

Towards training digitally-tied analog blocks *via* hybrid gradient computation



Timothy Nest*



Maxence Ernoult*

Context



A winning trio:

feedforward nets + backprop (BP) + GPUs

... yet extremely energy consuming

Context



An alternative:

energy-based models + equilibrium propagation + analog systems?

Context



An alternative:

energy-based models + equilibrium propagation + analog systems?
(EBMs)

“Forward pass” = energy minimization: $\nabla_1 E(s, \theta, x) = 0$

An alternative:

energy-based models + equilibrium propagation [1] + analog systems?
(EP)

Gradient computation with “forward passes” only
(beyond zeroth order [2] and without heuristics [3]):

$$\frac{dC}{d\theta} \approx_{\beta \rightarrow 0} \frac{1}{2\beta} (\nabla_2 E(s^\beta, \theta, x) - \nabla_2 E(s^{-\beta}, \theta, x))$$

with: $\nabla_1 E(s^{\pm\beta}, \theta, x) \pm \beta \ell(s^{\pm\beta}, y) = 0$

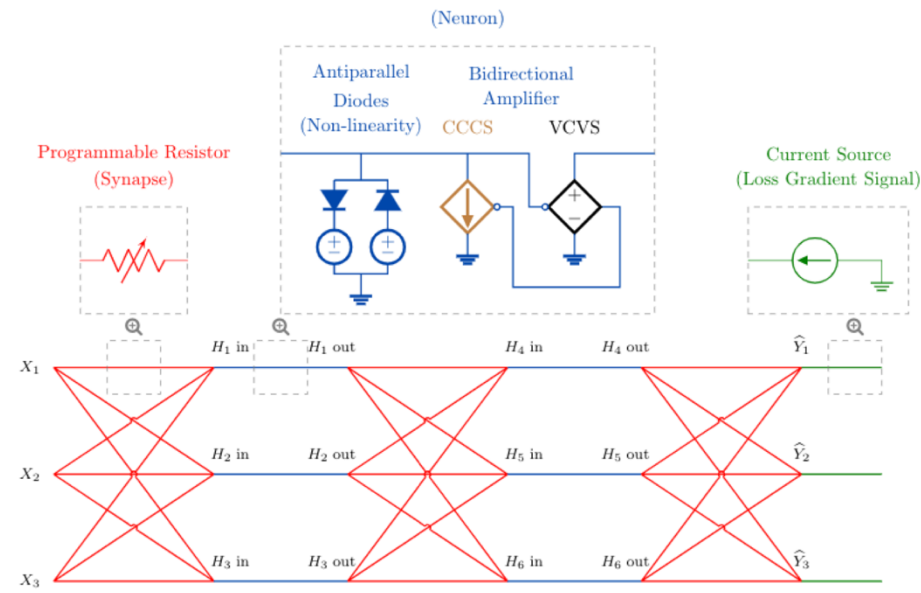
- [1] Scellier, B., & Bengio, Y. (2017). "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation"
- [2] Malladi, Sadhika, et al (2023). "Fine-tuning language models with just forward passes"
- [3] Hinton, G. (2022). "The forward-forward algorithm: Some preliminary investigations"

Context

An alternative:

energy-based models + equilibrium propagation + analog systems?

$$\nabla_1 E(s, \theta, x) = 0 \quad \equiv \quad [1,2]$$



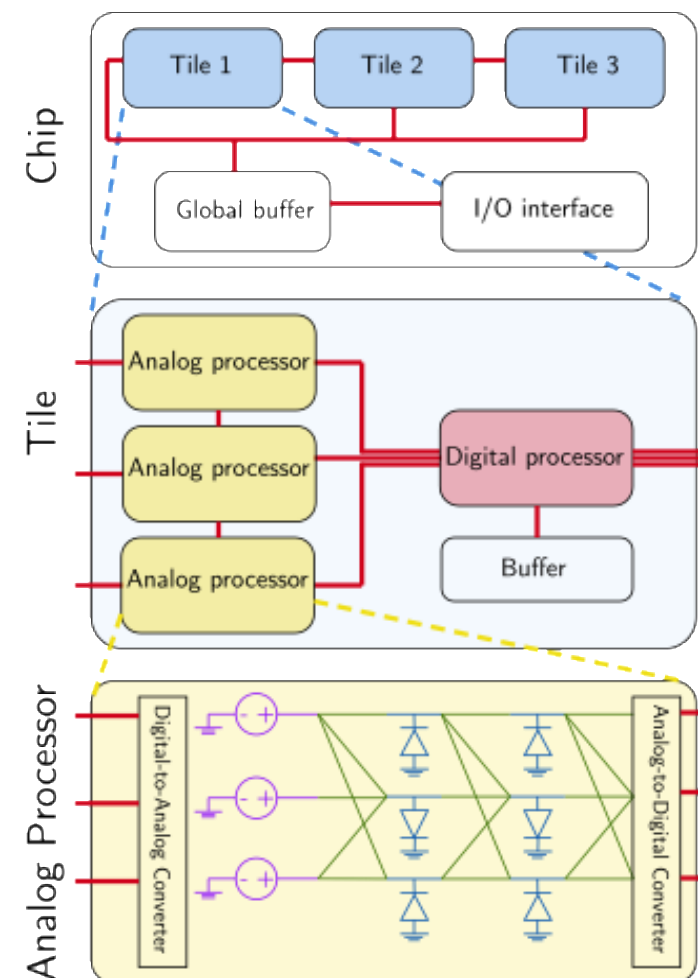
[1] Kendall, Jack, et al (2020). "Training end-to-end analog neural networks with equilibrium propagation"

[2] Scellier, B. (2024). "A Fast Algorithm to Simulate Nonlinear Resistive Networks"

Problem

Analog at scale requires digital circuitry [1]

