Stochastic Forward-Forward Learning through Representational Dimensionality Compression

Zhichao Zhu, Yang Qi, Hengyuan Ma, Wenlian Lu, Jianfeng Feng
Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, China

Motivation

- Backpropagation (BP) lies at the heat of the success of conventional deep learning but it is **biologically implausible**.

- The Forward-Forward (FF) algorithm offers **a bottom-up alternative** to BP for training neural networks. However,
  - It requires high-quality negative samples.
  - Existing “goodness” functions depend only on squared activations, ignoring **correlation among neurons**.

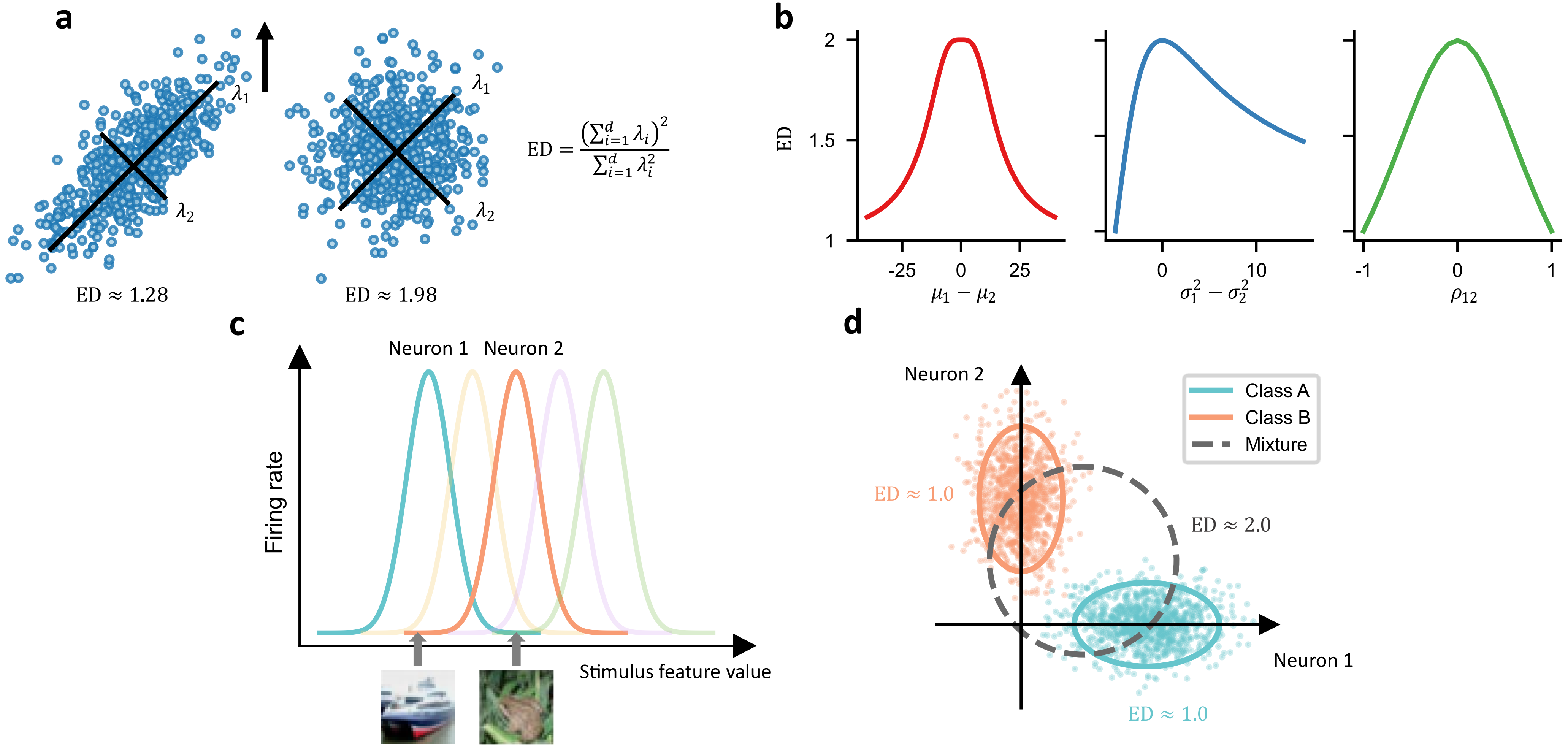
- Noise is a fundament property for the computation in biological brain.
  - Both the neural variability and the noise correlation can affect the quality of a neural code.

