WACV 2026
TUCSON, AZ 3/6 – 3/10

Feedback Alignment Meets Low-Rank Manifolds:

A Structured Recipe for Local Learning


Arani Roy


Marco P. E. Apolinario


Shristi Das Biswas

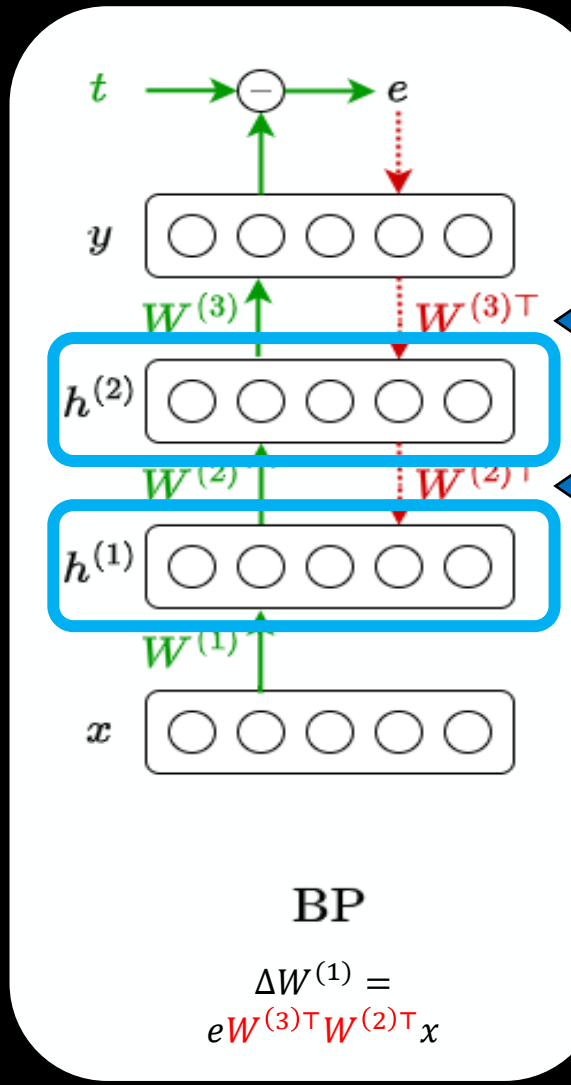

Kaushik Roy

PURDUE UNIVERSITY

CoCoSys
CENTER FOR THE CO-DESIGN OF COGNITIVE SYSTEMS

The Problem with Backpropagation

Why do we need an alternative training paradigm?



Update Locking: Layer i needs layer i+1 to complete, no parallelization

storage for batch size 256)

