Two Routes to Scalable Credit Assignment without Weight Symmetry

Daniel Kunin*, Aran Nayebi*, Javier Sagastuy-Brena*,
Surya Ganguli, Jon Bloom, Daniel L.K. Yamins

ICML 2020

Stanford
Neuro AI Lab
Stanford Neuroscience and Artificial Intelligence Lab

Agenda

- Motivation
  - The problems with backpropagation
  - Breaking weight symmetry: some previous proposals
- Regularization-inspired framework
  - Evaluation criteria
- Results
  - Local Learning Rules
  - Non-local Learning Rules
- Conclusion

The problems with backpropagation

The problems with backpropagation

- Relies on an error term which implies the existence of a target we are optimizing for.

The problems with backpropagation

- Relies on an error term which implies the existence of a target we are optimizing for.

- Requires the derivatives of the activation functions.

The problems with backpropagation

- Relies on an error term which implies the existence of a target we are optimizing for.

- Requires the derivatives of the activation functions.

- Requires separate forward and backward passes.

The problems with backpropagation

- Relies on an error term which implies the existence of a target we are optimizing for.

- Requires the derivatives of the activation functions.

- Requires separate forward and backward passes.

- The weight updates require access to transposes of the feedforward weights.

The problems with backpropagation

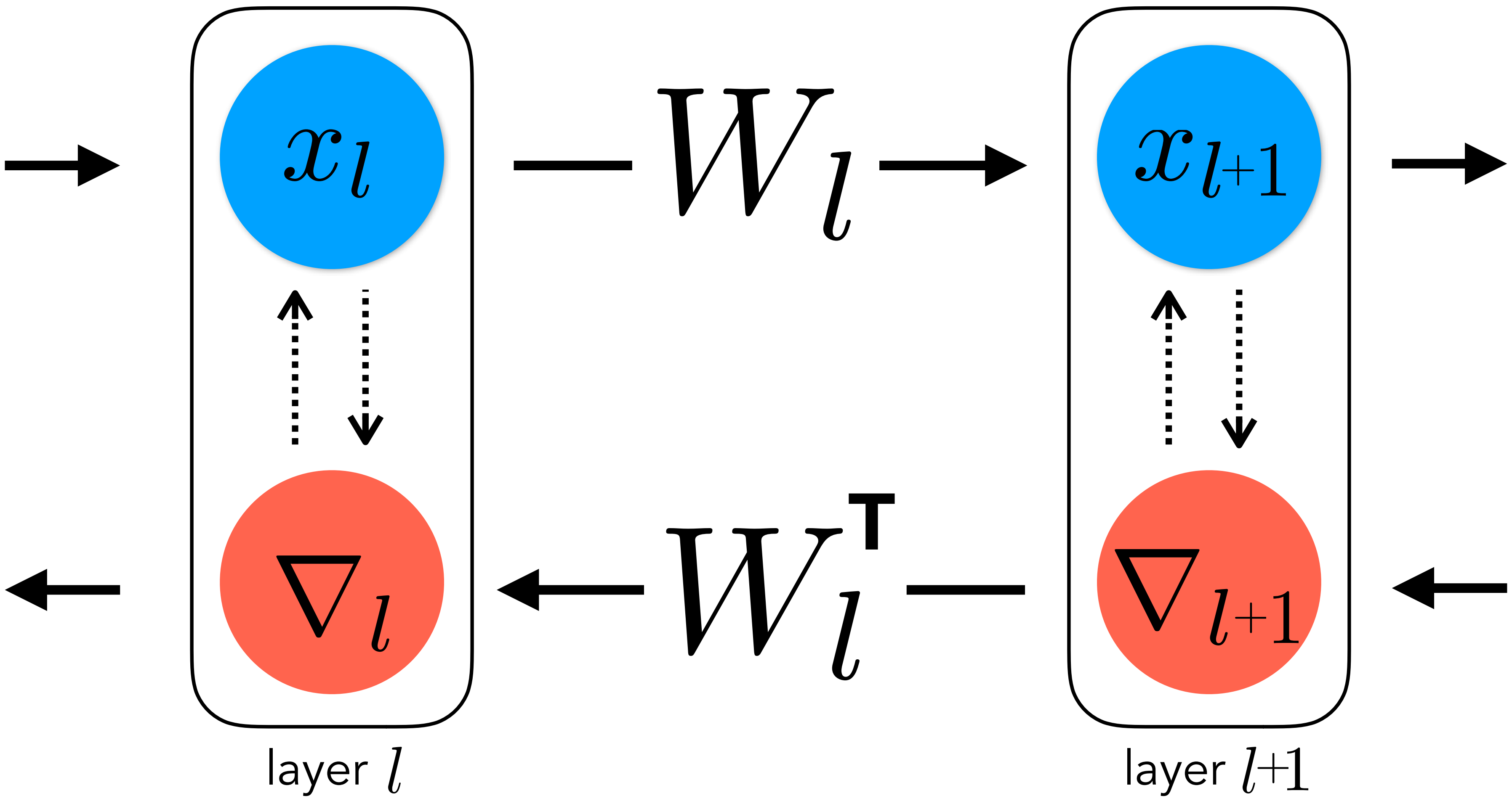
- Relies on an error term which implies the existence of a target we are optimizing for.

- Requires the derivatives of the activation functions.

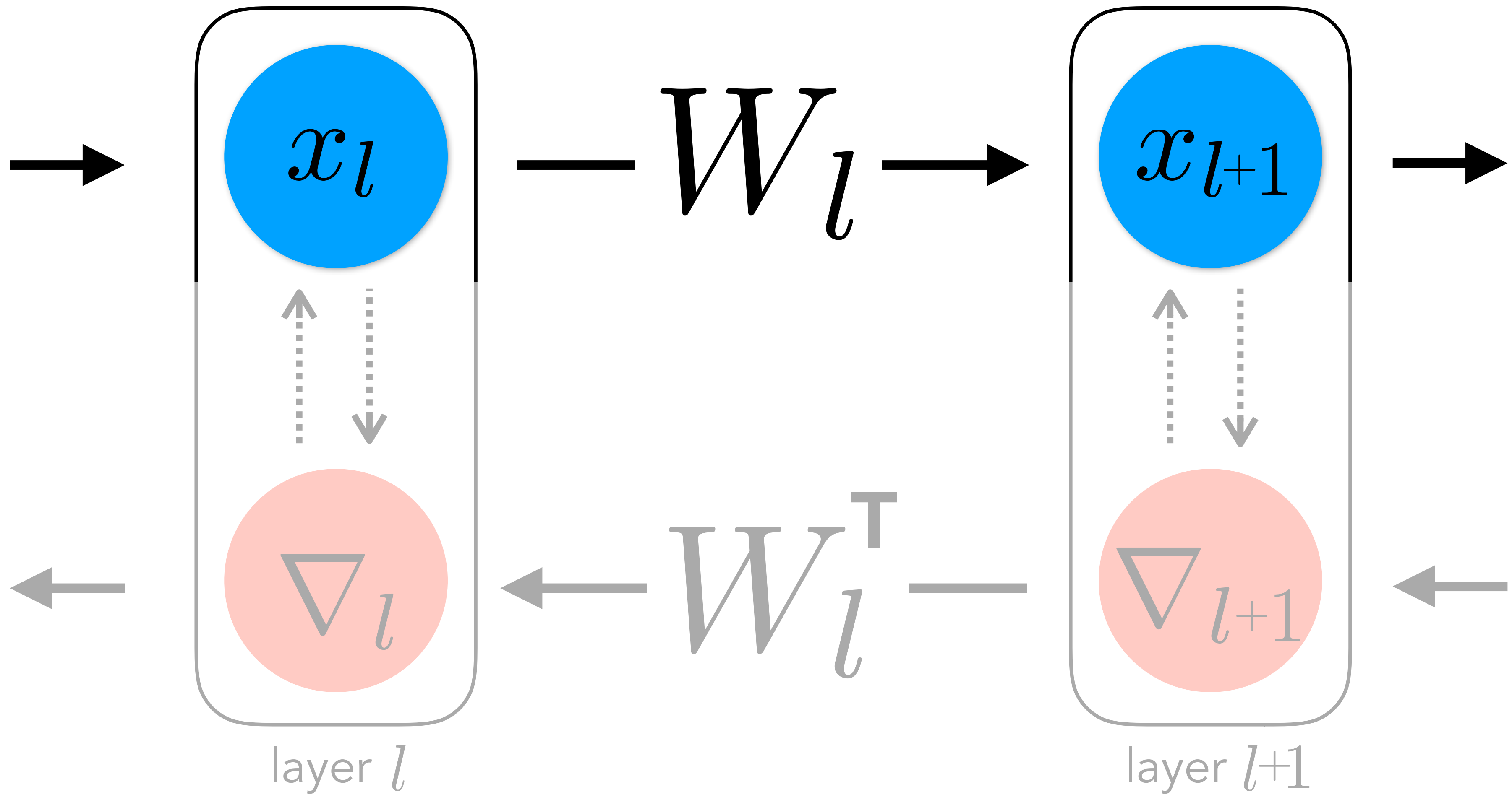
- Requires separate forward and backward passes.

- **The weight updates require access to transposes of the feedforward weights.**

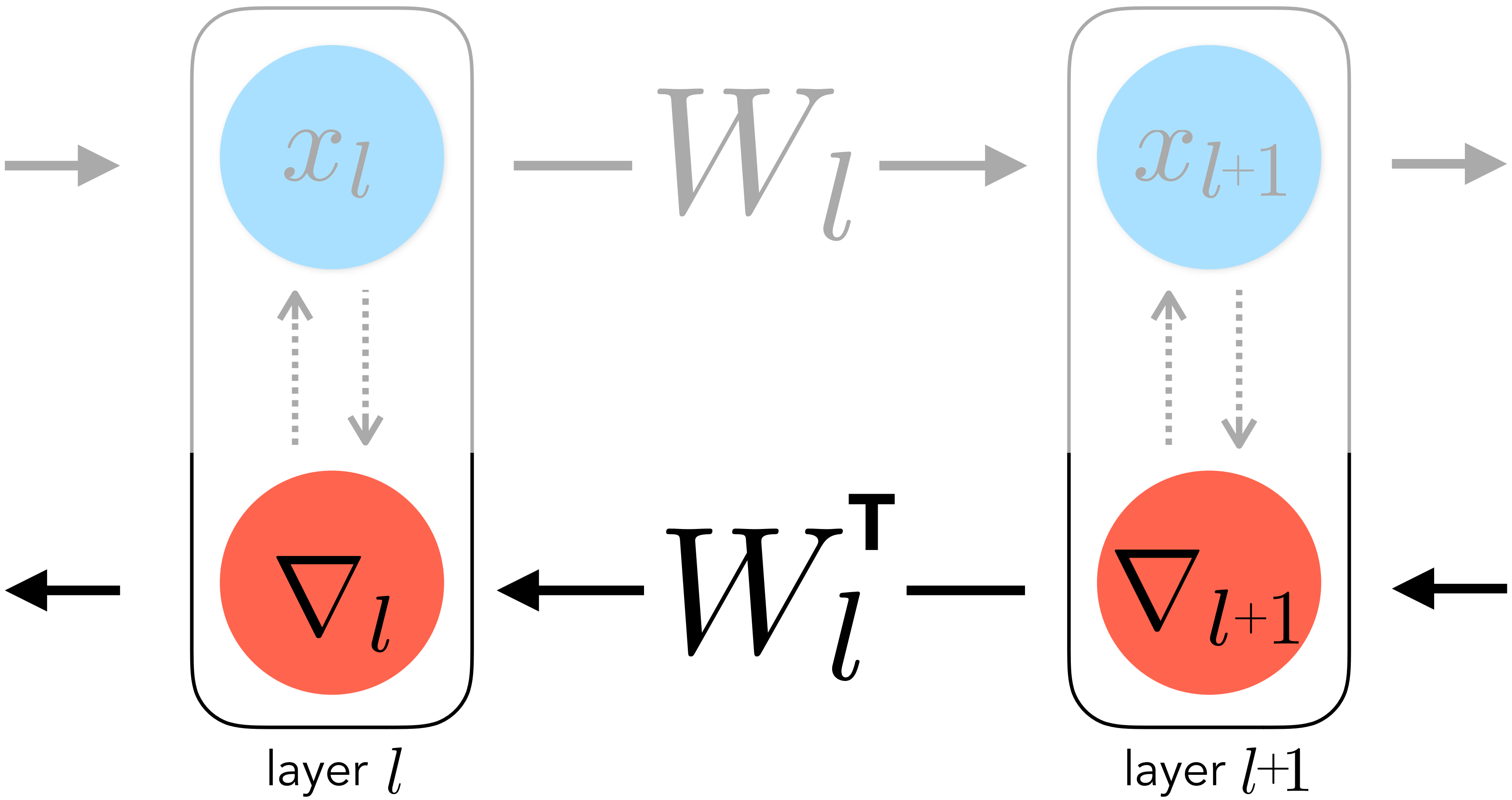
Breaking the backpropagation weight symmetry constraint



