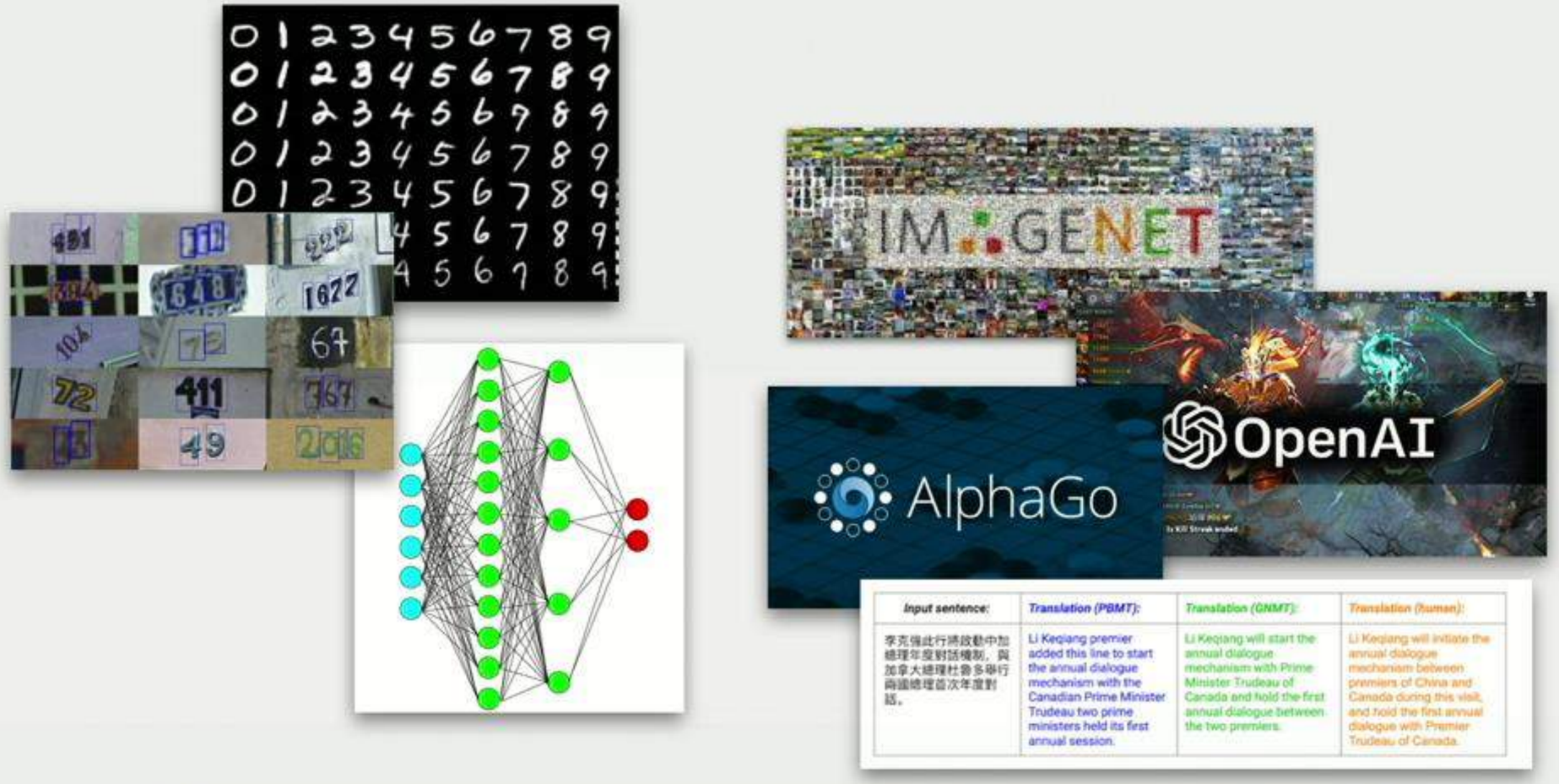
How Does Batch Normalization Help Optimization?

Andrew Ilyas

Joint work with Shibani Santurkar*, Dimitris Tsipras*, and Aleksander Mądry

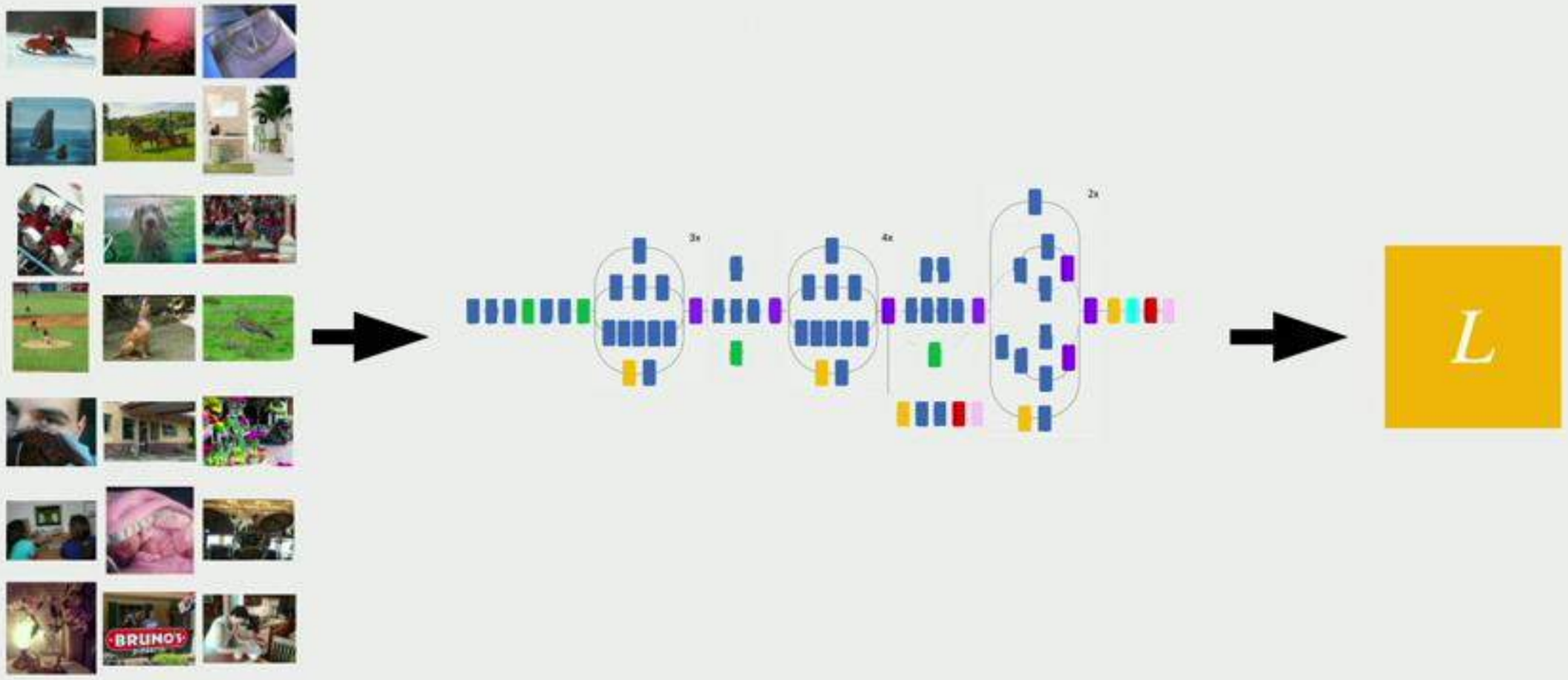
The Deep Learning Revolution



Advances in **Hardware, Data and Algorithms**

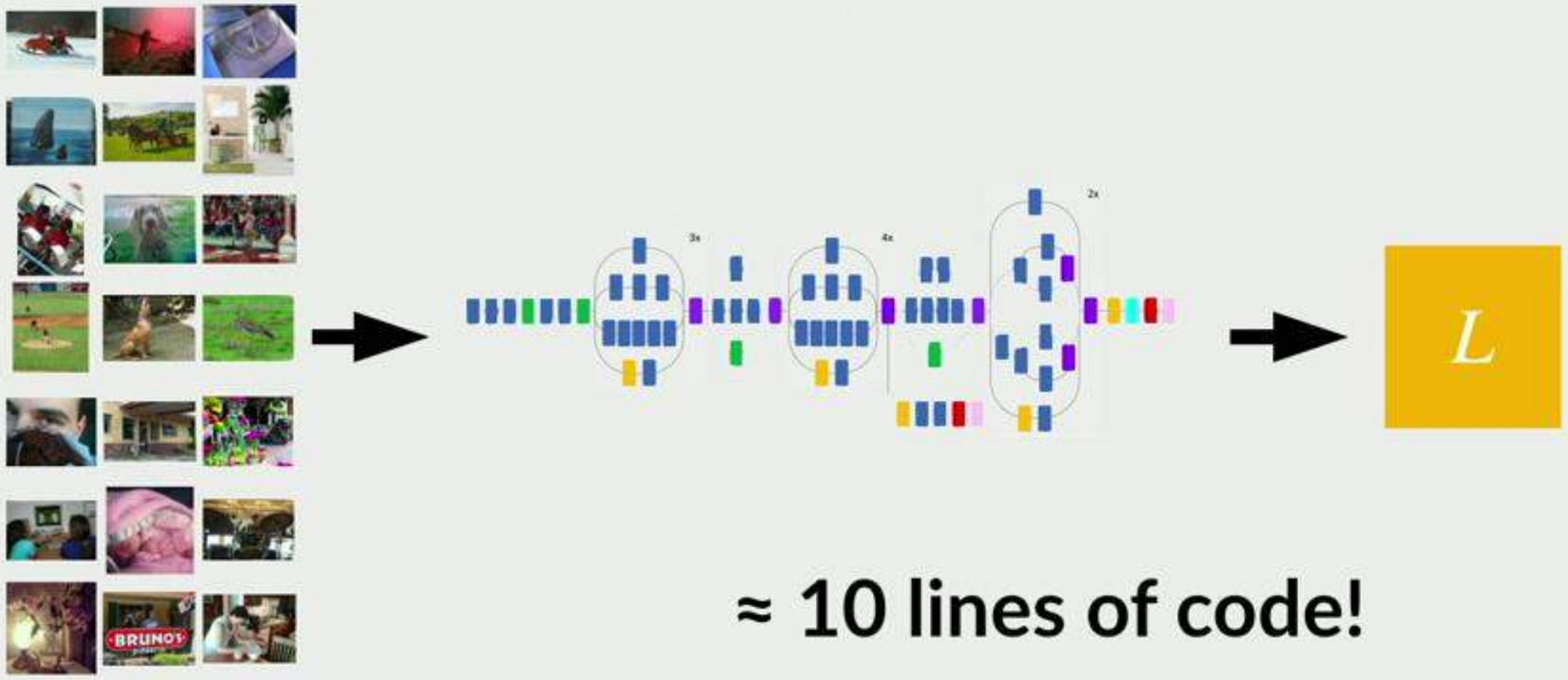
[Wu et al, 2016] [Deng et al, 2009] [OpenAI Five, 2018] [Vinyals et al, 2016]

At the Core: Deep Neural Networks



[Deng et al, 2009] [Szegedy et al., 2015]

At the Core: Deep Neural Networks



≈ 10 lines of code!

[Deng et al, 2009] [Szegedy et al., 2015]

Behind the Scenes

Training DNNs is *simple*, but *difficult (we just have a toolkit!)*

Many core components poorly understood (but do work)

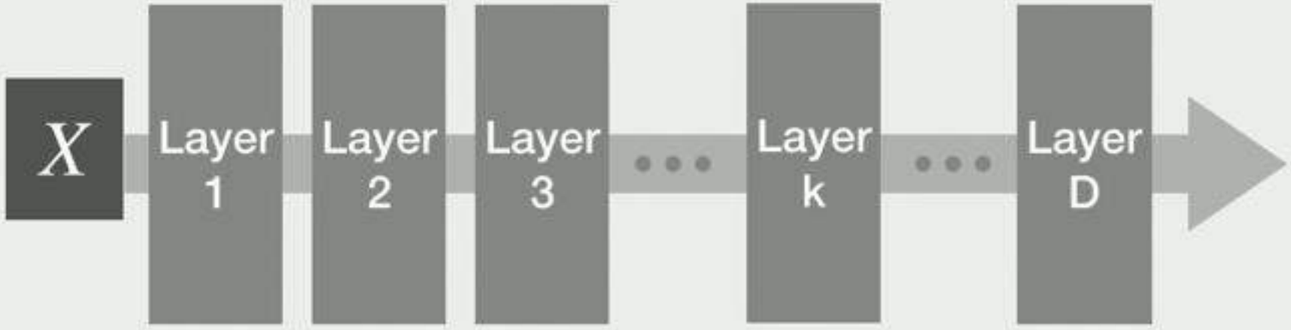
***General goal**: Build a better understanding of the modern machine learning toolkit*

Today:

A closer look at Batch Normalization

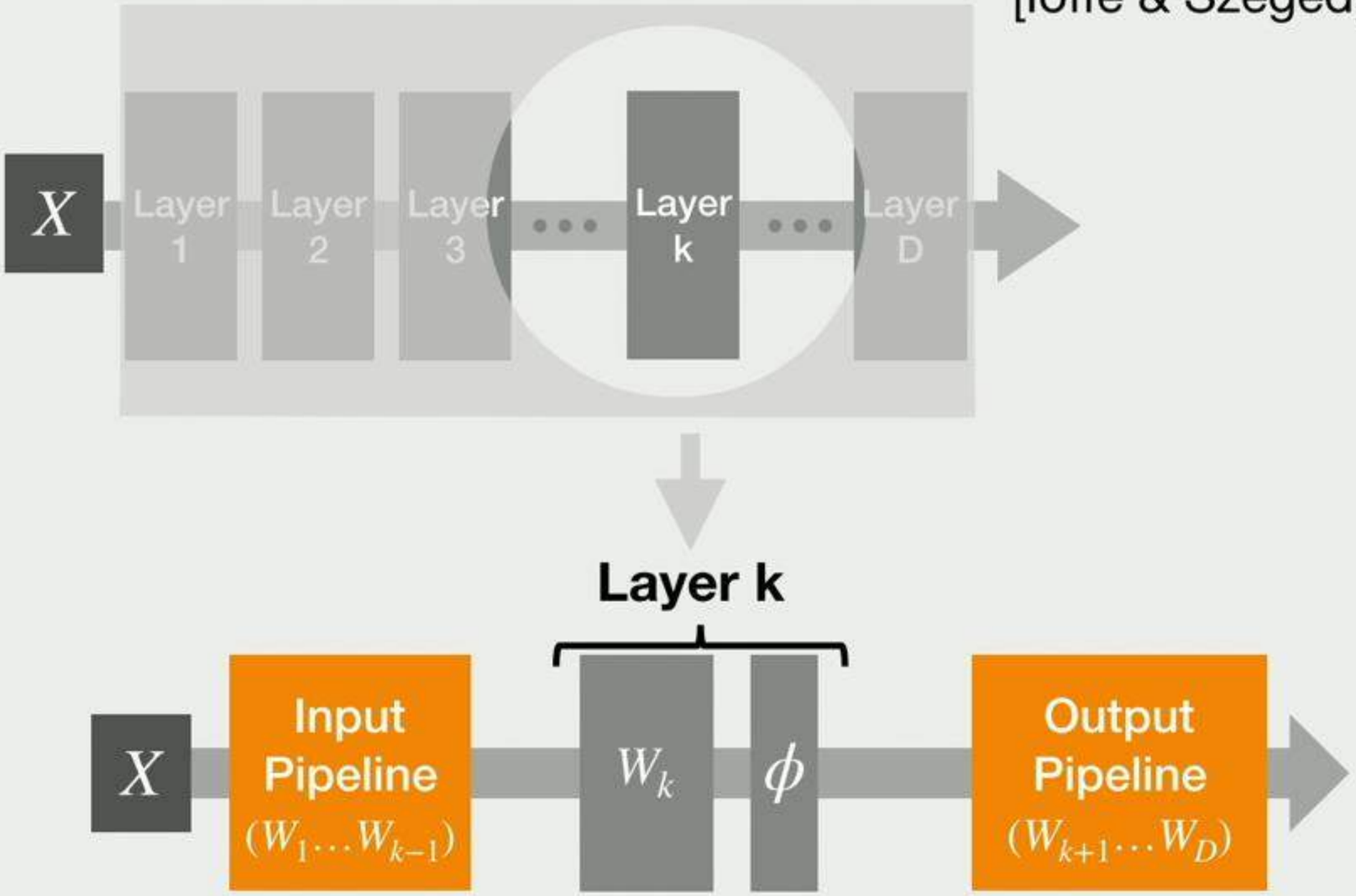
Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



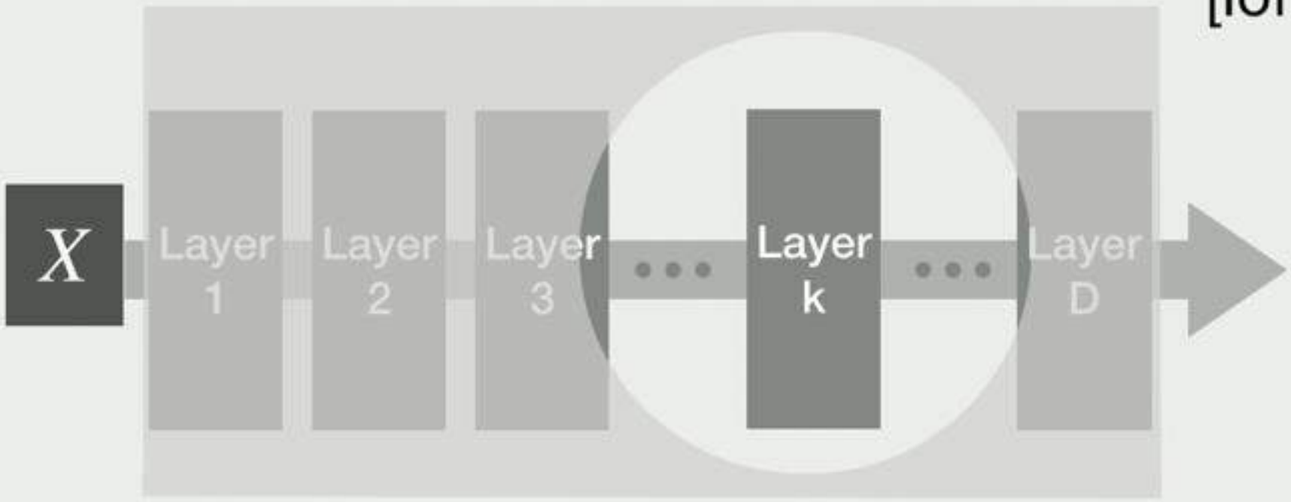
Batch Normalization (BatchNorm)

