Towards Scaling Difference Target Propagation by Learning Backprop Targets

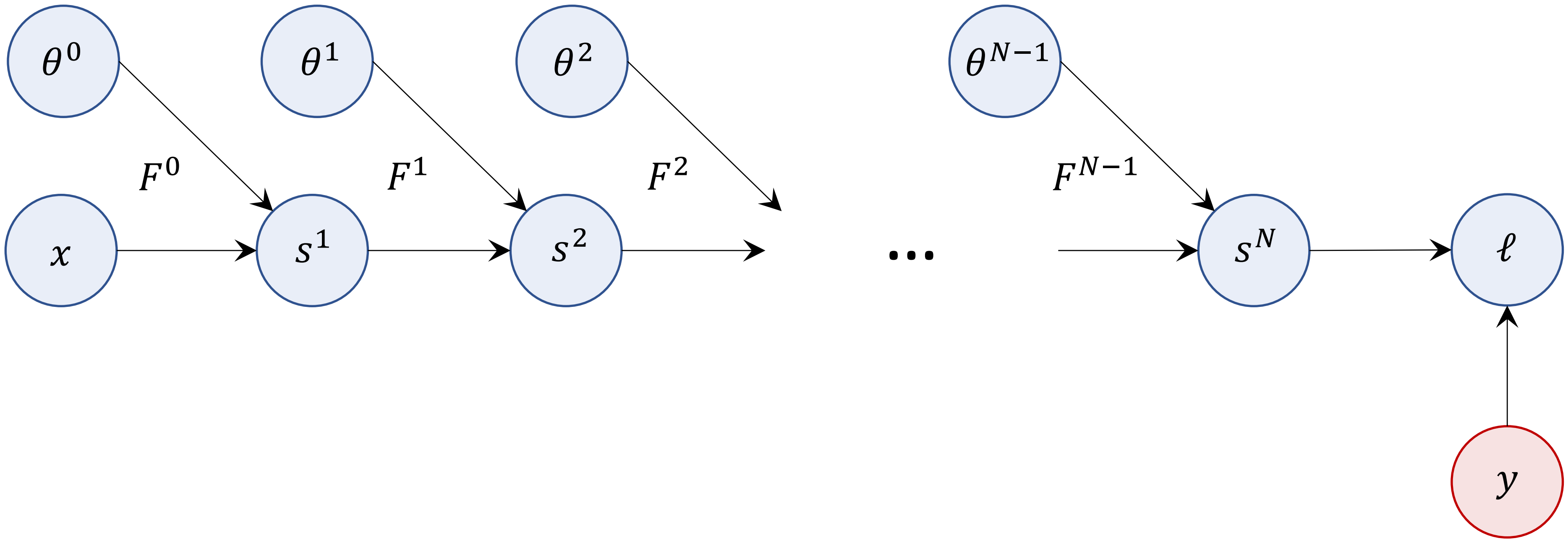



Maxence Ernoult, Fabrice Normandin*, Abhinav Moudgil*, Sean Spinney,