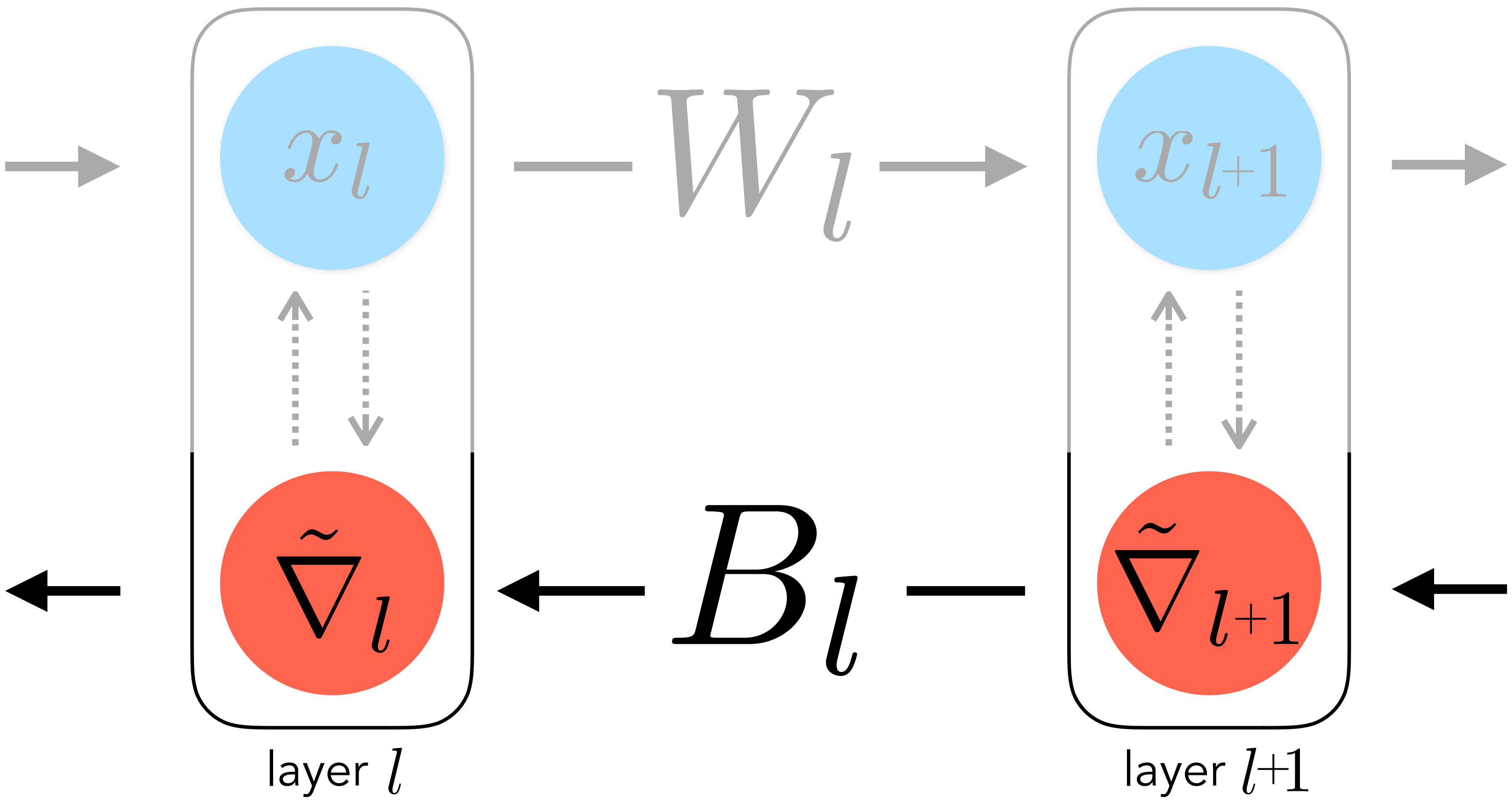
Breaking the backpropagation weight symmetry constraint



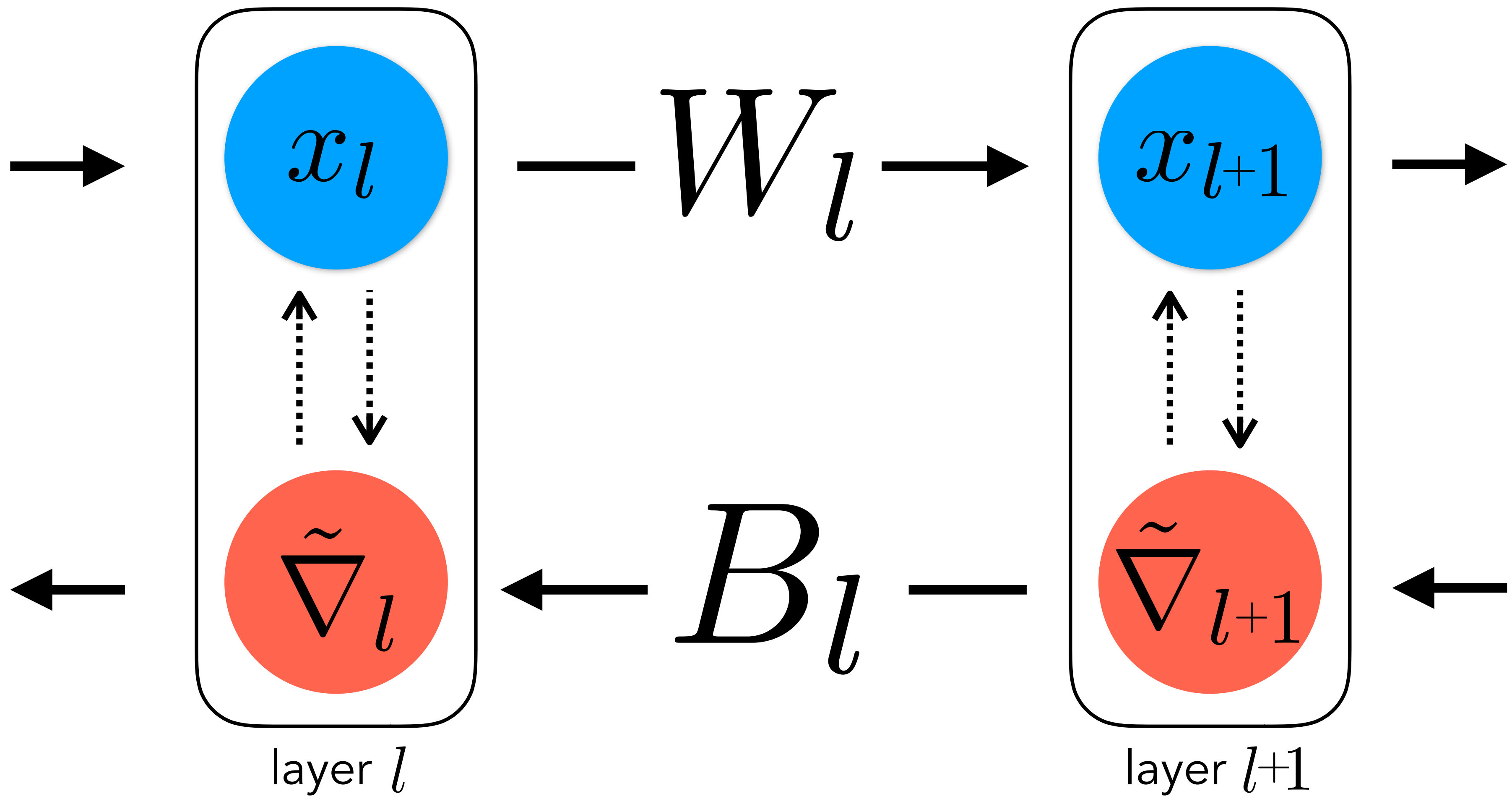
Breaking the backpropagation weight symmetry constraint



Breaking the backpropagation weight symmetry constraint



Breaking the backpropagation weight symmetry constraint



- What should the dynamics on the backward weights be?

Imposing dynamics on the backwards weights

Imposing dynamics on the backwards weights

- Some previous proposals:

Imposing dynamics on the backwards weights

- Some previous proposals:
  - **Feedback Alignment [1]:** no dynamics. B is fixed, random

$$\Delta B_l = 0$$

[1] Lillicrap, Timothy P., et al. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7.1 (2016): 1-10.
[2] Akrout, Mohamed, et al. "Deep learning without weight transport." *Advances in Neural Information Processing Systems*. 2019.
[3] Kolen, John F., and Jordan B. Pollack. "Backpropagation without weight transport." *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 3. IEEE, 1994.

Imposing dynamics on the backwards weights

- Some previous proposals:
  - **Feedback Alignment [1]:** no dynamics. B is fixed, random

$$\Delta B_l = 0$$

  - **Weight Mirror [2]:** feedforward neurons noisily discharge onto the backward path. Use a Hebbian update with this noise and add weight decay.

$$\Delta B_l = \eta x_l x_{l+1}^{\intercal} - \lambda_{\mathrm{WM}} B_l$$

[1] Lillicrap, Timothy P., et al. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7.1 (2016): 1-10.
[2] Akrout, Mohamed, et al. "Deep learning without weight transport." *Advances in Neural Information Processing Systems*. 2019.
[3] Kolen, John F., and Jordan B. Pollack. "Backpropagation without weight transport." *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 3. IEEE, 1994.

Imposing dynamics on the backwards weights

- Some previous proposals:
  - **Feedback Alignment [1]:** no dynamics. B is fixed, random

$$\Delta B_l = 0$$

  - **Weight Mirror [2]:** feedforward neurons noisily discharge onto the backward path. Use a Hebbian update with this noise and add weight decay.

$$\Delta B_l = \eta x_l x_{l+1}^\intercal - \lambda_{\mathrm{WM}} B_l$$

  - **Kolen-Pollack [2,3]:** use the same update on B as you would use on W and add weight decay.

$$\Delta B_l = -\eta x_l \tilde{\nabla}_{l+1}^\intercal - \lambda_{\mathrm{KP}} B_l$$

[1] Lillicrap, Timothy P., et al. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7.1 (2016): 1-10.
[2] Akrout, Mohamed, et al. "Deep learning without weight transport." *Advances in Neural Information Processing Systems*. 2019.
[3] Kolen, John F., and Jordan B. Pollack. "Backpropagation without weight transport." *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 3. IEEE, 1994.

Imposing dynamics on the backwards weights

- Some previous proposals:
  - **Feedback Alignment [1]:** no dynamics. B is fixed, random

$$\Delta B_l = 0$$

  - **Weight Mirror [2]:** feedforward neurons noisily discharge onto the backward path. Use a Hebbian update with this noise and add weight decay.

$$\Delta B_l = \eta x_l x_{l+1}^{\intercal} - \lambda_{\mathrm{WM}} B_l$$

  - **Kolen-Pollack [2,3]:** use the same update on B as you would use on W and add weight decay.

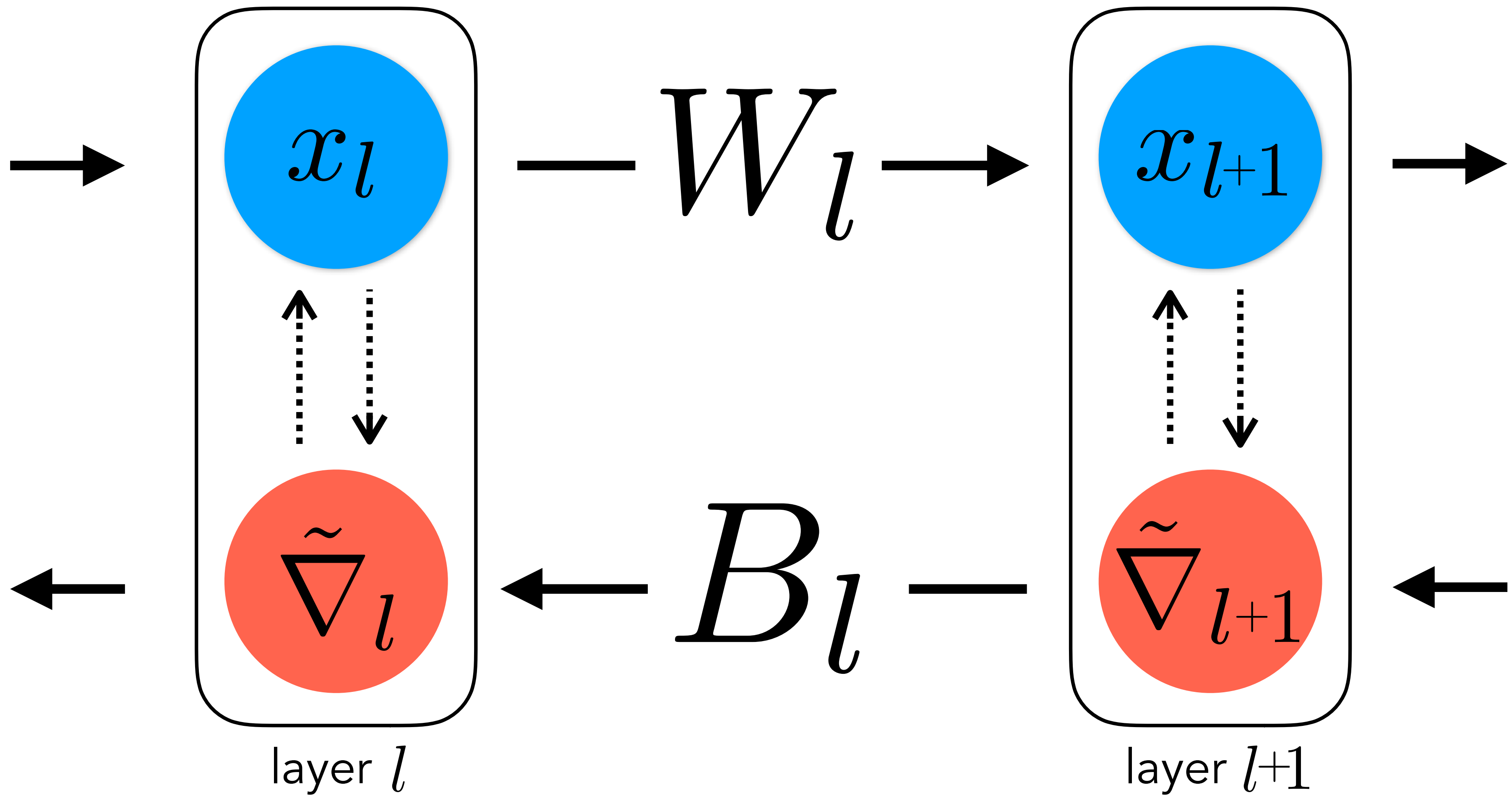
$$\Delta B_l = -\eta x_l \tilde{\nabla}_{l+1}^{\intercal} - \lambda_{\mathrm{KP}} B_l$$

- **Idea:** think of backwards weights updates as derivatives of a loss function
  - Integrates well with the current Deep Learning stack

[1] Lillicrap, Timothy P., et al. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7.1 (2016): 1-10.
[2] Akrout, Mohamed, et al. "Deep learning without weight transport." *Advances in Neural Information Processing Systems*. 2019.
[3] Kolen, John F., and Jordan B. Pollack. "Backpropagation without weight transport." *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 3. IEEE, 1994.