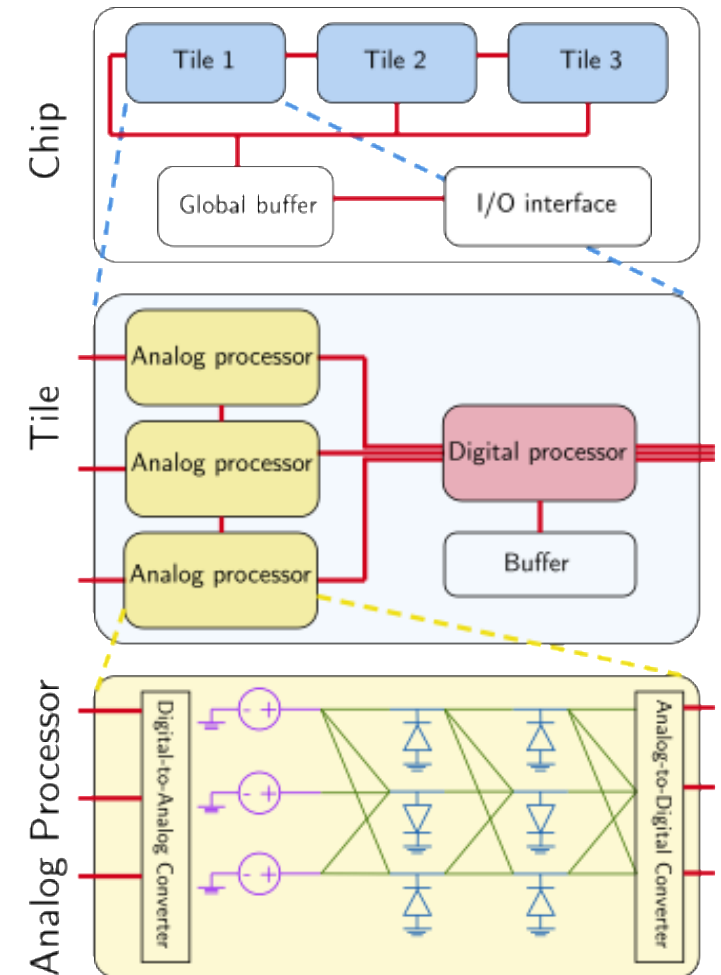
- Need for a new building block to model such systems
- Need for an associated algorithm to compute gradients end-to-end



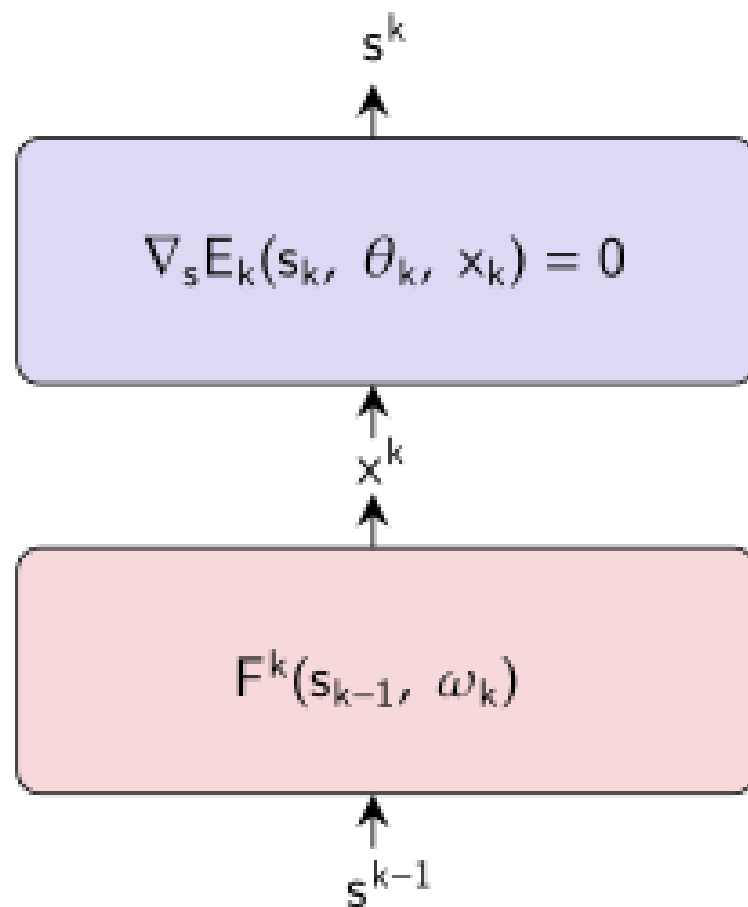
Problem

Analog at scale requires digital circuitry [1]

- Need for a new building block to model such systems
 - ff-EBMs
- Need for an associated algorithm to compute gradients end-to-end
 - EP-BP gradient chaining