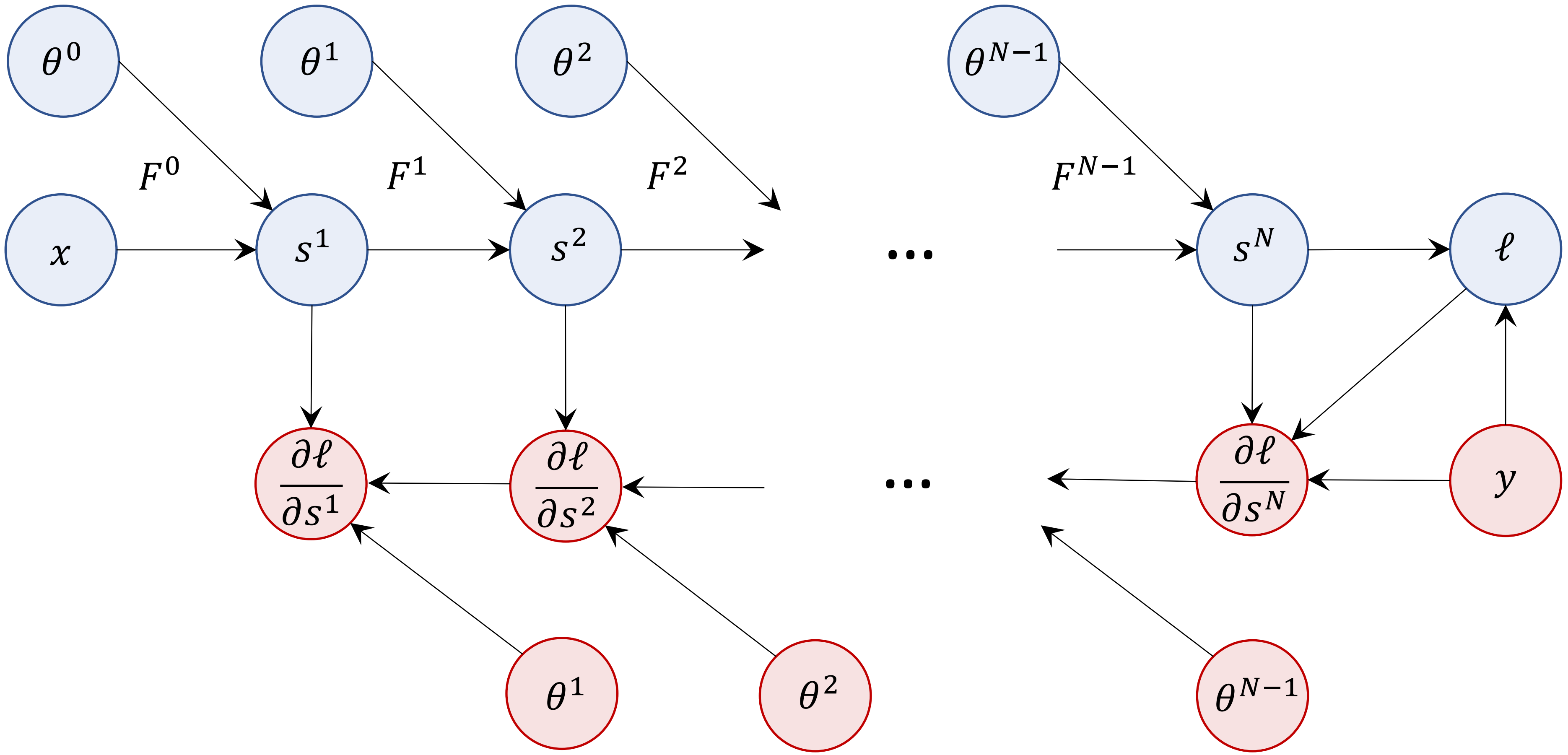



Eugene Belilovsky, Irina Rish, Blake Richards, Yoshua Bengio

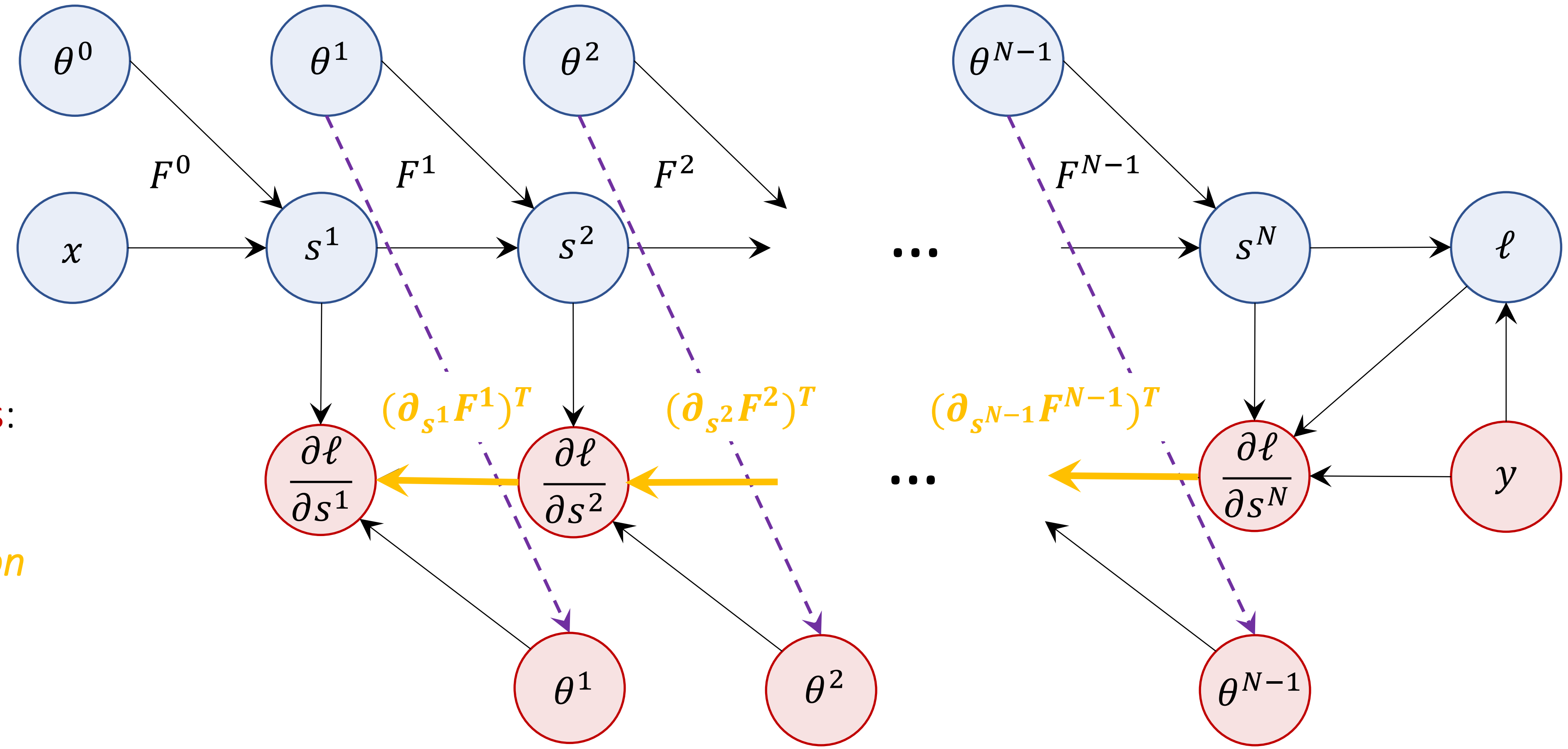
Motivation: biologically plausible implementation of backprop