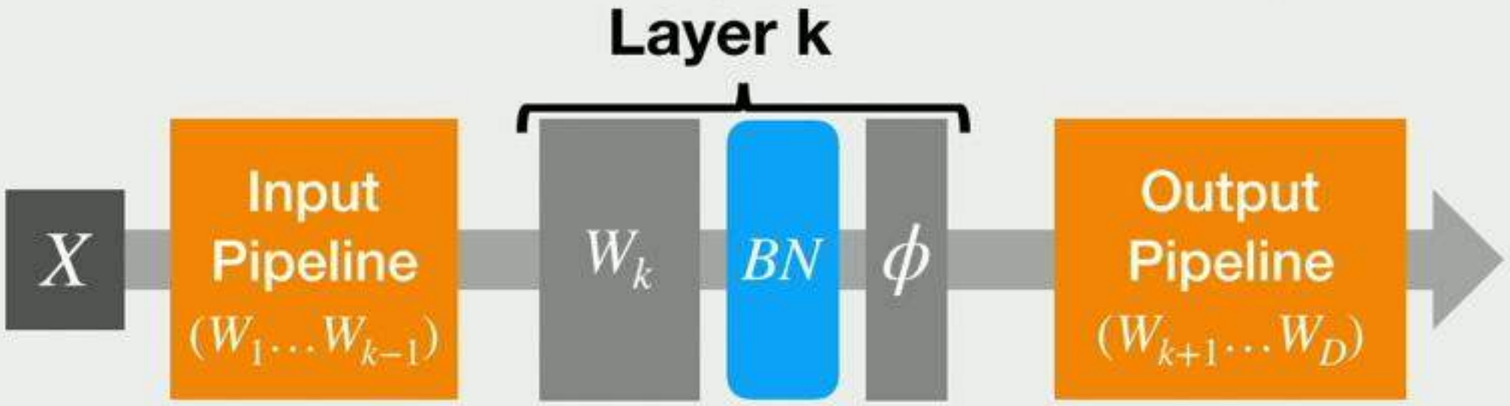
[Ioffe & Szegedy, 2015]



Batch Normalization (BatchNorm)

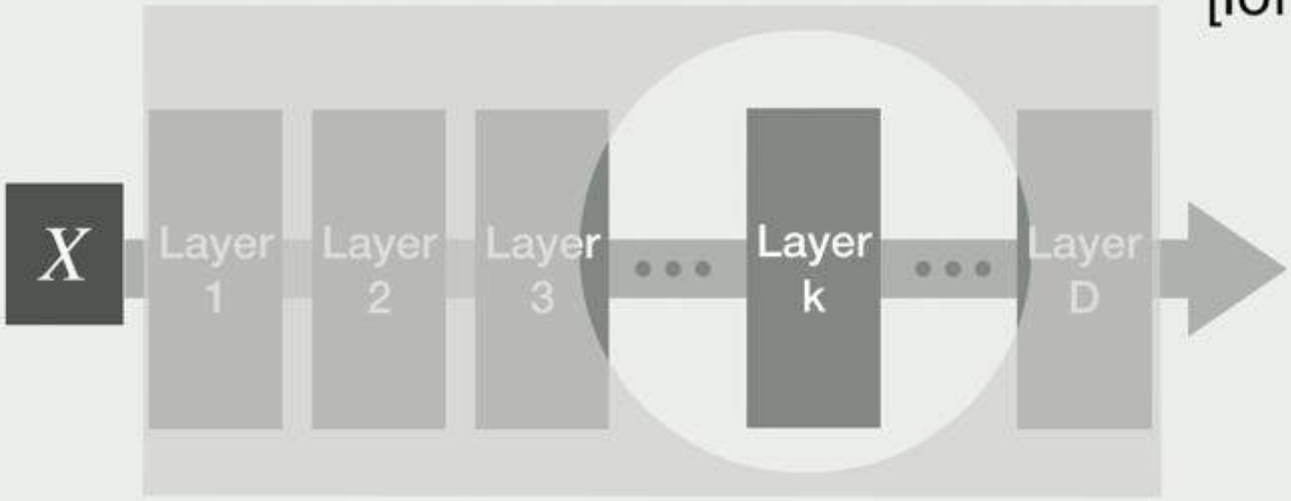
[Ioffe & Szegedy, 2015]



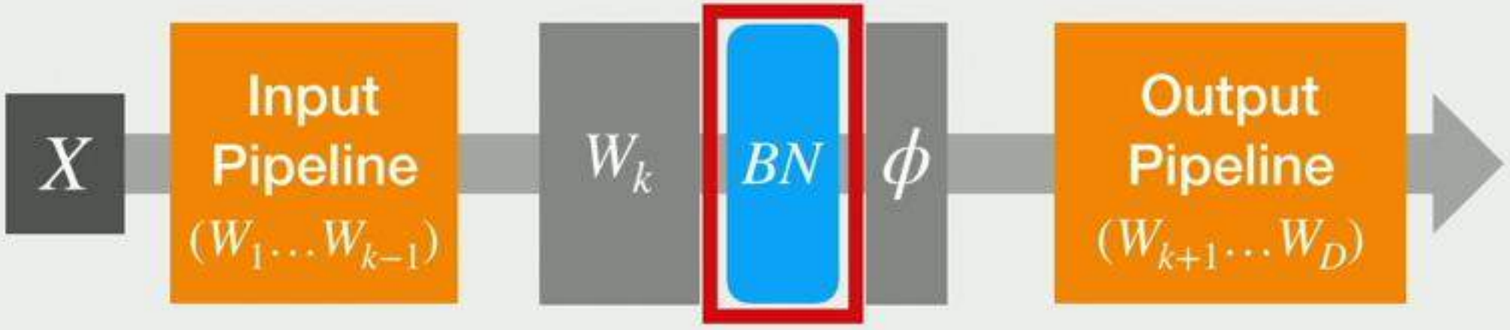


Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]

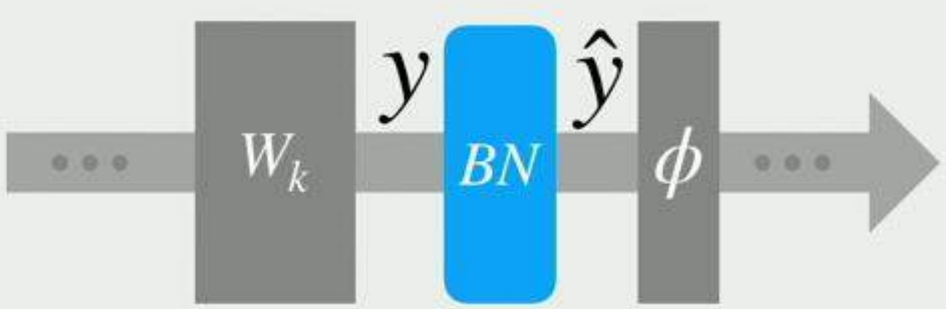


whitening transformation



Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



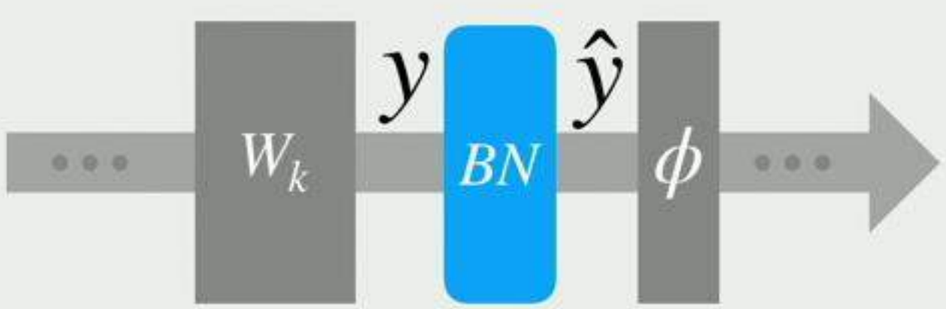
Make distribution "close" to **standard normal**

$$\hat{y} = \frac{y - \mu}{\sigma + \epsilon} \quad ,$$

where $\mu = \mathbb{E}[y]$ $\sigma^2 = Var(y)$

Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



Make distribution "close" to **standard normal**

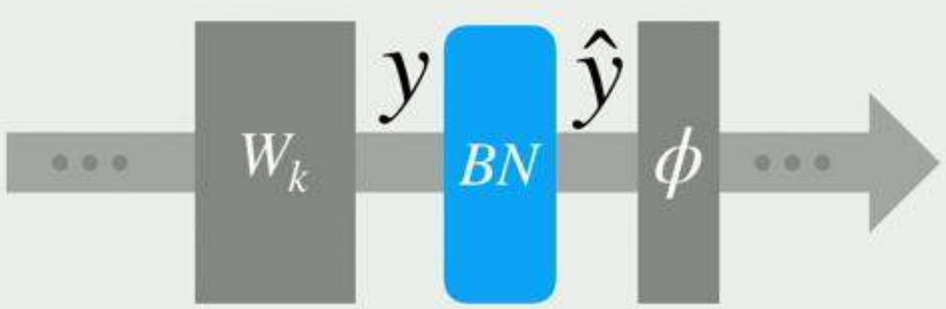
Batch Statistics

$$\hat{y} = \frac{y - \hat{\mu}}{\hat{\sigma} + \epsilon},$$

where $\hat{\mu} = \frac{1}{B}\sum_{i=1}^{B} y_i$ $\hat{\sigma}^2 = \frac{1}{B}\sum_{i=1}^{B} (y_i - \hat{\mu})^2$

Batch Normalization (BatchNorm)

[Ioffe & Szegedy, 2015]



Make distribution "close" to **standard normal**

Learnable

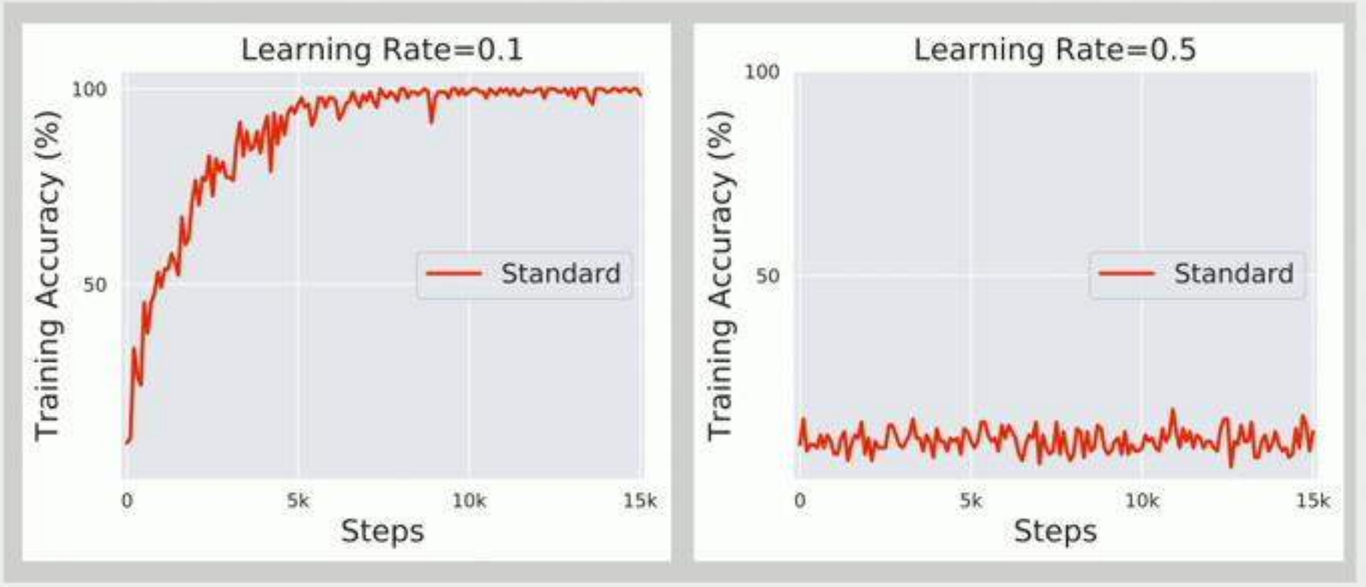
$$\hat{y} = \gamma \frac{y - \hat{\mu}}{\hat{\sigma} + \epsilon} + \beta,$$

where $\hat{\mu} = \frac{1}{B}\sum_{i=1}^{B} y_i$ $\hat{\sigma}^2 = \frac{1}{B}\sum_{i=1}^{B} (y_i - \hat{\mu})^2$

Why do we use BatchNorm?

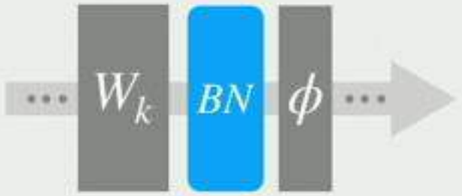
BatchNorm's Role in Optimization

Network **without** BatchNorm

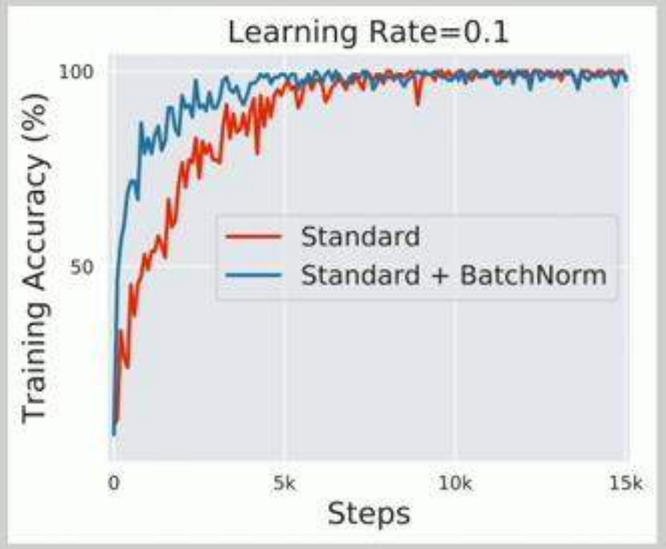


[Simoyan and Zisserman, 2014] [Krizhevsky and Hinton, 2009]

BatchNorm's Role in Optimization



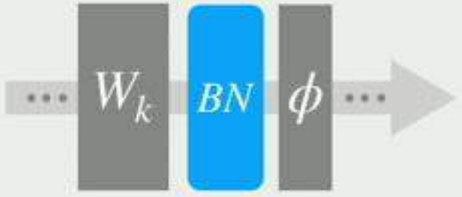
Network **with** BatchNorm



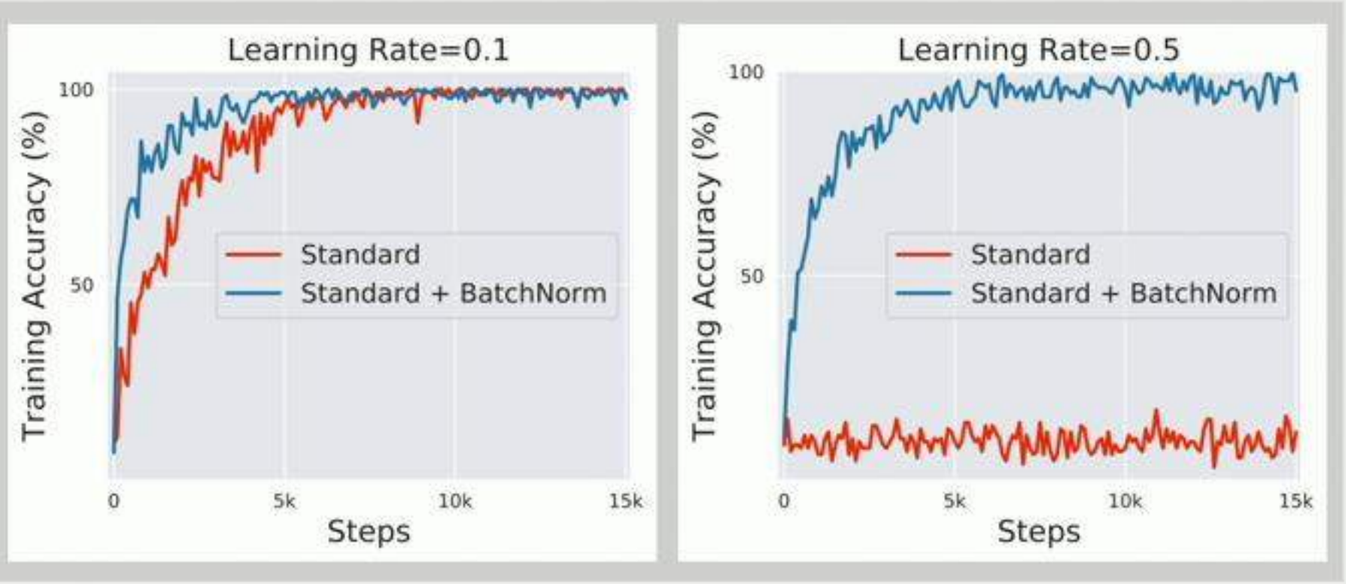
Faster Convergence

Robust to Hyperparameters

BatchNorm's Role in Optimization



Network **with** BatchNorm



Faster Convergence

Robust to Hyperparameters

One of the most influential techniques in DNN training

Default in almost all standard architectures

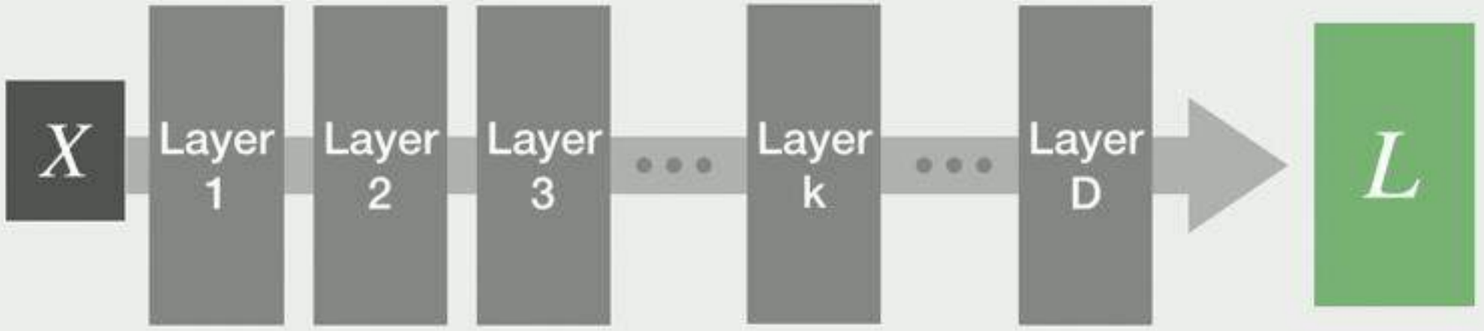
How does BatchNorm help?