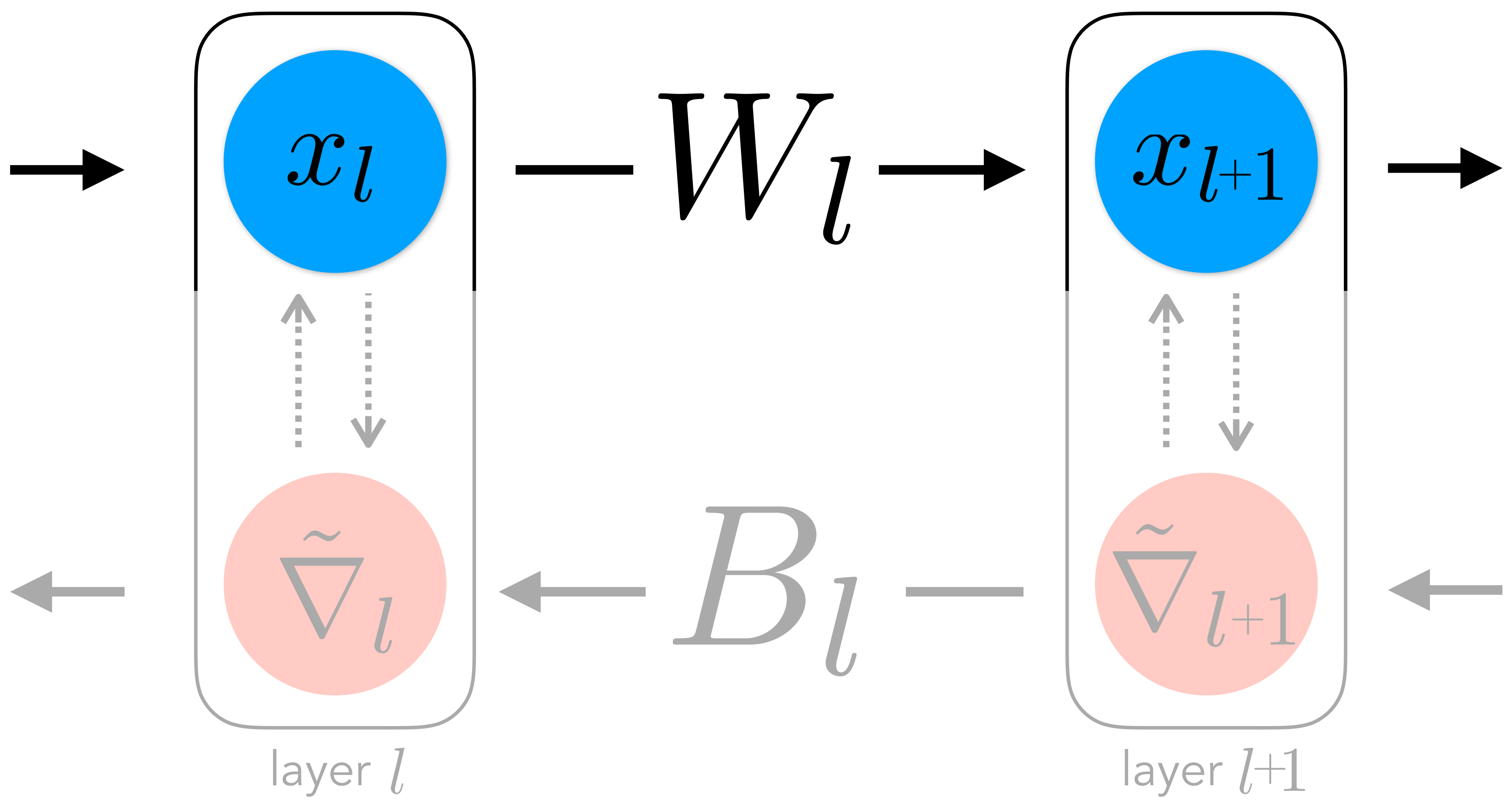
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



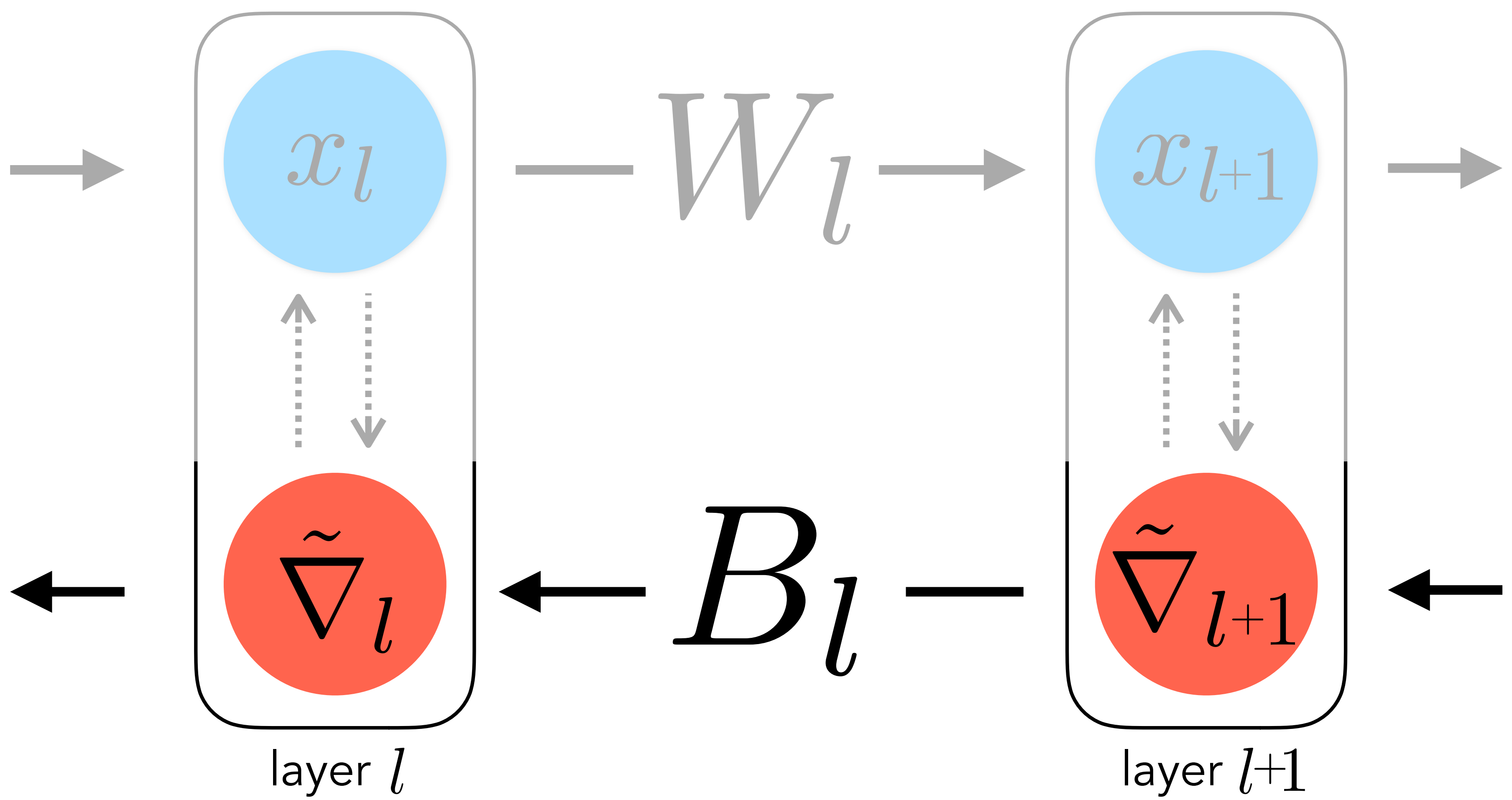
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



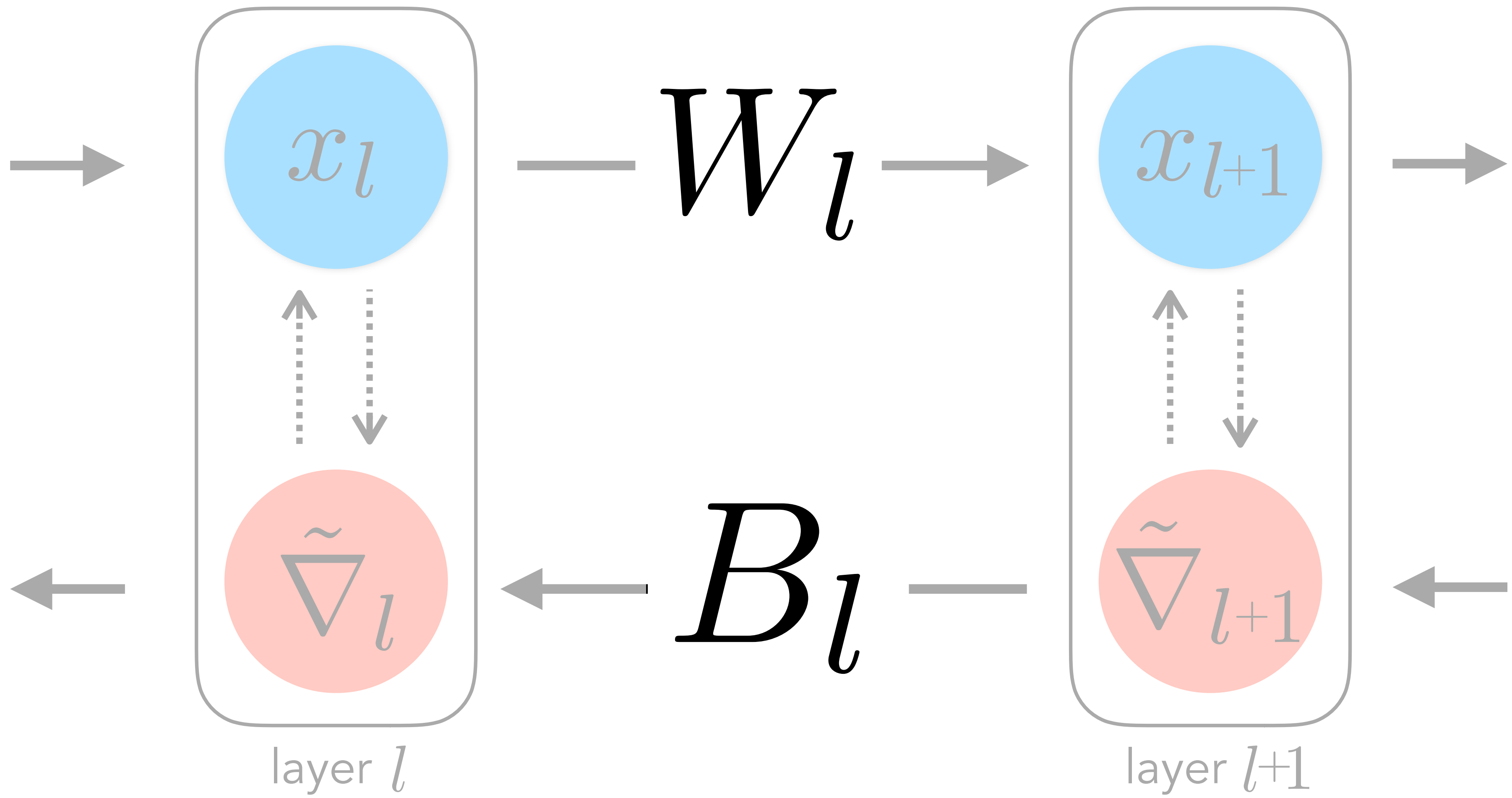
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



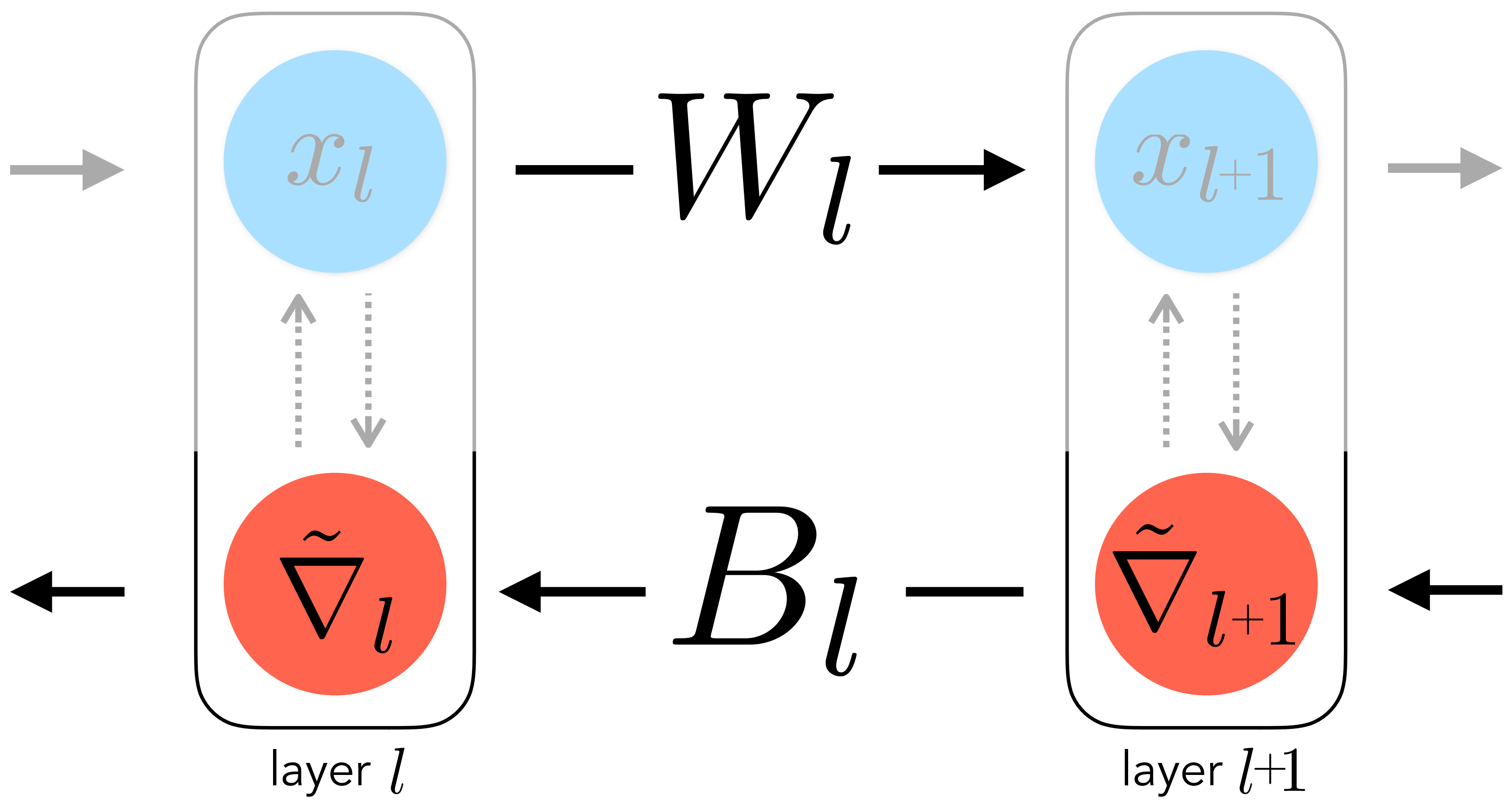
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W,B) = \mathcal{J}(W) + \mathcal{R}(B)$$