Motivation: biologically plausible implementation of backprop



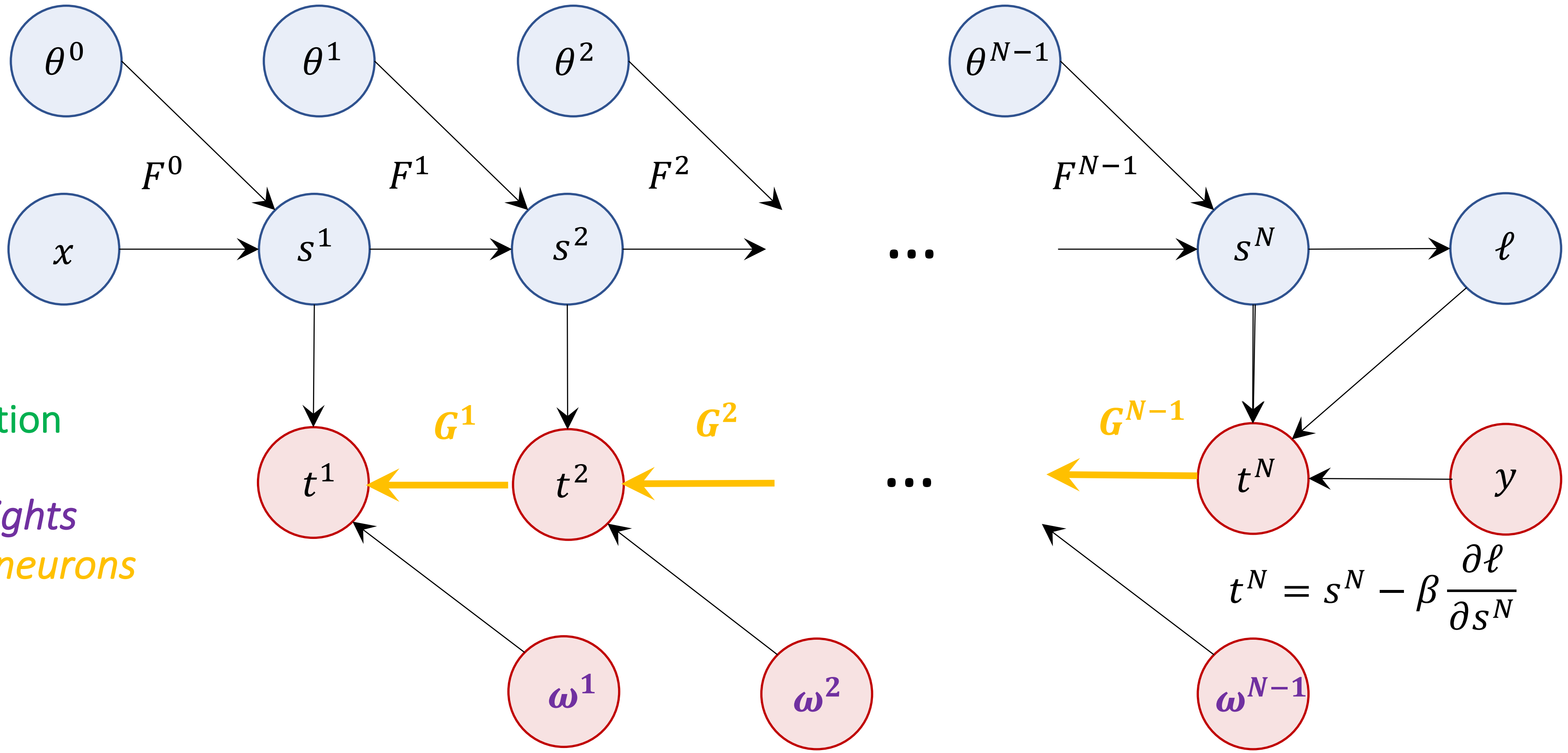
Motivation: biologically plausible implementation of backprop



