# Hybrid Equilibrium Propagation: Trading Estimator Bias for Compute

**Author:** Stefan Stoian **Supervisors**: Stephanie Tan, Yaqi Guo

## 1 INTRODUCTION

While **backpropagation** powers deep learning, it cannot run efficiently on **neuromorphic hardware**, is **biologically implausible**, and is **energy-costly** - driving interest in backpropagation-free learning.

**Equilibrium Propagation** (EP) is a backpropagation-free alternative: its **standard** estimator (*std*) computes the gradient by comparing the equilibrium states reached in two different phases. The **centered** estimator (*cep*) reduces this bias and improves accuracy, but adds a third phase, increasing the cost.

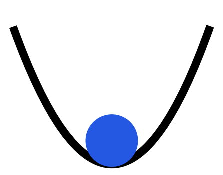
This work introduces **hybrid EP**, a new family of estimators that aims to combine the low cost of standard EP with the stability and accuracy of centered EP.

## 2 BACKGROUND

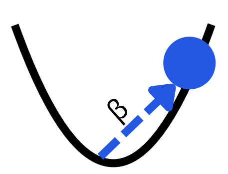
| Estimator | Phases | Bias | Meaning |
|---|---|---|---|
| Standard EP | 1 free + 1 nudged | O(β) | cheap, biased |
| Centered EP | 1 free + 2 nudged | O(β²) | accurate, costly |
| Hybrid EP *proposed* | std or cep | O(pβ + β²) | trade-off |

*Notation: **β** = nudging strength, **p** = probability of std update.

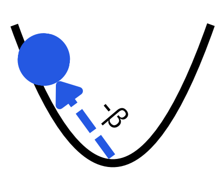
**FREE PHASE**



**NUDGED PHASE**




**EXTRA NUDGED PHASE**




## 3 RESEARCH QUESTION

Under what conditions can **hybrid Equilibrium Propagation** reduce the computational **cost** of centered EP while preserving training **stability** and classification **accuracy**?

## 4 METHOD

Per minibatch, apply **std** with probability ***p***, **cep** otherwise. All hybrids spend ≈ half their updates on std - they differ only in how those biased updates are **distributed** over training:

**Cosine** *primary* $\rightarrow p_{\text{std}}(e) = \frac{1}{2}\left(1 + \cos\left(\frac{\pi e}{E}\right)\right)$

**Cheap** first, **accurate** late.

**Inverse Cosine** $\rightarrow p_{\text{std}}(e) = \frac{1}{2}\left(1 - \cos\left(\frac{\pi e}{E}\right)\right)$

**Accurate** first, **cheap** late.

**Stochastic** $\rightarrow p_{\text{std}} = 0.5$

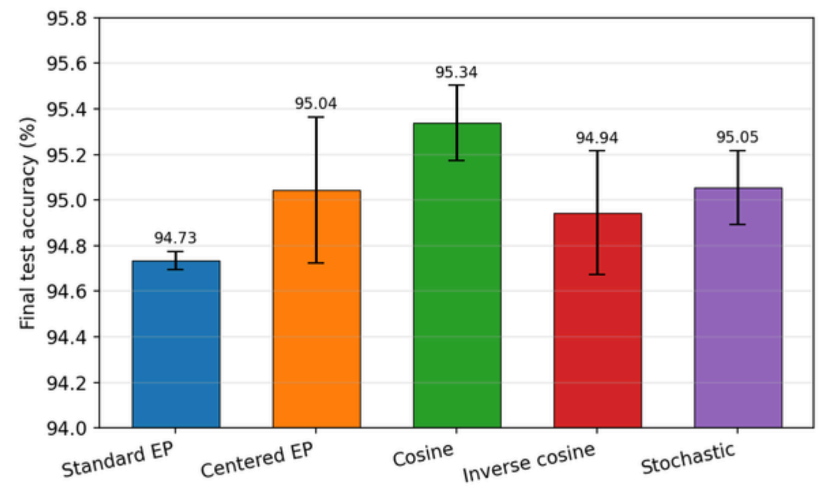
**Uniformly** mixed updates.

*Notation: **e** = current epoch, **E** = total epochs.

We compare **std**, **cep**, and the three **hybrid schedules** on **MNIST**, **Fashion-MNIST**, using **MLPs,** and on **CIFAR-10** using a **CNN**, measuring **accuracy** and **nudged-phase** count.
A **single** fixed optimization setup is used across all strategies. Results are averaged over 5 **seeds** (3 on CIFAR-10), with no runs discarded.

## 5 RESULTS - MNIST

Does **hybrid EP** help on **simple** benchmarks?