How does BatchNorm help?

The story so far

The Internal Covariate Shift Hypothesis

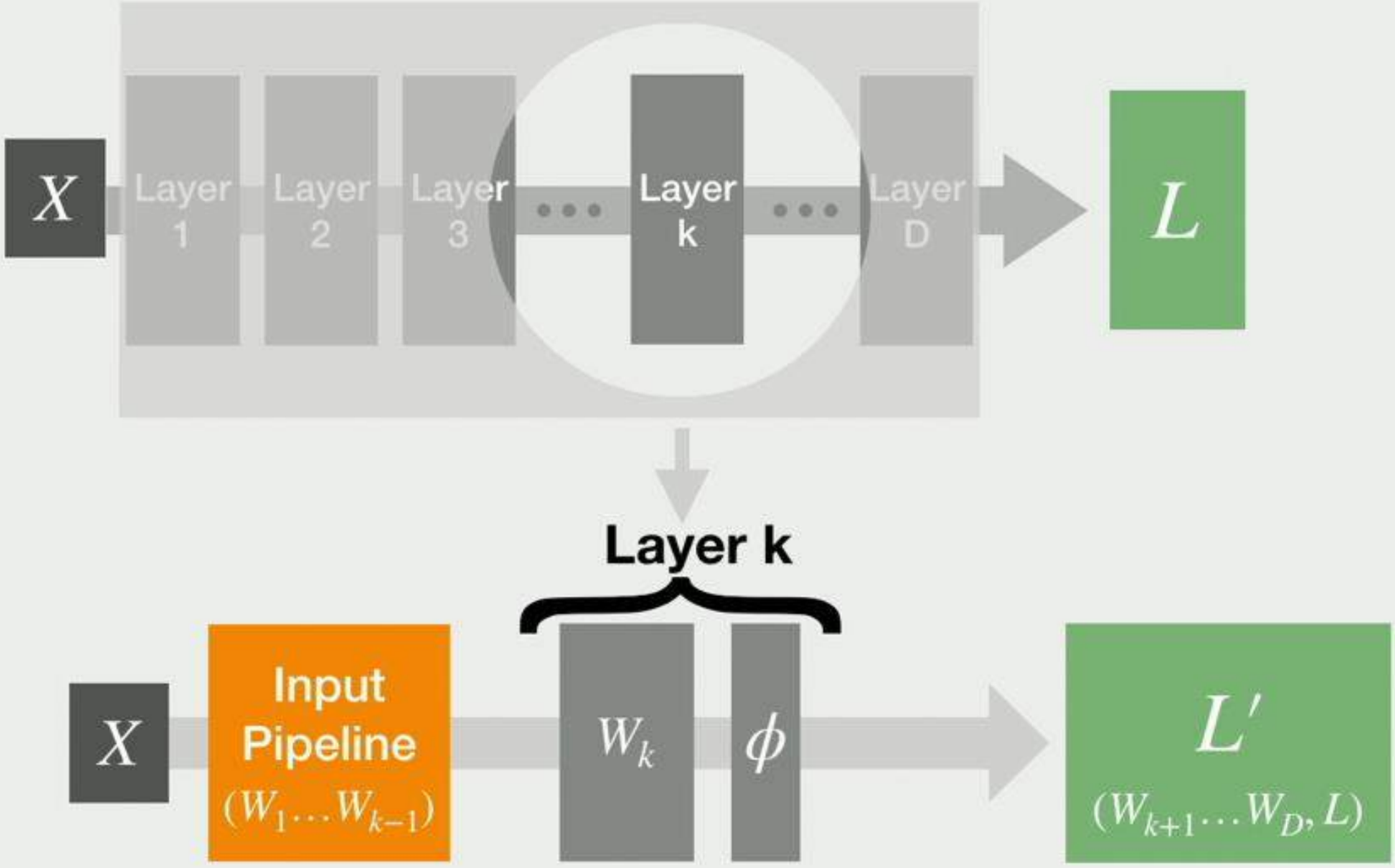
[Ioffe & Szegedy, 2015]



Training ≈ solving an optimization problem at **each** layer

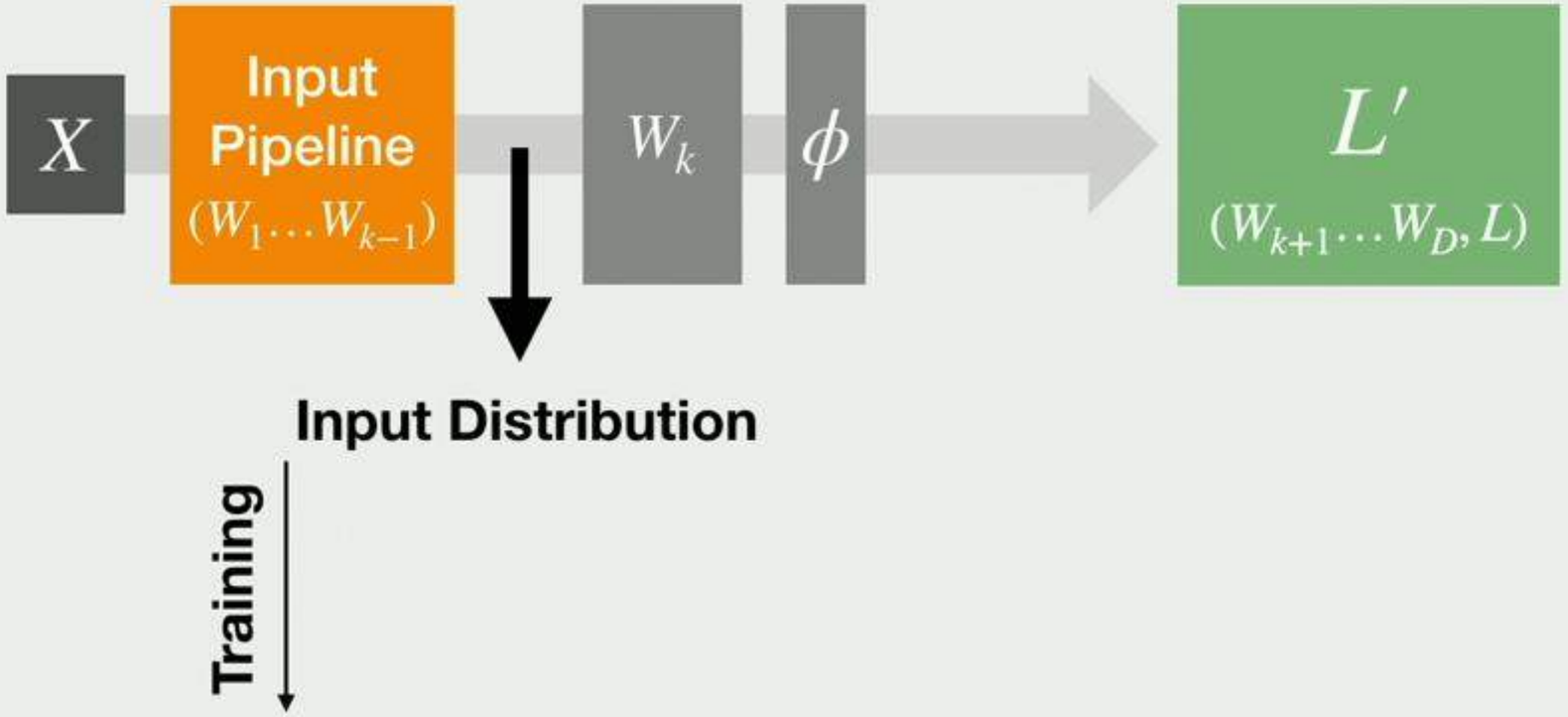
The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]



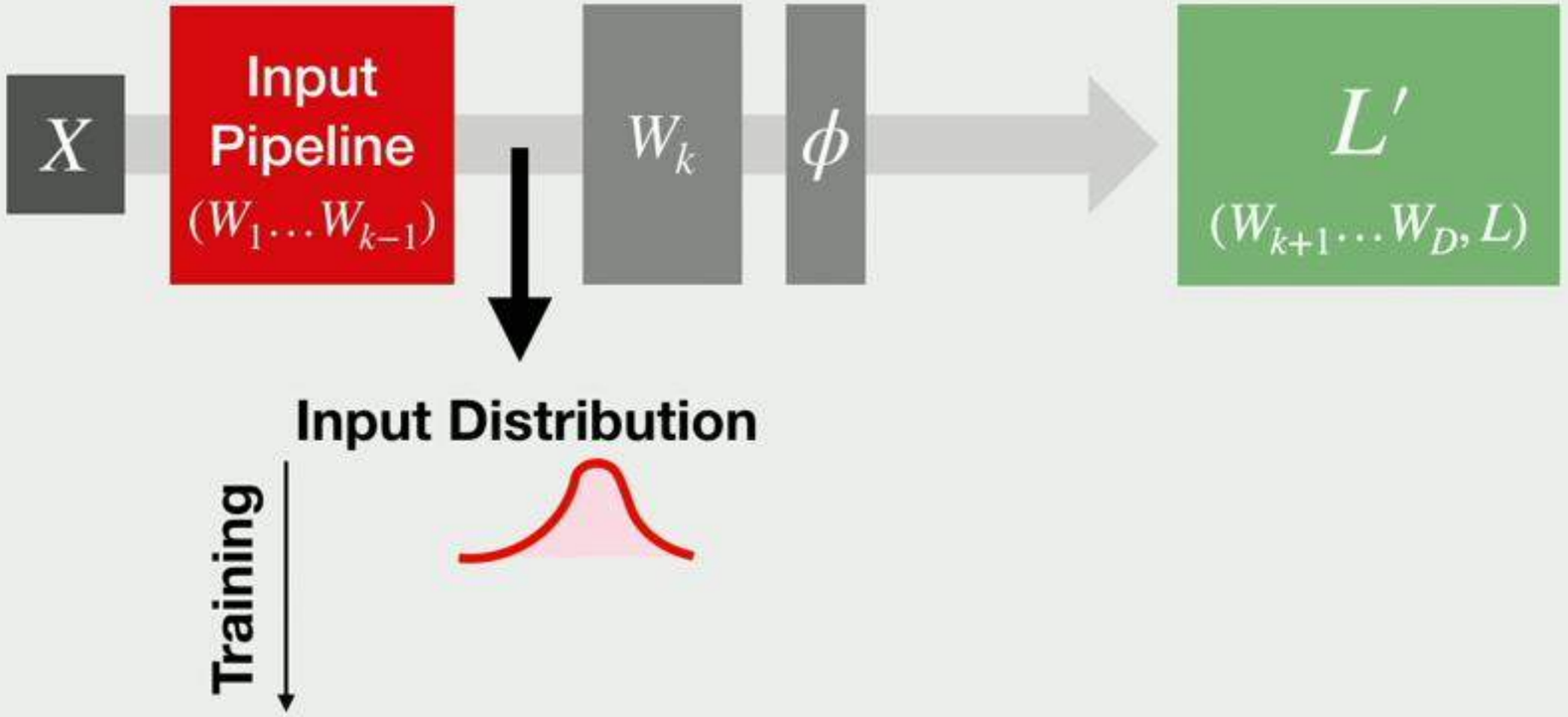
The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]



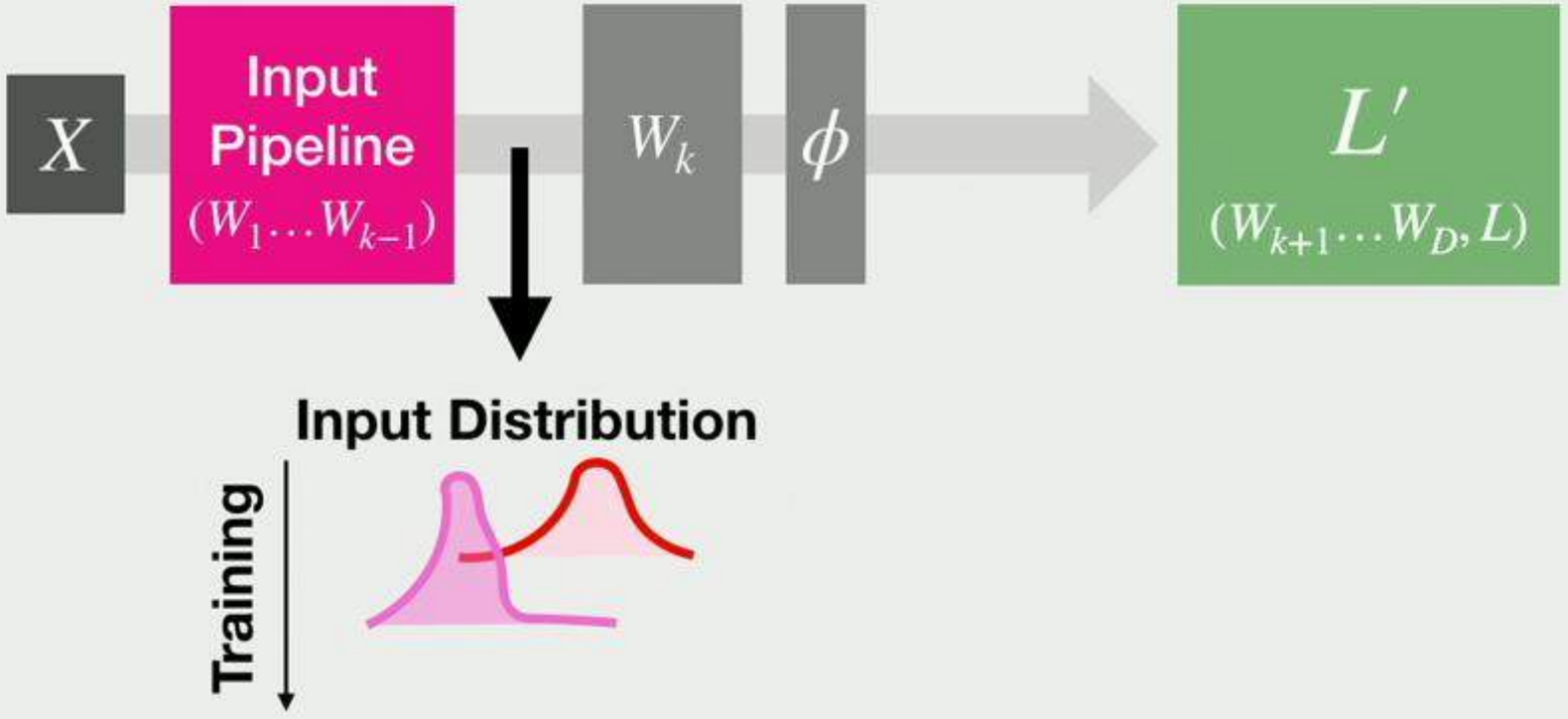
The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]



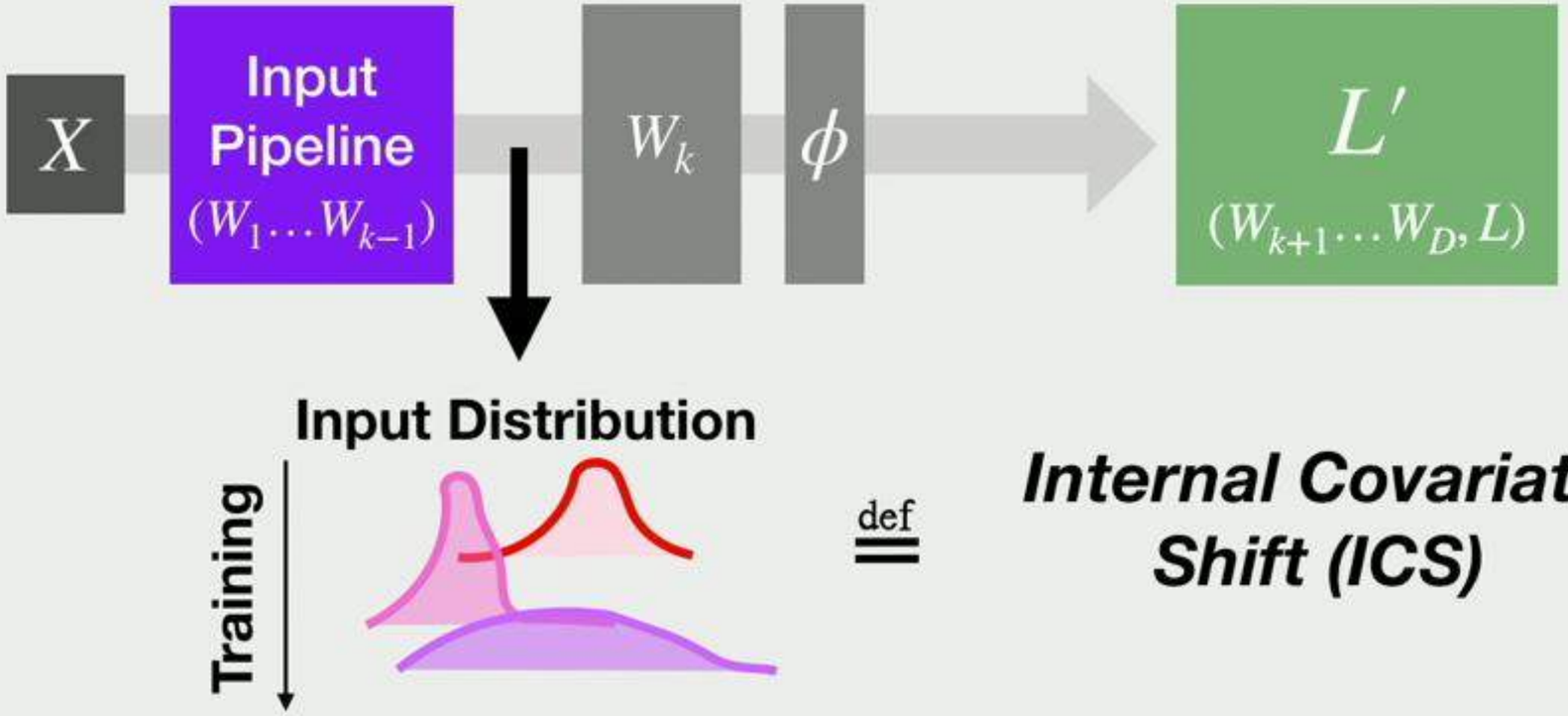
The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]



