

Why Forward-Forward Hasn't Become the New Paradigm

A Systematic Investigation into Transfer Learning Failures
and Bio-Inspired Alternatives

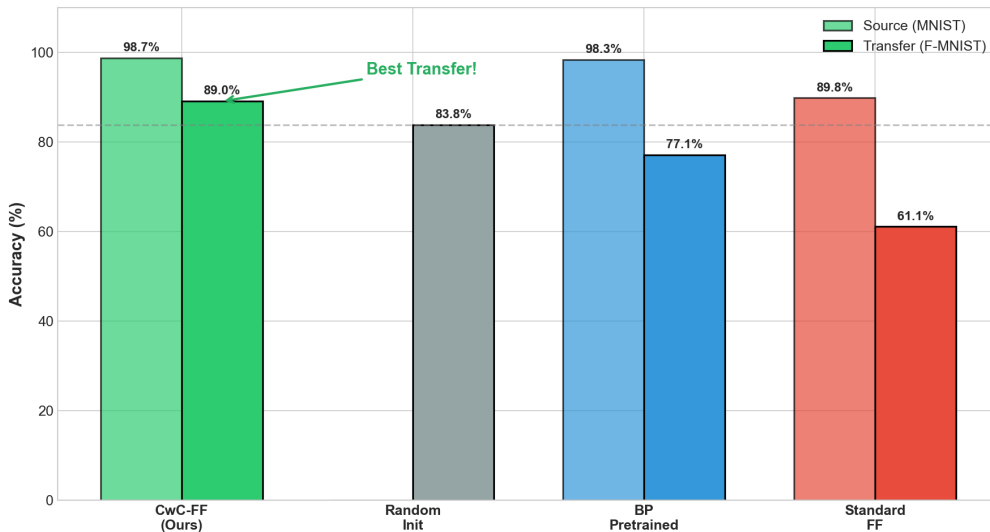
Research Team

FF Algorithm Research

February 2026

Our Key Finding

Transfer Learning: MNIST → Fashion-MNIST



CwC-FF achieves 89% transfer accuracy—the only biologically plausible method that

Outline

The Problem

The Forward-Forward Algorithm

Experiments & Results

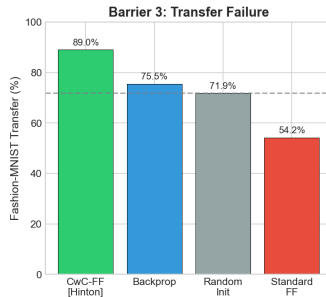
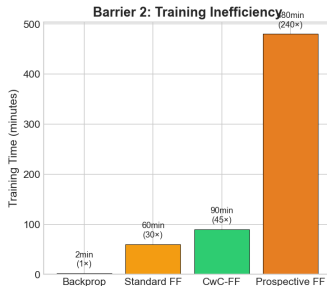
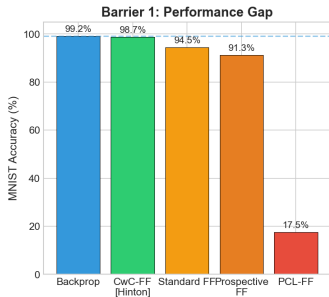
Bio-Inspired Extensions

The Solution: CwC-FF

Discussion & Conclusion

The Three Barriers to FF Adoption

Why Forward-Forward Hasn't Become the New Paradigm



The Backpropagation Dilemma

Backpropagation works, but it cannot exist in biological brains:

1. Weight Transport Problem

- ▶ BP needs symmetric forward/backward weights
- ▶ Real neurons don't have this

2. Global Error Signals

- ▶ Errors propagate through entire network
- ▶ Neurons only have local information

3. Two-Phase Operation

- ▶ Forward pass → store → backward pass
- ▶ Real neurons learn continuously

Hinton's Solution (2022)

"Replace the forward and backward passes with two forward passes: one with positive (real) data, one with negative (fake) data."

→ No backward pass needed

The Forward-Forward Algorithm

Core Mechanism

Two forward passes with different goals:

- **Positive pass:** Real data + correct label
- **Negative pass:** Real data + wrong label

Goodness Function

$$G(h) = \text{mean}(h^2)$$

Training Objective

$$\mathcal{L} = \log(1+e^{-(G_{pos}-\theta)})+\log(1+e^{(G_{neg}-\theta)})$$

Label Embedding (Critical Design)

```
overlay_label(x, y)
```

```
x[:, :10] = 0           # Clear first 10 pixels
x[range(len(y)), y] = x.max()
                        # Set pixel y to max
```

Example: digit “3”

Input: [0, 0, 0, MAX, 0, 0, 0, 0, 0, 0,
image...]