- **Goal:** Design a *second-order* local loss that actively take noise into computation and bypass the need of generating negative samples.

Effective dimensionality as a goodness function

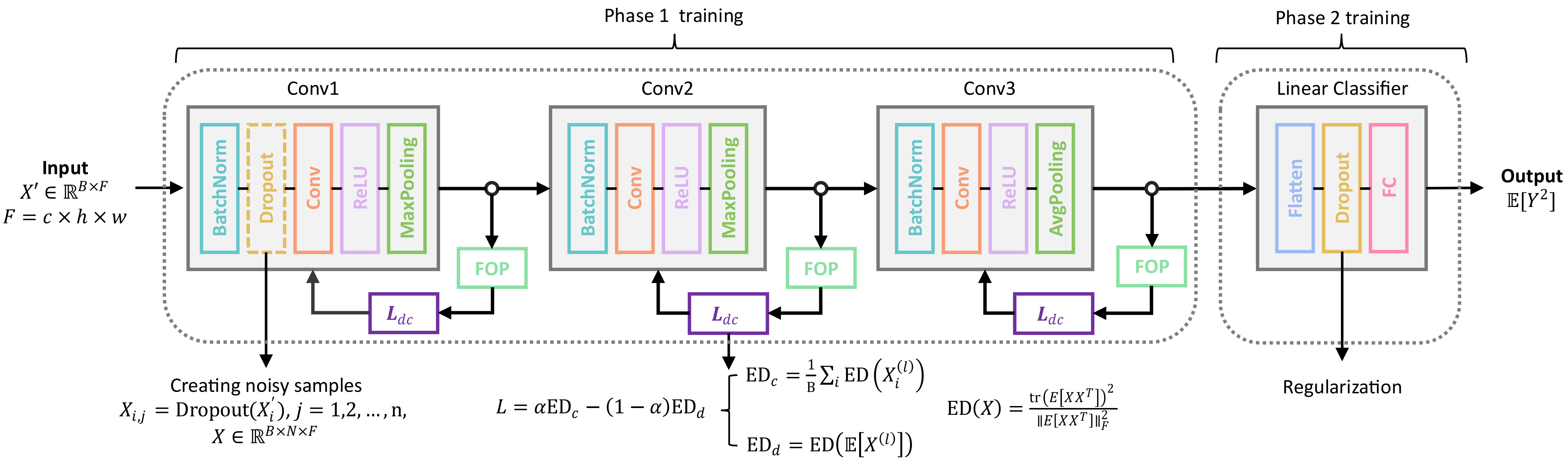
ED in data analysis is primarily based on centered covariance.

$$\mathrm{ED}\left(X^{(l)}\right) = \frac{\mathrm{tr}\left(\mathbb{E}\left[X^{(l)T}X^{(l)}\right]\right)^2}{\left\|\mathbb{E}\left[X^{(l)T}X^{(l)}\right]\right\|_F^2} = \frac{\left(\sum_{i=1}^{d}\lambda_i\right)}{\sum_{i=1}^{d}\lambda_i^2}$$



$$L = \alpha \mathrm{ED}_c - (1-\alpha)\mathrm{ED}_d, \qquad \mathrm{ED}_c = \frac{1}{B}\sum_{i=1}^{B}\mathrm{ED}\left(X_i^{(l)}\right) \qquad \mathrm{ED}_d = \mathrm{ED}\left(\mathbb{E}\left[X^{(l)}\right]\right)$$

Training and inference protocols



- Using dropout to create a set of noisy variants for each input.
- Two phase training: first train convolutional blocks using L block by block, then train the linear classifier solely to estimate the how well the network has learnt.
- For the phase 1 training, project $X^{(l)}$ to a lower dimension through a fixed orthonormal projection (FOP) module before computing ED.
- Using the energy term $E[Y^2]$ as the predictive score rather than $E[Y]$.
- Network architecture remained the same for all experiments.