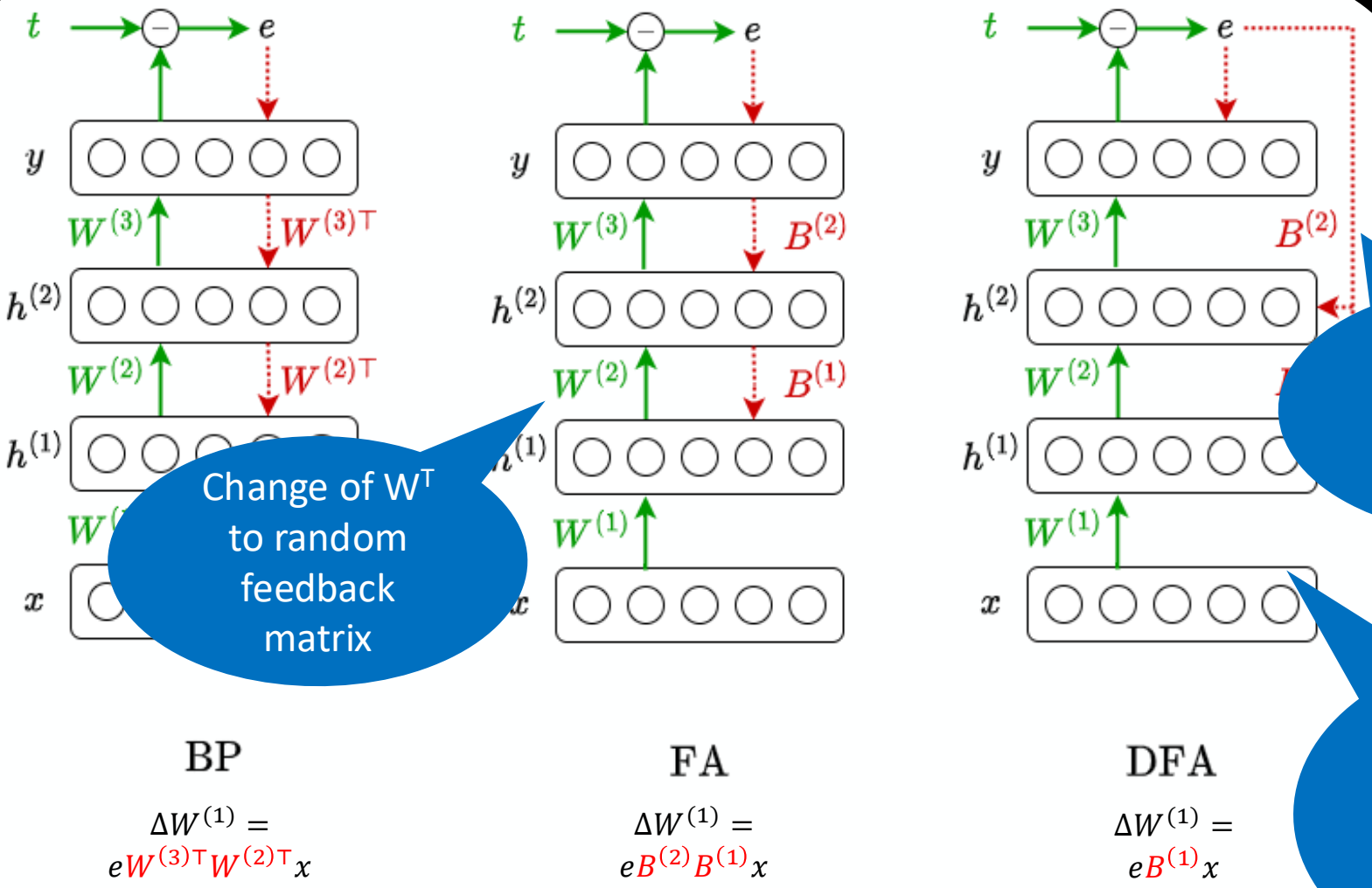
Memory Intensive

Weight Transport

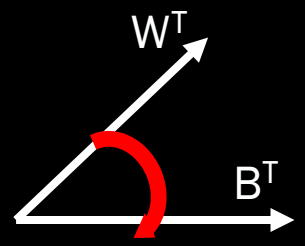
Sequential bottleneck

Local Learning Paradigms

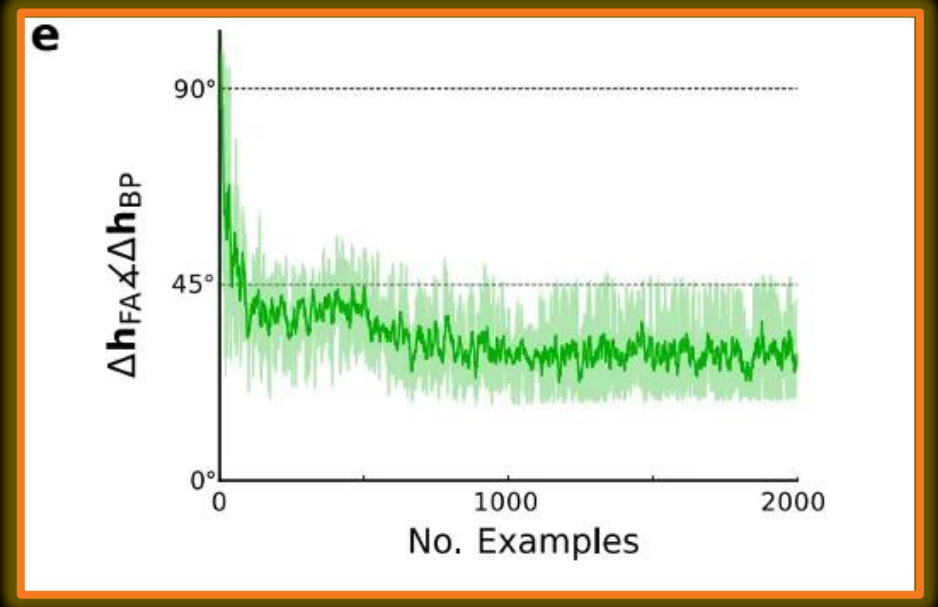
Alternatives to BP



How does DFA learn?



The weight tries to align itself to the fixed random matrix



So, for gaining improvements in this local method, we need to improve the alignment.

The gradients between alignment methods and BP need to be **< 90 degrees for convergence**

DFA: Benefits and Issues

Benefits of DFA

✓ **No Activation Storage**
Forward activations not needed

✓ **Parallel Updates**
All layers update simultaneously

✓ **No Weight Transport**
Fixed random feedback matrices B

↓

Issues with DFA

Struggles with Depth

- Limited to ~10 layers
- ImageNet: 82% error for 18-layer DNNS

Struggles with conv layers

- Random feedback has no spatial structure.
- Mismatch of conv kernels and feedback weights.

