EBERHARD KARLS UNIVERSITÄT TÜBINGEN
Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

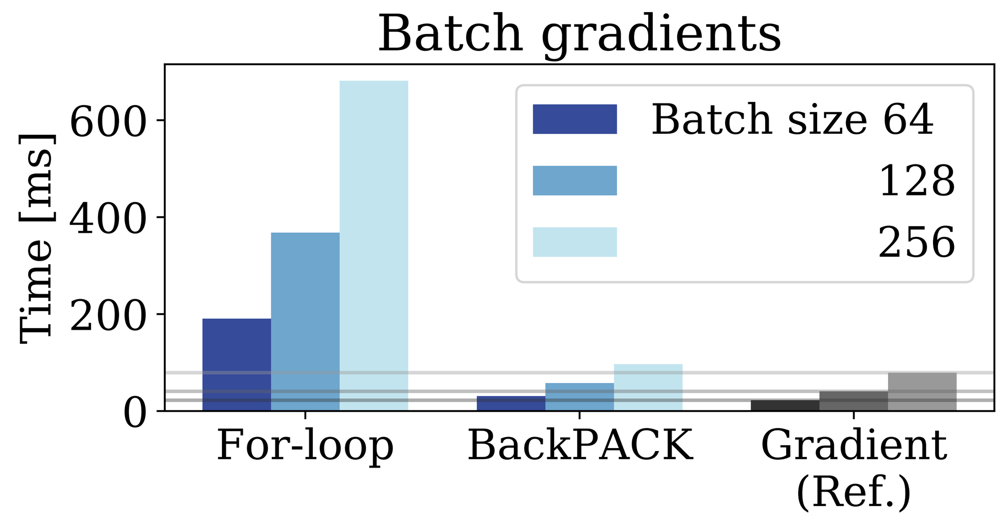
BackPACK: Packing more into Backprop

Felix Dangel, Frederik Künstner, and Philipp Hennig
University of Tübingen & Max Planck Institute for Intelligent Systems, Tübingen, Germany

(Code available at `https://github.com/f-dangel/backpack`)

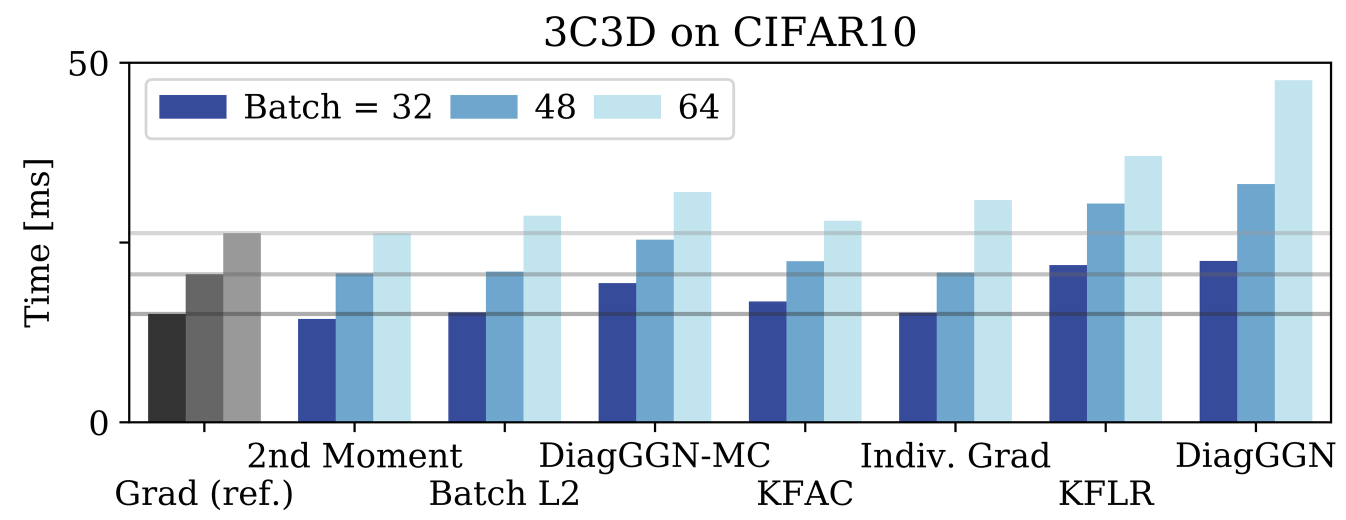
Max Planck Institute for Intelligent Systems
imprs-is

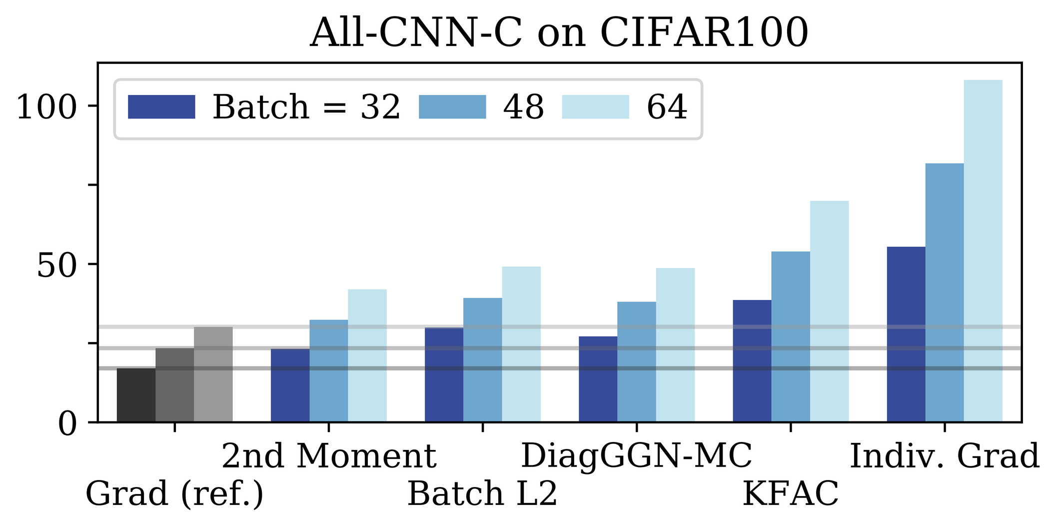
Why BackPACK?