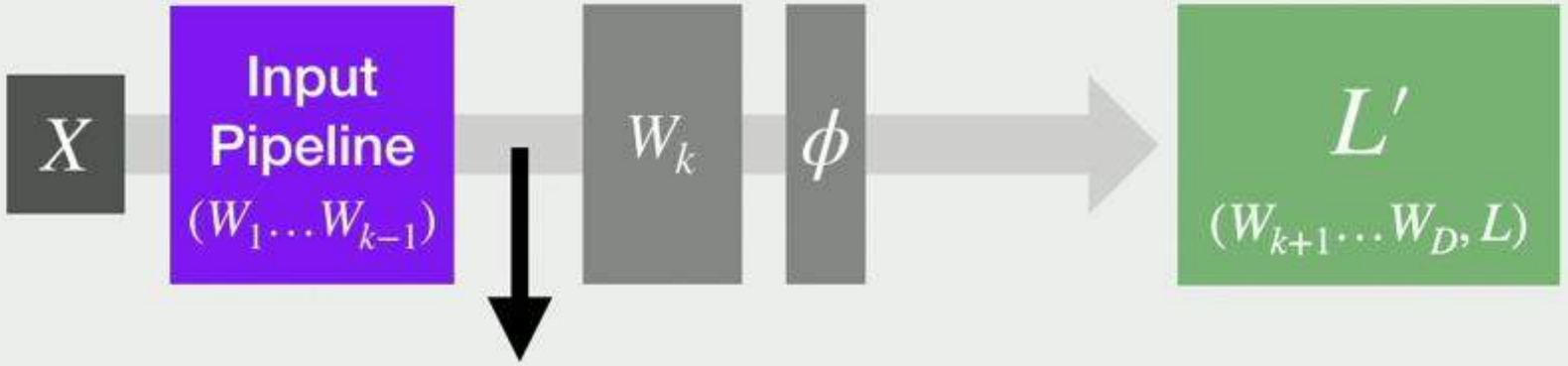
The Internal Covariate Shift Hypothesis

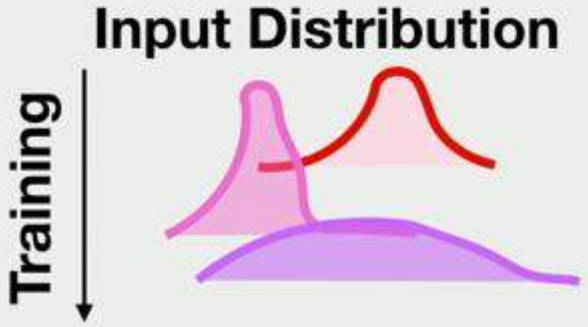
[Ioffe & Szegedy, 2015]



The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]





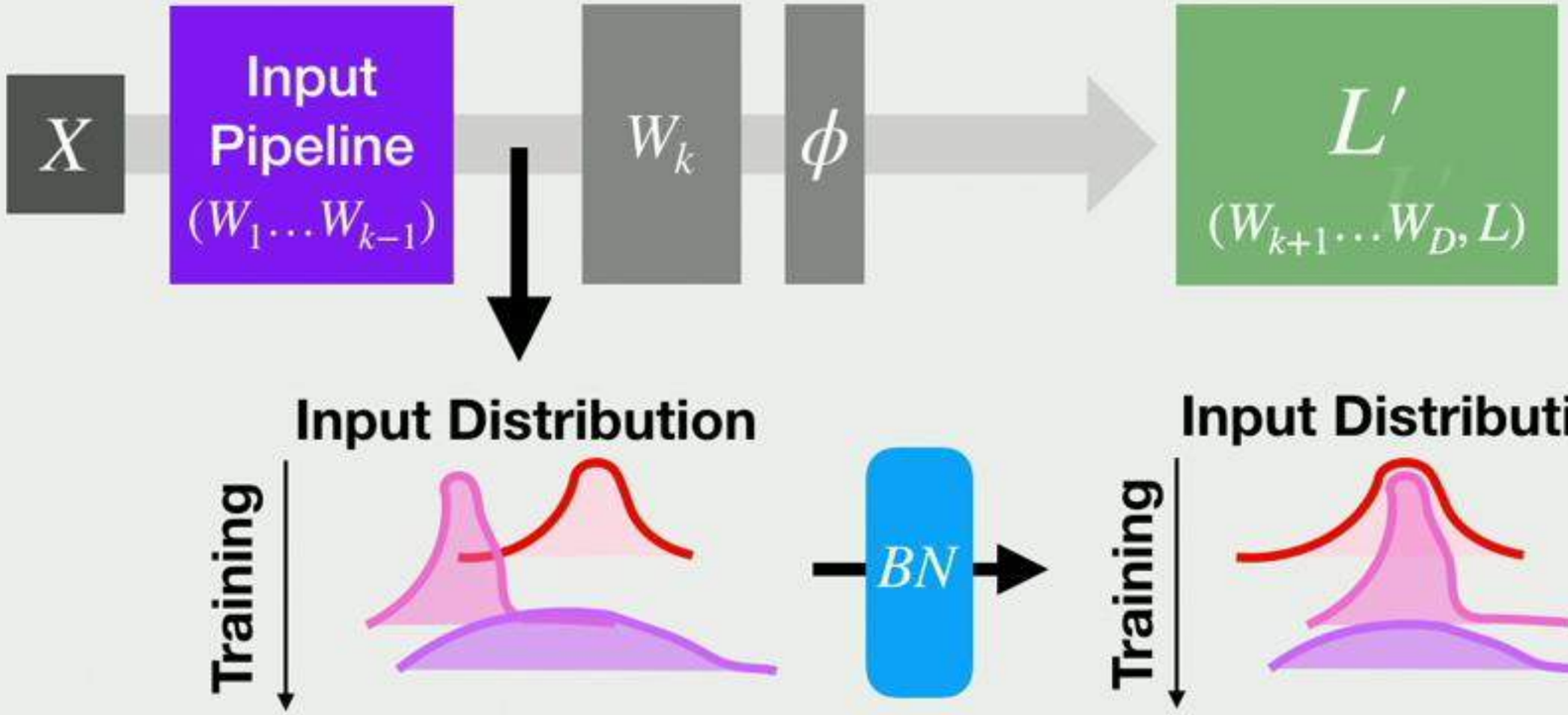
$\overset{\text{def}}{=}$

Internal Covariate Shift (ICS)

[IS15]: **Layers need to continually adapt.**

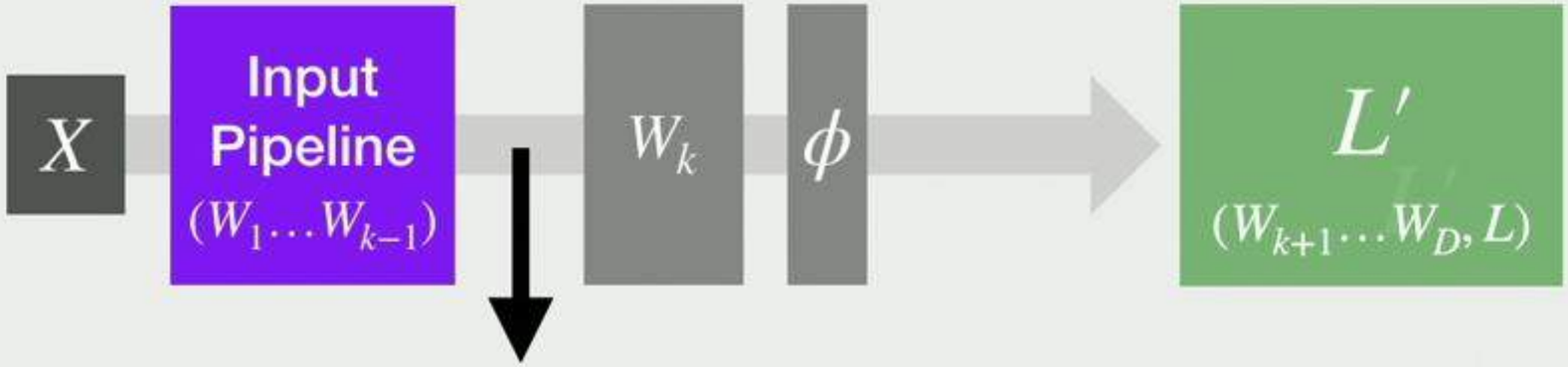
The Internal Covariate Shift Hypothesis

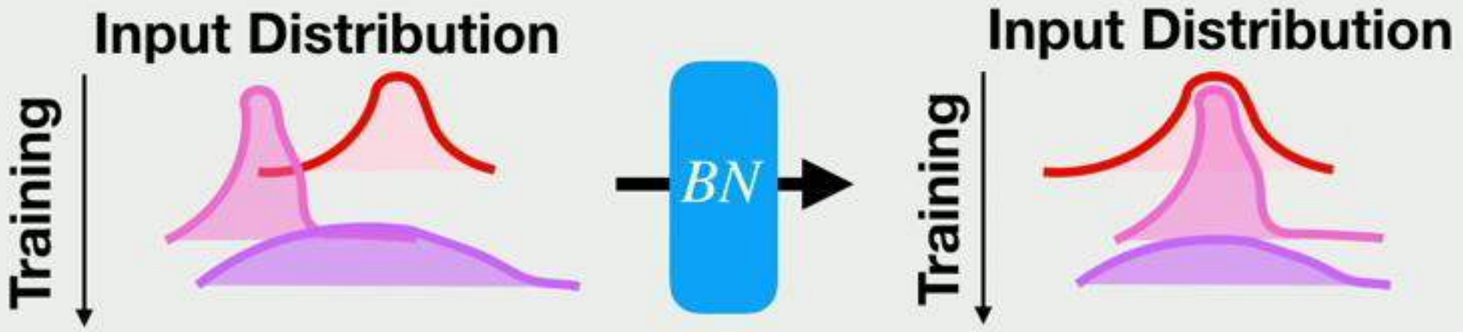
[Ioffe & Szegedy, 2015]



The Internal Covariate Shift Hypothesis

[Ioffe & Szegedy, 2015]

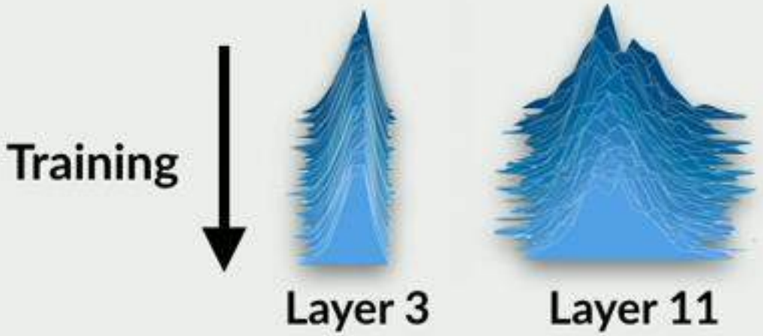


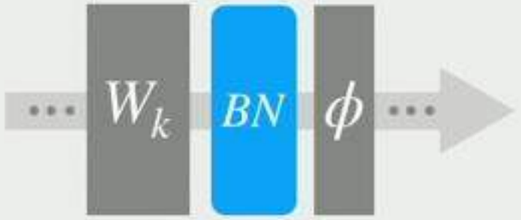


[IS15]: **Reducing internal covariate shift** is the key to BN's **success**

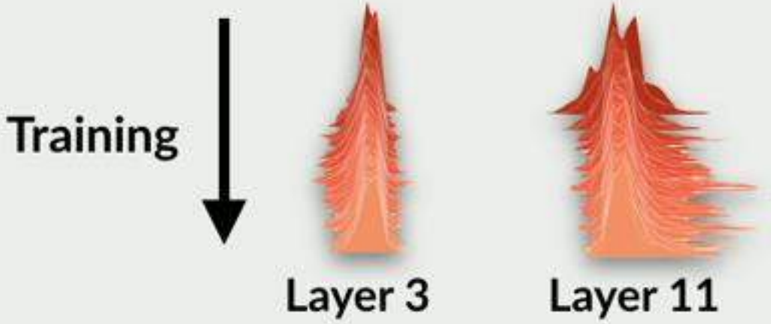
A Closer Look at Internal Covariate Shift

Network **with** BatchNorm:



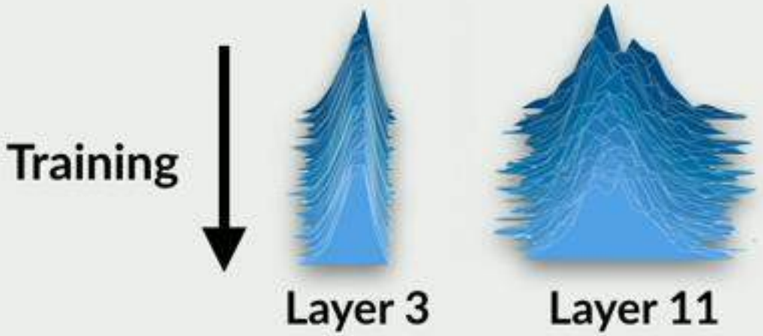


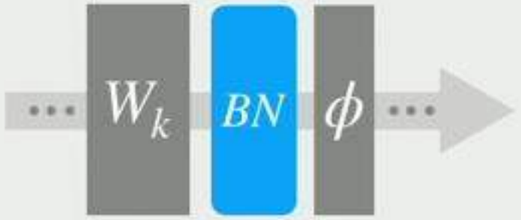
Network **without** BatchNorm:



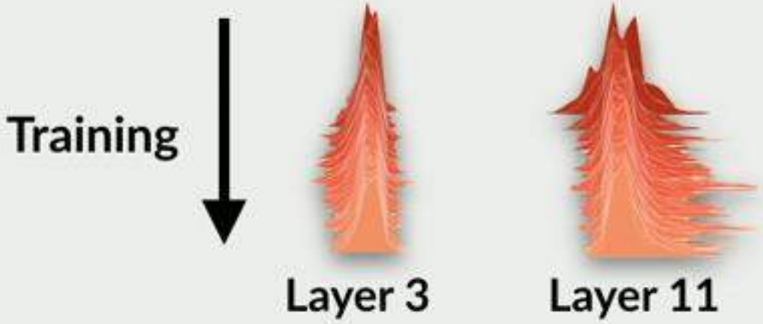
A Closer Look at Internal Covariate Shift

Network **with** BatchNorm:





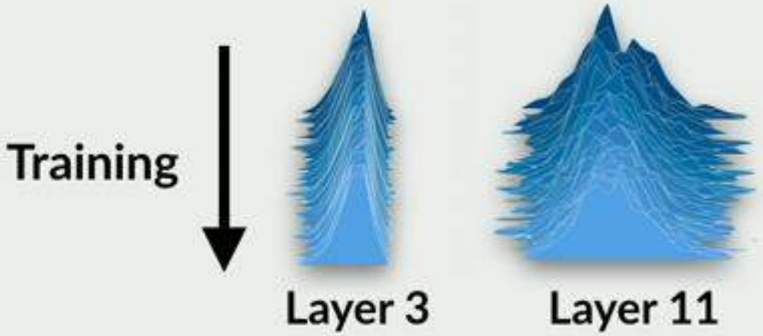
Network **without** BatchNorm:



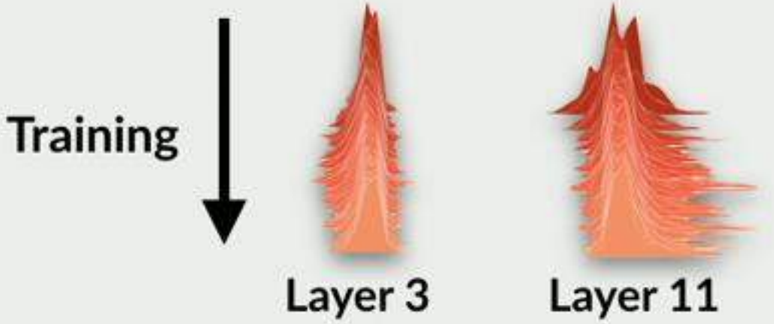
No difference in stability ...