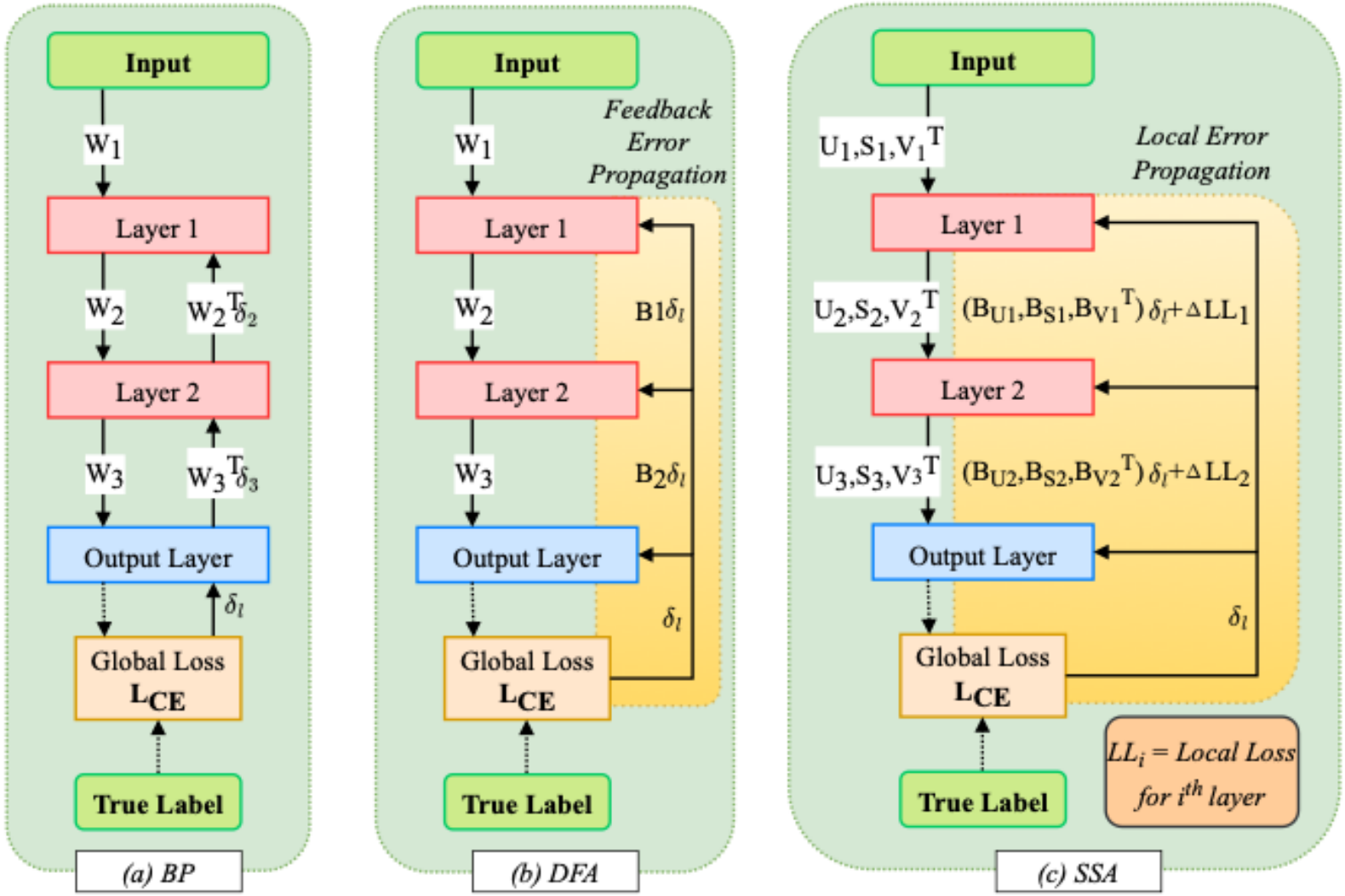
Root cause: misaligned feedback

The cosine angle between DFA pseudo-gradients and true BP gradients grows with depth, especially in convolutional architectures.

Key Question:

Could **low-rank decompositions** of weight and feedback spaces enable better alignment and extend DFA to convolutional layers? Could it make DFA more **scalable**? Could we alleviate further **memory and compute**?

SSA: SVD-Space Alignment



(a) BP
Sequential backward pass with W^T transport. Activations stored at every layer.

(b) DFA
Fixed random B sends error directly to each layer. No W^T. No storage. But no structure.

(c) SSA
Weights decomposed as U,S,V^T before training. Feedback B also SVD-factored. Local loss L_i = L_CE + L_align + L_ortho applied at each layer.

Fig. 1: BP requires symmetric weight transport and sequential updates. DFA adds random direct feedback. SSA structures both forward and feedback in SVD-space with local losses.

Local Loss Design

Composite Loss Function

$$LL_i(\theta_i) = \alpha \cdot L_{CE}(\theta_i) + \beta \cdot L_{align}(\theta_i) + \gamma \cdot L_{ortho}(\theta_i)$$

L_{CE}: Cross-Entropy

$$\Delta L_{CE} = y_{predict} - y_{label}$$

- Standard global error
- Gradient obtained from the global error
- Input locally, parallelly to each layer

L_{align}: Alignment

$$L_{align}(\theta_i) = \|U_i - B_{U_i}\|_F^2 + \|S_i - B_{S_i}\|_F^2 + \|V_i^T - B_{V_i^T}\|_F^2$$