Backprop has two issues:

- *Weight transport*
- *Jacobian computation*

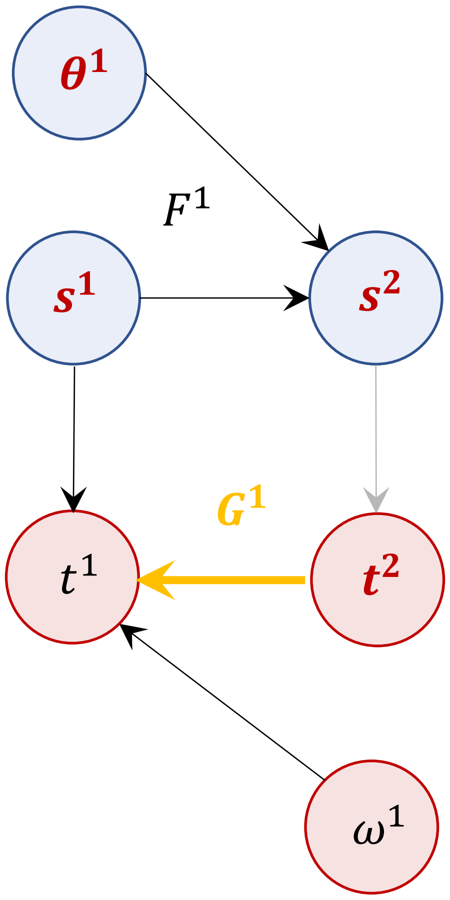
Motivation: biologically plausible implementation of backprop



Difference Target Propagation (Lee et al, 2015):

- *Distinct feedback weights*
- *Compute targets for neurons*

Difference Target Propagation (DTP)

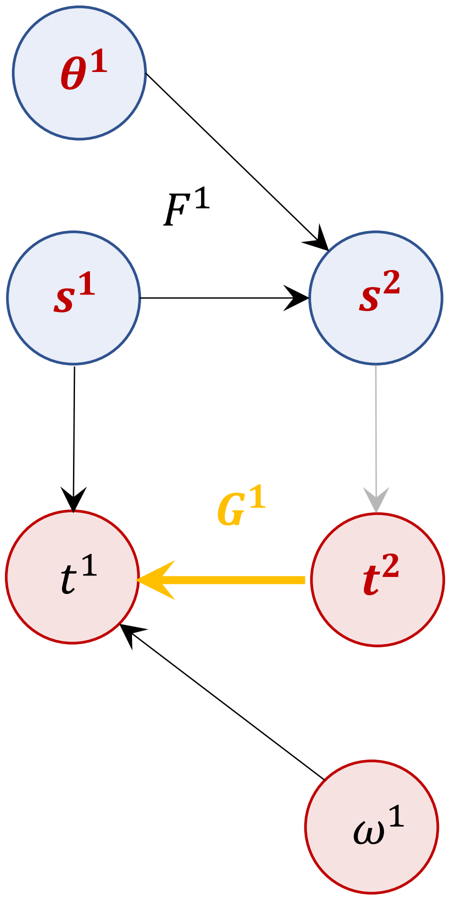


Difference Target Propagation (DTP)