A Closer Look at Internal Covariate Shift

Network **with** BatchNorm:



Network **without** BatchNorm:



... despite large difference in performance

The Impact of Internal Covariate Shift

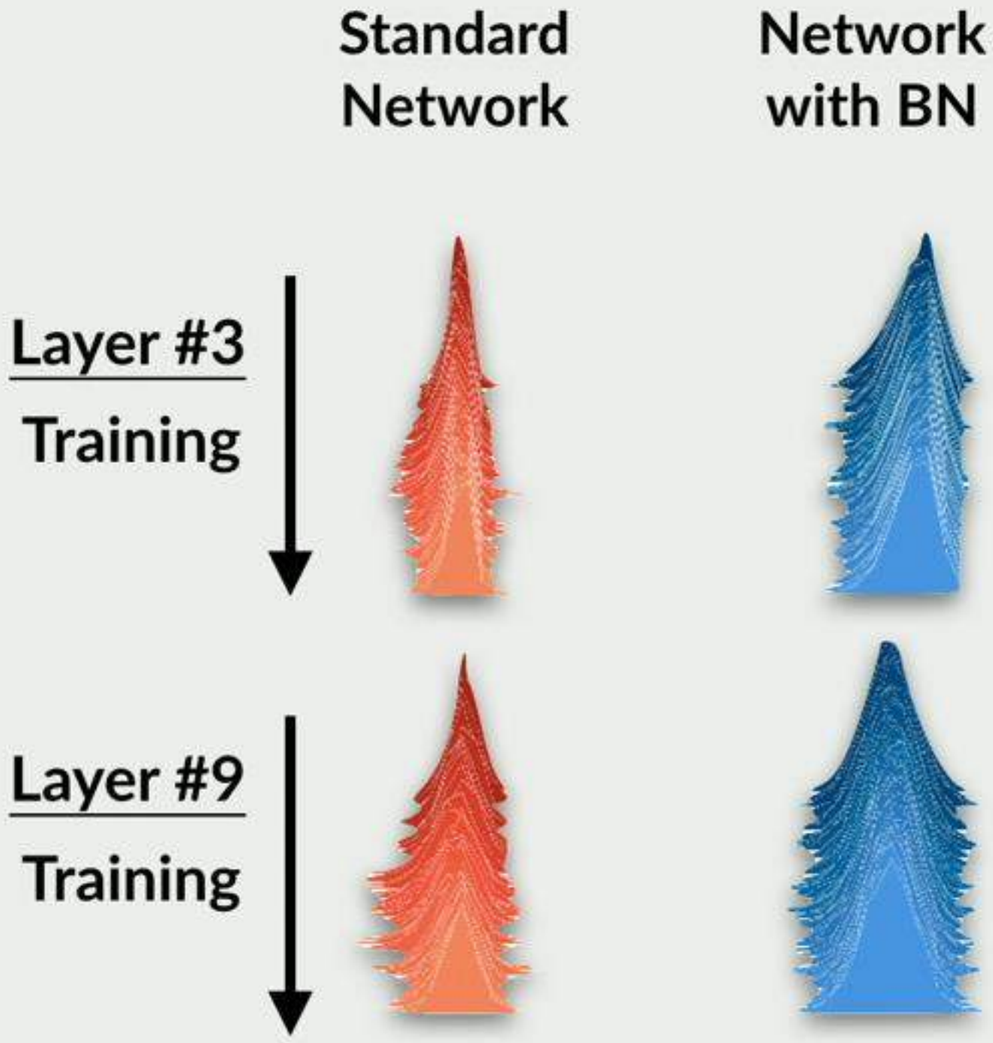
*What happens if we **increase** internal covariate shift ?*

Network **with** BN

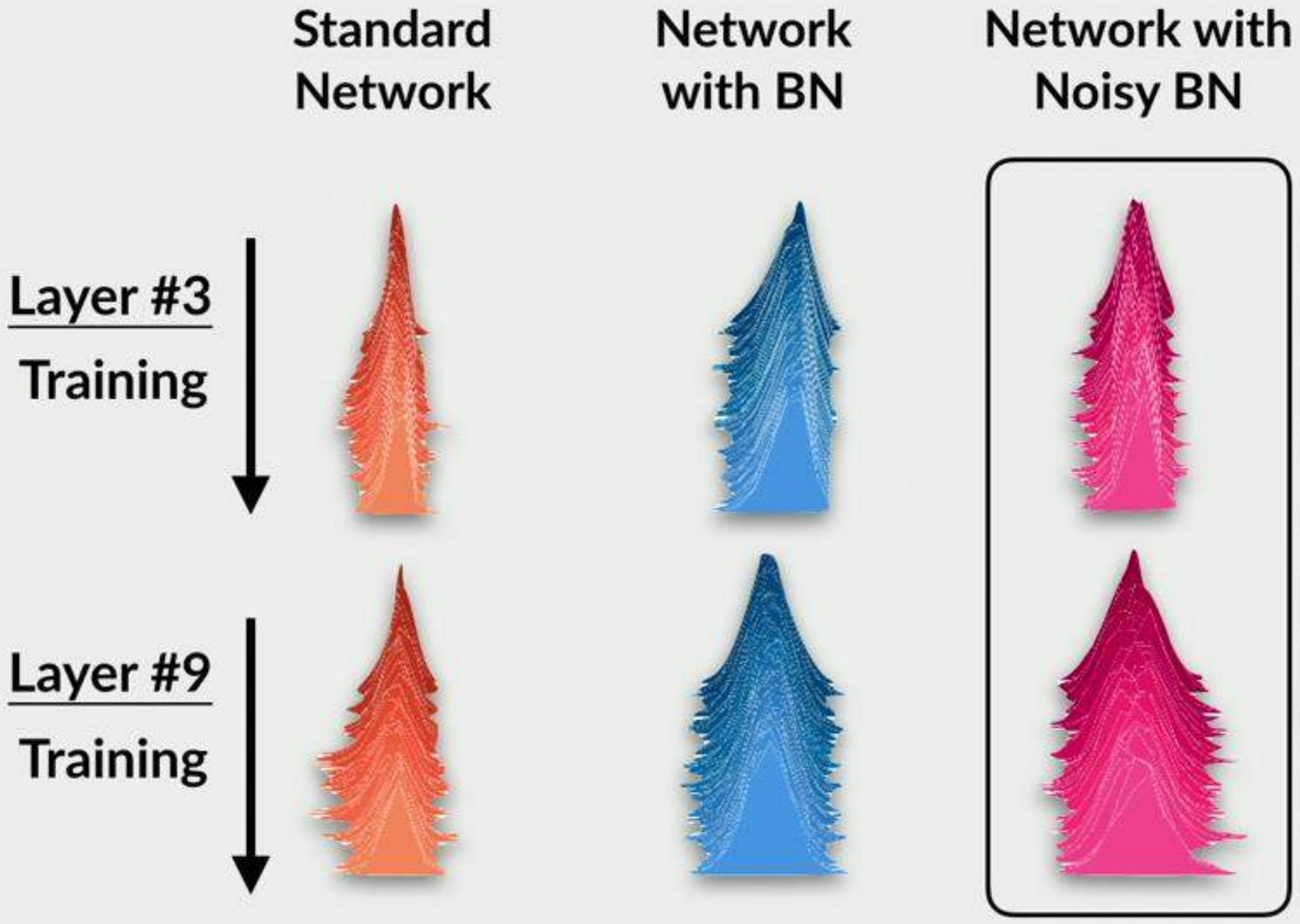
Non-stationary noise
(non-zero mean and variance)