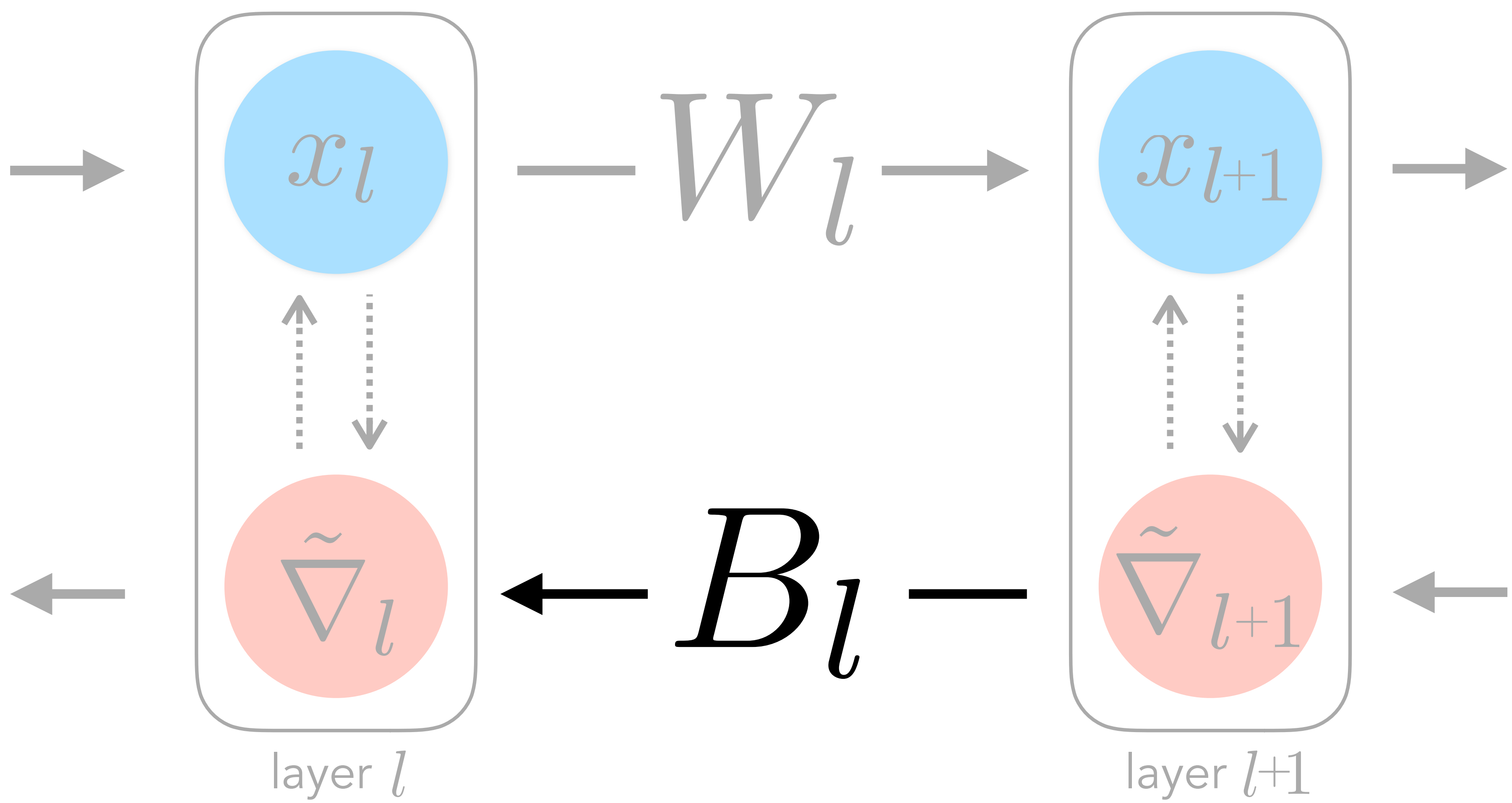
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



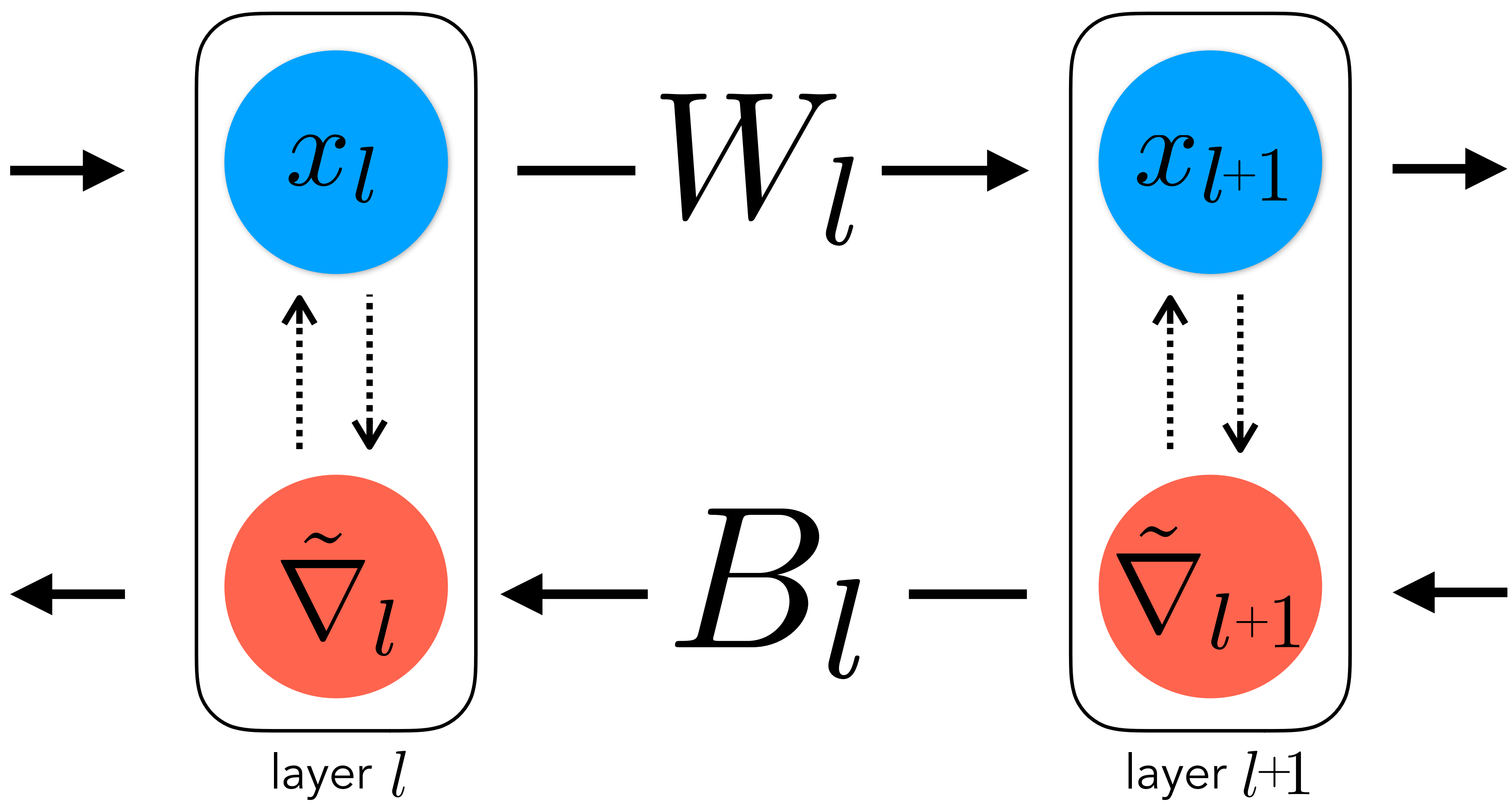
Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$



Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\mathrm{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\mathrm{tr}(B_l W_l)$	$-W_l^\intercal$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

$$\mathcal{R}_{\text{FA}} \equiv 0$$

Local	$\mathcal{P}_l$	$\nabla\mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\text{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla\mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\text{tr}(B_l W_l)$	$-W_l^\intercal$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

$$\mathcal{R}_{\text{FA}} \equiv 0$$

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\text{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\text{tr}(B_l W_l)$	$-W_l^\intercal$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Previous proposals

$$\mathcal{R}_{\text{FA}} \equiv 0$$

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\text{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\text{tr}(B_l W_l)$	$-W_l^\intercal$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\mathrm{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\mathrm{tr}(B_l W_l)$	$-W_l^\intercal$

Previous proposals

$$\mathcal{R}_{\mathrm{FA}} \equiv 0$$

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

$$\mathcal{R}_{\mathrm{IA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}} + \gamma \mathcal{P}_l^{\text{null}}$$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\text{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\text{tr}(B_l W_l)$	$-W_l^\intercal$

Previous proposals

$$\mathcal{R}_{\text{FA}} \equiv 0$$

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

$$\mathcal{R}_{\text{IA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}} + \gamma \mathcal{P}_l^{\text{null}}$$

$$\mathcal{R}_{\text{SA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{self}} + \beta \mathcal{P}_l^{\text{decay}}$$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Local	$\mathcal{P}_l$	$\nabla\mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\mathrm{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla\mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\mathrm{tr}(B_l W_l)$	$-W_l^\intercal$

Previous proposals

$$\mathcal{R}_{\mathrm{FA}} \equiv 0$$

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$

$$\mathcal{R}_{\mathrm{IA}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}} + \gamma \mathcal{P}_l^{\mathrm{null}}$$

$$\mathcal{R}_{\mathrm{SA}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{self}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$

$$\mathcal{R}_{\mathrm{AA}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{sparse}}$$

Regularization Inspired Learning Rule Framework

$$\mathcal{L}(W, B) = \mathcal{J}(W) + \mathcal{R}(B)$$

Local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
decay	$\frac{1}{2}\|\|B_l\|\|^2$	B_l
amp	$-\text{tr}(x_l^\intercal B_l x_{l+1})$	$-x_l x_{l+1}^\intercal$
null	$\frac{1}{2}\|\|B_l x_{l+1}\|\|^2$	$B_l x_{l+1} x_{l+1}^\intercal$
Non-local	$\mathcal{P}_l$	$\nabla \mathcal{P}_l$
sparse	$\frac{1}{2}\|\|x_l^\intercal B_l\|\|^2$	$x_l x_l^\intercal B_l$
self	$-\text{tr}(B_l W_l)$	$-W_l^\intercal$

Previous proposals

$$\mathcal{R}_{\text{FA}} \equiv 0$$

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

$$\mathcal{R}_{\text{IA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}} + \gamma \mathcal{P}_l^{\text{null}}$$

$$\mathcal{R}_{\text{SA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{self}} + \beta \mathcal{P}_l^{\text{decay}}$$

$$\mathcal{R}_{\text{AA}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{sparse}}$$

Novel proposals

Evaluating Learning Rules

Evaluating Learning Rules

- **1. Task Performance**

Evaluating Learning Rules

- **1. Task Performance**
  - Animal must perform well at a behavior. Bartunov et al. (2018) point out lack of scalability of prior proposals.

Evaluating Learning Rules

- **1. Task Performance**
  - Animal must perform well at a behavior. Bartunov et al. (2018) point out lack of scalability of prior proposals.
  - ImageNet top-1 validation accuracy, since performance-optimized CNNs provide the most effective model of neural responses throughout the primate ventral visual pathway (Yamins et al., 2014; Cadena et al., 2019).

Evaluating Learning Rules

- **I. Task Performance**
  - Animal must perform well at a behavior. Bartunov et al. (2018) point out lack of scalability of prior proposals.
  - ImageNet top-1 validation accuracy, since performance-optimized CNNs provide the most effective model of neural responses throughout the primate ventral visual pathway (Yamins et al., 2014; Cadena et al., 2019).
- **II. Metaparameter Robustness**

Evaluating Learning Rules

- **I. Task Performance**
  - Animal must perform well at a behavior. Bartunov et al. (2018) point out lack of scalability of prior proposals.
  - ImageNet top-1 validation accuracy, since performance-optimized CNNs provide the most effective model of neural responses throughout the primate ventral visual pathway (Yamins et al., 2014; Cadena et al., 2019).
- **II. Metaparameter Robustness**
  - Metaparameters that work well for a particular learning rule should transfer well to different and deeper architectures.

Evaluating Learning Rules

- **I. Task Performance**
  - Animal must perform well at a behavior. Bartunov et al. (2018) point out lack of scalability of prior proposals.
  - ImageNet top-1 validation accuracy, since performance-optimized CNNs provide the most effective model of neural responses throughout the primate ventral visual pathway (Yamins et al., 2014; Cadena et al., 2019).
- **II. Metaparameter Robustness**
  - Metaparameters that work well for a particular learning rule should transfer well to different and deeper architectures.
  - ImageNet top-1 validation accuracy *across* models for *fixed* metaparameters.