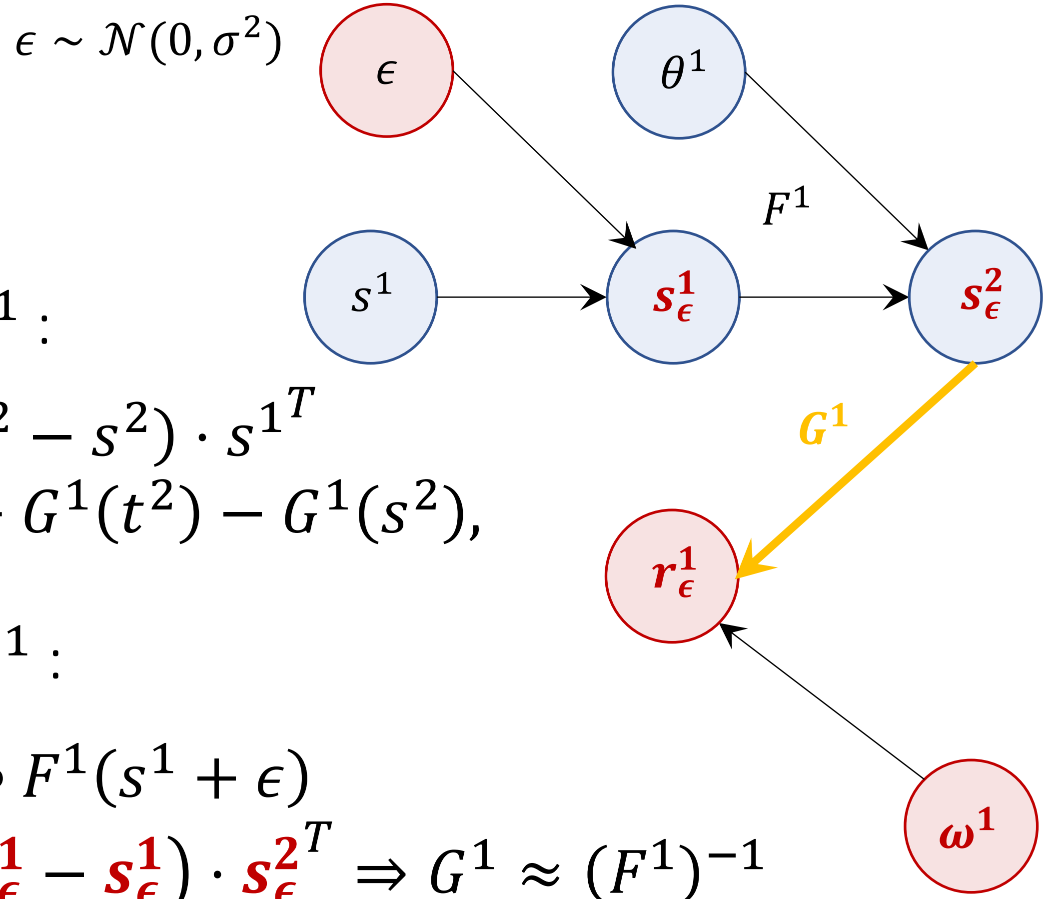
Training θ^1 :

$$\Delta\boldsymbol{\theta}^{\mathbf{1}} \propto \left(\boldsymbol{t}^{\mathbf{2}} - \boldsymbol{s}^{\mathbf{2}}\right) \cdot {\boldsymbol{s}^{\mathbf{1}}}^{T}$$

$$t^1 = s^1 + G^1(t^2) - G^1(s^2),$$



Difference Target Propagation (DTP)



Training θ^1 :

$$\Delta\theta^1 \propto (t^2 - s^2) \cdot {s^1}^T$$
$$t^1 = s^1 + G^1(t^2) - G^1(s^2),$$