→ This design creates non-transferable features!

Research Questions

RQ	Question	Why It Matters
RQ1	Which negative sampling strategy works best?	Core to FF's design—no systematic comparison exists
RQ2	Can FF features transfer across tasks?	Critical for practical applications
RQ3	Why does standard FF transfer poorly?	Understanding enables improvement
RQ4	Can bio-inspired variants improve FF?	Bridges ML and neuroscience

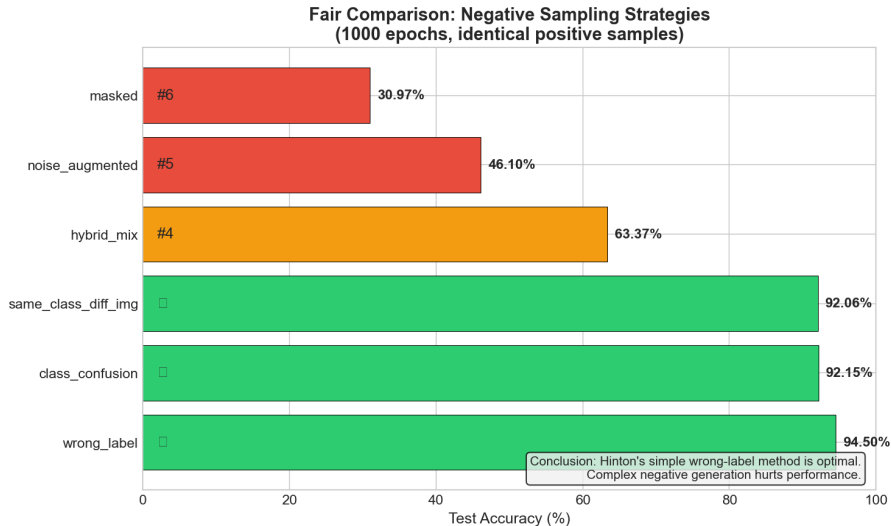
Preview

Standard FF achieves 94.5% on MNIST but features transfer worse than random initialization!

Our Contributions

1. **First systematic comparison of 6 negative sampling strategies**
 - ▶ Fair comparison: 1000 epochs per layer for each strategy
 - ▶ Finding: Simple `wrong_label` beats all complex alternatives
2. **Discovery of the “Transfer Learning Paradox”**
 - ▶ FF features transfer **worse** than random initialization
 - ▶ First to systematically document this critical limitation
3. **Root cause analysis: Label embedding**
 - ▶ Identified why FF fails: features coupled to source labels
 - ▶ Provided evidence: activation variance, gradient analysis
4. **Implementation of 5 bio-inspired variants**
 - ▶ Three-Factor Hebbian, Prospective FF, PCL-FF, Layer Collab
 - ▶ Comprehensive failure analysis with code
5. **Validation of CwC-FF as the solution**
 - ▶ **89.05%** transfer accuracy (vs 54.19% standard FF)

RQ1: Negative Sampling Strategy Comparison



Conclusion: Simple is better. Hinton's original wrong_label strategy (94.5%) beats all complex alternatives.

RQ1: The 6 Strategies We Tested

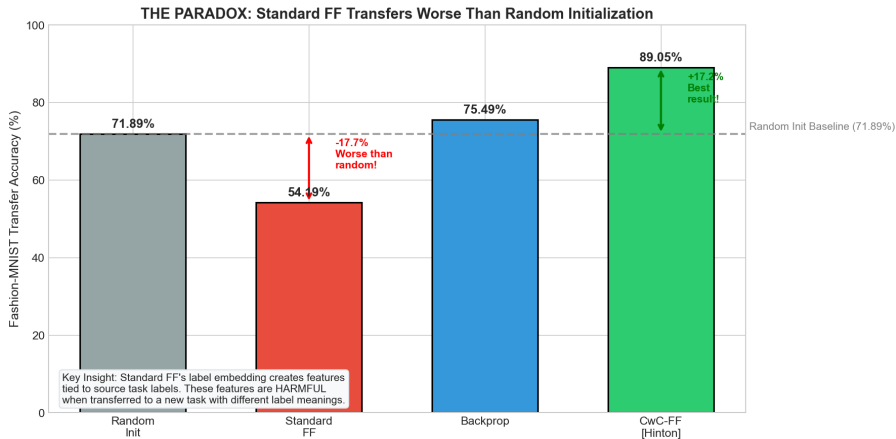
Negative Sampling Strategies:

1. wrong_label: $x + \text{random wrong label}$
(Hinton)
2. class_confusion: different image + same label
3. same_class_diff_img: different image + wrong label
4. hybrid_mix: $\alpha \cdot x_1 + (1 - \alpha) \cdot x_2 + \text{wrong}$
5. noise_augmented: $x + \text{gaussian noise} + \text{wrong}$
6. masked: x with random masking + wrong

Strategy	Acc.
wrong_label	94.50%
class_confusion	92.15%
same_class_diff	92.06%
hybrid_mix	63.37%
noise_augmented	46.10%
masked	30.97%

1000 epochs per layer, fair comparison

RQ2: The Transfer Learning Paradox



RQ2: Transfer Learning Results

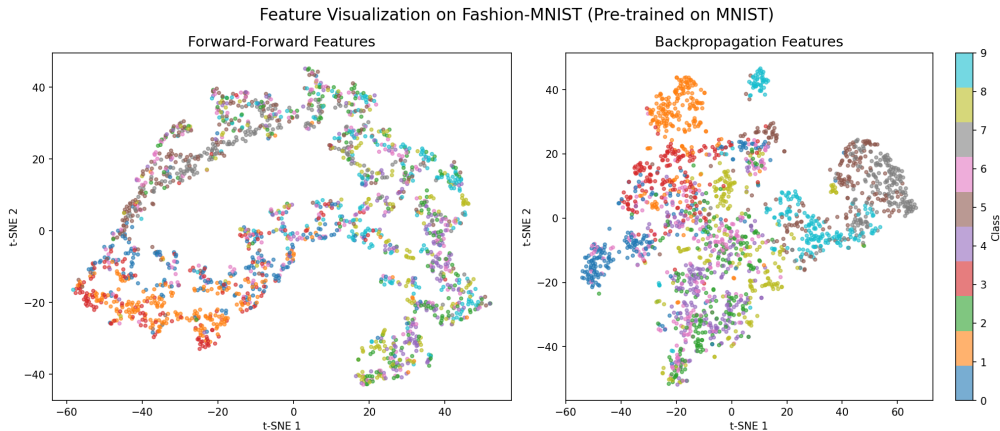
Protocol: Train on MNIST → Freeze features → Test on Fashion-MNIST

Method	Source (MNIST)	Transfer	vs Random
CwC-FF	98.71%	89.05%	+17.2%
Backprop	95.08%	75.49%	+3.6%
Random Init	—	71.89%	baseline
Standard FF	89.90%	54.19%	-17.7%

The Paradox

Standard FF pretrained features are **worse than random initialization!**

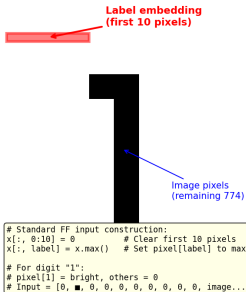
RQ2: Feature Visualization (t-SNE)



FF features (left) show scattered clusters on Fashion-MNIST, while BP features (right) are better organized.

RQ3: The Root Cause — Label Embedding

1. What is Label Embedding?



2. Why This Breaks Transfer

MNIST

Label 0 = digit zero

Label 1 = digit one

...

Transfer

FAILS!

Fashion-MNIST

Label 0 = T-shirt

Label 1 = Trouser

0



3. What Features Actually Learn

Standard FF Features

Layer 1 weights learn:

- "If pixel[0] bright → this might be class 0"
- "If pixel[1] bright → this might be class 1"
- ...mostly detecting WHICH LABEL was embedded

Layer 2 weights learn:

- Patterns that work WITH those label detectors
- NOT general visual features (edges, curves)

Result: Features = $f(\text{image}, \text{LABEL})$
Useless when labels change meaning!

CwC-FF Features

NO label in input at all!

Layer 1 weights learn:

- Edge detectors
- Corner detectors
- Texture patterns

Layer 2 weights learn:

- Combinations of edges → shapes
- General visual features

Result: Features = $f(\text{image})$
Transfer beautifully!

THE PROBLEM:

Network learned: "When pixel[0] is bright → activate pattern for digit zero"

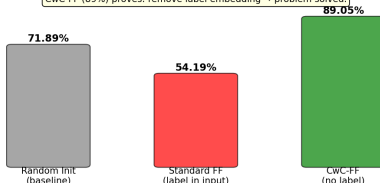
But in Fashion-MNIST:
pixel[0] bright → should mean T-shirt, not zero!