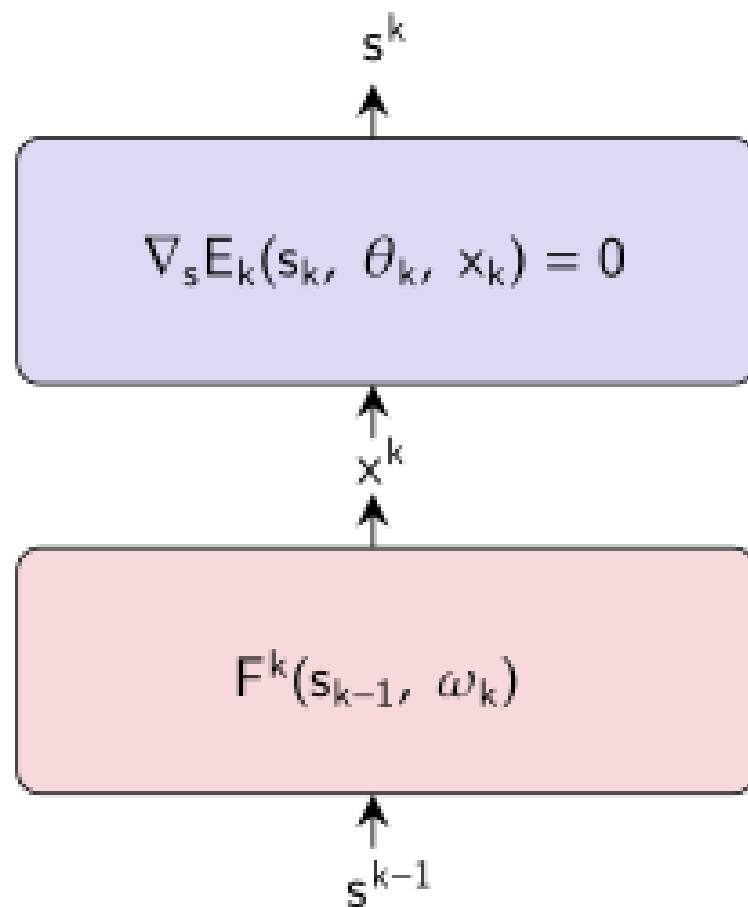
Feedforward-tied EBMs (ff-EBMs)



$\triangleq$ analog parts = EB block

$\triangleq$ digital parts = Feedforward (ff) block

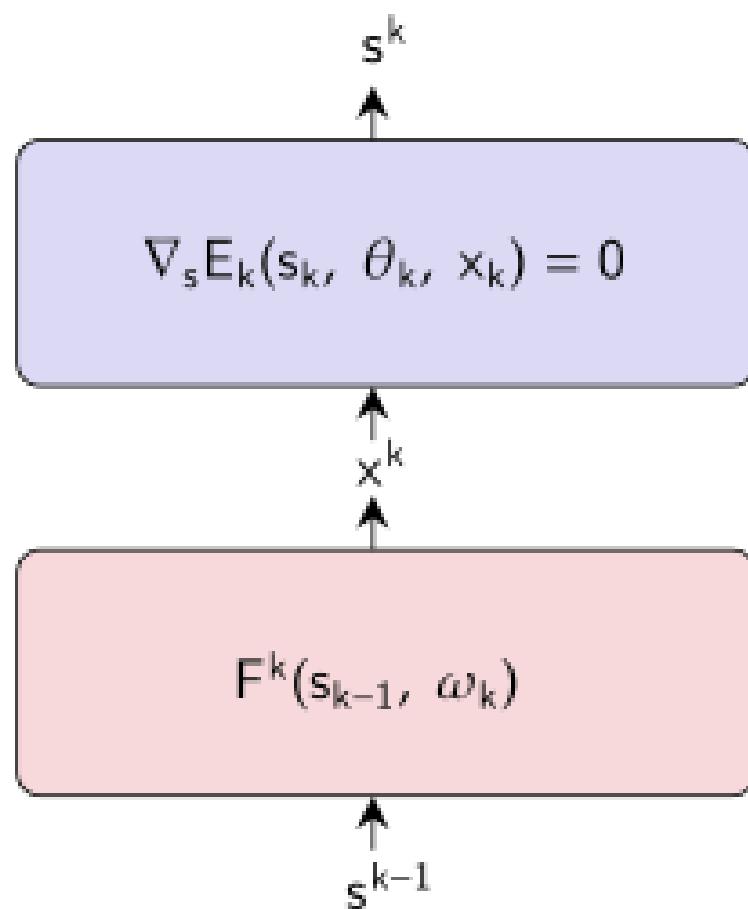
Feedforward-tied EBMs (ff-EBMs)



Algorithm 1 ff-EBM inference

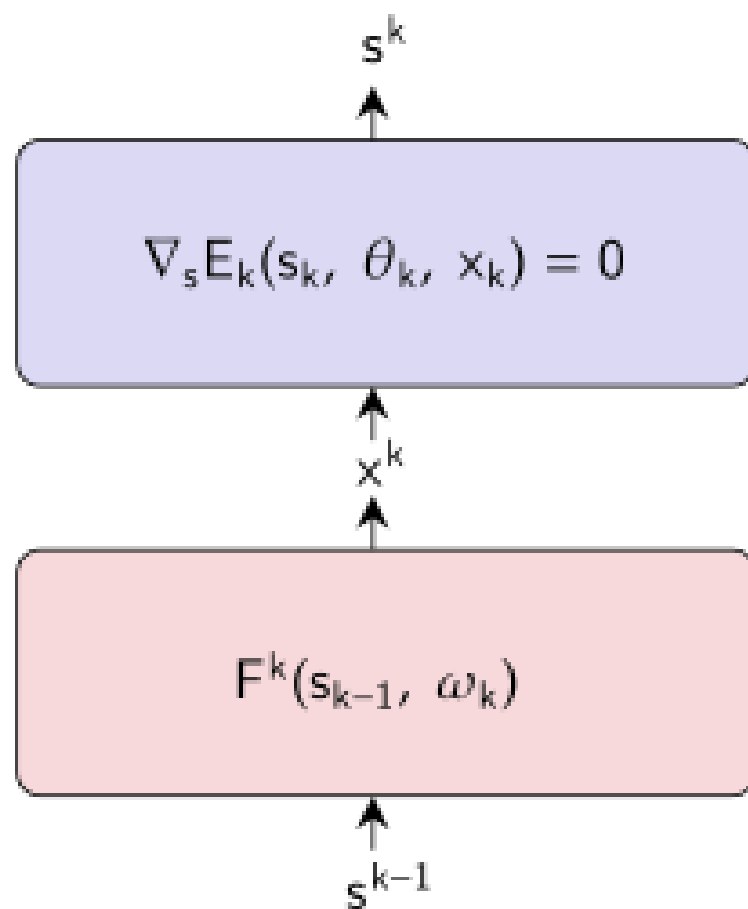
- 1: $s \leftarrow x$
 - 2: **for** $k = 1 \dots N - 1$ **do**
 - 3: $x \leftarrow F^k(s, \omega^k)$
 - 4: $s \leftarrow \underset{s}{\text{Optim}} [E^k(s, \theta^k, x)]$
 - 5: **end for**
 - 6: $\hat{o} \leftarrow F^N(s, \omega^N)$
-