Training ω^1 :

$$r_\epsilon^1 = G^1 \circ F^1(s^1 + \epsilon)$$
$$\Delta\boldsymbol{\omega}^1 \propto \left(\boldsymbol{r}_{\epsilon}^{1} - \boldsymbol{s}_{\epsilon}^{1}\right) \cdot {\boldsymbol{s}_{\epsilon}^{2}}^T \Rightarrow G^1 \approx (F^1)^{-1}$$

Emulating backprop by DTP

DTP ≈ Gauss-Newton Optimization

(Meulemans et al 2020, Bengio 2020)

Training θ^n :

$$\Delta\theta^n \propto \partial_{\theta^n} F^{n^T} \cdot (t^{n+1} - s^{n+1})$$

$$t^n - s^n \approx \partial_{s^{n+1}} G^n \cdot (t^{n+1} - s^{n+1})$$

Training ω^n :

$$G^n \approx (F^n)^{-1}$$

Emulating backprop by DTP

DTP ≈ Gauss-Newton Optimization
(Meulemans et al 2020, Bengio 2020)

Training θ^n :

$$\Delta\theta^n \propto \partial_{\theta^n} F^{n^T} \cdot (t^{n+1} - s^{n+1})$$

$$t^n - s^n \approx \partial_{s^{n+1}} G^n \cdot (t^{n+1} - s^{n+1})$$

Training ω^n :

$$G^n \approx (F^n)^{-1}$$

Our work: DTP ≈ Backprop

$$\Delta\theta^n \propto \partial_{\theta^n} F^{n^T} \cdot (t^{n+1} - s^{n+1})$$

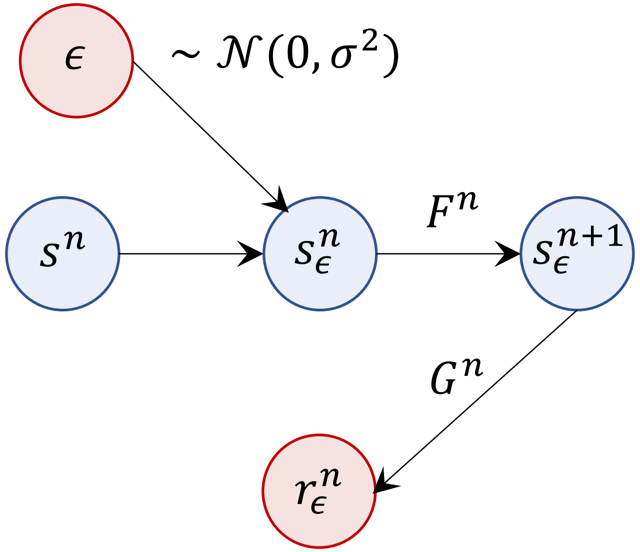
$$t^n - s^n \approx \boldsymbol{(\partial_{s^n} F^n)^T} \cdot (t^{n+1} - s^{n+1})$$

Training $\boldsymbol{\omega^n}$:

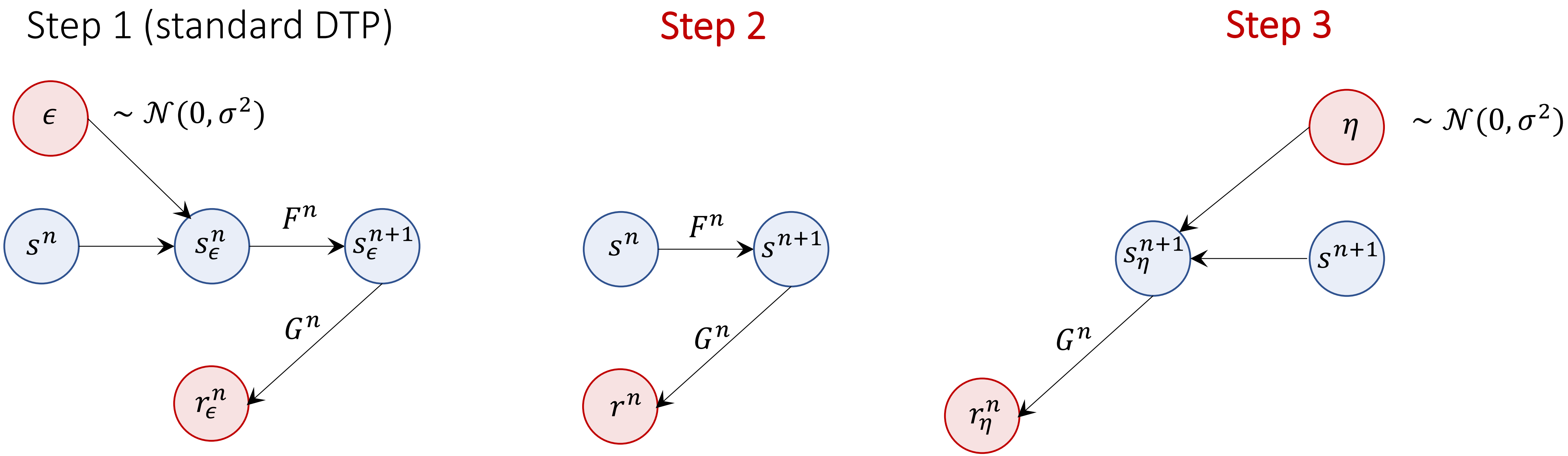
$$\boldsymbol{\partial_{s^{n+1}} G^n \approx (\partial_{s^n} F^n)^T}$$