Route I: Local Learning Rules

Local Learning Rules: Weight Mirror

Weight Mirror: literature

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

Learning Rule	Top-1 Val Accuracy
Backprop.	70.06%
$\mathcal{R}_{\text{WM}}$	63.5%

Local Learning Rules: Weight Mirror

Weight Mirror: literature

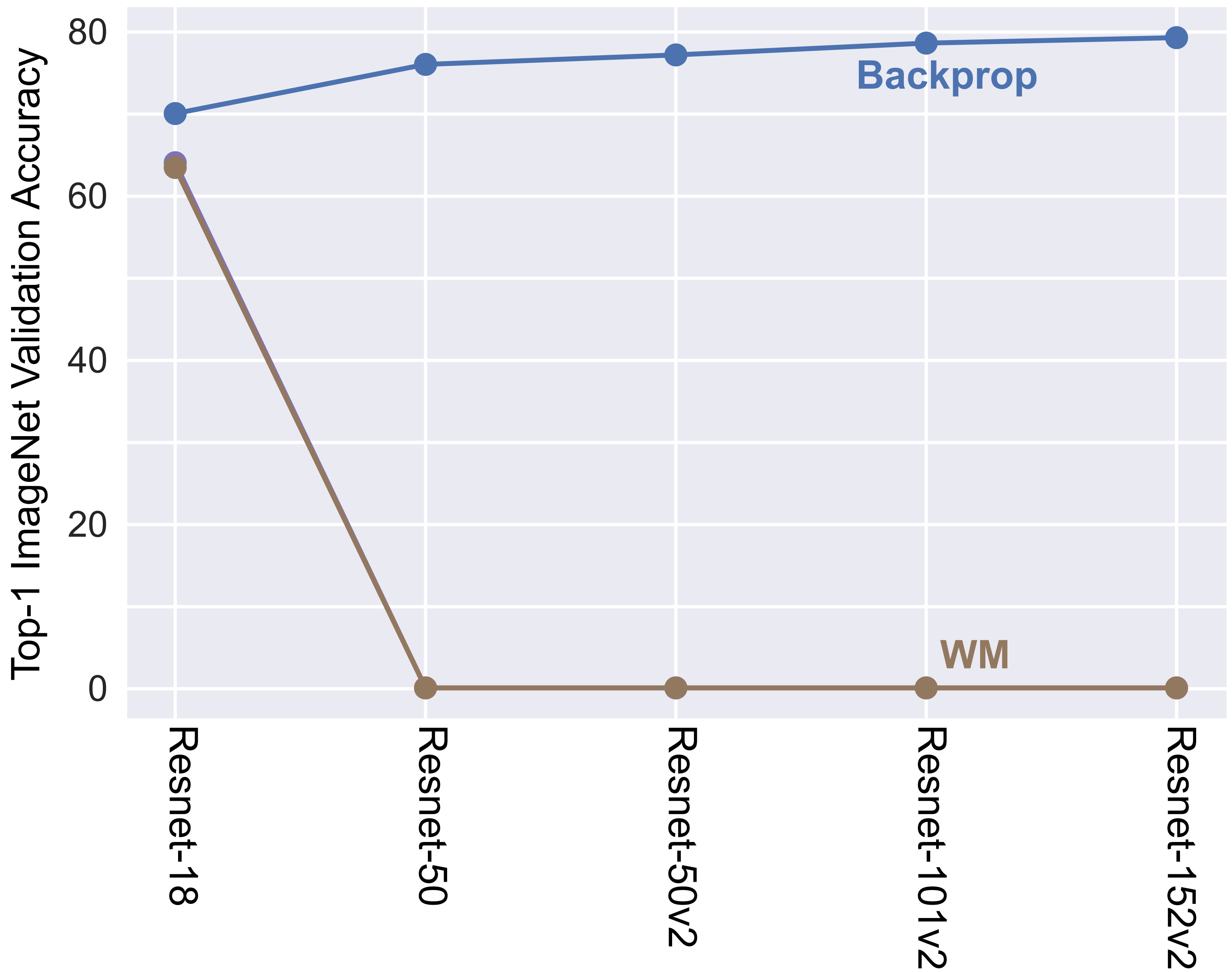
$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

Learning Rule	Top-1 Val Accuracy
Backprop.	70.06%
$\mathcal{R}_{\text{WM}}$	63.5%
Akrout et al. 2019	**69.73%**

Local Learning Rules: Weight Mirror

Weight Mirror: literature

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

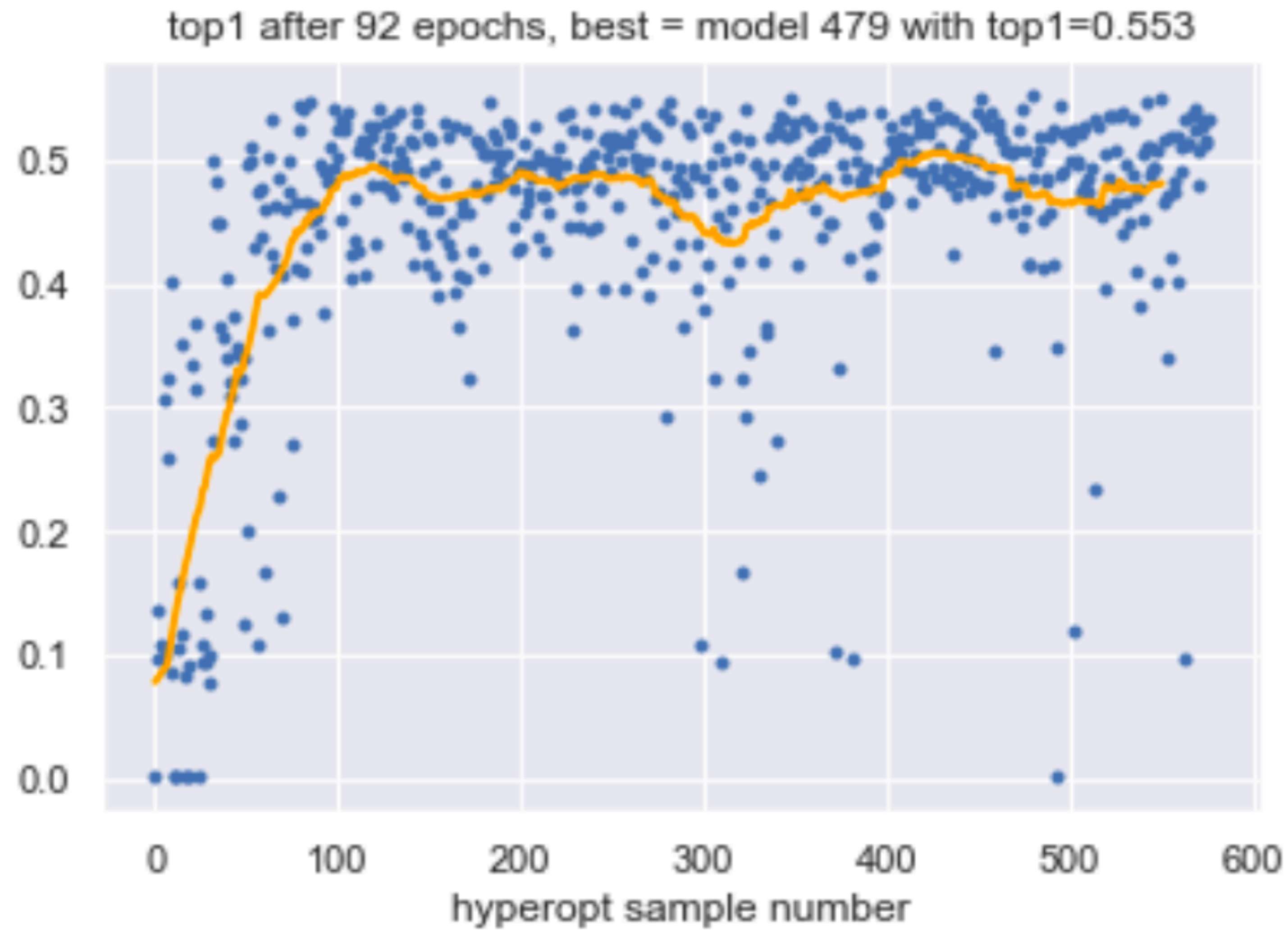


Local Learning Rules: Weight Mirror

Weight Mirror: optimized metaparameters

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$

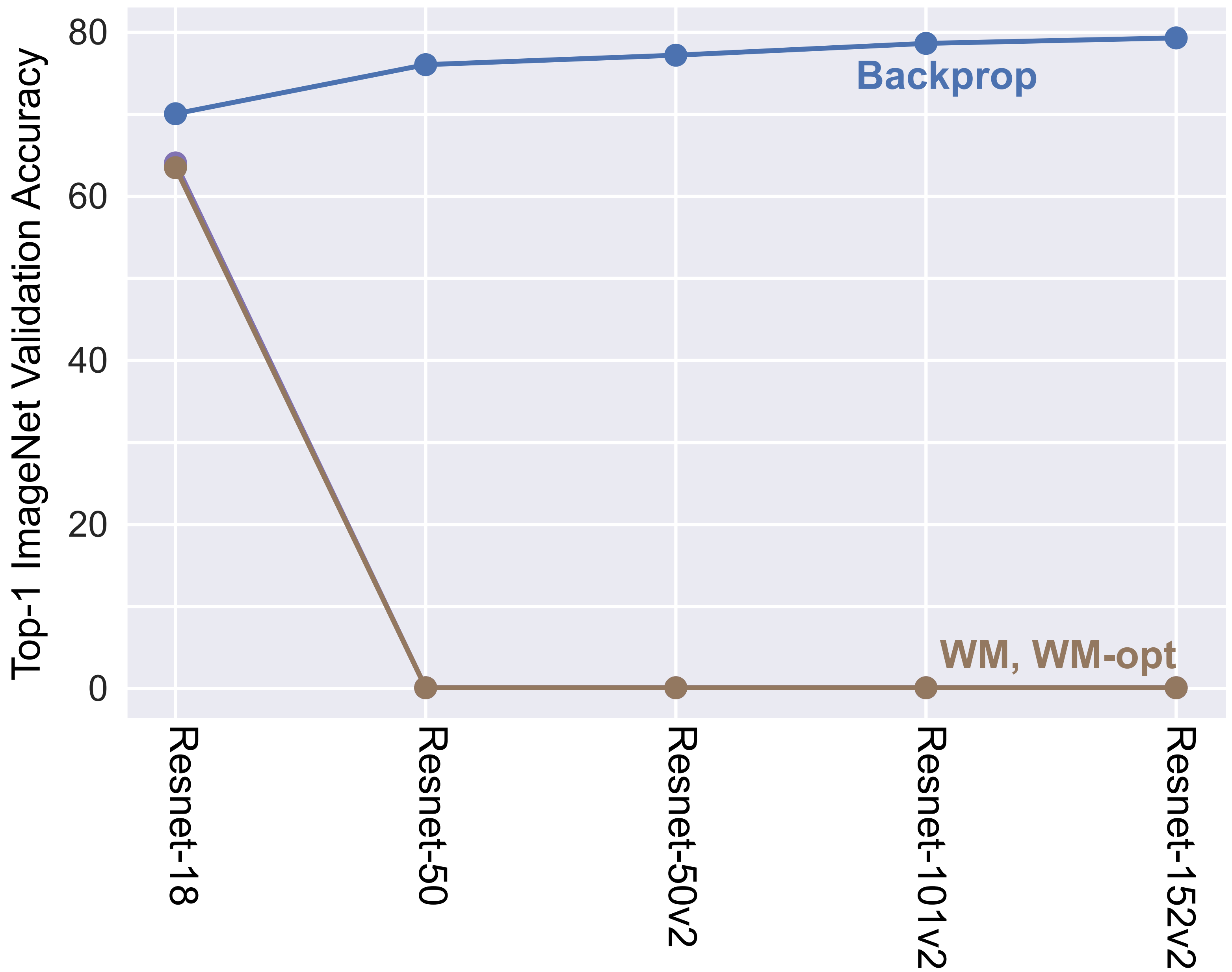
- TPE Search over alpha, beta and the variance of the noise used in mirror mode on ResNet-18.



Local Learning Rules: Weight Mirror

Weight Mirror: optimized metaparameters

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

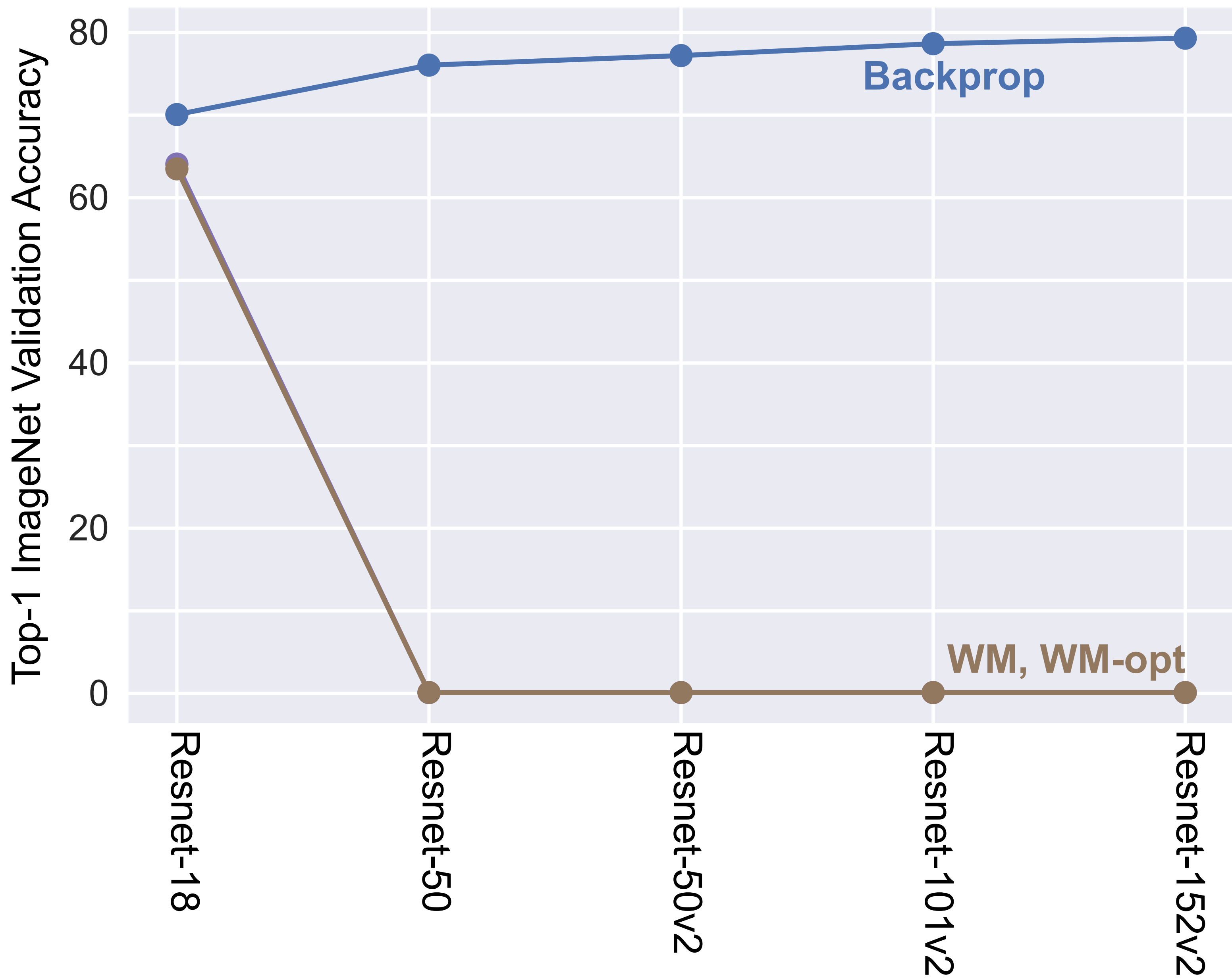


Local Learning Rules: Weight Mirror

Weight Mirror: optimized metaparameters

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$

- Why is weight mirror unstable?