Network with **"Noisy"** BN

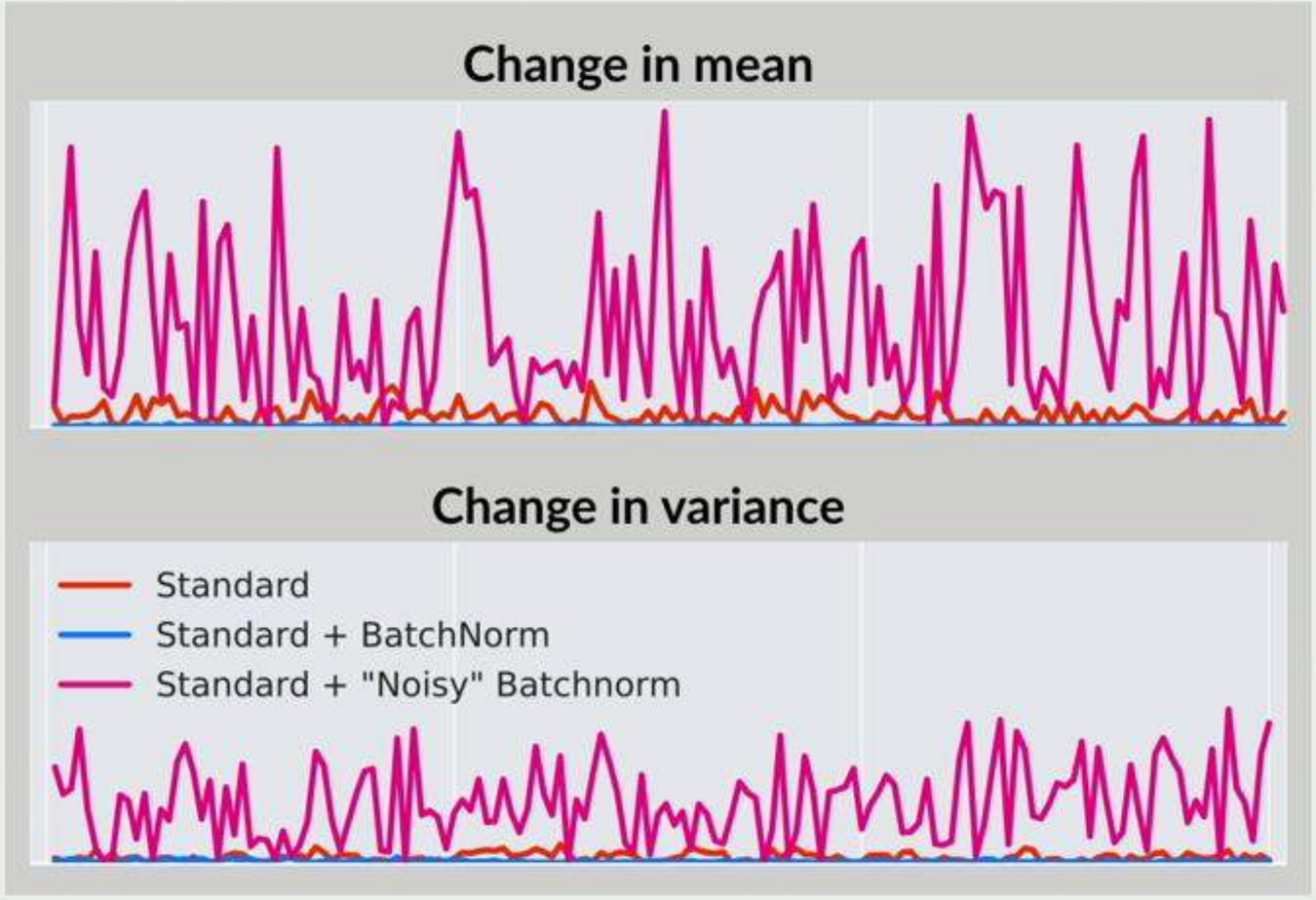
“Noisy” BatchNorm Activations



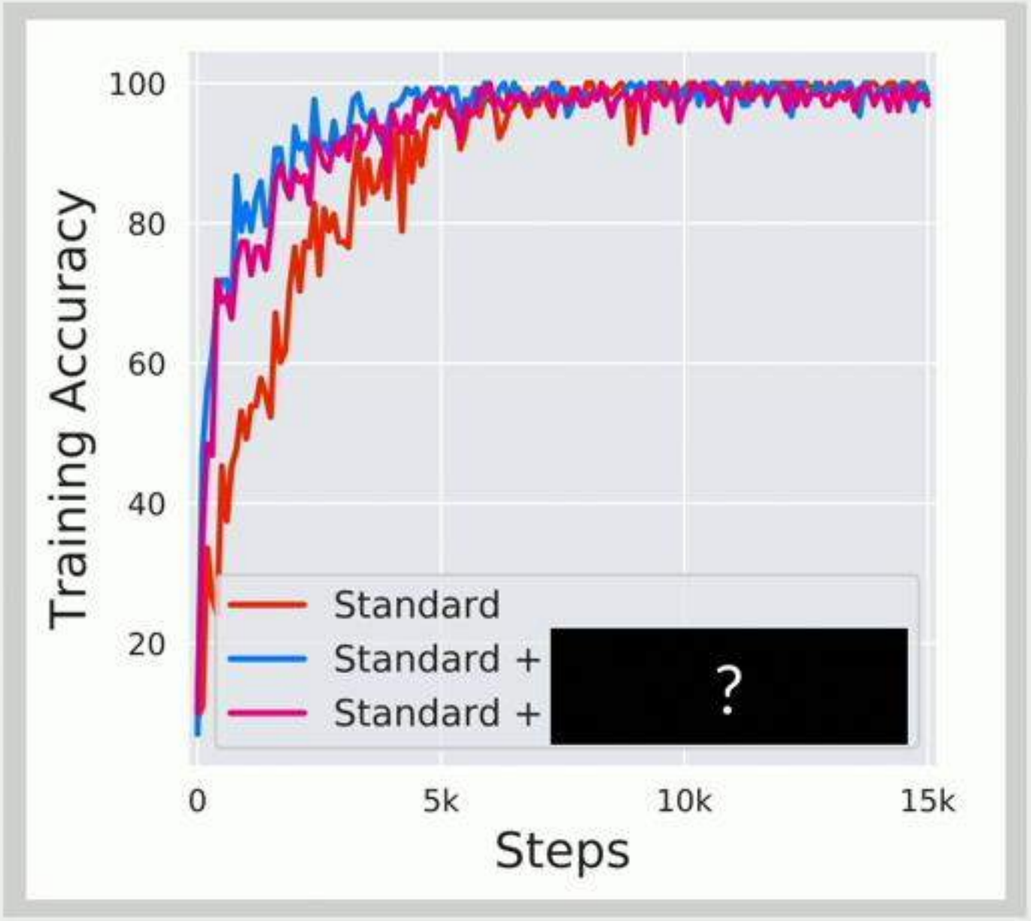
“Noisy” BatchNorm Activations



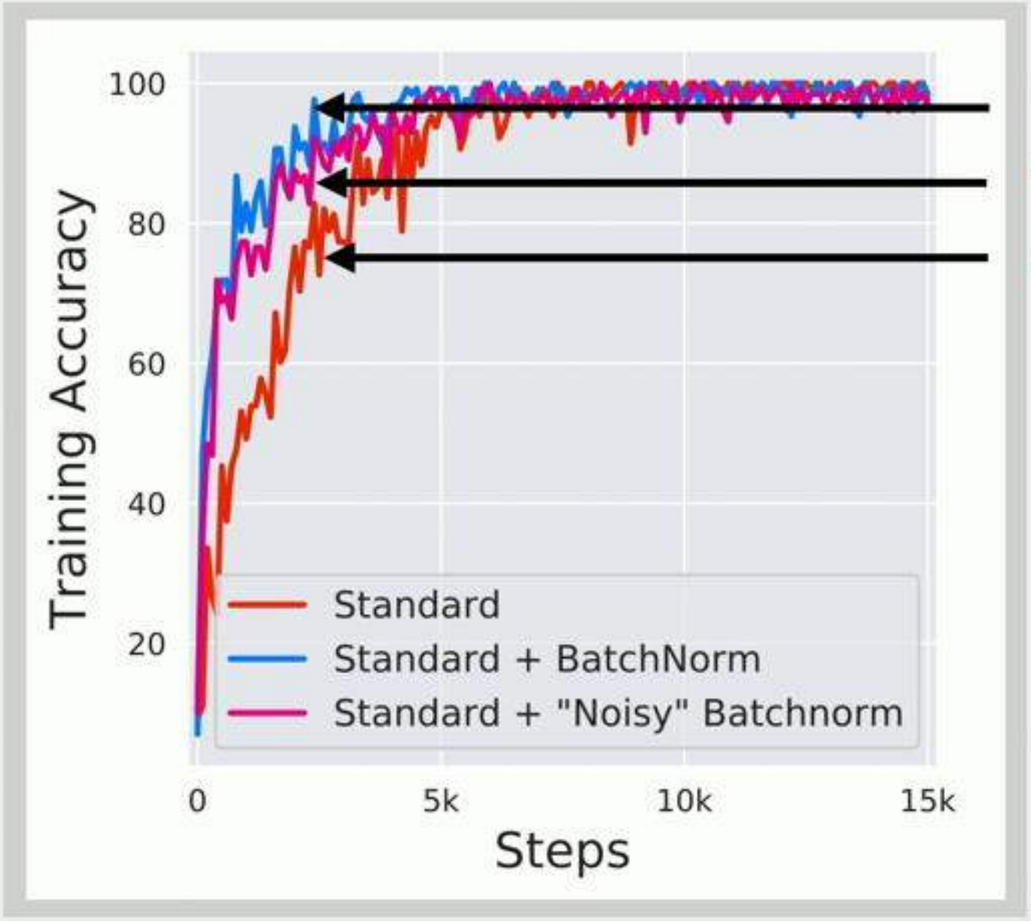
“Noisy” BatchNorm Activations



“Noisy” BatchNorm Activations



“Noisy” BatchNorm Activations

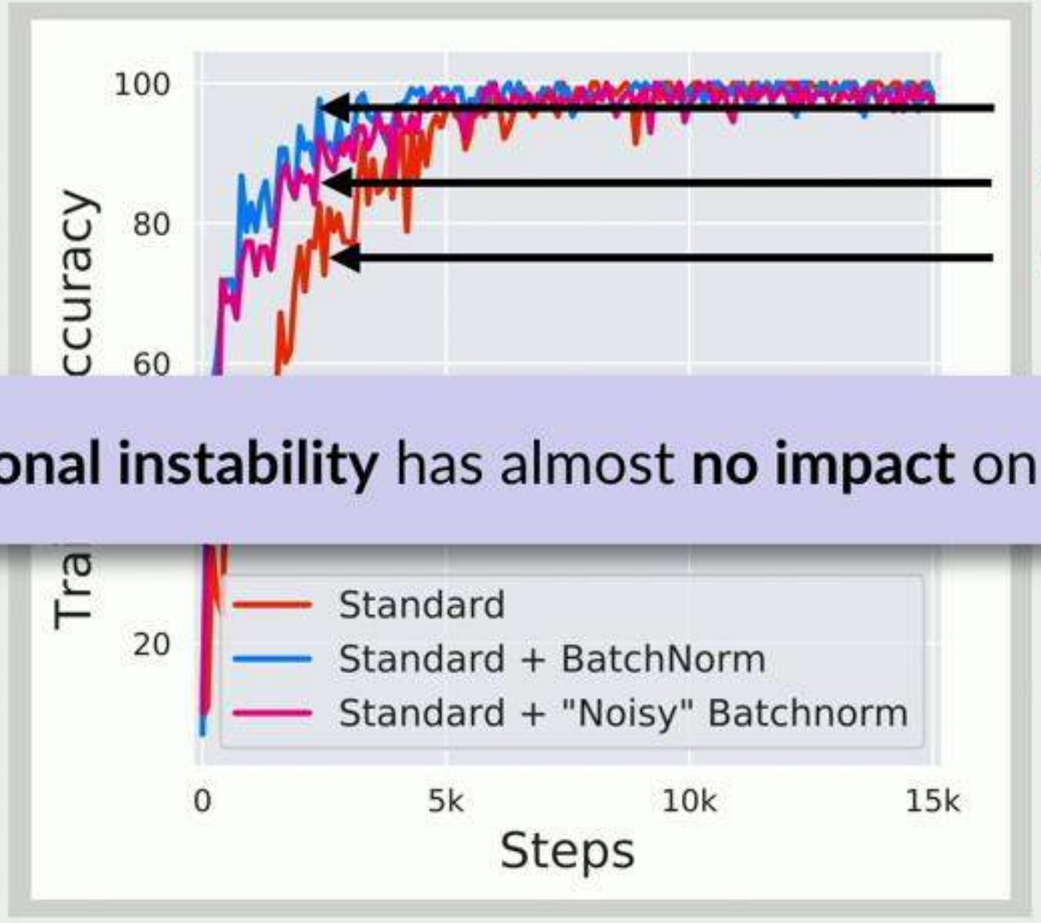


BatchNorm

“Noisy” BatchNorm

Standard

“Noisy” BatchNorm Activations



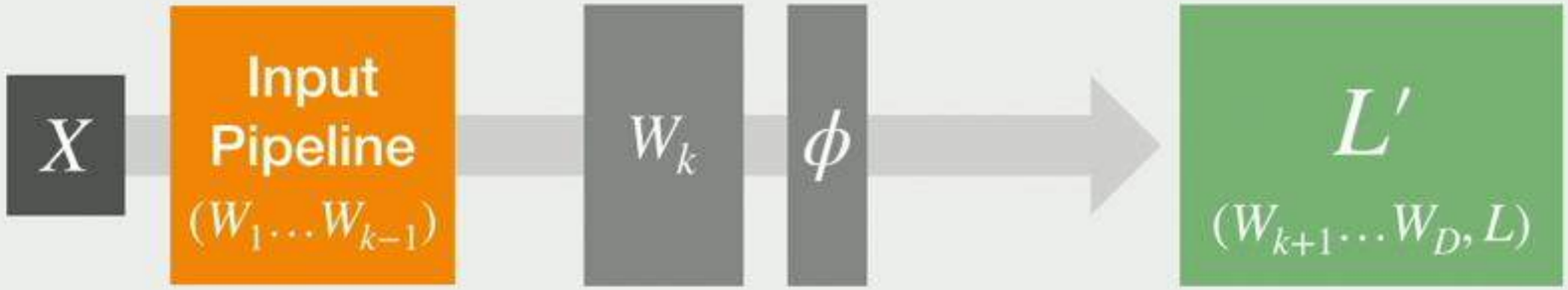
BatchNorm

“Noisy” BatchNorm

Standard

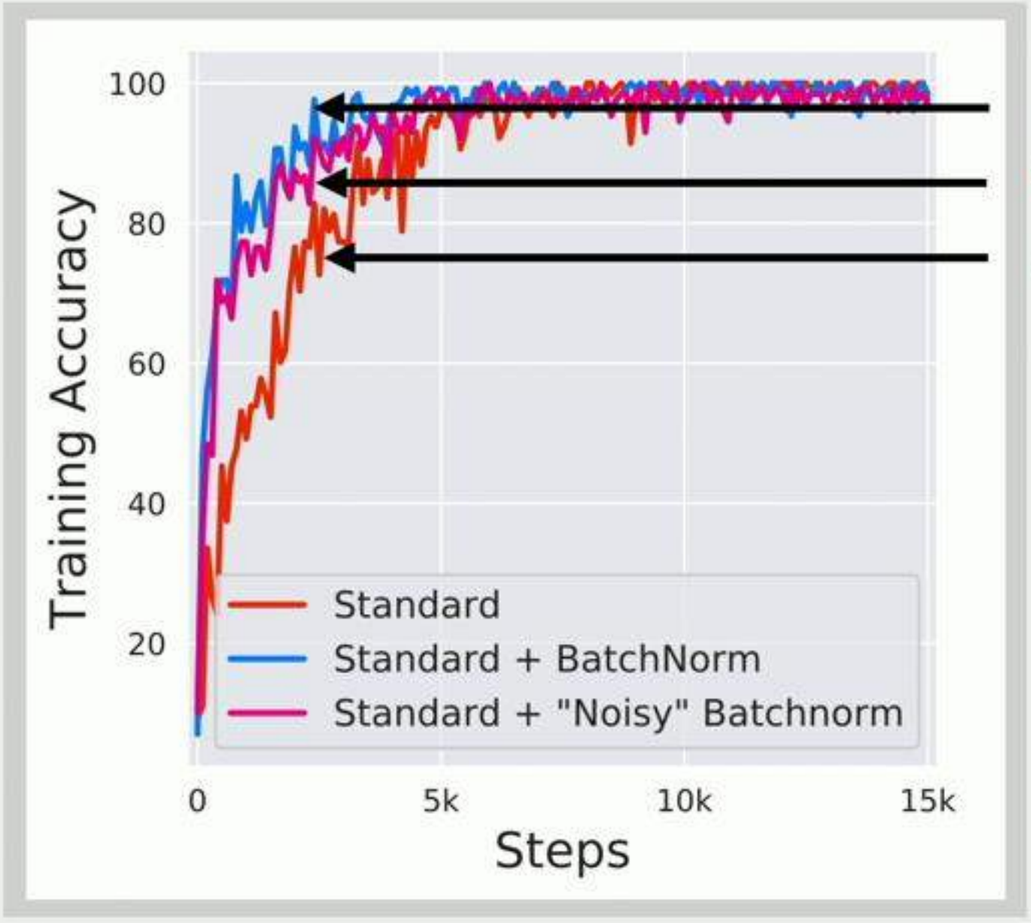
Distributional instability has almost **no impact** on performance!

Different View of Internal Covariate Shift



We train our models with first-order methods

“Noisy” BatchNorm Activations

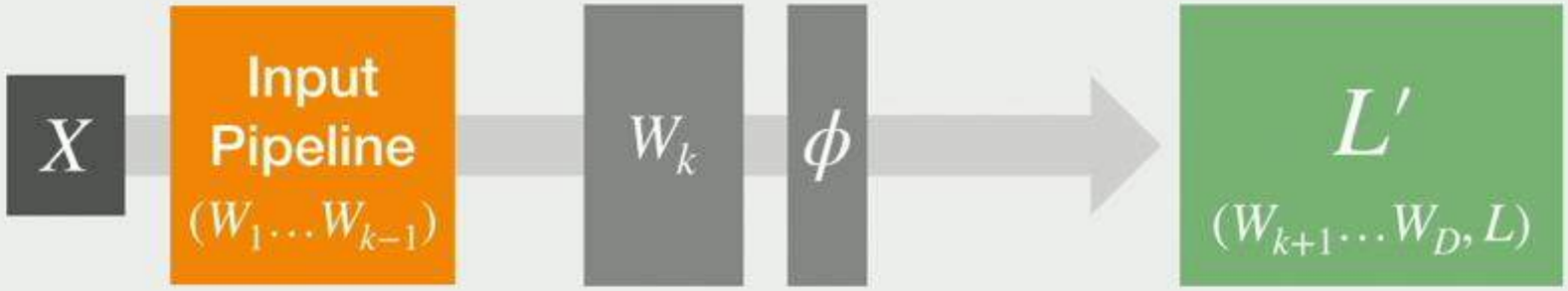


BatchNorm

“Noisy” BatchNorm

Standard

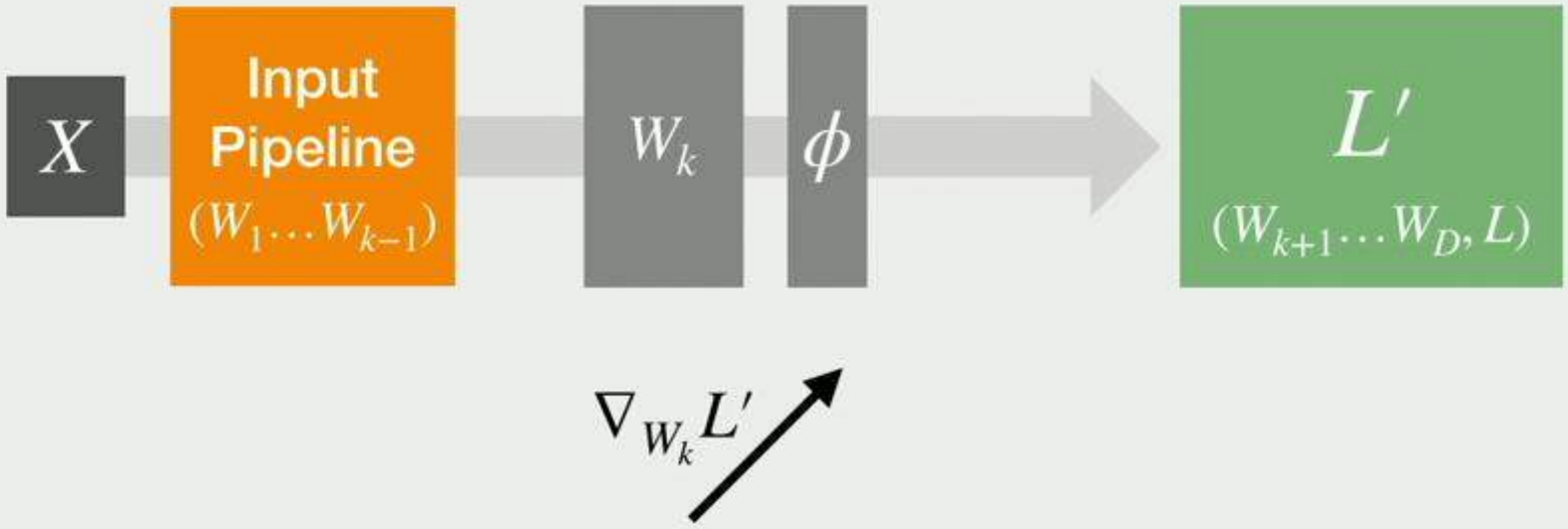
Different View of Internal Covariate Shift



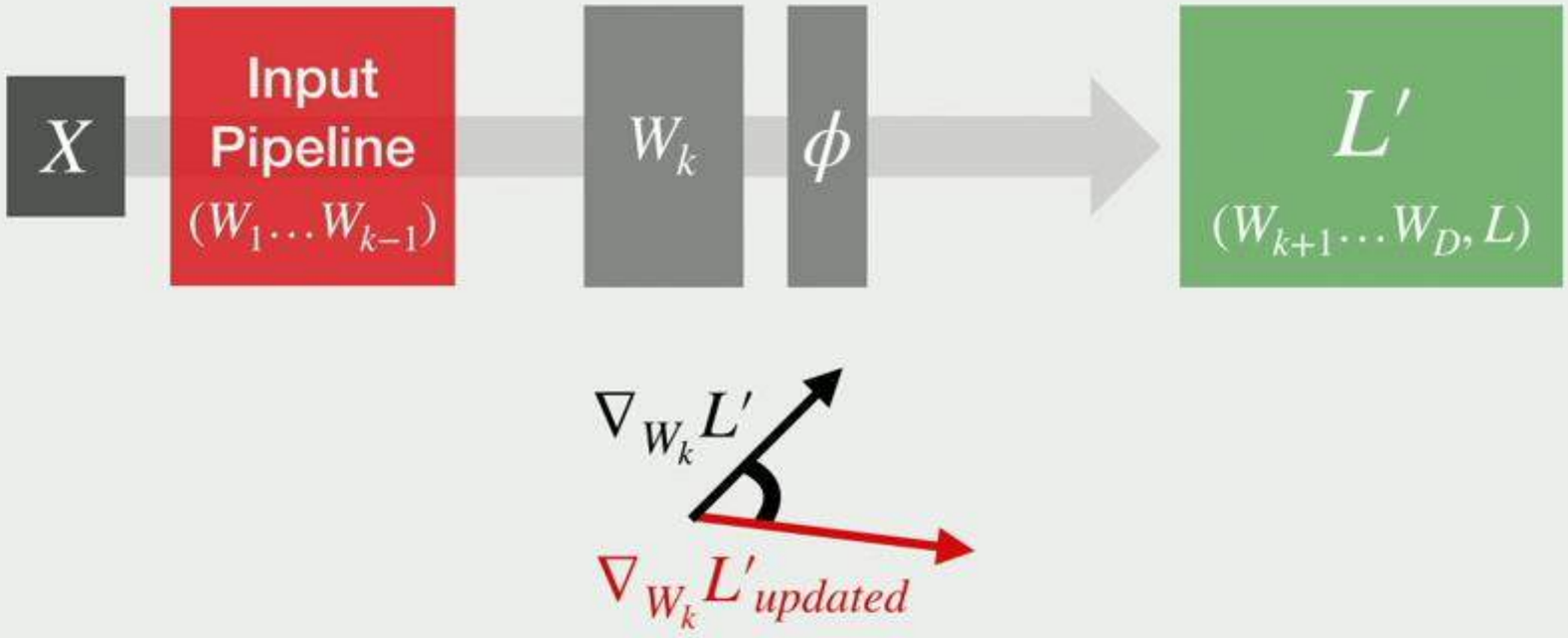
We train our models with first-order methods

*How do updates to previous layers affect the **gradient** for this layer?*

Different View of Internal Covariate Shift

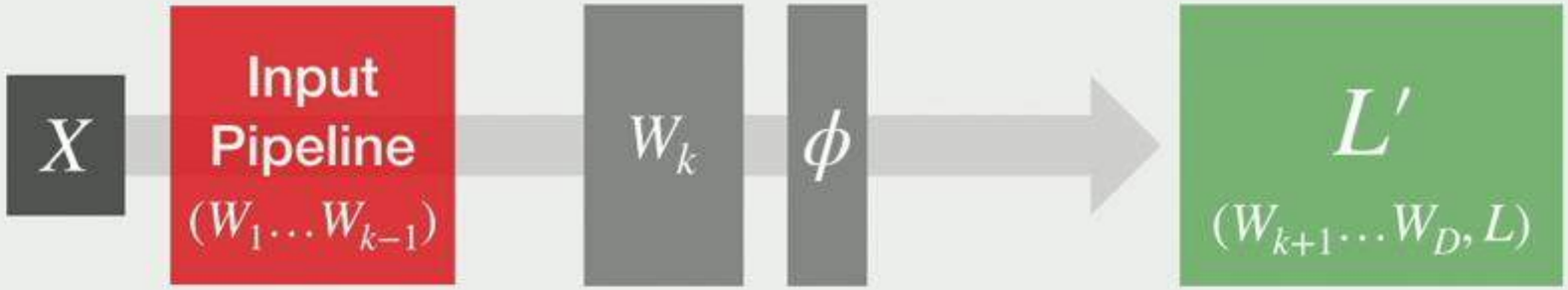


Different View of Internal Covariate Shift



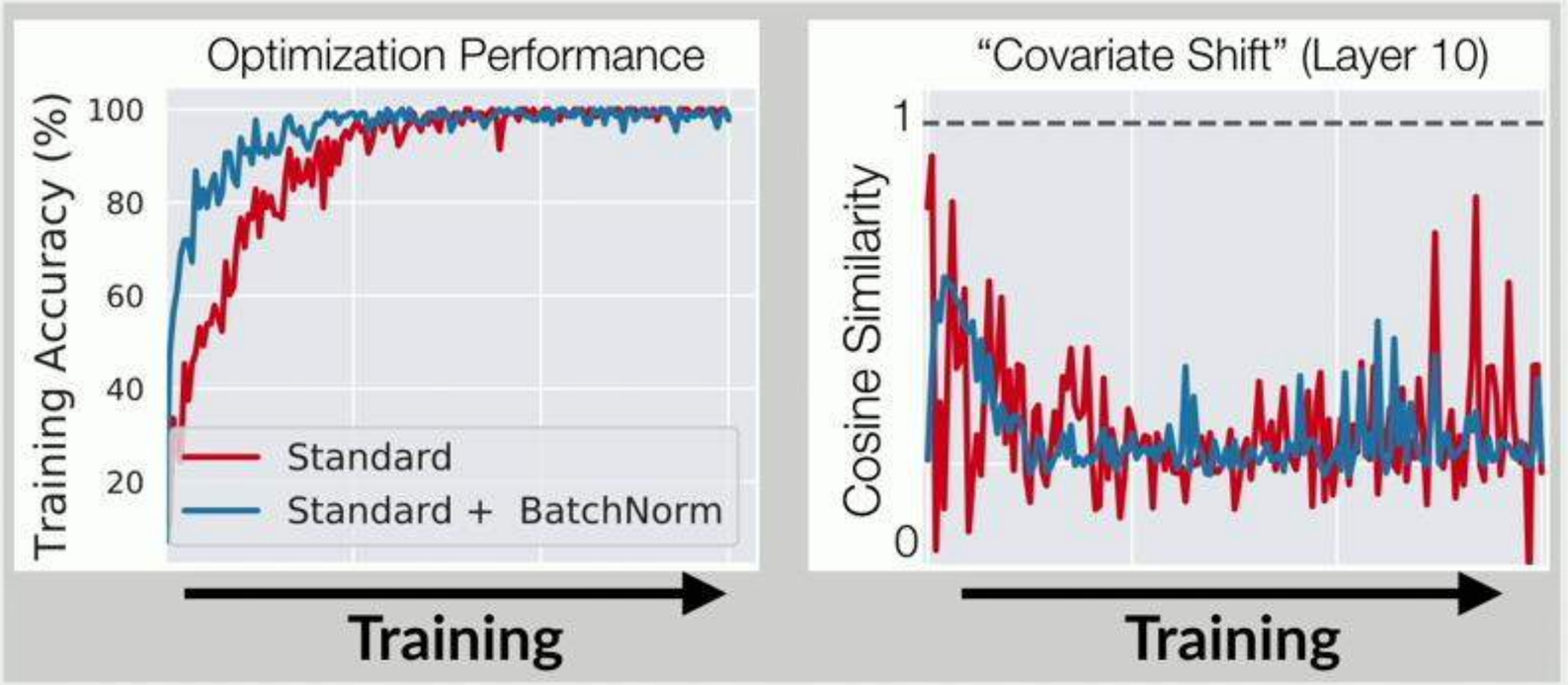
Different View of Internal Covariate Shift