BP-EP gradient chaining

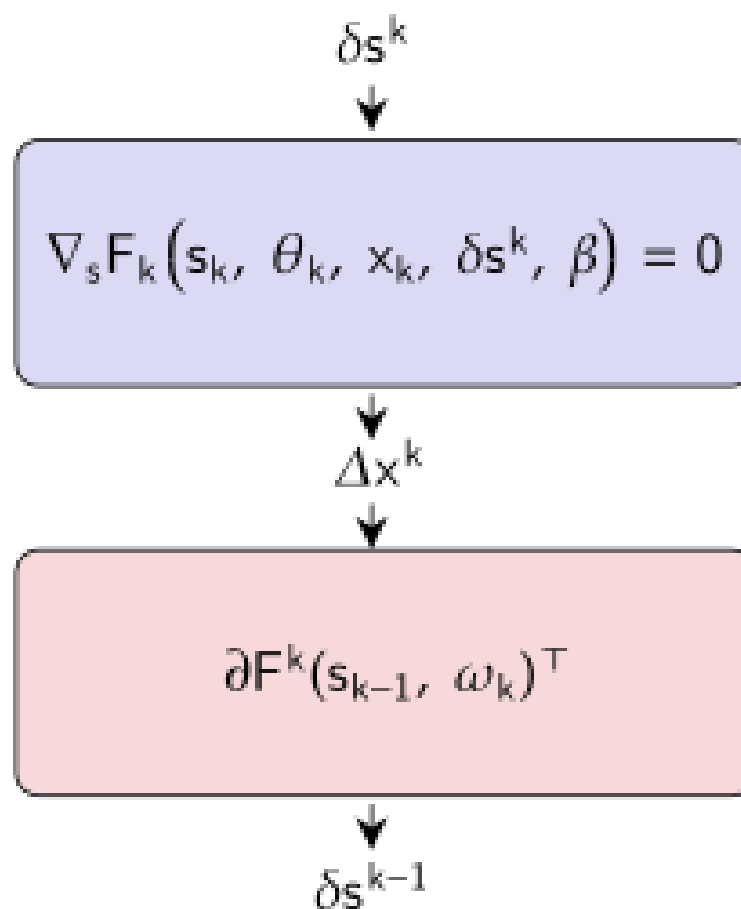


ff-EBM inference

BP-EP gradient chaining



ff-EBM inference

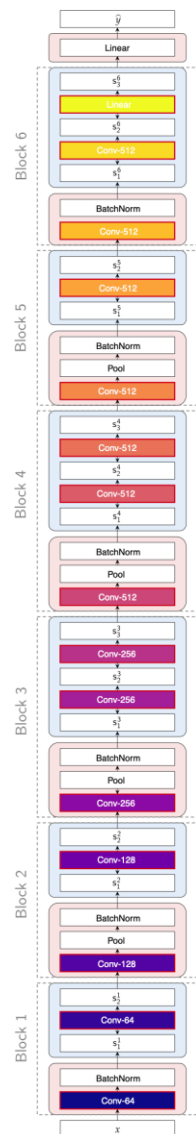


ff-EBM gradient computation (Theorem 3.1)

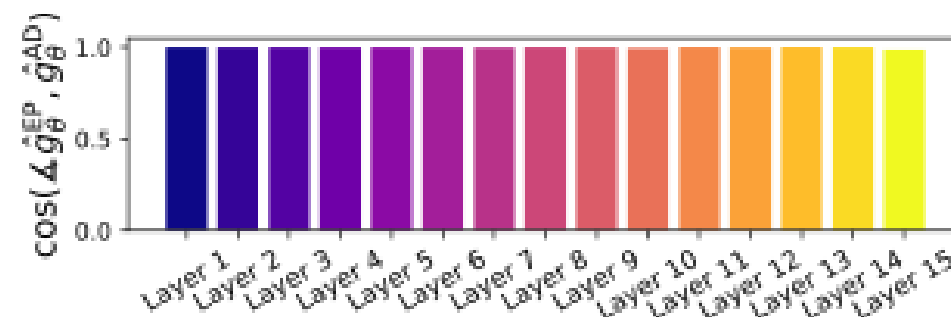
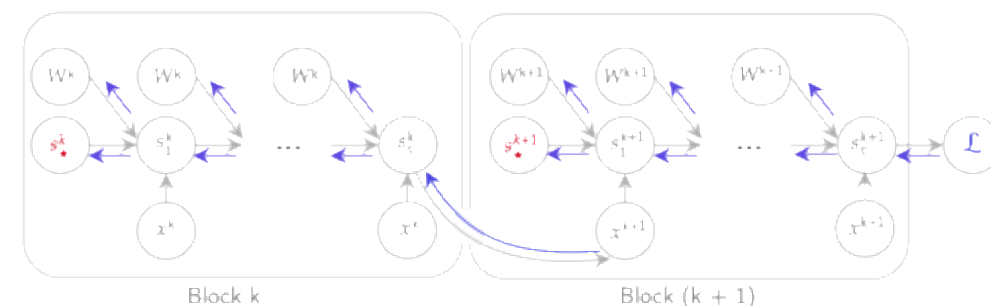
EP through EB blocks...