Comparable performance with other non-BP methods

Method	Validation Accuracy (%)		
	MNIST	CIFAR10	CIFAR100
BP♠	99.33±0.04	82.50±0.09	61.28±0.25
DFA	98.98±0.05	73.10±0.50	41.00±0.25
Original FF	98.73	59	-
CaFo FF	98.95	69.49	42.13
CwC FF	99.42 ± 0.08	78.11 ± 0.44	51.32
Soft Hebbian♠	99.35±0.03	80.31±0.14	56.00
Hard Hebbian♠	-	76.00	-
GIM ♠♣	99.29±0.03	78.19±0.34	50.09±0.45
EBL♦	99.56	89.6	65.8
Proposed method	99.31±0.07	76.96±0.73	53.29±1.02

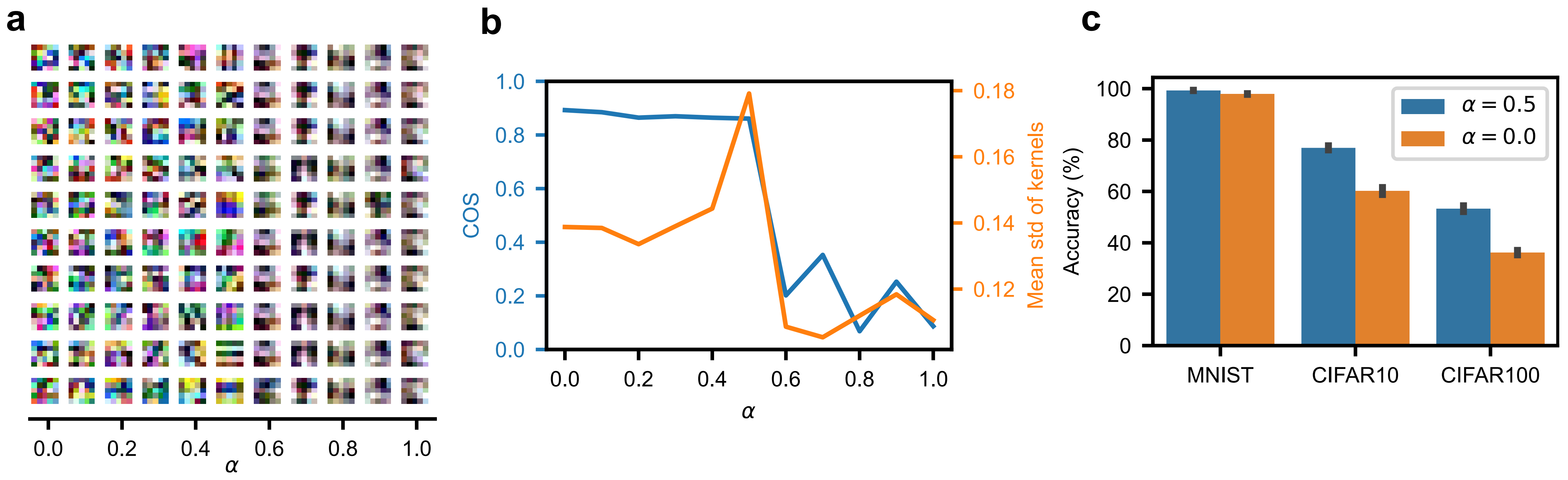
♠: comparable architecture.
♣: reimplementation results.
♦: used a deeper network, but adding more than three convolutional blocks with our method leads to a performance drop.

Optimizing ED leads to orthogonal weights

- When $\alpha > 0.5$, convolutional kernels collapse to the similar patterns.
- When $\alpha \leq 0.5$, it leads to almost orthogonal weights ($\mathrm{COS} > 0.8$).

