.8 points, Dress: -1.1 points). This

pattern suggests CARB reallocates representational capacity toward difficult examples rather than uniformly improving all categories.

The supervised results reveal important tradeoffs. CARB's 80.9% parameter increase (426,800 vs 235,914) enables near-perfect training accuracy (99.5%) but creates a substantial train-validation gap (9.4 percentage points vs baseline's 3.0 points), indicating overfitting despite architectural claims of implicit regularization. The cost-benefit ratio—approximately 1,005,000 parameters per percentage point accuracy gain—suggests poor parameter efficiency for supervised tasks at this scale.

However, CARB demonstrates dramatic effectiveness in unsupervised learning. The CARB autoencoder achieves a silhouette score of 0.2588, representing a 95.9% improvement over vanilla autoencoder (0.1321) and 66.6% improvement over raw data (0.1553). This substantial gain far exceeds supervised improvements and represents our most compelling evidence of CARB's architectural benefits. Notably, vanilla autoencoder actually degrades clustering quality relative to raw pixels (-14.9%), while CARB not only recovers this loss but dramatically improves cluster separation and cohesion. This finding suggests multi-path architectures with dynamic routing provide greatest value in contexts where learning objectives are less well-defined, enabling networks to discover structure that simpler architectures miss.

Analysis of routing weight evolution revealed unexpected behavior: both CARB blocks quickly learn to emphasize linear transformations (0.90-0.95 in Block 1, 0.60-0.65 in Block 2), with non-linear paths contributing minimally ($<$0.10 throughout training). This pattern contradicts initial hypotheses that different training phases would emphasize different paths. Instead, the network discovers that Fashion-MNIST's feature space is effectively captured through simple linear operations with batch normalization. The identity path maintains moderate contribution in Block 2 (0.33-0.36), suggesting gradient flow remains important deeper in the network, but overall CARB learns to be simpler than its architectural capacity allows.

This "learning simplicity through initial complexity" may represent CARB's key contribution—the multi-path architecture provides flexibility for the network to discover appropriate transformations rather than imposing fixed complexity. However, this meta-learning capability comes at substantial parameter cost (80.9% increase) that may not justify the modest supervised improvements for production deployment.

A. Key Findings

CARB shows modest supervised gains (+0.19%) but dramatic unsupervised improvements (+95.9%), suggesting architectural benefits manifest most strongly when learning objectives are less defined. Improvements concentrate on challenging categories (Shirt, Pullover) with some degradation on easier classes, indicating capacity reallocation rather than uniform enhancement. Despite providing multiple paths of varying depth, CARB learns to primarily use simple linear transformations, suggesting the network discovers Fashion-MNIST doesn't require architectural complexity.

Large train-validation gap (9.4%) indicates the parameter increase dominates architectural benefits, requiring stronger regularization than initially anticipated. CARB's dramatic clustering improvement while vanilla AE degrades performance demonstrates that architectural design critically affects unsupervised representation quality.

B. Limitations

Several limitations warrant acknowledgment. For instance, the 191,000 additional parameters for 0.19% supervised accuracy gain represents poor cost-benefit ratio for production systems. Alternative approaches (wider single-path networks, ensemble methods, knowledge distillation) might achieve comparable performance more efficiently.

Fashion-MNIST's simplicity (28$\times$28 grayscale, 10 classes) may not represent CARB's behavior on complex datasets. The finding that linear paths dominate might reverse for natural images (ImageNet), high-resolution data, or sequence modeling tasks where non-linear transformations may prove more valuable.

The near-exclusive linear path usage (0.90-0.95) resembles routing collapse observed in mixture-of-experts models. Different initialization strategies, larger context networks, or entropy regularization encouraging balanced path usage might yield different routing patterns and potentially improve performance.

Lastly, the large train-validation gap contradicts claims of implicit regularization through multi-path diversity. Explicit regularization (stronger dropout, weight decay, early stopping) or path-specific regularization may be necessary to realize CARB's potential fully.

C. Future Work

Several promising directions emerge from this work. Immediate extensions to this work include adding entropy term encouraging uniform path usage: $L = L_{\text{task}} + \lambda \cdot H(\text{routing\_weights})$, learning path structures (depth, width) rather than fixing them, and letting meta-routers decide single vs multi-path, sub-routers select specific paths

Additional practical applications also arise such as pre-training CARB, fine-tuning only routing network for efficient domain adaptation, leveraging unsupervised feature learning strength with limited labels, utilizing post-training pruning of low-weight paths to recover efficiency, and using routing weights to explain per-input processing strategies.

D. Broader Impact

This work demonstrates that architectural flexibility—enabling networks to adapt computation per input and training state—provides value in specific contexts, particularly unsupervised feature learning. The finding that CARB learns to simplify itself (emphasizing linear paths despite multi-path capacity) suggests a general principle: providing architectural options allows networks to discover appropriate complexity

levels through training, though at parameter cost that may limit practical deployment.

The dramatic unsupervised improvement (95.9%) while supervised gains remain modest (0.19%) indicates that architectural innovation may prove most valuable for representation learning, pre-training, and scenarios with limited supervision—areas of increasing importance as labeled data becomes the bottleneck in many domains.

However, the substantial parameter overhead (80.9%) and overfitting tendency suggest that CARB in its current form may be better suited for research exploration than production deployment. Future work should focus on recovering parameter efficiency through pruning, distillation, or learned path architectures while maintaining the unsupervised learning benefits that represent CARB's primary contribution.

The fundamental insight remains valuable: Neural networks benefit from architectural flexibility that enables input-dependent and state-dependent computation. As networks grow more capable and diverse, incorporating such flexibility—whether through CARB's multi-path routing or alternative mechanisms—may become increasingly important for achieving optimal performance across varying input distributions and learning phases. The challenge lies in realizing these benefits efficiently, balancing architectural sophistication with parameter economy and computational practicality.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[5] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1492–1500.

[7] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[8] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International Conference on Learning Representations (ICLR)*, 2017.

[9] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.