...BP through ff blocks

Static gradient analysis

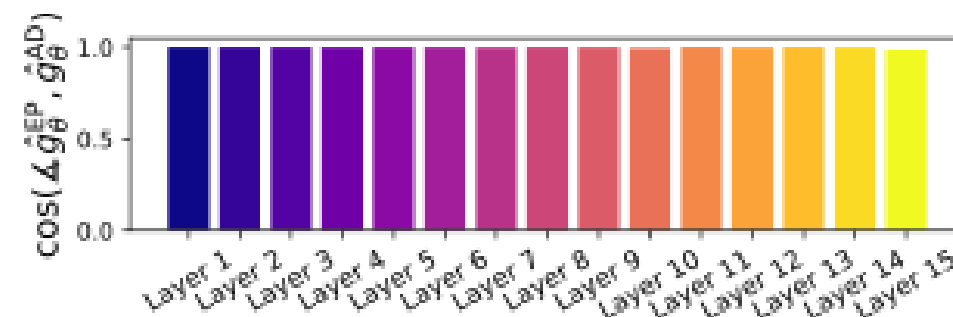
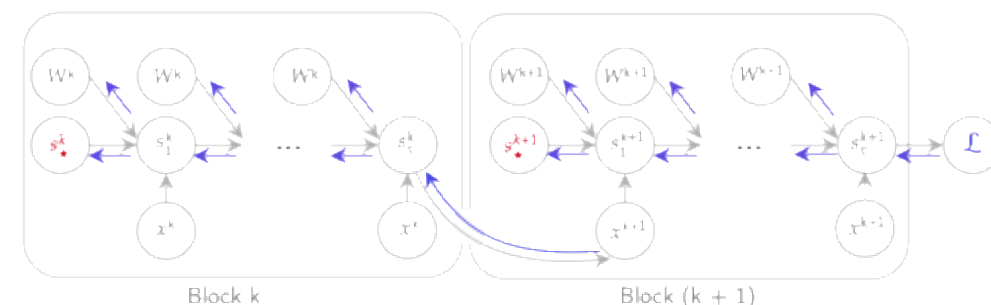
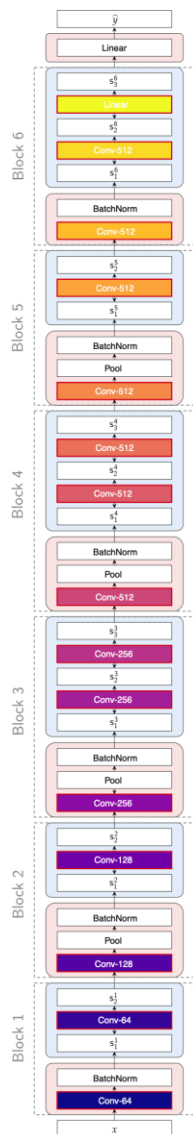


- **Architecture :**
15 layers in total, 6 EB blocks and 6 ff blocks with heterogenous block sizes.
- **Algorithmic baseline:**
end-to-end automatic differentiation (AD) through equilibrium computation
- **Experiment:**
pick random (x, y) and compare BP-EP chaining gradients to AD gradients



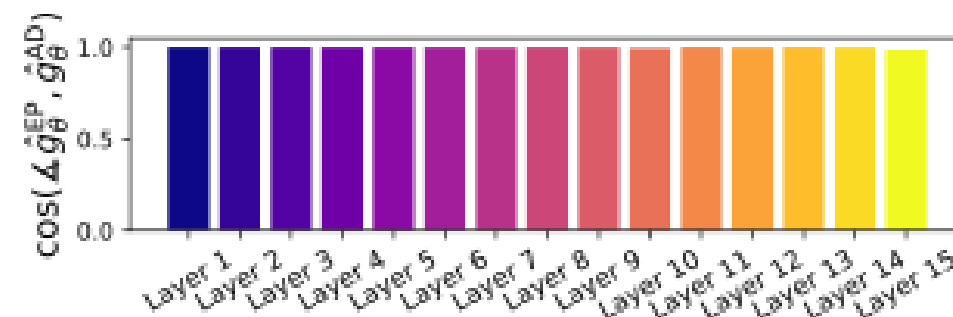
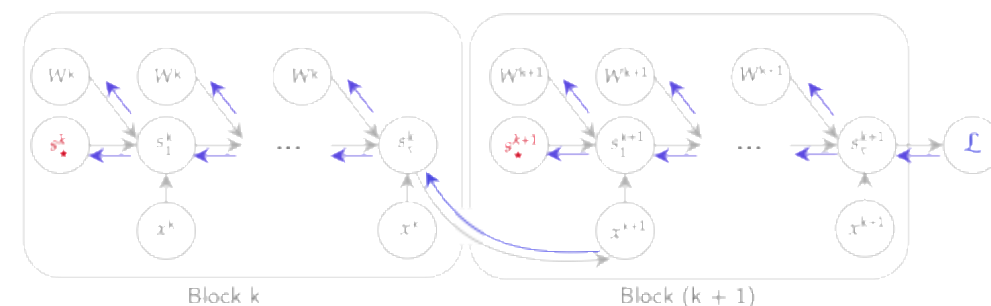
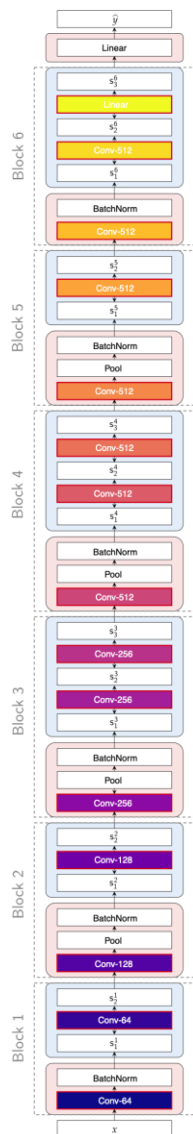
Static gradient analysis