Our new algorithm to train ω^n

Step 1 (standard DTP)



Our new algorithm to train ω^n

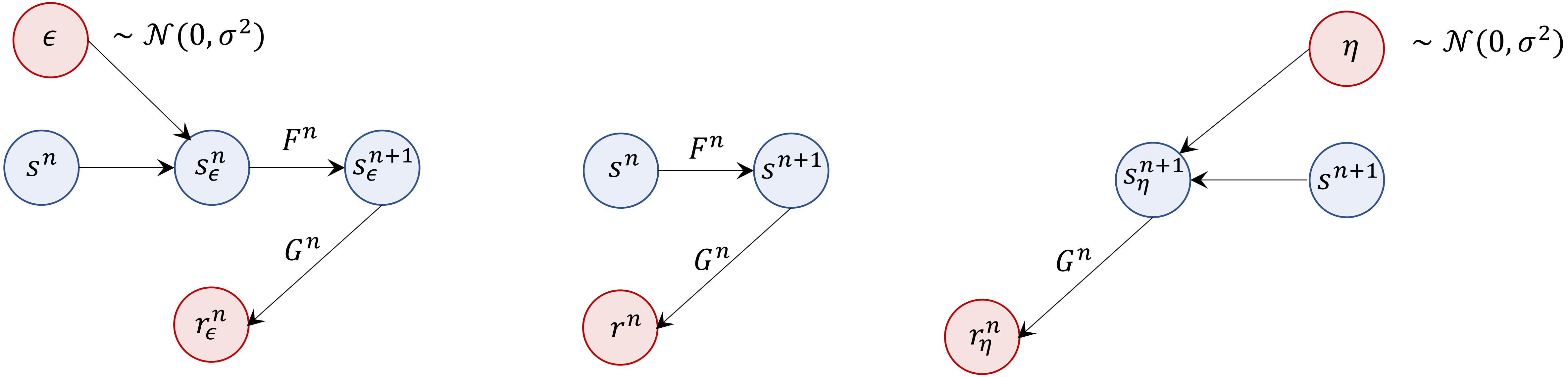


Our new algorithm to train ω^n

Step 1 (standard DTP)