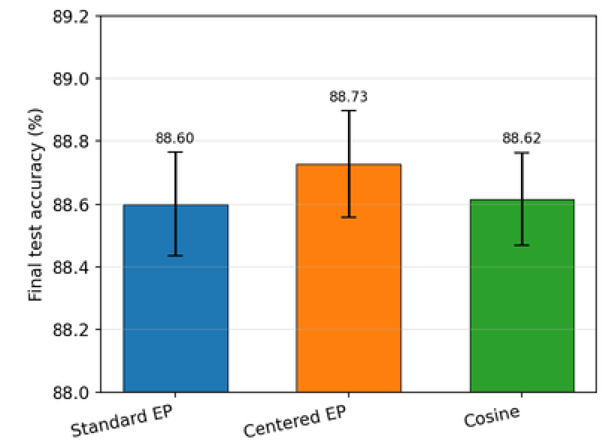



As expected **by design**, all hybrid strategies sit near the **midpoint** between std and cep in nudged-phase count.

At equal compute, cosine outperforms its inverse: **95.34%** vs **94.94%.** This shows that **order matters**, not just mixing.

## 6 RESULTS - Fashion-MNIST

Does the **cosine hybrid** still work under **domain shift**?

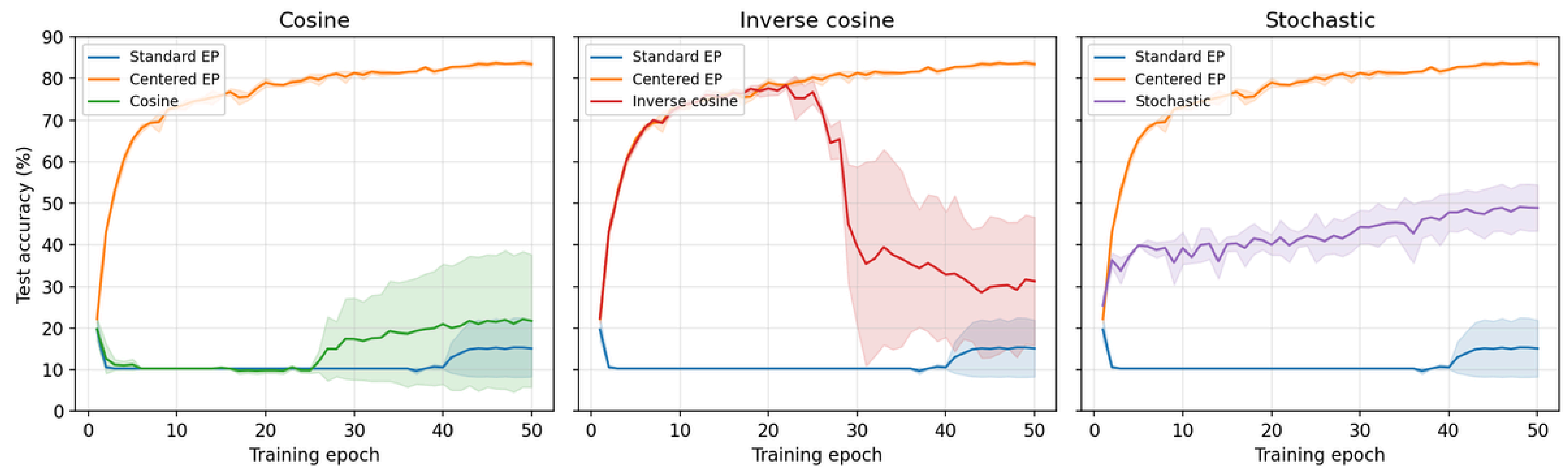



| Strategy | Test acc. (%) | Nudged phases |
|---|---|---|
| Standard EP | $88.60 \pm 0.16$ | 23,450 |
| Centered EP | $\mathbf{88.73 \pm 0.17}$ | 46,900 |
| Cosine hybrid | $88.62 \pm 0.15$ | 34,962 |

All three strategies reach almost the **same accuracy**, with differences inside seed-to-seed **variation**.

## 7 RESULTS - CIFAR-10

What happens when **hybrid EP** is used on a **harder** task?




Std **collapses** on CIFAR-10, and long std-dominated hybrid phases collapse with it. Only stochastic mixing stays **stable**, showing biased updates must be **spread out**, not concentrated.

## 8 CONCLUSION

Compute savings are real but **task-dependent** - realized cleanly only when both estimators are individually **viable**.

**Hybrid EP** uses the mixing probability ***p*** to trade off **std cost** against **cep accuracy**, with bias **O(pβ + β²)**.

Stability is governed by the distribution of biased updates in time, not their number.

## 9 REFERENCES

[1] Scellier, Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation.
[2] Laborieux et al. Scaling EP to deep ConvNets by reducing gradient estimator bias.
[3] Loshchilov, Hutter. Sgdr: Stochastic gradient descent with warm restarts.
[4] Scellier et al. Energy-based learning algorithms for analog computing: a comparative study.