- **Architecture :**
15 layers in total, 6 EB blocks and 6 ff blocks with heterogenous block sizes.
- **Algorithmic baseline:**
end-to-end automatic differentiation (AD) through equilibrium computation
- **Experiment:**
pick random (x, y) and compare BP-EP chaining gradients to AD gradients

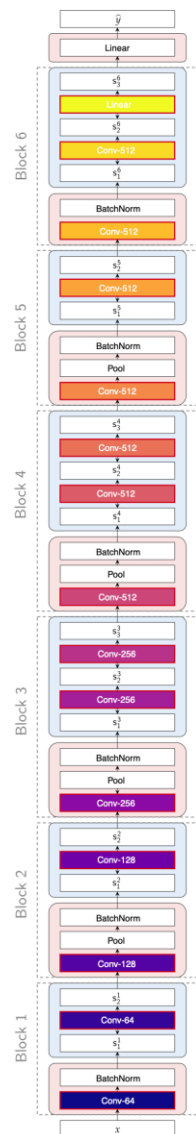


Static gradient analysis

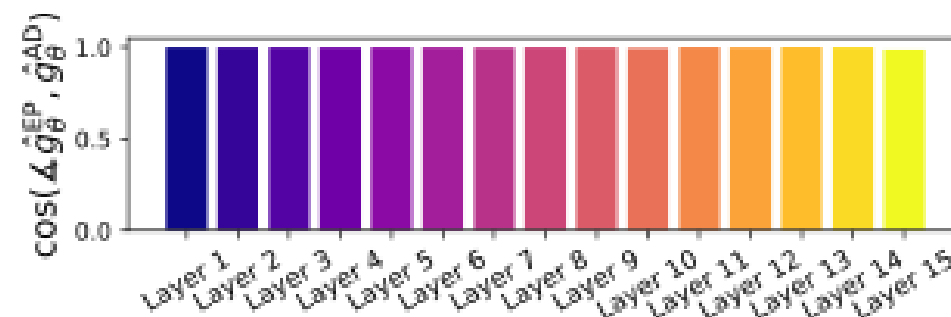
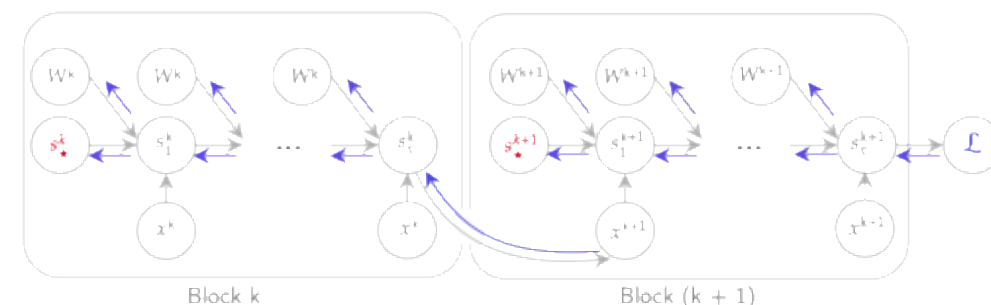
- Architecture :
15 layers in total, 6 EB blocks and 6 ff blocks with heterogenous block sizes.
- Algorithmic baseline:
end-to-end automatic differentiation (AD) through equilibrium computation
- Experiment:
pick random (x, y) and compare BP-EP chaining gradients to AD gradients



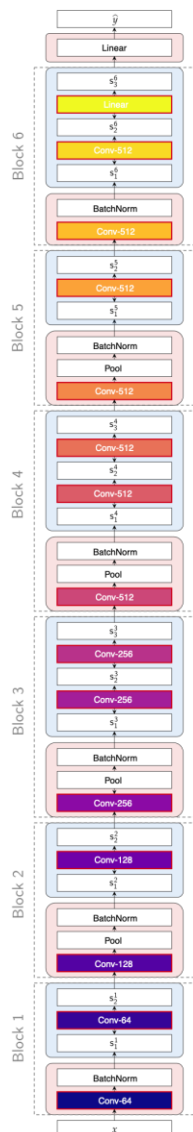
Static gradient analysis



- Architecture :
15 layers in total, 6 EB blocks and 6 ff blocks with heterogenous block sizes.
- Algorithmic baseline:
end-to-end automatic differentiation (AD) through equilibrium computation
- Experiment:
pick random (x, y) and compare BP-EP chaining gradients to AD gradients

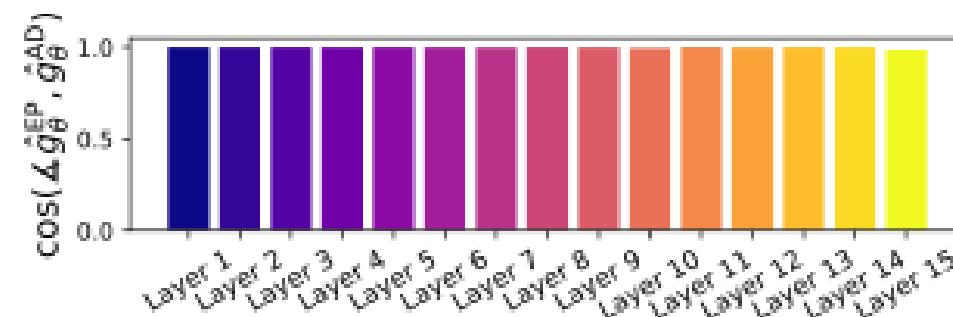
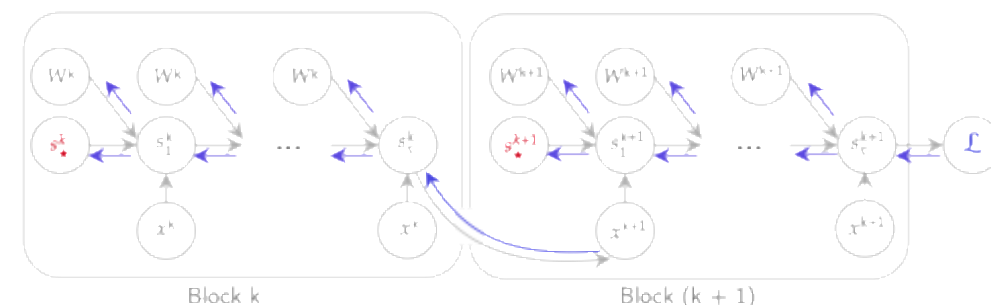