- Alignment between forward and feedback SVD components
- Reduces mismatch in SVD-space

L_{ortho}: Orthogonality

$$L_{ortho}(\theta_i) = \|U_i^T U_i - I\|_F^2 + \|V_i^T V_i - I\|_F^2$$

- Ensures orthogonality of the parameters, otherwise SVD properties gets disrupted

SSA: SVD-Space Alignment

Core Idea

1. Decompose
$W = USV^T$
2. Align Feedback
$B \rightarrow B_U, B_S, B_V{}^T$
3. Train in SVD-space
Update U, S, V^T directly

SVD-Space Parameterization

Fully Connected Layers:

$$\mathcal{W}_i = \mathcal{U}_i \Sigma_i \mathcal{V}_i^\top, \quad \mathcal{U}_i \in \mathbb{R}^{m \times r}, \quad \Sigma_i \in \mathbb{R}^{r \times r}, \quad \mathcal{V}_i \in \mathbb{R}^{n \times r}$$

Convolutional Layers:

$$\mathcal{K}' = \mathcal{U}\Sigma\mathcal{V}^\top, \quad \mathcal{K}' \in \mathbb{R}^{NW \times CH}$$
$$\mathcal{K}_1 = \text{reshape}(\mathcal{U}\sqrt{\Sigma}) \in \mathbb{R}^{r \times C \times H \times 1},$$
$$\mathcal{K}_2 = \text{reshape}(\sqrt{\Sigma}\mathcal{V}^\top) \in \mathbb{R}^{N \times r \times 1 \times W}$$

Local Loss Design

- ***Cross-Entropy:*** Learns from feedback error
- ***Alignment:*** Keeps forward and feedback paths structurally similar
- ***Orthogonality:*** Maintains stability of SVD components

Local Updates

- Calculate gradients parallelly per layer for SVD components.
- Project the updates on **Stiefel Manifold**, for better orthogonality preservation.