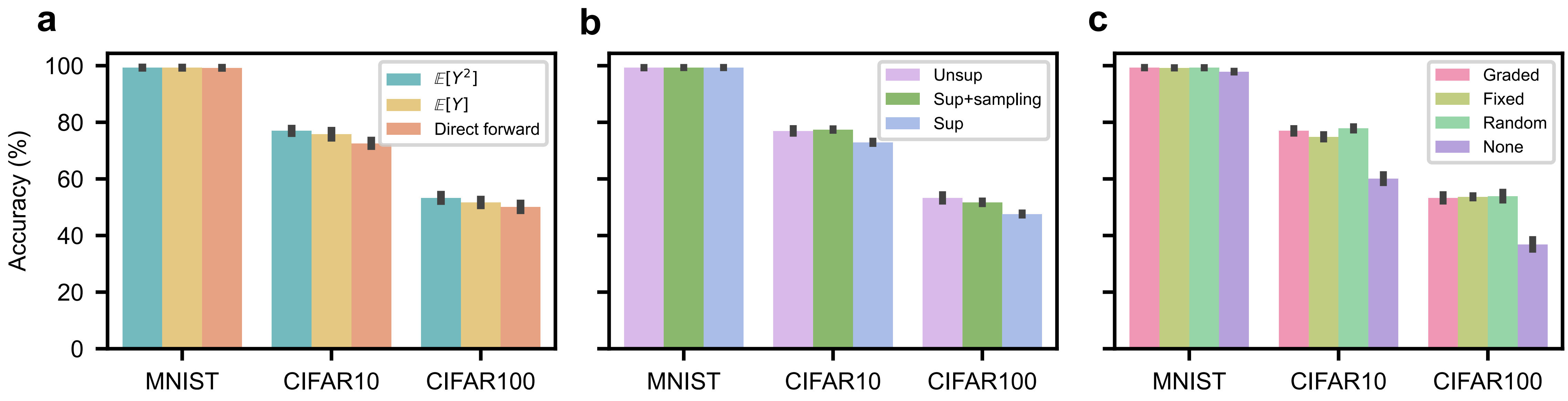
$$\mathrm{COS} = \frac{1}{K}\left(\sum_{j<i} 1 - \left|\frac{\langle w_i, w_j\rangle}{\|w_i\|\|w_j\|}\right|\right)$$

- Both ED_c and ED_d necessitate for achieving a good performance.



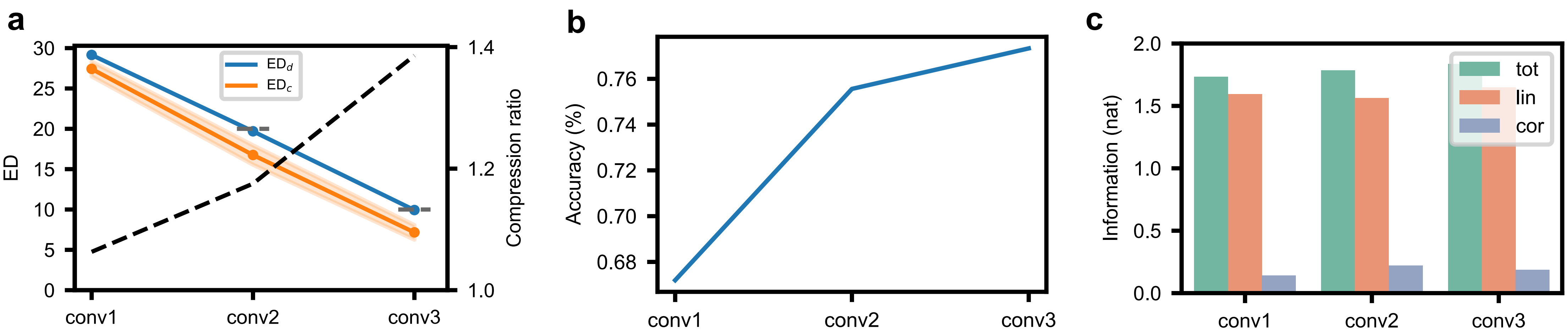
Factors that affecting task performance

- The linear classifier can have a slightly better performance when using $E[Y^2]$
  - Both the mean and variance can carry the information about the class.
- Unsupervised learning is enough.
  - Use the label information to actively group $X^{(l)}$ for ED_c is unnecessary.
  - Noisy sampling improves the performance.
- Projecting $X^{(l)}$ to lower dimension facilitate learning, no matter what projecting scheme is used.



Higher compression ratio leads to better performance

- The ratio $\frac{\mathrm{ED}_d}{\mathrm{ED}_c}$ computed in the projected space increase with the network depth
- The linear separability of the internal representations also increases.
- Correlation among neurons also carried some information about the label. However, such information cannot be readout using a linear method.



Discussion

Theoretical connections:

- The dual objective of minimizing ED_c and maximizing ED_d parallels **predictive coding**:
  - ED_c : stable responses to noisy inputs → *prediction consistency*
  - ED_d : diverse representations across inputs → *novelty encoding*

Biological feasibility:

- Winner-Take-All (WTA) circuits reduce ED_c .
- Inhibitory competition & divisive normalization can expand ED_d .

Next steps:

- Scaling to deeper architectures and larger datasets.
- Developing **local circuit models** implementing ED_c /ED_d trade-off.
- Exploring implementation on **neuromorphic hardware** where noise is inevitable .

Thanks!