

## Introduction

- Approximations of the **Fisher Information Matrix (FIM)** are crucial for deep neural networks.
- The **Kronecker-Factored Approximate Curvature (K-FAC)** approximates the FIM with a block-diagonal Kronecker-factored matrix using independence assumptions that are usually **not met in practice**.
- We propose the **Kronecker-Factored Optimal Curvature (K-FOC)** which is an **optimal** and **scalable** block-diagonal Kronecker-factored approximation of the FIM.

## Background

### Neural Networks

- **a**: activation of layer  $l - 1$ , **s**: pre-activation of layer  $l$
- Fully-connected layer:  

$$\mathbf{s} = \mathbf{W} \begin{pmatrix} \mathbf{a} \\ 1 \end{pmatrix} = \bar{\mathbf{a}}, \quad \mathbf{W} \in \mathbb{R}^{d_l \times (d_{l-1} + 1)}$$
- Convolutional layer:  

$$\mathbf{s}_{k,t} = \mathbf{b}_k + \sum_{k'=1}^{c_{l-1}} \sum_{\delta \in \Delta} \mathbf{W}_{k,k',\delta} \mathbf{a}_{k',\zeta(\mathbf{t},\delta)}, \quad \mathbf{W} \in \mathbb{R}^{c_l \times c_{l-1} \times h_l^\Delta \times w_l^\Delta}$$

$$\hat{\mathbf{s}} = \hat{\mathbf{a}}(\hat{\mathbf{W}})^T, \quad \hat{\mathbf{W}} \in \mathbb{R}^{c_l \times (c_{l-1}|\Delta| + 1)} =: \mathbb{R}^{c_l \times d_l} \quad (\text{extended vectorized form})$$

### Fisher Information Matrix

- Diagonal block for layer  $l$ :  

$$\mathbf{F}_l = \mathbb{E}_{\mathbf{x} \sim Q_x} \mathbb{E}_{\mathbf{y} \sim p(\cdot | \mathbf{x}, \theta)} \left[ \frac{d \ln p(\mathbf{y} | \mathbf{x}, \theta)}{d\theta_l} \frac{d \ln p(\mathbf{y} | \mathbf{x}, \theta)}{d\theta_l} \right]$$

$$=: \mathbb{E} \left[ \mathcal{D}\theta_l (\mathcal{D}\theta_l)^T \right]$$
- Fully-connected layer:  

$$\mathbf{F}_l = \mathbb{E} [\mathcal{D}\mathbf{s} (\mathcal{D}\mathbf{s})^T \otimes \bar{\mathbf{a}} (\bar{\mathbf{a}})^T]$$

$$\approx \sum_{k=1}^{K_p} \mathcal{D}\mathbf{s}^k (\mathcal{D}\mathbf{s}^k)^T \otimes \frac{1}{K_p} \bar{\mathbf{a}}^k (\bar{\mathbf{a}}^k)^T \quad (K_p: \text{batch size})$$
- Convolutional layer:  

$$\mathbf{F}_l = \mathbb{E} \left[ \sum_{\mathbf{t} \in \mathcal{T}} \sum_{\mathbf{t}' \in \mathcal{T}} \mathcal{D}\mathbf{s}_{\mathbf{t}} (\mathcal{D}\mathbf{s}_{\mathbf{t}'})^T \otimes \hat{\mathbf{a}}_{\mathbf{t}} (\hat{\mathbf{a}}_{\mathbf{t}'})^T \right]$$

$$\approx \sum_{k=1}^{K_p} \sum_{\mathbf{t} \in \mathcal{T}} \sum_{\mathbf{t}' \in \mathcal{T}} \mathcal{D}\mathbf{s}_{\mathbf{t}}^k (\mathcal{D}\mathbf{s}_{\mathbf{t}'}^k)^T \otimes \frac{1}{K_p} \hat{\mathbf{a}}_{\mathbf{t}}^k (\hat{\mathbf{a}}_{\mathbf{t}'}^k)^T$$

⇒ For both layer types, a sum of Kronecker products needs to be computed.