Dynamic Rank Reduction

- Hoyer Regularizer every 10 epochs for first 30% of the epochs, sparsifies weights $\mathcal{L}_{\text{Hoyer}}(S_i) = \frac{\|S_i\|_1}{\|S_i\|_2}$
- Rest epochs, threshold-based pruning, keeping 95% of energy intact

Result

- A conv network upto 32 layers giving similar results as BP
- Lightweight inference

Results: Gradient Alignment

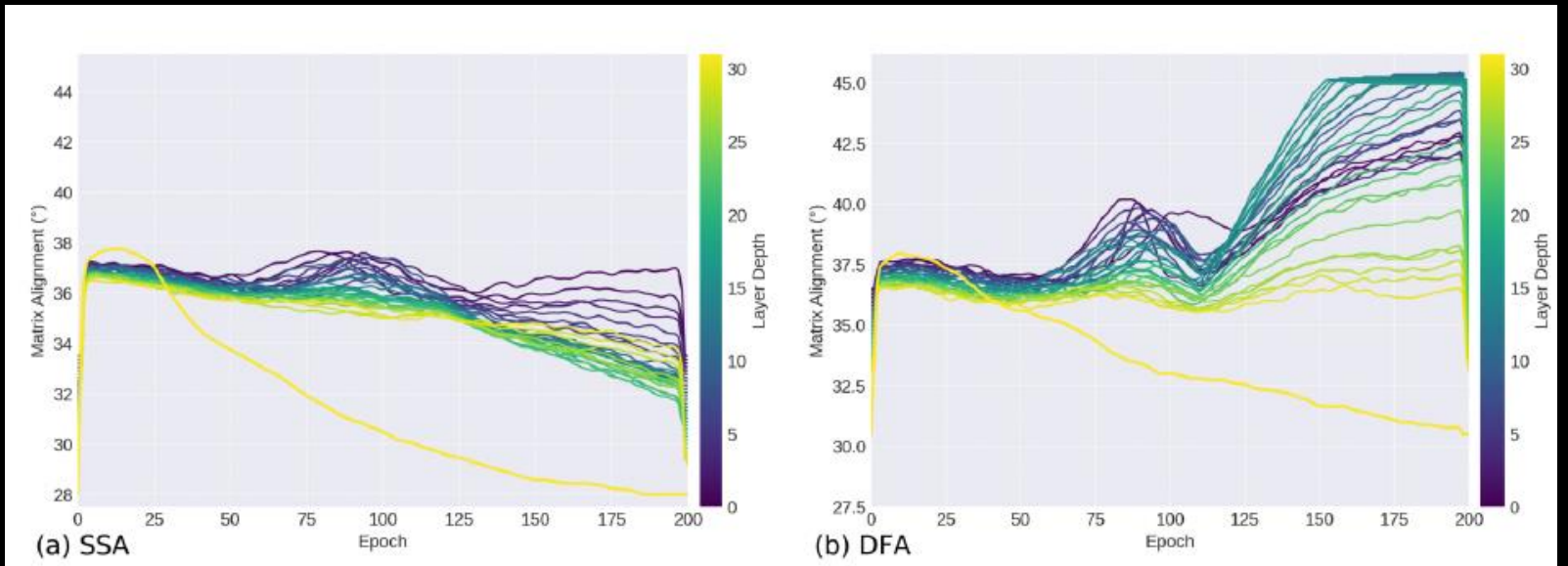
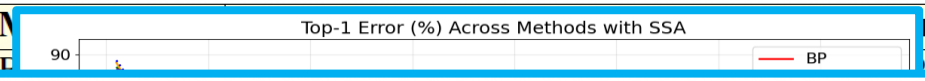