Static gradient analysis



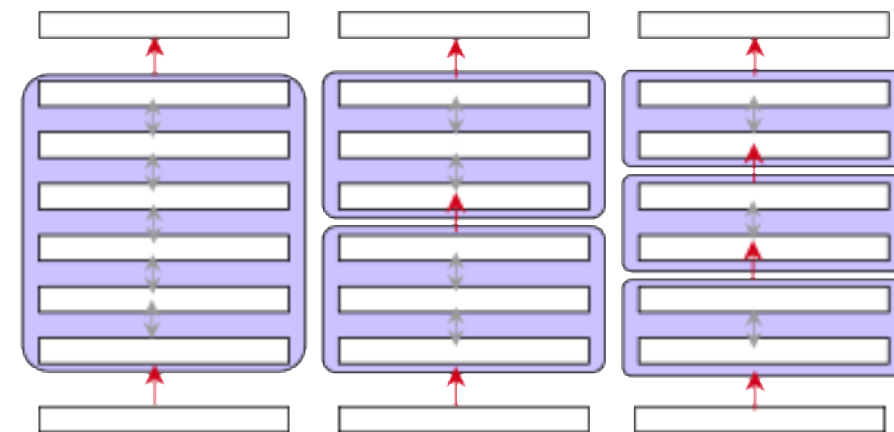
- Architecture :
15 layers in total, 6 EB blocks and 6 ff blocks with heterogenous block sizes.
- Algorithmic baseline:
end-to-end automatic differentiation (AD) through equilibrium computation
- Experiment:
pick random (x, y) and compare BP-EP chaining gradients to AD gradients

→ Near-perfect alignment



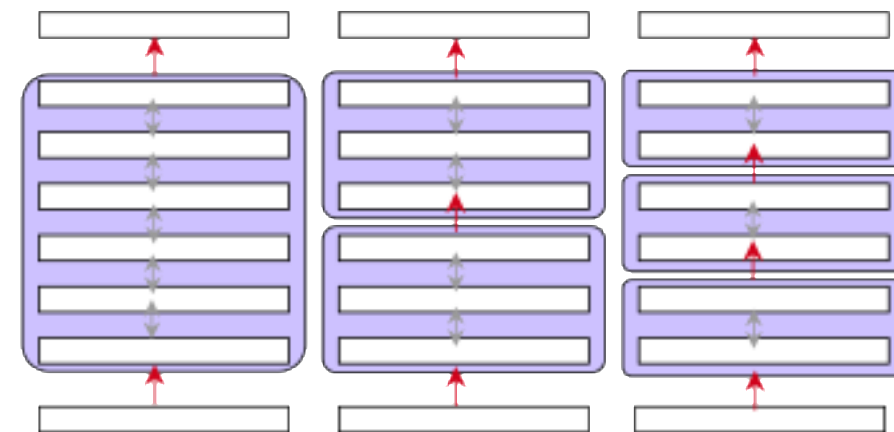
Splitting experiment

- **Models :**
various EB block sizes with *fixed* depth ($L=6$ or 12)
- **Setup:**
CIFAR-10 training experiments with our algorithm
and end-to-end AD
- **Results:**



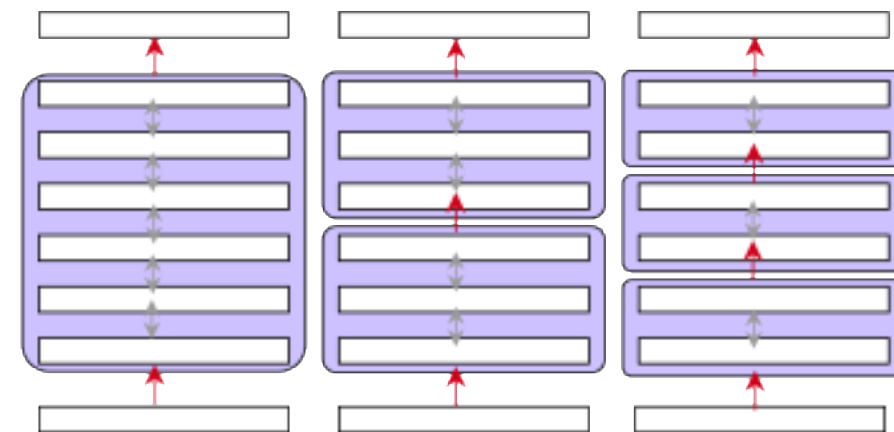
Splitting experiment

- **Models :**
various EB block sizes with *fixed* depth ($L=6$ or 12)
- **Setup:**
CIFAR-10 training experiments with our algorithm
and end-to-end AD
- **Results:**