(paraphrasing deep learning researchers, in particular on optimization)
"If A were accessible, then B **could be computed**, adding only **little overhead**. But unfortunately, this is **not possible in popular implementations**."

(Discussions and feature requests from the community)
`discuss.pytorch.org/t/`
➥ 1433, 8405, 17204, 24955: Individual gradients
➥ 15270: Hessian
➥ 19350: Variance





Get the Gradient with PyTorch...

```
X, y     = load_mnist_data()
model    = Linear(784, 10)
lossfunc = CrossEntropyLoss()

loss     = lossfunc(model(X), y)


loss.backward()

for param in model.parameters():
    print(param.grad)
```

github.com/f-dangel/backpack

...and the Variance with BackPACK

```
X, y     = load_mnist_data()
model    = extend(Linear(784, 10))
lossfunc = extend(CrossEntropyLoss())

loss     = lossfunc(model(X), y)

with backpack(Variance()):
    loss.backward()

for param in model.parameters():
    print(param.grad)
    print(param.var)
```

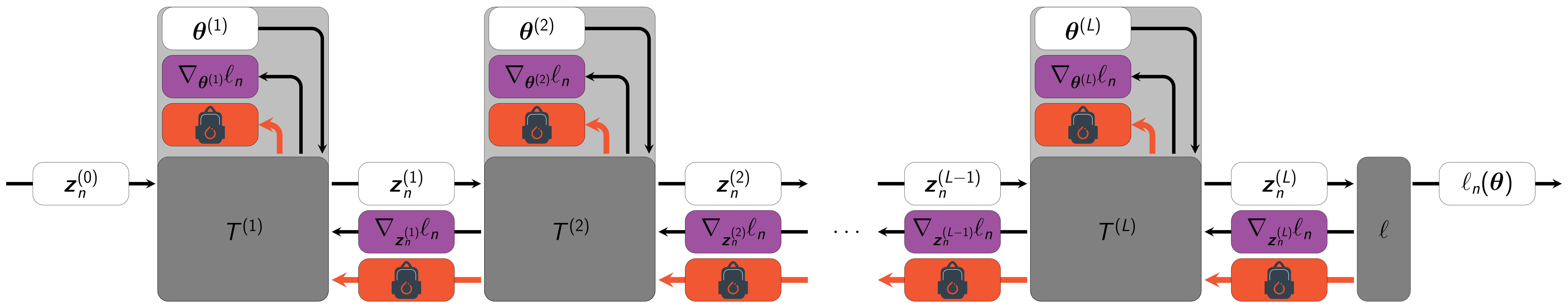
Internals & Assumptions

- Feedforward network

$$z_n^{(0)} = x_n,$$
$$z_n^{(L)} = (T^{(L)} \circ \ldots \circ T^{(1)})(z_n^{(0)}, \theta)$$

- Empirical risk

$$\tfrac{1}{N}\Sigma_{n=1}^{N} \ell(z_n^{(L)}, y_n) = \tfrac{1}{N}\Sigma_{n=1}^{N} \ell_n(\theta)$$



- First-order backprop

$$\nabla_{\theta^{(i)}}\ell_n(\theta) = (\mathrm{J}_{\theta^{(i)}} z_n^{(i)})^\top (\nabla_{z_n^{(i)}}\ell_n(\theta))$$

- Second-order backprop

$$\nabla^2_{\theta^{(i)}}\ell_n(\theta) = (\mathrm{J}_{\theta^{(i)}} z_n^{(i)})^\top (\nabla^2_{z_n^{(i)}}\ell_n(\theta)) (\mathrm{J}_{\theta^{(i)}} z_n^{(i)}) + \Sigma_j (\nabla^2_{\theta^{(i)}}[z_n^{(i)}]_j)[\nabla_{z_n^{(i)}}\ell_n(\theta)]_j,$$

...and more with BackPACK

- First-order extensions
  - `BatchGrad` (Individual gradients)
  - `BatchL2Grad` (Individual L_2 norm)
  - `SumGradSquared` (Gradient 2nd moment)
  - `Variance` (Gradient variance)
- Second-order extensions
  - `DiagGGNExact`, `DiagGGNMC` (Exact and Monte-Carlo sampled diagonal of the generalized Gauss-Newton (GGN))
  - `KFAC`, `KFLR`, `KFRA` (Kronecker-factored GGN approximations)
  - `DiagHessian` (Exact Hessian diagonal)

1. F. Dangel, F. Künstner, P. Hennig: BackPACK: *Packing more into backprop* (2019) [openreview.net/forum?id=BJlrF24twB]
2. F. Dangel, S. Harmeling, P. Hennig: *Modular Block-diagonal Curvature Approximations for Feedforward Architectures* (2019) [arxiv.org/abs/1902.01813]