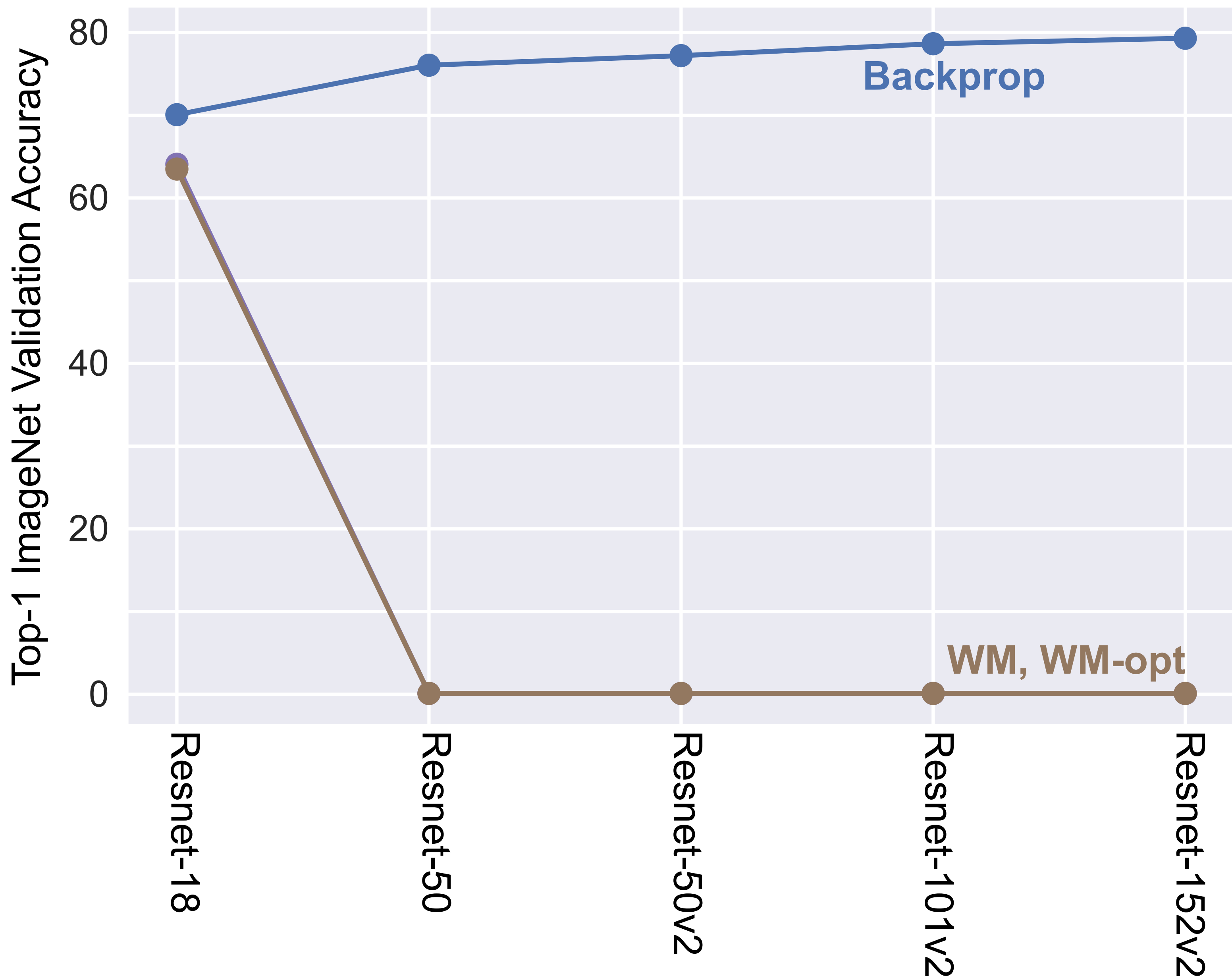


Local Learning Rules: Weight Mirror

Weight Mirror: optimized metaparameters

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

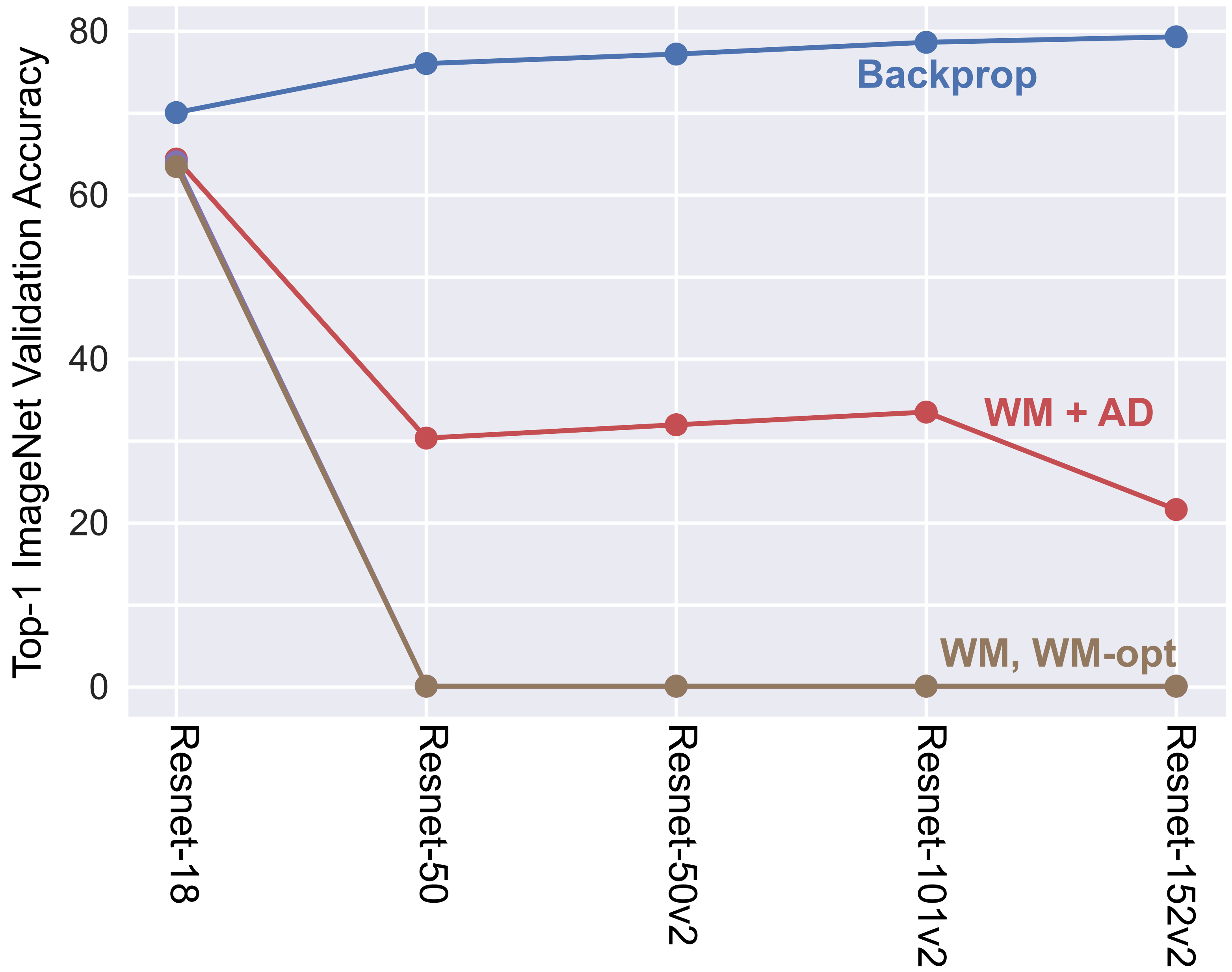
- Why is weight mirror unstable?
- Could this be helped by using an adaptive optimizer?



Local Learning Rules: Improved Metaparameter Robustness

Weight Mirror: adding an adaptive optimizer

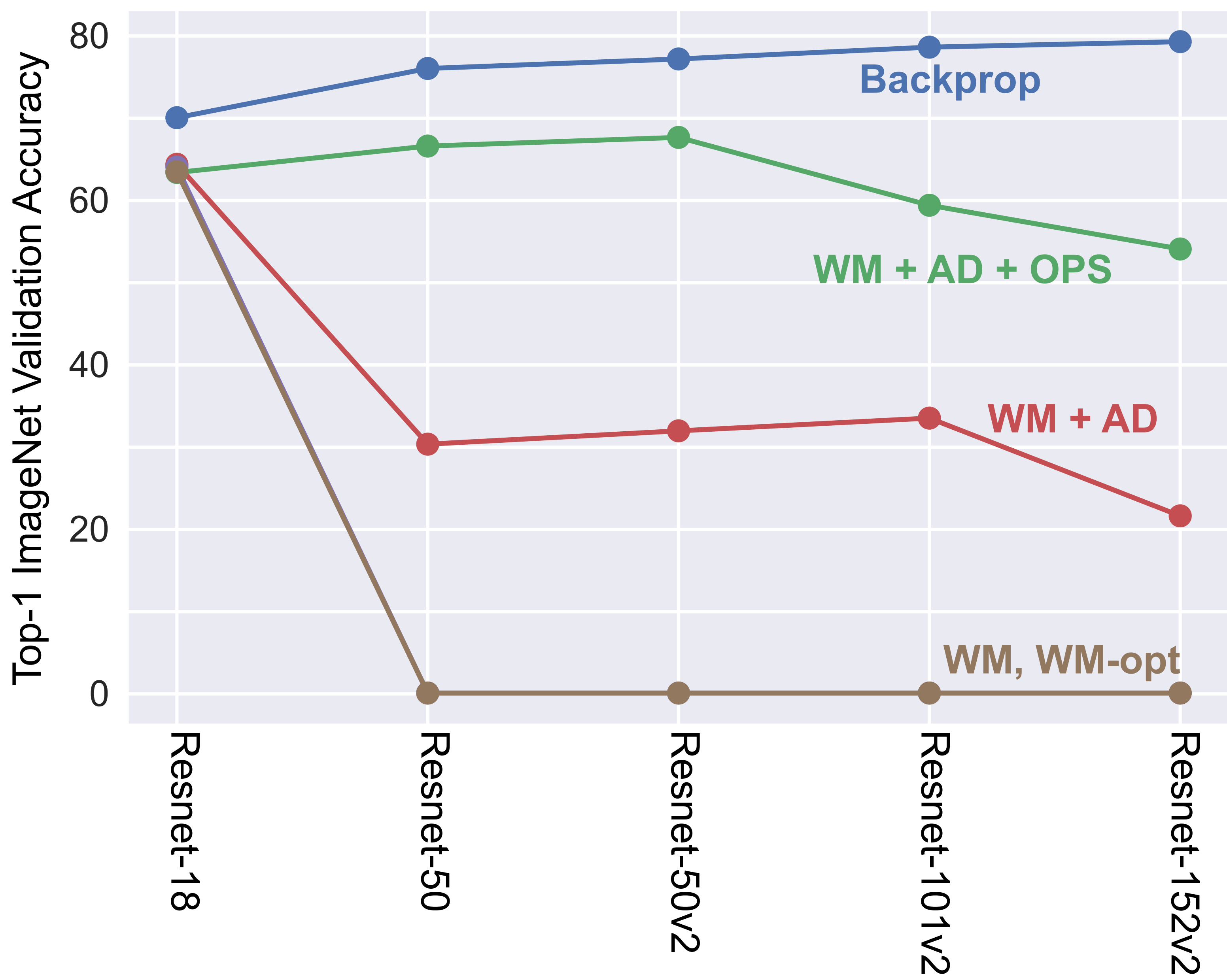
$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$



Local Learning Rules: Improved Metaparameter Robustness

Weight Mirror: adding an adaptive optimizer and normalizing operations

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$



Oja-style Stabilization of Weight Mirror

- The update given by WM (without decay) is Hebbian
- Purely Hebbian learning rules are unstable
- WM adds weight decay to prevent diverging norms
- An alternative strategy to stabilizing Hebbian dynamics given by Oja (1982) for learning dynamics of linear neurons

Normalizing operation:

$$B_l^{(t+1)} = \frac{B_l^{(t)} + \eta x_l x_{l+1}^\intercal}{||B_l^{(t)} + \eta x_l x_{l+1}^\intercal||}$$

Taylor series expansion:

$$B_l^{(t+1)} = B_l^{(t)} + \eta \left(x_l x_{l+1}^\intercal - B_l^{(t)} x_l^\intercal B_l^{(t)} x_{l+1} \right) + O(\eta^2)$$

New update:

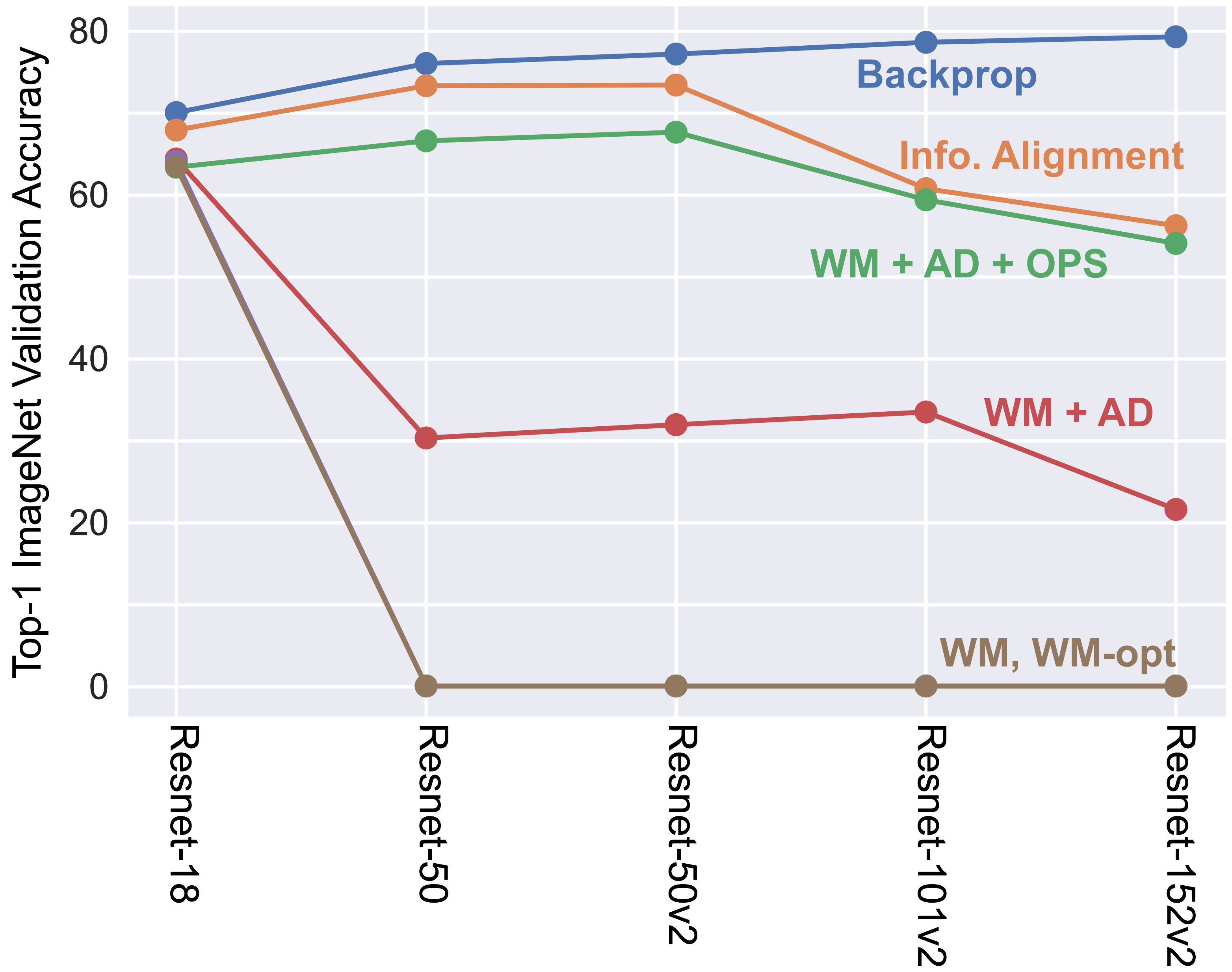
$$\Delta B_l = \eta \left(x_l x_{l+1}^\intercal - \underbrace{B_l x_l^\intercal B_l x_{l+1}}_{\approx \nabla \mathcal{P}_{\text{null}}} \right)$$

A More Robust Local Learning Rule

Weight Mirror

Information Alignment

$$\mathcal{R}_{\mathrm{WM}} = \sum_{l \in \mathrm{layers}} \alpha \mathcal{P}_l^{\mathrm{amp}} + \beta \mathcal{P}_l^{\mathrm{decay}}$$



Route II: Non-Local Learning Rules

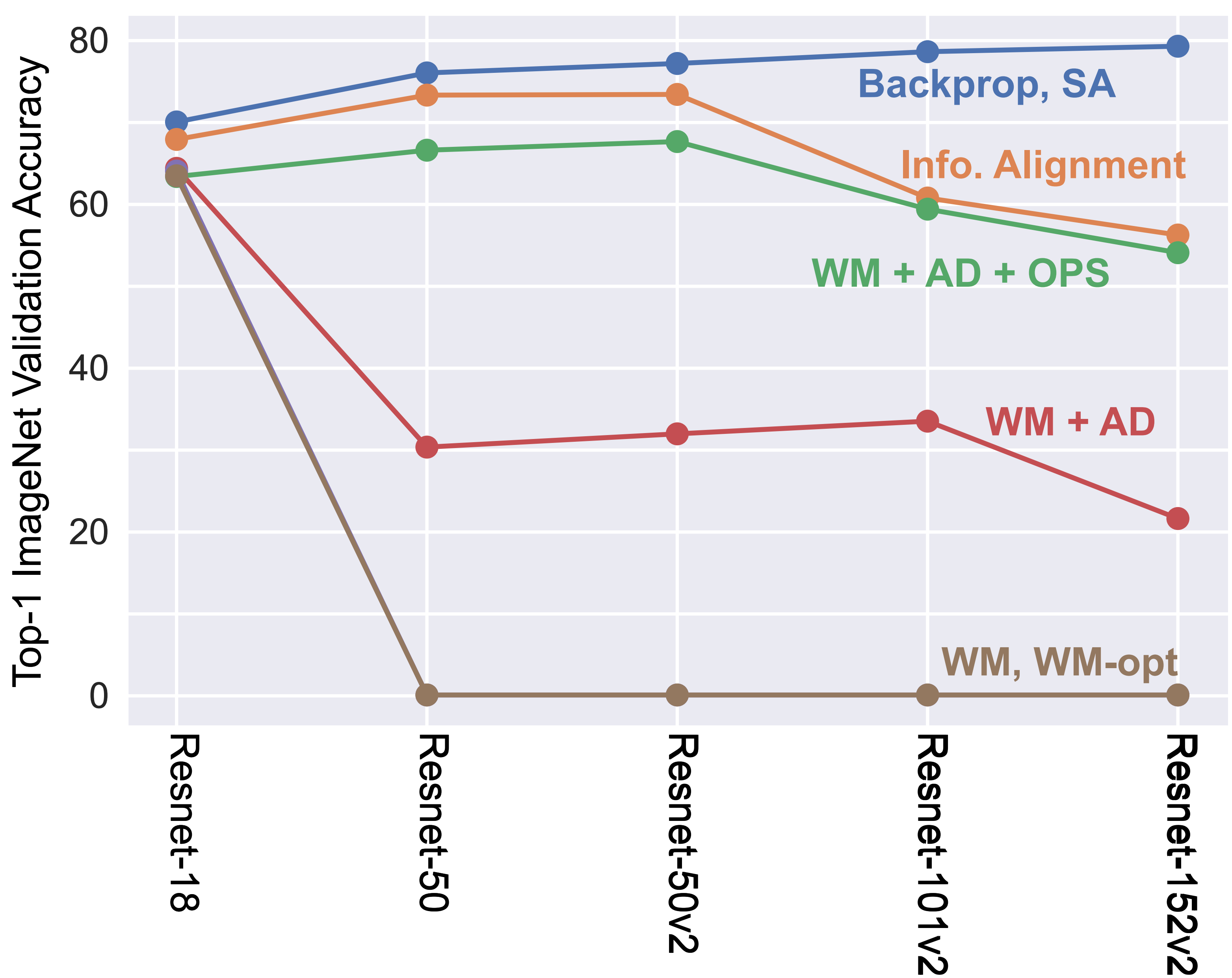
Non-Local Learning Rules

Weight Mirror

Symmetric Alignment

$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

$$\propto \sum_{l \in \text{layers}} \frac{1}{2} ||W_l - B_l^{\intercal}||^2$$



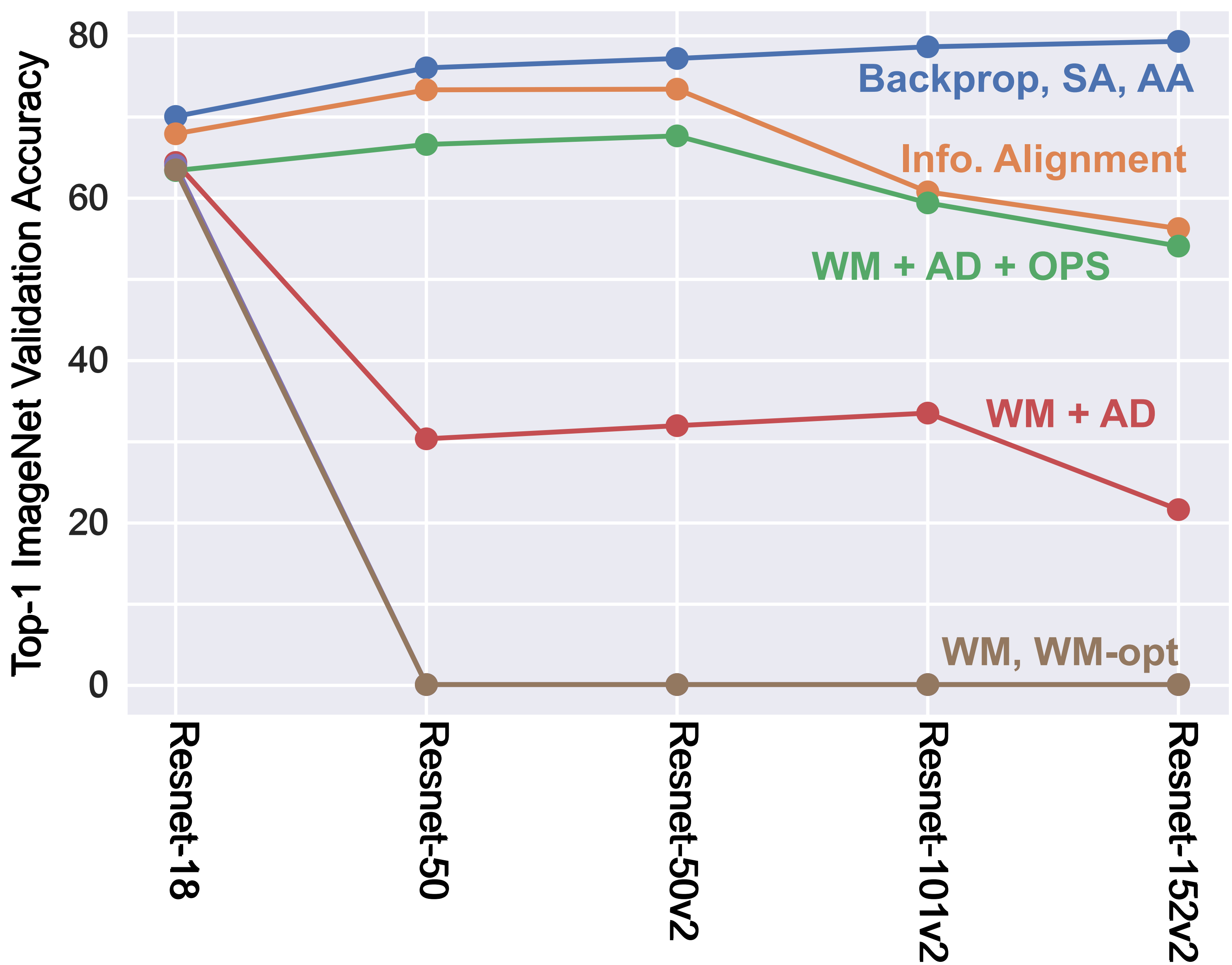
Non-Local Learning Rules

Weight Mirror

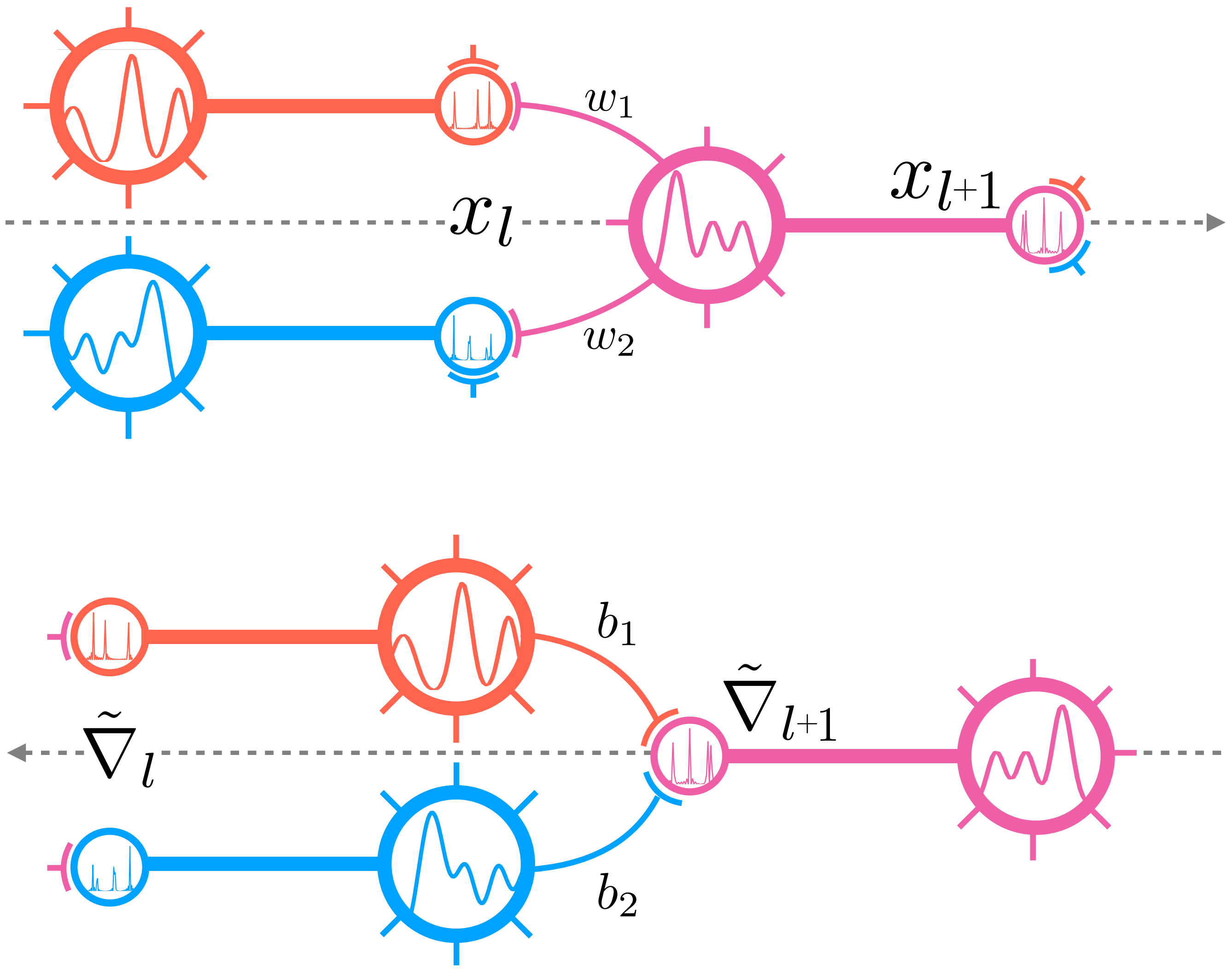
$$\mathcal{R}_{\text{WM}} = \sum_{l \in \text{layers}} \alpha \mathcal{P}_l^{\text{amp}} + \beta \mathcal{P}_l^{\text{decay}}$$