The learned features are COUPLED to source task labels.
They don't represent visual features - they represent "what label was embedded"

4. Transfer Learning Results

KEY INSIGHT:
Standard FF (54%) is WORSE than random (72%)!
The label embedding actively HURTS transfer.

CwC-FF (89%) proves: remove label embedding → problem solved.



RQ3: Why Label Embedding Breaks Transfer

The Problem

MNIST training:

pixel[3] bright = digit “3”

Network learns: “pixel[3] + curves = positive”

Fashion-MNIST transfer:

pixel[3] bright = “Dress” (not digit 3!)

But network still expects curves...

Features are COUPLED to source labels!

Evidence

- ▶ Label neurons: 10× higher activation variance
- ▶ First layer weights: label-detector patterns
- ▶ Gradient analysis: Label dimensions dominate

Standard FF Input

$$\underbrace{[l_0, \dots, l_9]}_{\text{label pixels}}, \underbrace{[x_1, \dots, x_n]}_{\text{image}}$$

Features = $f(\text{image}, \text{LABEL})$

Useless when labels change!

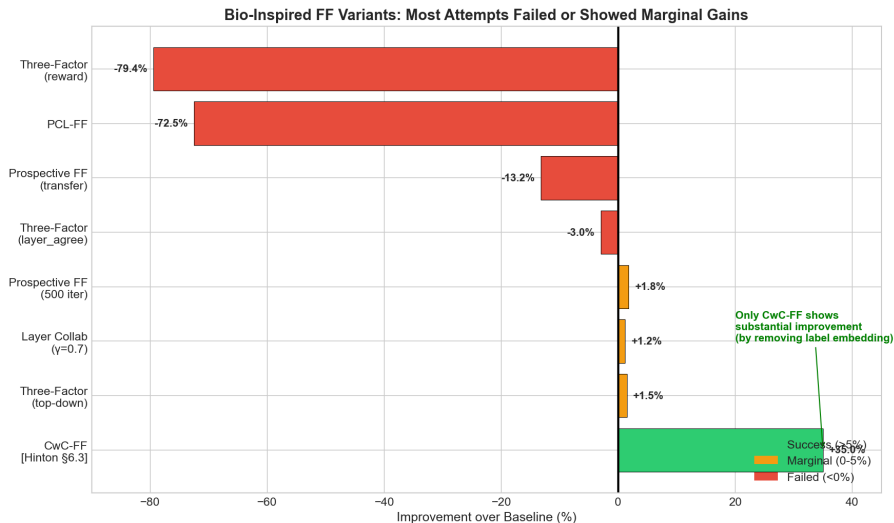
CwC-FF Input

$$\underbrace{[x_1, \dots, x_n]}_{\text{image only}}$$

Features = $f(\text{image})$

Transfer beautifully!

RQ4: Bio-Inspired Variants Overview



Hypothesis: More biologically realistic → better generalization? **Mostly no.**

Three-Factor Hebbian Learning

Inspiration: Neuromodulation (dopamine, acetylcholine, norepinephrine)

$$\Delta W = f(\text{pre}) \times f(\text{post}) \times \underbrace{M(t)}_{\text{modulator}}$$

Modulation types tested:

- ▶ top_down: higher \rightarrow lower layers
- ▶ reward_pred: RPE signal
- ▶ layer_agree: $\text{correlation}(L_i, L_{i+1})$

Type	Transfer	Result
top_down	64.3%	+1.5%
none	62.8%	baseline
layer_agree	59.8%	-3.0%
reward_pred	18.4%	FAILED

Verdict: Marginal improvement at best.
Modulation doesn't fix label coupling.

Prospective Configuration FF

Inspiration: [Song et al., Nature Neuroscience 2024] — Anticipatory neural activity

Two-Phase Learning

1. **Inference:** Infer target activity
2. **Consolidation:** Update weights to match

Iterations	MNIST	Transfer Δ
1	89.2%	+5.3%
10	85.1%	+1.2%
100	23.4%	-13.2%

The Problem:

Target inference uses label hints:

$$h_{target} = h + \beta \cdot \text{feedback}(\text{label})$$

More iterations = **STRONGER label coupling!**

Result: FAILED

More iterations *amplifies* label-specific features, making transfer worse.