Step 2

Step 3

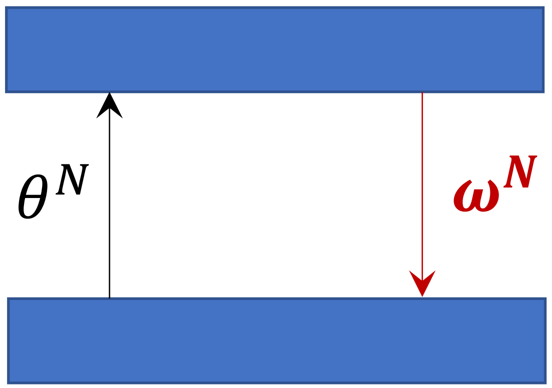


Step 4

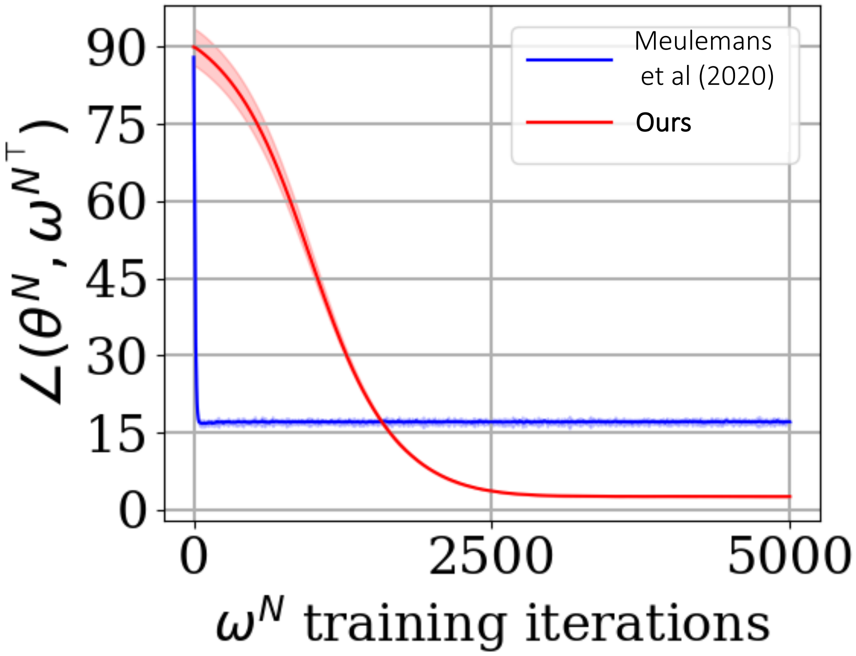
Update ω^n with $\mathcal{L}_\omega^n = -\frac{1}{\sigma^2}\epsilon^\top \cdot (r_\epsilon^n - r^n) + \frac{1}{2\sigma^2}\left\|r_\eta^n - r^n\right\|^2$

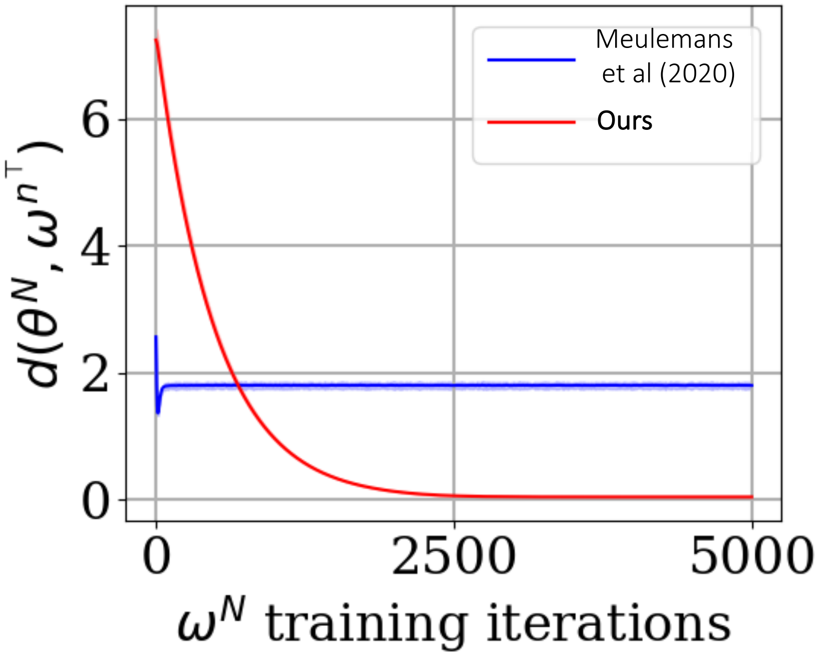
$$\approx \frac{1}{2}\left\|\partial_{s^{n+1}} G^n - \left(\partial_{s^n} F^n\right)^T\right\|^2$$

Testing the feedback training algorithm



FC layer
(frozen feeforward weights)





Training experiments

LeNet (4 layers)	MNIST	F-MNIST	CIFAR-10
Meulemans et al (2020)	98.6	88.9	76.3
Ours	98.9	90.3	85.3
BP	98.9	91.4	86.3

VGG (6 layers)	CIFAR-10	ImageNet 32x32 (Top 5)
Ours	89.4	60.6
BP	89.0	61.3

Conclusion

- Better to learn backprop targets than Gauss-Newton targets
- Repo:
  https://github.com/ernoult/scalingDTP

- *(New)* Distributed implementation parallelizing feedback weight training:
  https://github.com/amoudgl/distributed-dtp