## Our Method

### Approximation of Sums of Kronecker Products

*Problem.*

$$\hat{\mathbf{L}}, \hat{\mathbf{R}} \in \arg \min_{\mathbf{L} \in \mathbb{R}^{M \times M}, \mathbf{R} \in \mathbb{R}^{N \times N}} \left\| \sum_{k=1}^K \mathbf{L}^k \otimes \mathbf{R}^k - \mathbf{L} \otimes \mathbf{R} \right\|_F. \quad (1)$$

**Lemma 3.1.** Let  $M, N, K \in \mathbb{N}$ ,  $\mathbf{L}^k \in \mathbb{R}^{M \times M}$  and  $\mathbf{R}^k \in \mathbb{R}^{N \times N}$  for  $k \in [K]$ . Then

$$\left\| \sum_{k=1}^K \mathbf{L}^k \otimes \mathbf{R}^k - \mathbf{L} \otimes \mathbf{R} \right\|_F = \left\| \sum_{k=1}^K \text{vec}(\mathbf{L}^k) \text{vec}(\mathbf{R}^k)^T - \text{vec}(\mathbf{L}) \text{vec}(\mathbf{R})^T \right\|_F.$$

**Lemma 3.2.** Let  $\mathbf{A} = \sum_{k=1}^K \text{vec}(\mathbf{L}^k) \text{vec}(\mathbf{R}^k)^T$  and  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$  be its singular value decomposition with  $\sigma_1 \geq \dots \geq \sigma_r > 0$  and  $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \mathbb{1}[i=j]$ . Then there is a solution of Equation 1 with

$$\text{vec}(\hat{\mathbf{L}}) = \mathbf{u}_1, \text{vec}(\hat{\mathbf{R}}) = \sigma_1 \mathbf{v}_1.$$

If  $\sigma_1 > \sigma_2$ , the solution is unique up to changing the sign of both factors and Algorithm 1 converges almost surely to this solution.

### Kronecker-Factored Optimal Curvature

As a consequence of Lemma 3.2, we can find optimal factors with the **power method**. However, computing  $\mathbf{A}\mathbf{A}^T \text{vec}(\hat{\mathbf{L}})$  has a complexity of  $\mathcal{O}(n^{\max} K(N^2 + M^2))$  with  $\mathcal{O}(K(N^2 + M^2))$  memory. Therefore, we incorporate the structure of the matrix and its factors to **reduce the complexity**:

- **Algorithm 1:** Power method for sums of Kronecker products

**Main idea:**

- $\mathbf{A} = \sum_{k=1}^K \text{vec}(\mathbf{L}^k) \text{vec}(\mathbf{R}^k)^T$  is a **sum of outer products**.

$$\Rightarrow \hat{\mathbf{R}} \leftarrow \sum_{k=1}^K \langle \mathbf{L}^k, \hat{\mathbf{L}} \rangle_F \mathbf{R}^k, \quad \hat{\mathbf{L}} \leftarrow \sum_{k=1}^K \langle \mathbf{R}^k, \hat{\mathbf{R}} \rangle_F \mathbf{L}^k$$

- **Algorithm 2:** Adaption for **convolutions** (and fully-connected layers viewed as convolutions with  $\mathcal{T} = \{(1, 1)\}$ )

**Main idea:**

- $\mathbf{L}^{k,t,t'} = \mathcal{D}\mathbf{s}_{\mathbf{t}}^k (\mathcal{D}\mathbf{s}_{\mathbf{t}'}^k)^T$  and  $\mathbf{R}^{k,t,t'} = \hat{\mathbf{a}}_{\mathbf{t}}^k (\hat{\mathbf{a}}_{\mathbf{t}'}^k)^T$  are **outer products** of vectors.
- The summation is over **all combinations** of  $\mathbf{t}, \mathbf{t}' \in \mathcal{T}$ .