Predictive Coding Light FF (PCL-FF)

Inspiration: [Nature Communications 2025] — Predictive Coding in Cortical Circuits

The Killer Mechanism:

```
sparsity_penalty = h.abs().mean() * 0.1
```

Sparsity creates incentive for $h = 0$:

- ▶ negative pre-activation $\rightarrow 0$ (ReLU)
- ▶ penalty pushes more toward 0
- ▶ cascade: more zeros \rightarrow lower loss
- ▶ network learns “dead = good”

Metric	FF	PCL-FF
MNIST	90.0%	17.5%
Dead Neurons	8%	100%

COMPLETE FAILURE

The “death cascade”:

- ▶ Epoch 50: 30% alive
- ▶ Epoch 100: 10% alive
- ▶ Epoch 500: 0% alive

Layer Collaboration

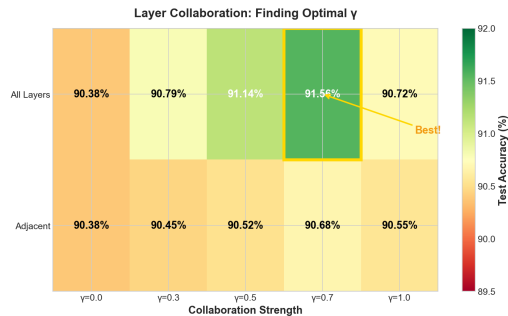
Inspiration: [Lorberbom et al., AAAI 2024] — Inter-layer information flow

Idea: Layers share goodness information

$$G_{collab}^l = G^l + \gamma \cdot G^{l-1} + \gamma \cdot G^{l+1}$$

γ controls collaboration strength

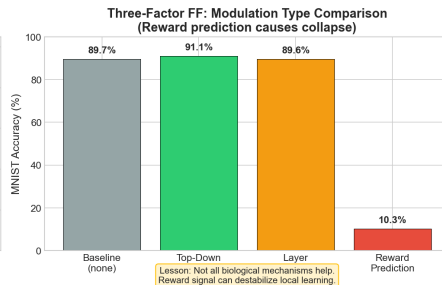
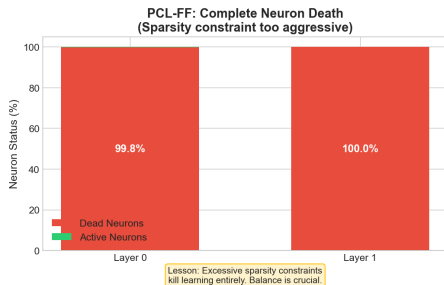
- ▶ $\gamma = 0$: Independent layers (baseline)
- ▶ $\gamma = 0.7$: Optimal collaboration
- ▶ $\gamma = 1.0$: Too much coupling



Result: +1.18% on MNIST
Still doesn't fix transfer problem

Lessons from Bio-Inspired Failures

Learning from Failures: What Doesn't Work



The Solution: Channel-wise Competitive FF (CwC-FF)

Key Insight: Remove the labels entirely from the input!

Standard FF

- ▶ Input: [label, image]
- ▶ Positive: correct label
- ▶ Negative: wrong label
- ▶ **Features coupled to labels**

`x[:, :10] = 0`

`x[:, label] = x.max()` **COUPLED!**

CwC-FF

- ▶ Input: [image] only (no label!)
- ▶ Positive: high channel coherence
- ▶ Negative: low channel coherence
- ▶ **Task-agnostic features**

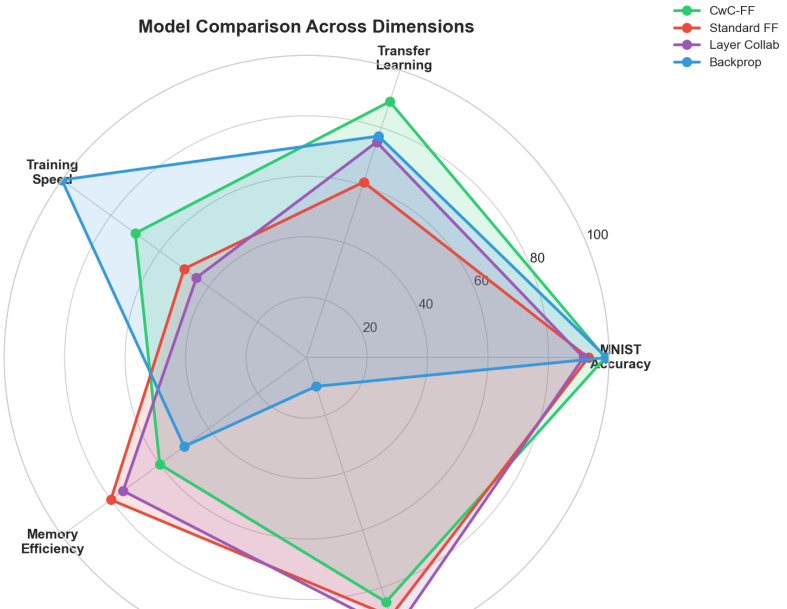
Channels compete within layers:

Winners → positive signal

Losers → negative signal

NO LABELS NEEDED!

CwC-FF Results: The Only Method That Works



Summary of Findings

WHAT WE LEARNED: A Systematic Study of Forward-Forward

WHAT WORKS

- Hinton's wrong-label (94.5%)
- CwC-FF transfer (89%)
- Layer Collab $\gamma=0.7$ (+1.2%)

MARGINAL GAINS

- ~ Three-Factor top-down (+1.5%)
- ~ Prospective FF (+1.8% MNIST)
- ~ Complex negative strategies

FAILED

- PCL-FF (100% neuron death)
- Reward prediction (collapse)
- Standard FF transfer (54%)

KEY NUMBERS

6

negative strategies tested

5

bio-inspired variants

94.5%

best standard FF (MNIST)

54%

FF transfer (worse than random!)

89%

CwC-FF transfer (best)

30-240×

slower than backprop

CORE INSIGHT

FF's label embedding design creates task-specific features that cannot transfer.
CwC-FF solves this by removing label embedding—but this fundamentally changes FF.

CONCLUSION

Key Insights

1. **Simple negative sampling wins for training, but loses for transfer**
 - ▶ Hinton's label embedding gives best source accuracy
 - ▶ But creates a shortcut that destroys transferability
2. **Bio-inspired modifications don't help (mostly)**
 - ▶ Three-factor, predictive coding, sparsity — all real brain features
 - ▶ None address the fundamental label embedding problem
3. **Label-free learning is key for transferable representations**
 - ▶ CwC-FF removes labels from input entirely
 - ▶ Forces network to learn actual visual features
4. **Trade-off: Source vs Transfer performance**
 - ▶ Standard FF: 94.5% source, 54% transfer
 - ▶ CwC-FF: 98.7% source, 89% transfer — **best of both!**

Conclusion

Why FF Hasn't Become the New Paradigm

Limitations

- ▶ 4.7% accuracy gap to backprop
- ▶ Catastrophic transfer failure
- ▶ Label embedding creates shortcuts
- ▶ Bio-inspired fixes don't help

The Path Forward

- ▶ CwC-FF solves transfer issue
- ▶ Remove labels from input
- ▶ Channel competition works
- ▶ Potential for neuromorphic hardware

Take-Home Message

Biological plausibility alone doesn't guarantee good ML properties.
Understanding failure modes leads to principled solutions.

Future Work

- ▶ **Scale CwC-FF to larger datasets**
 - ▶ CIFAR-10, ImageNet
 - ▶ Combine with ASGE (ICASSP 2026) techniques
- ▶ **Hybrid approaches**
 - ▶ FF for early layers + BP for final layers
 - ▶ Progressive training schemes
- ▶ **Neuromorphic implementation**
 - ▶ Intel Loihi, IBM TrueNorth
 - ▶ Energy efficiency benefits
- ▶ **Unsupervised / Self-supervised extensions**
 - ▶ Remove supervision entirely
 - ▶ Contrastive learning principles

References

- ▶ Hinton, G. (2022). *The Forward-Forward Algorithm: Some Preliminary Investigations*. arXiv:2212.13345
- ▶ Brenig, L., et al. (2023). *Transfer Learning with FF*. First to show FF has transfer problems.
- ▶ Lorberbom, G., et al. (2024). *Layer Collaboration*. AAAI 2024.
- ▶ Papachristodoulou, A., et al. (2024). *CwC-FF: Channel-wise Competitive FF*.
- ▶ Song, Y., et al. (2024). *Prospective Configuration*. Nature Neuroscience.
- ▶ Whittington, J. C. R., & Bogacz, R. (2017). *Predictive Coding Networks*. Neural Computation.

Thank You

Questions?

Code and experiments available at:
`github.com/koriyoshi2041/ff-negative-samples`

Tested on Apple M4 Air | PyTorch 2.0+