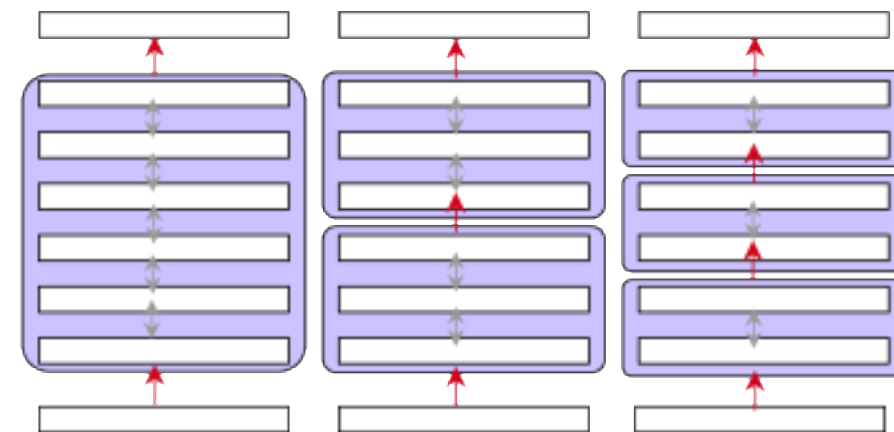
Splitting experiment

- Models :
various EB block sizes with *fixed* depth (L= 6 or 12)
- Setup:
CIFAR-10 training experiments with our algorithm
and end-to-end AD
- Results:



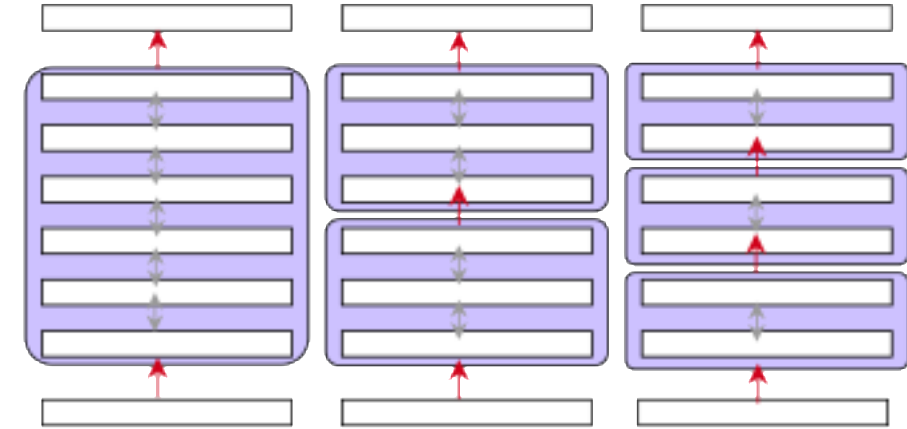
Splitting experiment

- Models :
various EB block sizes with *fixed* depth ($L=6$ or 12)
- Setup:
CIFAR-10 training experiments with our algorithm
and end-to-end AD
- Results:



Splitting experiment

- Models :
various EB block sizes with *fixed* depth (L= 6 or 12)
- Setup:
CIFAR-10 training experiments with our algorithm
and end-to-end AD
- Results:
 - For a given depth, performance is maintained across all splits
 - Our algorithm is on par with end-to-end AD on all models
 - For a given depth, simulating ff-EBMs with smaller block sizes results
in 4x speed up



Scaling experiment



- **Models:**
ff-EBM with EB blocks of size 2, with up to 15 layers in total
- **Setup:**
ImageNet32 and CIFAR100 training experiments with our algorithm and end-to-end AD
- **Results:**

Scaling experiment

- **Models:**
ff-EBM with EB blocks of size 2, with up to 15 layers in total
- **Setup:**
ImageNet32 and CIFAR100 training experiments with our algorithm and end-to-end AD
- **Results:**

Scaling experiment



- Models:
ff-EBM with EB blocks of size 2, with up to 15 layers in total
- Setup:
ImageNet32 and CIFAR100 training experiments with our algorithm and end-to-end AD
- Results:

Scaling experiment



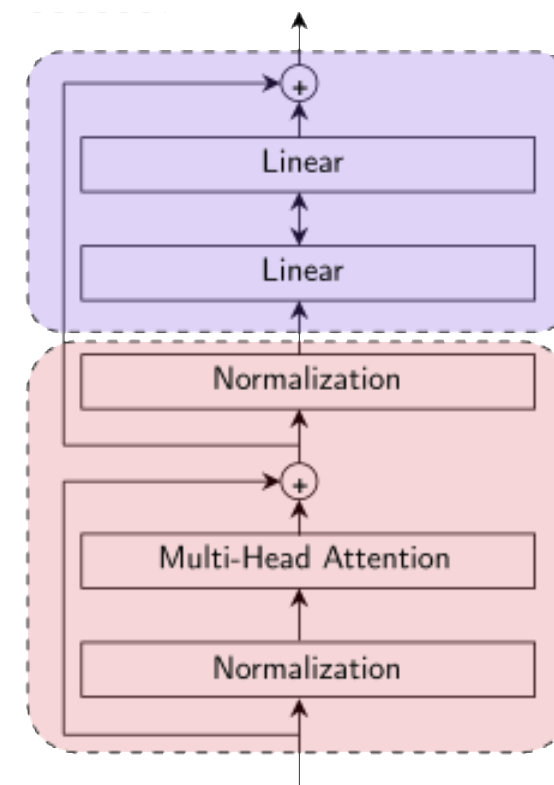
- Models:
ff-EBM with EB blocks of size 2, with up to 15 layers in total
- Setup:
ImageNet32 and CIFAR100 training experiments with our algorithm and end-to-end AD
- Results:

Scaling experiment

- Models:
ff-EBM with EB blocks of size 2, with up to 15 layers in total
- Setup:
ImageNet32 and CIFAR100 training experiments with our algorithm and end-to-end AD
- Results:
 - New EP SOTA on CIFAR100 (~71.2 % top1 val)
 - New EP SOTA on ImageNet32 (~46 % top1 val)
 - Our algorithm still on par with end-to-end AD on all models

Conclusion

- Our work enables the gradual integration of analog (energy-based) parts into existing digital accelerators
- Also promising to scale up EP simulations to deeper architectures
- Possible extensions of our work:
 - more hardware realistic simulations
 - ff-EBM counterparts of transformers



See you in Vancouver! :)