⇒ Pre-compute  $\mathbf{X}^k = (\mathcal{D}\mathbf{s}^k)^T \hat{\mathbf{a}}^k$  for  $k \in [K]$

$$\hat{\mathbf{R}} \leftarrow \sum_{k=1}^{K_p} \mathbf{X}^k \hat{\mathbf{L}} (\mathbf{X}^k)^T, \quad \hat{\mathbf{L}} \leftarrow \sum_{k=1}^{K_p} (\mathbf{X}^k)^T \hat{\mathbf{R}} \mathbf{X}^k$$

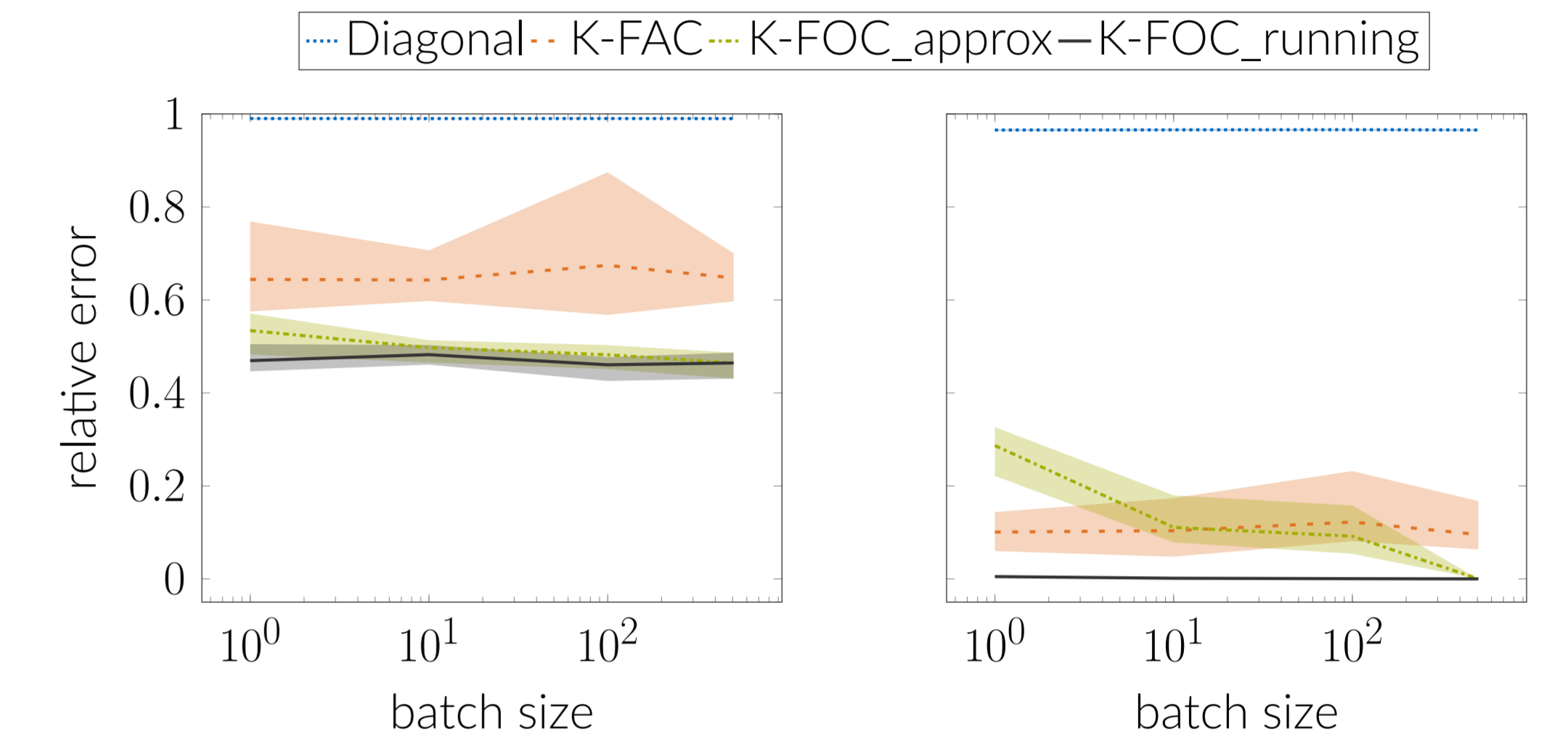
The algorithms have a **similar complexity** as K-FAC for practical applications.