Change in gradients ← change in optimization problem

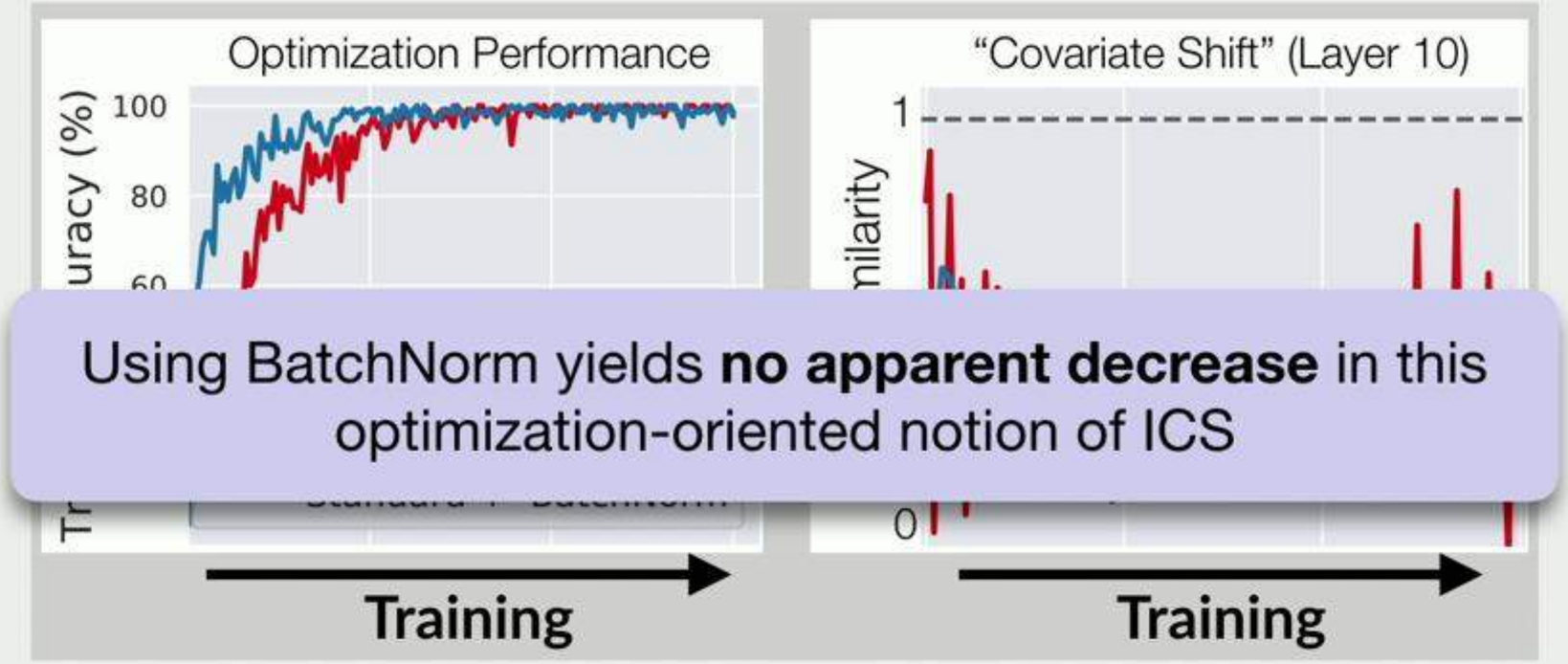


Does BatchNorm increase this notion of stability?

Different View of Internal Covariate Shift

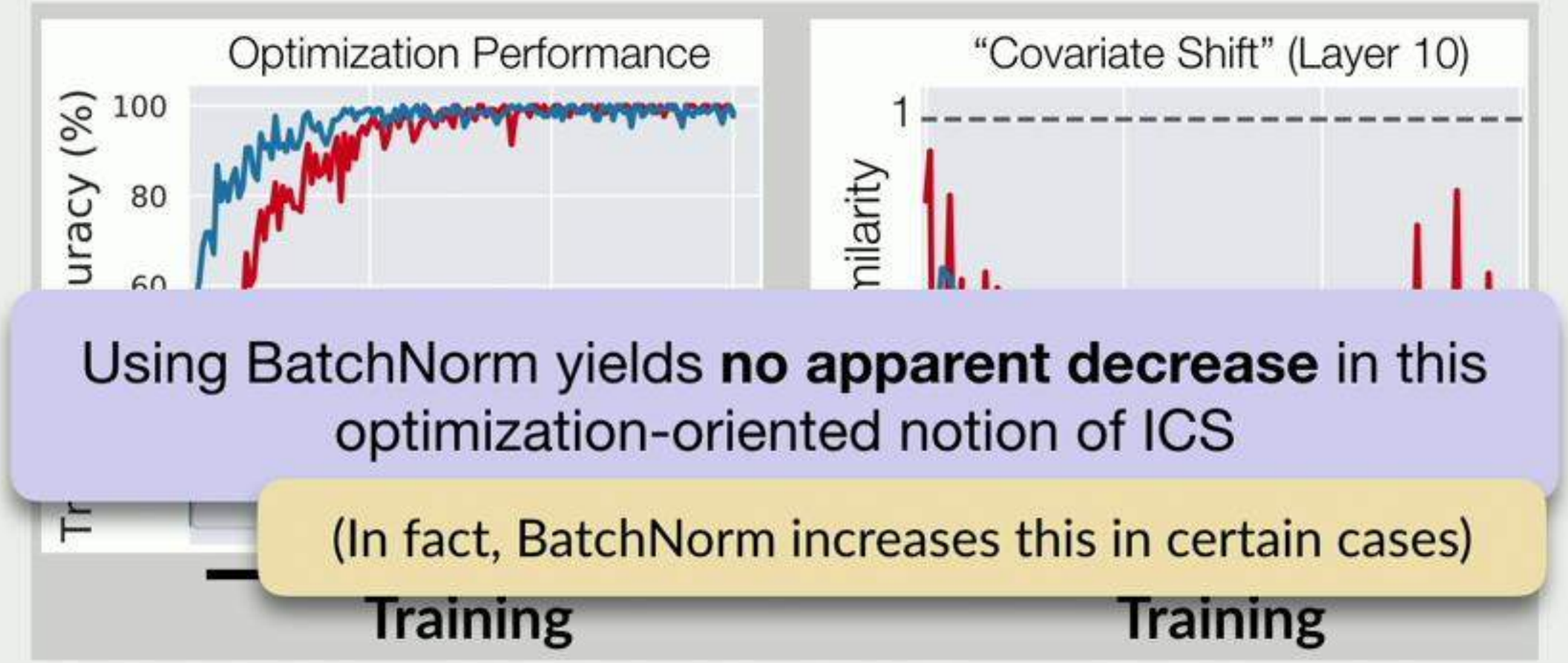


Different View of Internal Covariate Shift



Using BatchNorm yields **no apparent decrease** in this optimization-oriented notion of ICS

Different View of Internal Covariate Shift



Using BatchNorm yields **no apparent decrease** in this optimization-oriented notion of ICS

(In fact, BatchNorm increases this in certain cases)

How does BatchNorm help?

How does BatchNorm help?

So far: Internal covariate shift connection unclear

But BatchNorm is effective: Why?

Back to Optimization Primitives

Recall: We use first-order methods in practice

Back to Optimization Primitives

Recall: We use first-order methods in practice

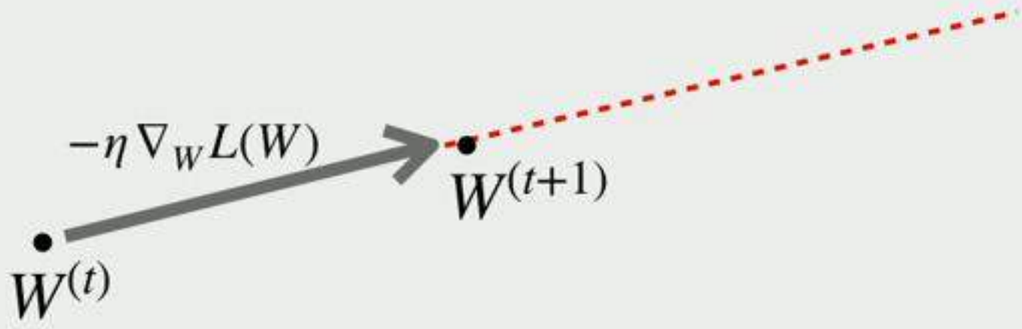
$$\dot{W}^{(t)}$$

Back to Optimization Primitives

Recall: We use first-order methods in practice

Back to Optimization Primitives

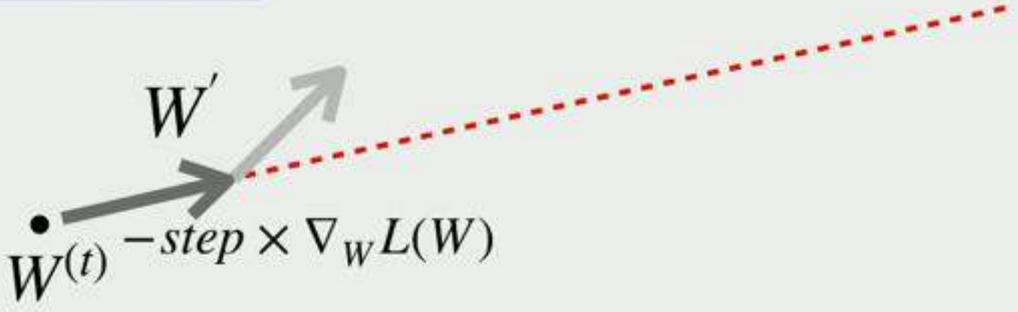
Recall: We use first-order methods in practice

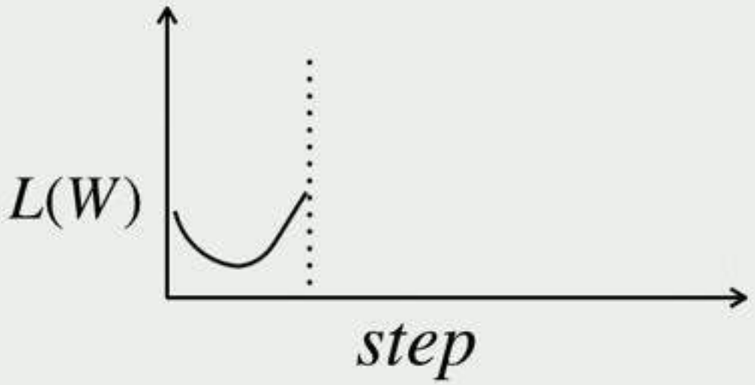


We rely on our loss being *locally* well-behaved

A Simple Experiment

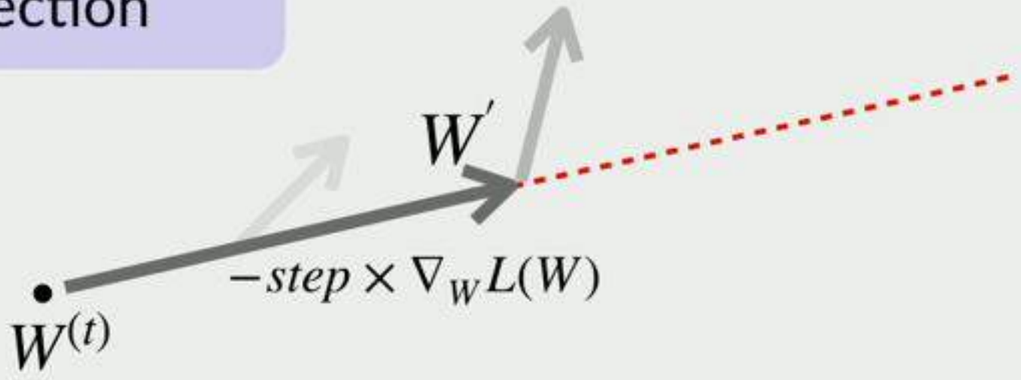
Explore landscape in the gradient direction

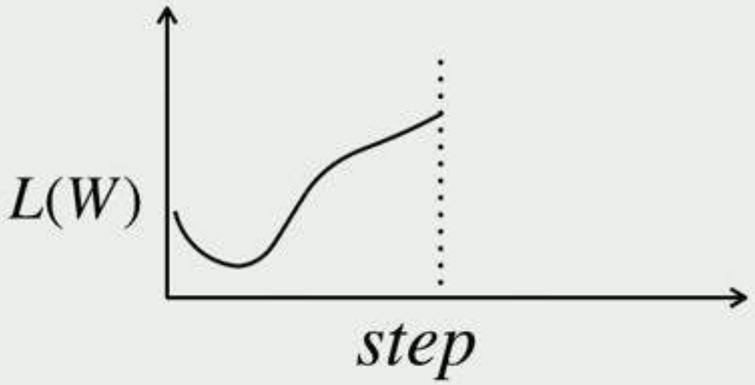




A Simple Experiment

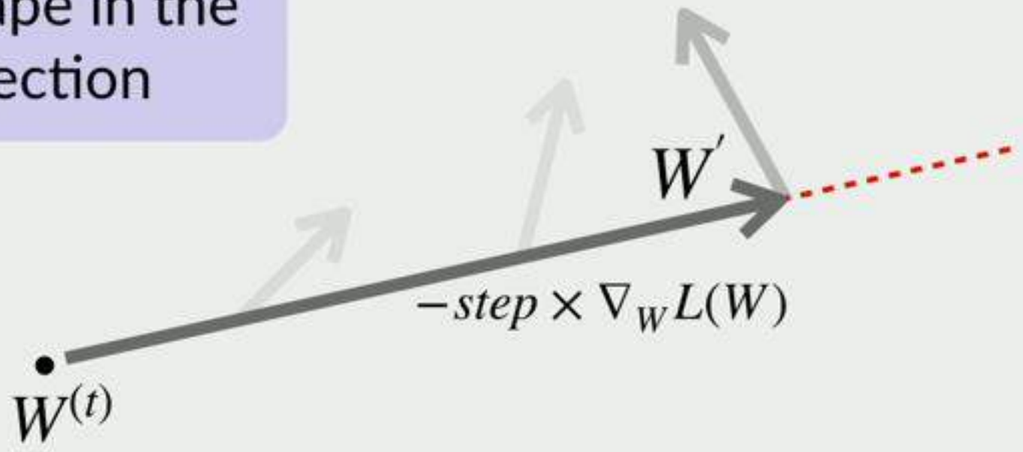
Explore landscape in the gradient direction

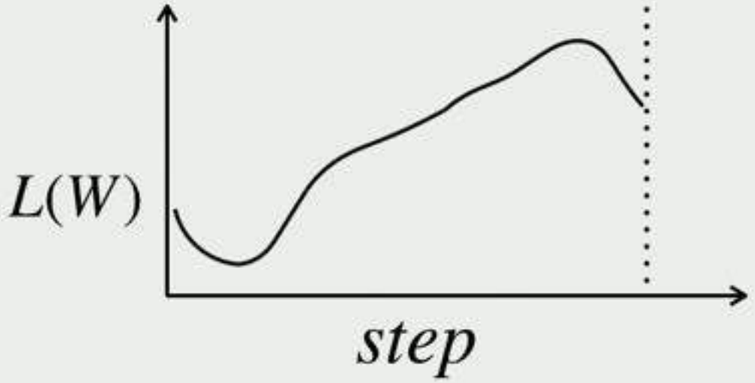




A Simple Experiment

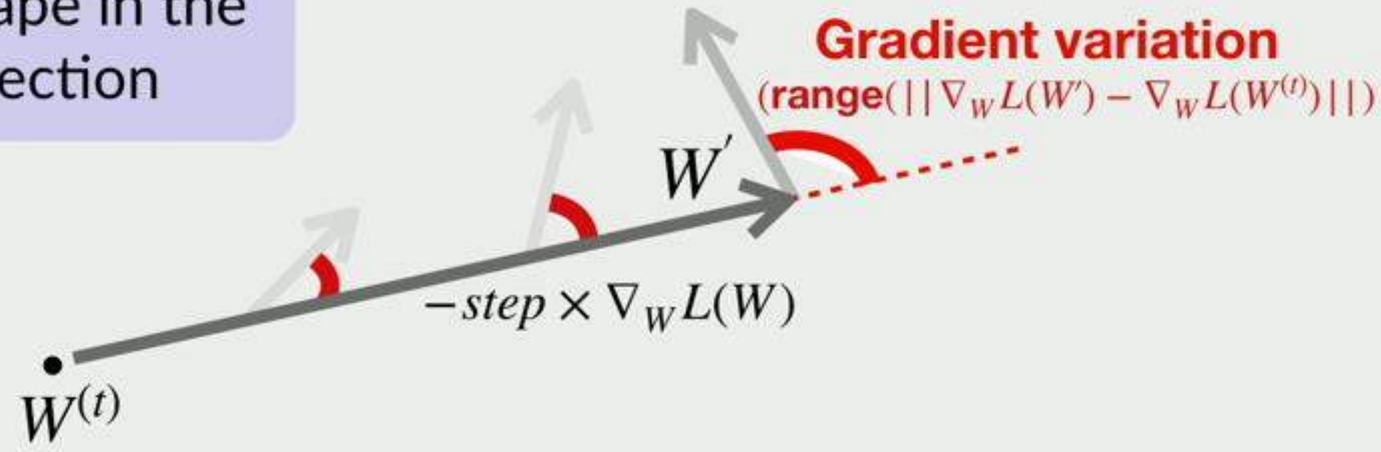
Explore landscape in the gradient direction