Activation Alignment

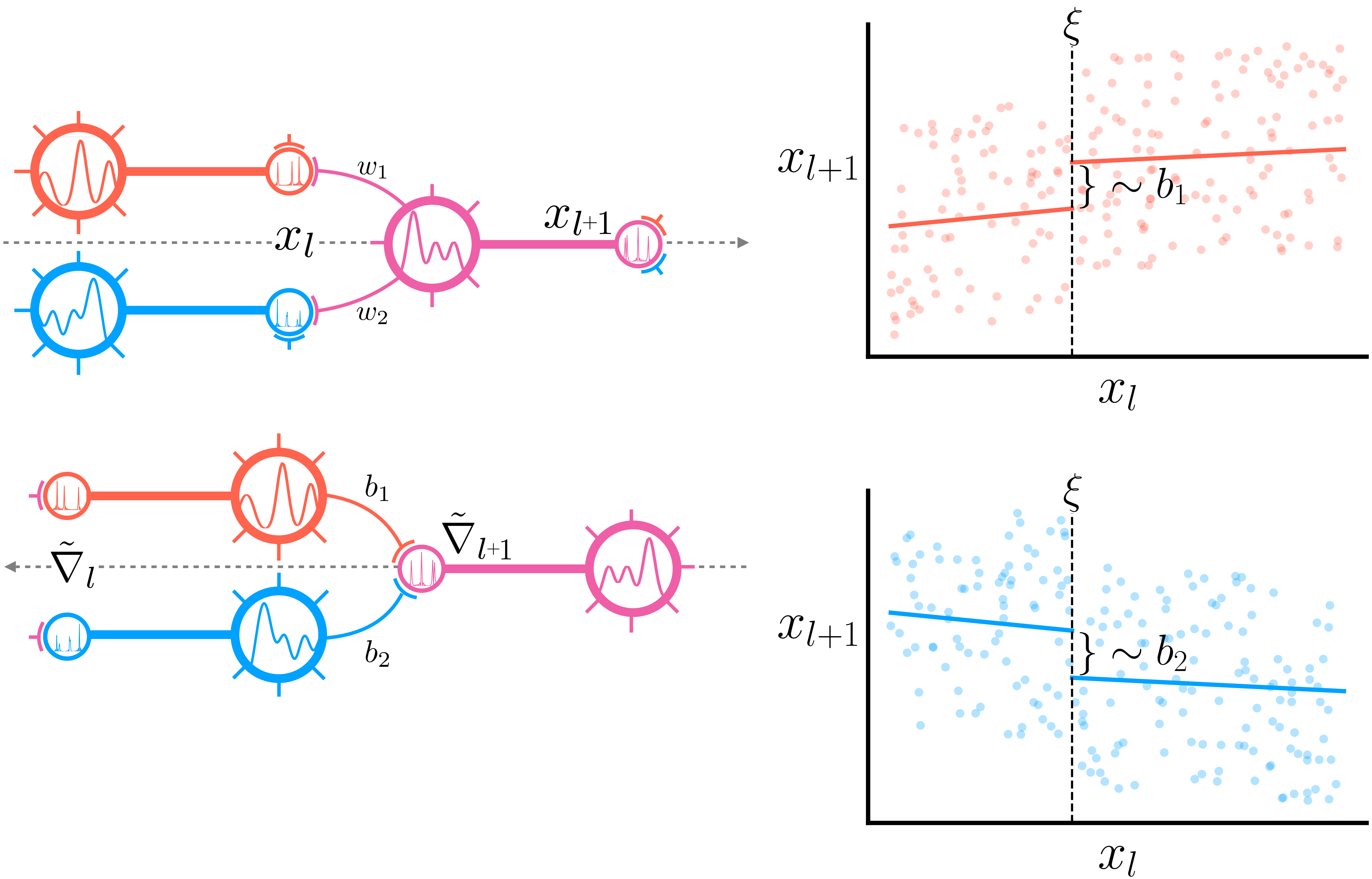
$$\propto \sum_{l \in \text{layers}} \frac{1}{2} ||W_l x_l - B_l^{\intercal} x_l||^2$$



Weight Estimation



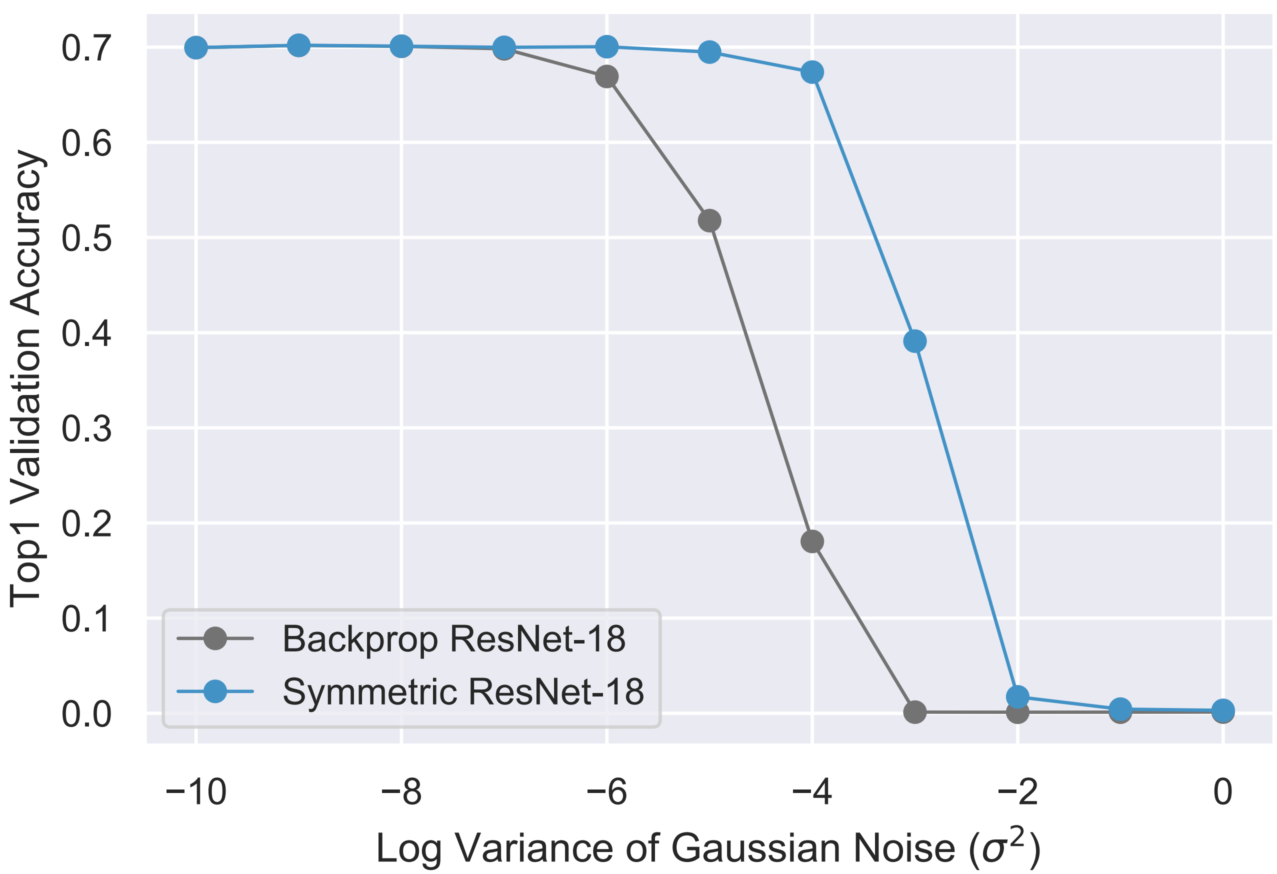
Weight Estimation: Regression Discontinuity Design



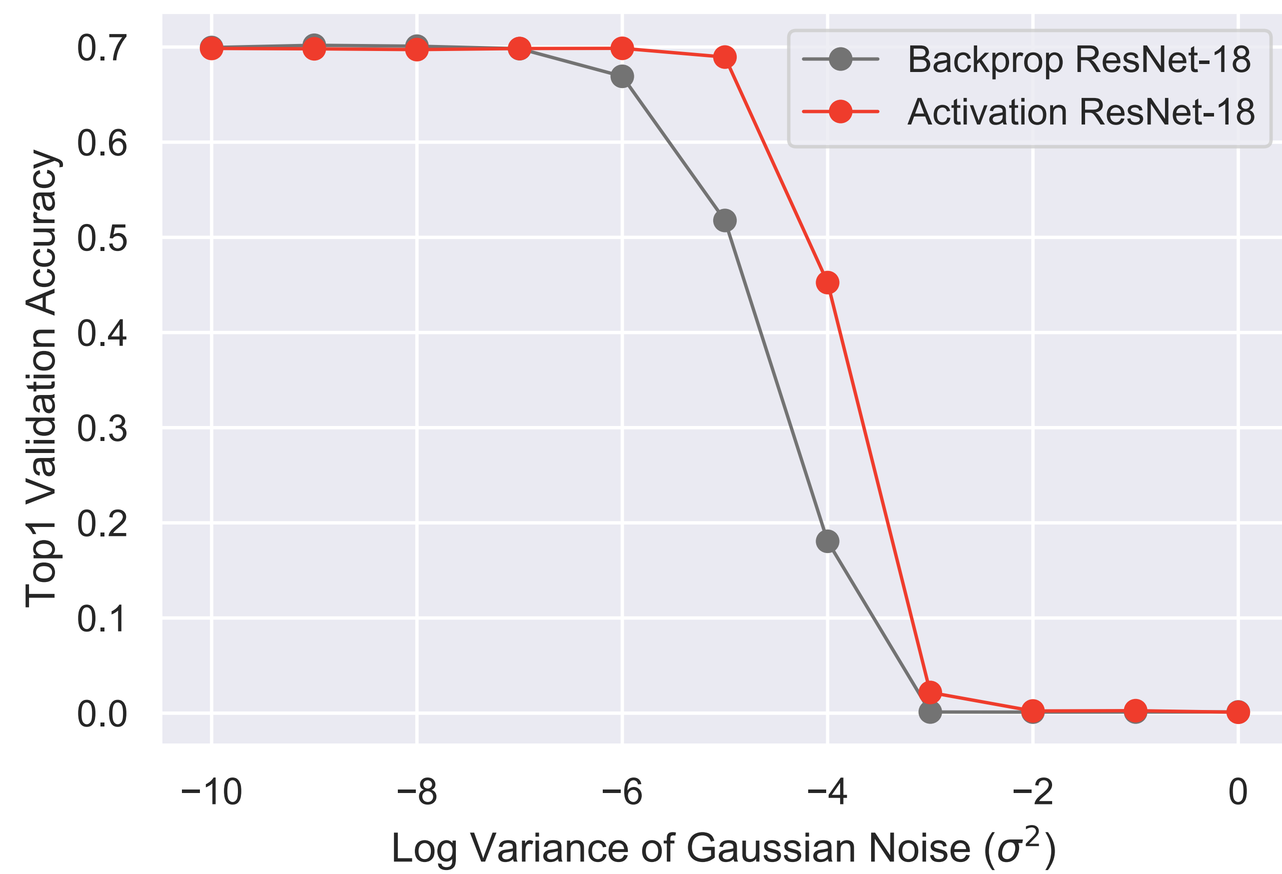
[1] Lansdell, B. J. and Kording, K. P. Spiking allows neurons to estimate their causal effect. bioRxiv, pp. 253351, 2019.
[2] Guerguiev, J., Kording, K. P., and Richards, B. A. Spike-based causal inference for weight alignment. arXiv:1910.01689 [cs, q-bio], October 2019.

Non-Local Learning Rules: Robust to noisy updates

Symmetric Alignment



Activation Alignment



$$\Delta B = \nabla \mathcal{R} + \mathcal{N}(0, \sigma^2)$$

Conclusions

Conclusions

- Unifying framework allowing the systematic identification of novel proposals

Conclusions

- Unifying framework allowing the systematic identification of novel proposals
- Local learning rule that transfers more robustly across architectures than previous proposals

Conclusions

- Unifying framework allowing the systematic identification of novel proposals

- Local learning rule that transfers more robustly across architectures than previous proposals

- Non-local learning rules perform competitively, robust to noisy updates

Conclusions

- Unifying framework allowing the systematic identification of novel proposals
- Local learning rule that transfers more robustly across architectures than previous proposals
- Non-local learning rules perform competitively, robust to noisy updates
- **Route I:** further improvement of local rules to close the gap with respect to backpropagation

Conclusions

- Unifying framework allowing the systematic identification of novel proposals

- Local learning rule that transfers more robustly across architectures than previous proposals

- Non-local learning rules perform competitively, robust to noisy updates

- **Route I:** further improvement of local rules to close the gap with respect to backpropagation

- Github library: https://github.com/neuroailab/neural-alignment

Conclusions

- Unifying framework allowing the systematic identification of novel proposals
- Local learning rule that transfers more robustly across architectures than previous proposals
- Non-local learning rules perform competitively, robust to noisy updates
- **Route I:** further improvement of local rules to close the gap with respect to backpropagation
- **Route II:** further refinement and characterization of scalable biological implementations of weight estimation mechanisms for non-local rules
- Github library: https://github.com/neuroailab/neural-alignment