Furthermore, we compare two methods to aggregate the batches in an online setting:

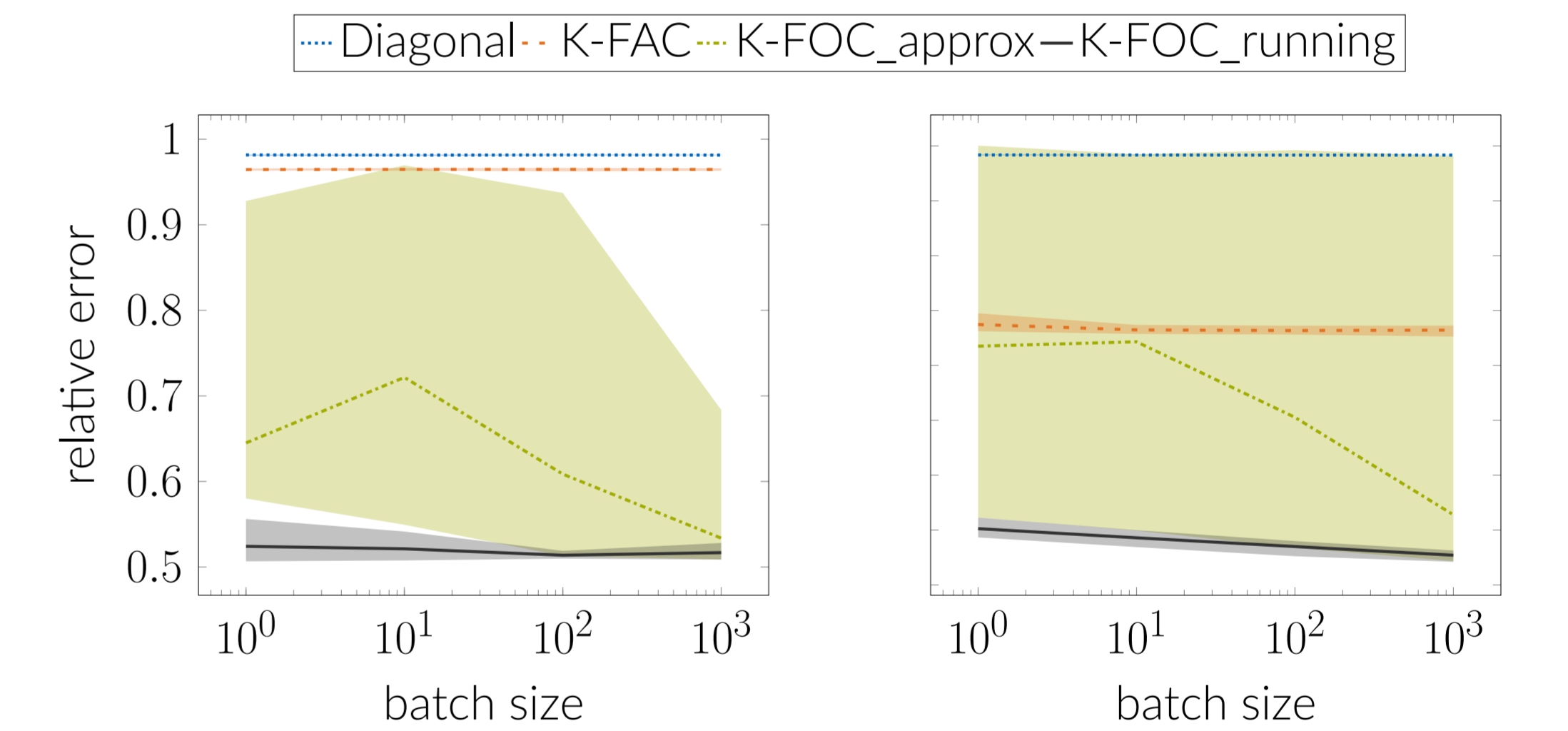
- **K-FOC\_approx:** similar aggregation as K-FAC
- **K-FOC\_running:** use Algorithm 1 to combine current estimate with a new batch

## Results

### Fully-connected layers



### Convolutional layers



## Conclusion

### The Kronecker-Factored Optimal Curvature

- uses the **power iteration** to approximate the FIM,
- is **tractable** for convolutional and fully-connected layers and
- approximates the FIM **more accurately** than K-FAC.

## References

- [1] Dominik Schnaus, Jongseok Lee, and Rudolph Triebel. Kronecker-factored optimal curvature. In *Bayesian Deep Learning Workshop*, 2021.
- [2] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417, Lille, France, 2015. PMLR.