Figure 3. Matrix alignment angles across ResNet-32 layers over training epochs, between (a) SSA (b) DFA, and BP

$$\cos(\nabla_W L_{\text{true}}, \nabla_W L_{\text{SSA}}) > 0$$

Results: Accuracy



Network	Method	CIFAR10 (Top-1, mean ± std)	ImageNet (Top-1, mean ± std)	ImageNet (Top-5, mean ± std)
VGG-13	BP	93.75 ± 0.12	71.59 ± 0.25	90.39 ± 0.14
	SVD-BP [29]	92.80 ± 0.15	71.37 ± 0.20	90.20 ± 0.18
	PredSim [22]	86.49 ± 0.53	NA	NA
	AugLocal [19]	93.72 ± 0.10	70.93 ± 0.22	90.16 ± 0.16
	SSA (ours)	92.70 ± 0.13	69.68 ± 0.24	88.84 ± 0.15
ResNet-32	BP	92.14 ± 0.11	74.28 ± 0.30	91.76 ± 0.12
	SVD-BP [29]	91.77 ± 0.14	72.91 ± 0.27	89.27 ± 0.20
	PredSim [22]	79.31 ± 0.45	NA	NA
	AugLocal [19]	93.47 ± 0.12	73.95 ± 0.25	91.70 ± 0.17
	SSA (ours)	88.02 ± 0.18	69.38 ± 0.23	87.72 ± 0.19

Table 2. Comparison of classification accuracy (mean ± standard deviation) over 5 independent runs for CIFAR-10 and ImageNet datasets.

Results: Loss Component Contribution & Efficiency

Component Removal Impact (Table 3, VGG-13 CIFAR-10)

Configuration	Accuracy	Impact
Full SSA (all components)	**92.70%**	**Baseline**
No Cross-Entropy Loss	27.05%	**-65.65%**
No Alignment Loss	83.12%	-9.58%
No Orthogonality Regularizer	85.44%	-7.26%

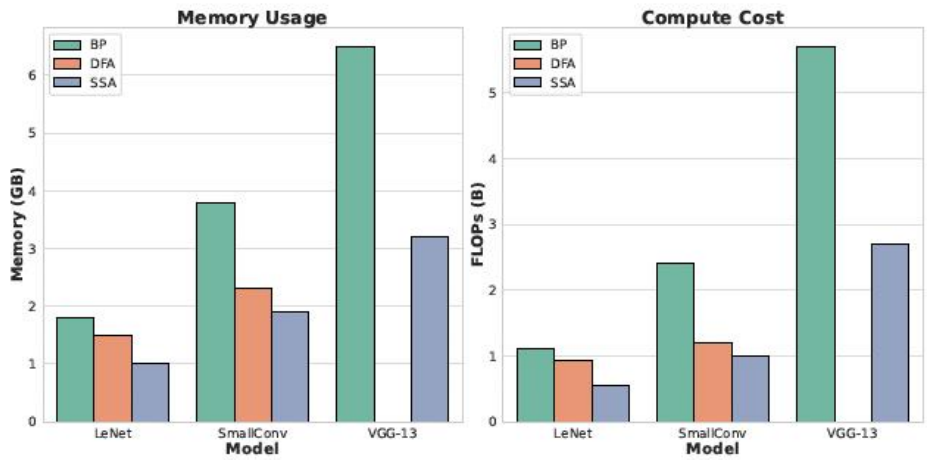


Figure 5. Compute and memory cost comparison of BP, DFA, and SSA across different architectures. SSA achieves lowest inference-time cost through structured low-rank training.

Key Takeaways

SVD-Space Alignment (SSA) → a principled, scalable local learning framework

1

Structured beats random

SVD-structured feedback + local alignment loss closes the DFA scalability gap. Gradient alignment angle drops 40–45° vs DFA.

2

Accuracy on par with BP

86.2% CIFAR-10, 92.7% VGG-13, 88.0% ResNet-32 — without backpropagation, weight transport, or activation storage.

3

Compact and efficient

Lowest inference FLOPs of any method.

Future work: Transformer blocks with SSA · Theoretical convergence bounds

Thank You!