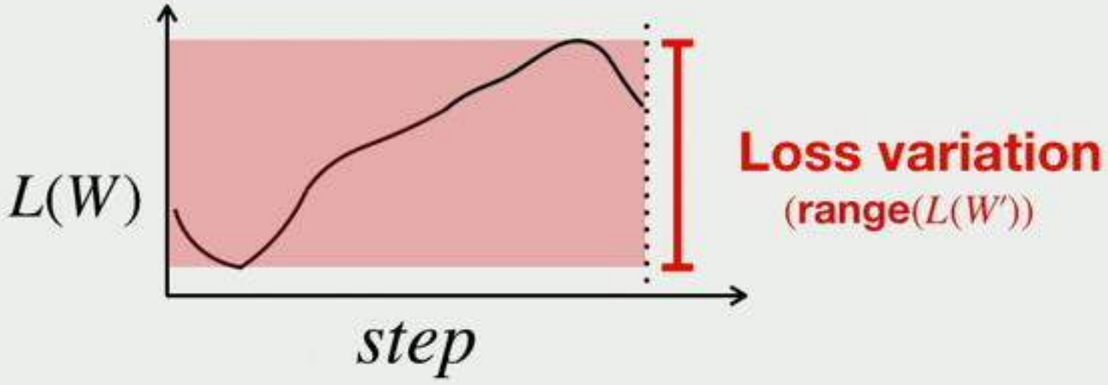




A Simple Experiment

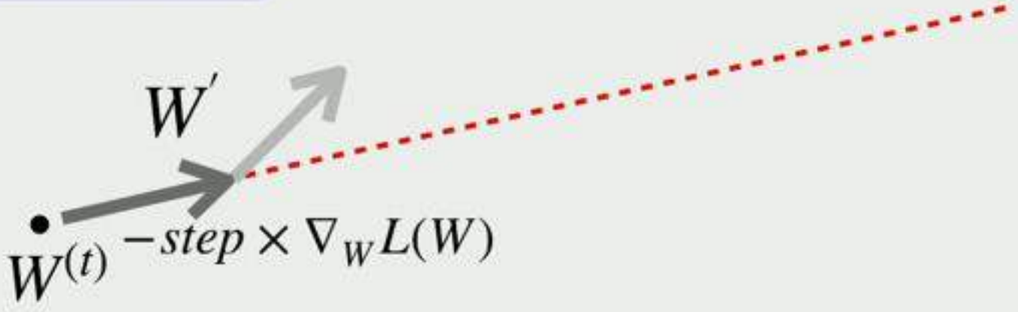
Explore landscape in the gradient direction

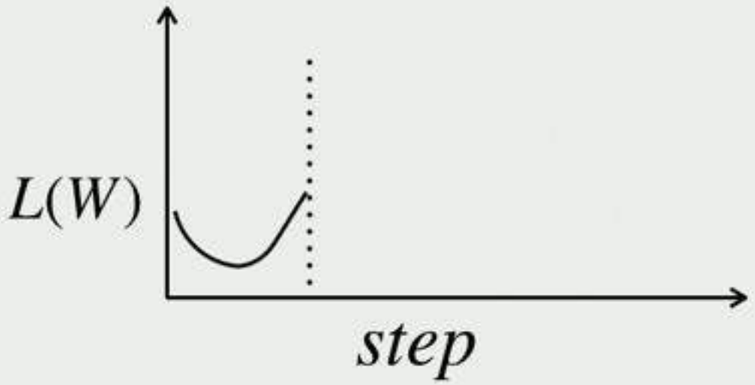




A Simple Experiment

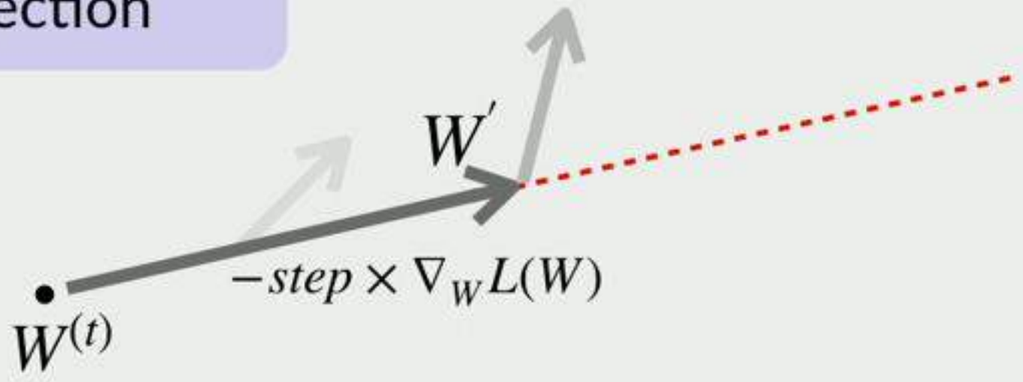
Explore landscape in the gradient direction

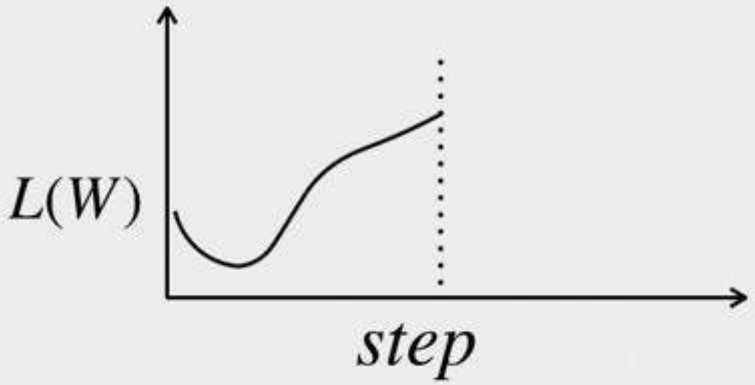




A Simple Experiment

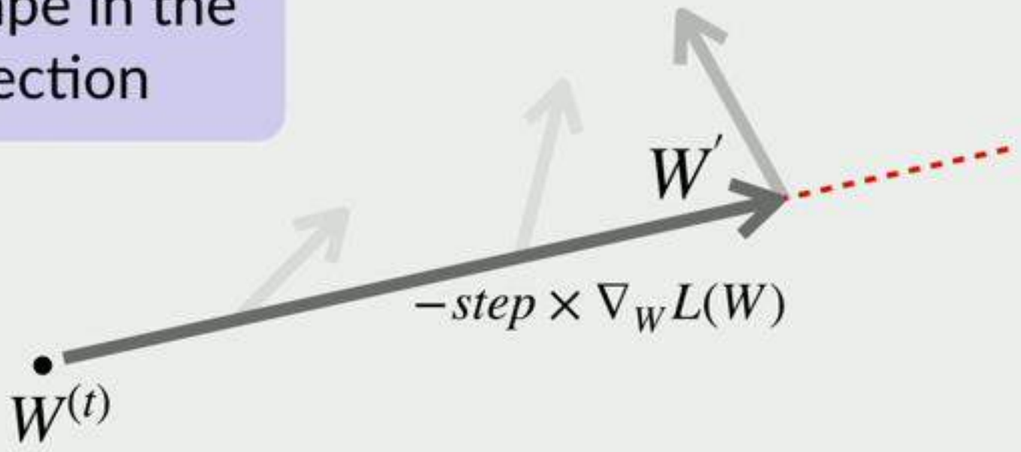
Explore landscape in the gradient direction

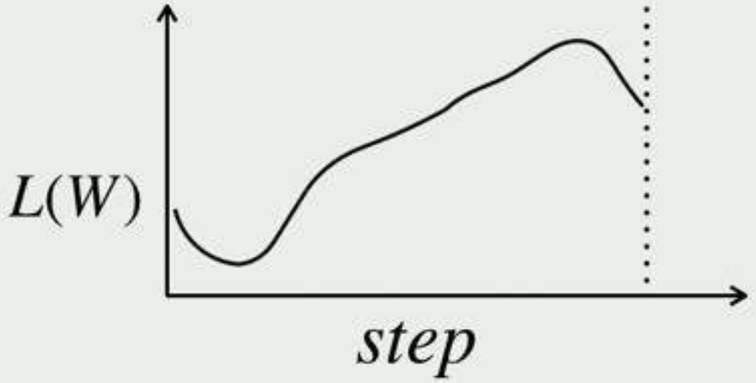




A Simple Experiment

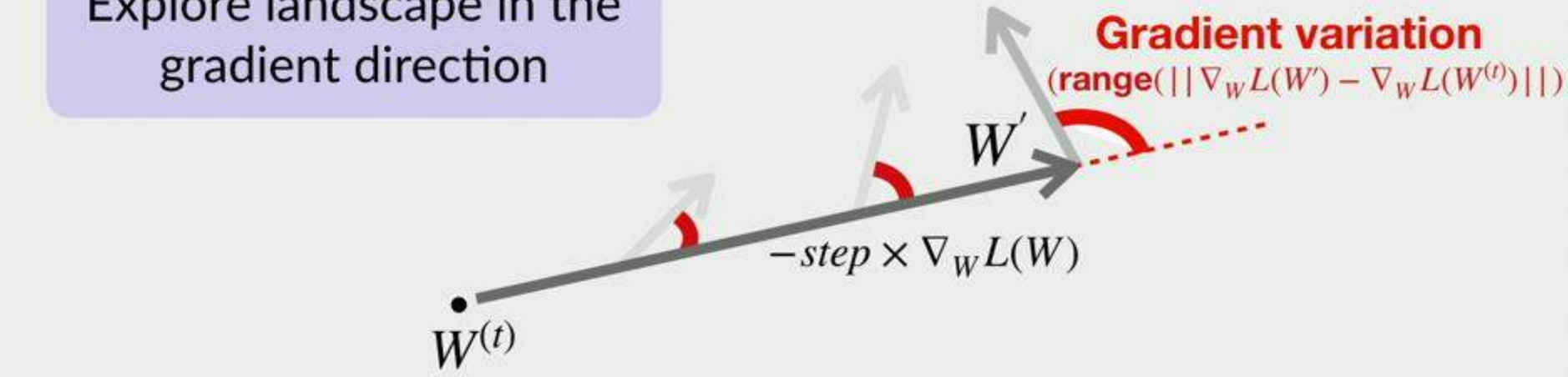
Explore landscape in the gradient direction

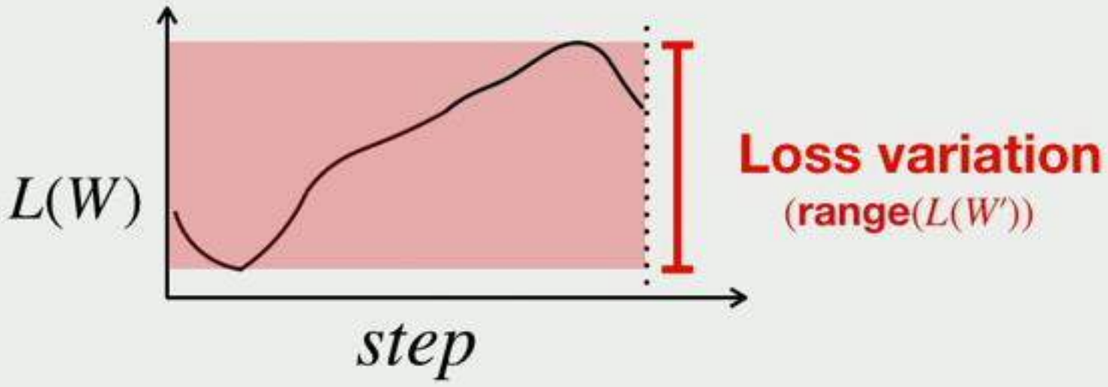




A Simple Experiment

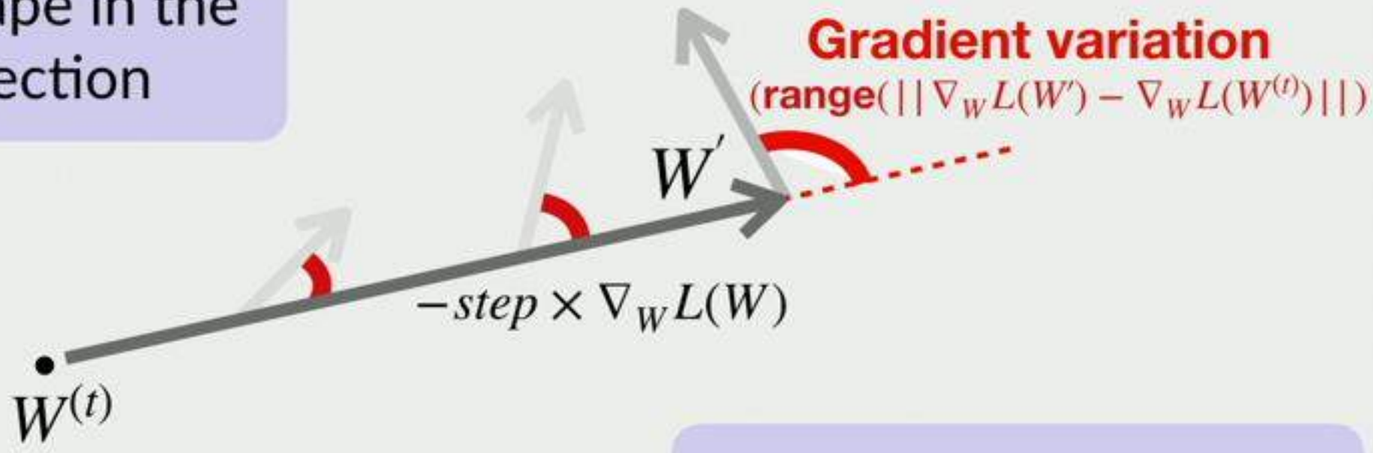
Explore landscape in the gradient direction

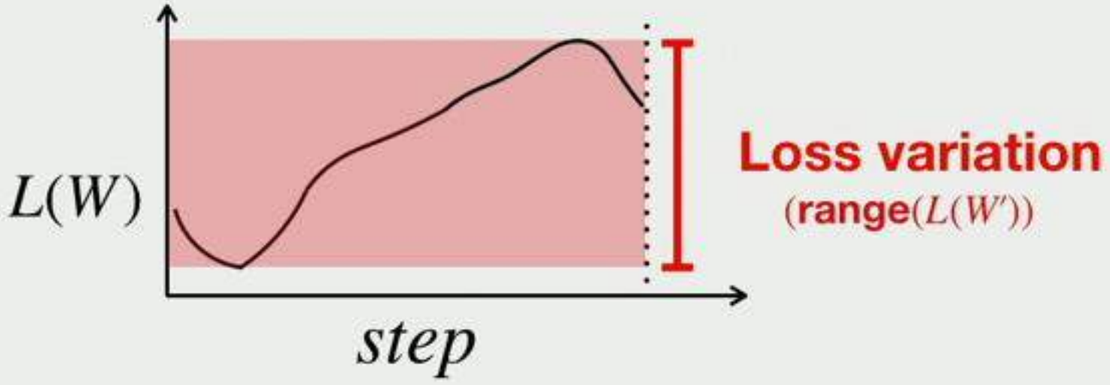




A Simple Experiment

Explore landscape in the gradient direction





Loss variation

Large fluctuations make optimization hard

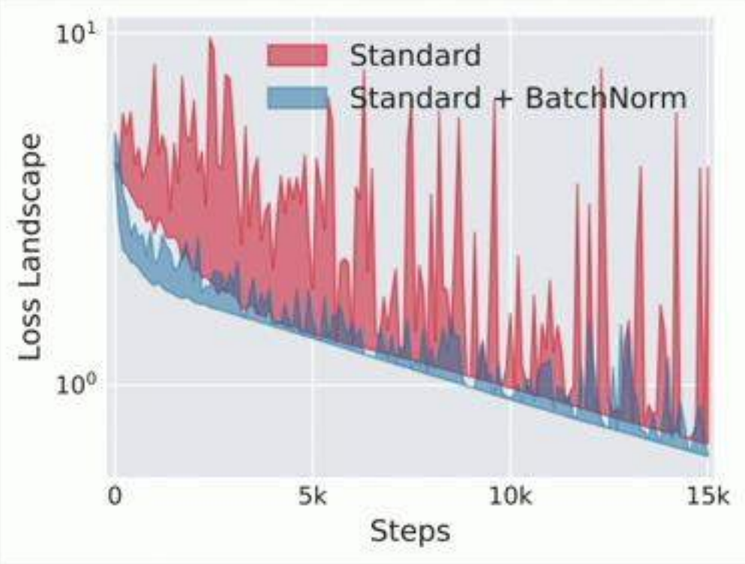
Changes in gradient:

Big differences imply less reliable gradients

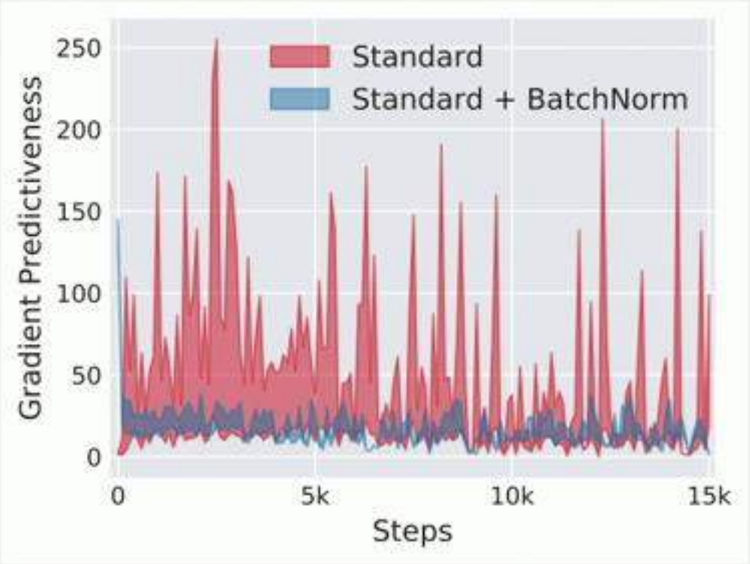
Landscape Induced by BatchNorm

Measure this variation at different points during training

Variation in Loss $(L(W))$



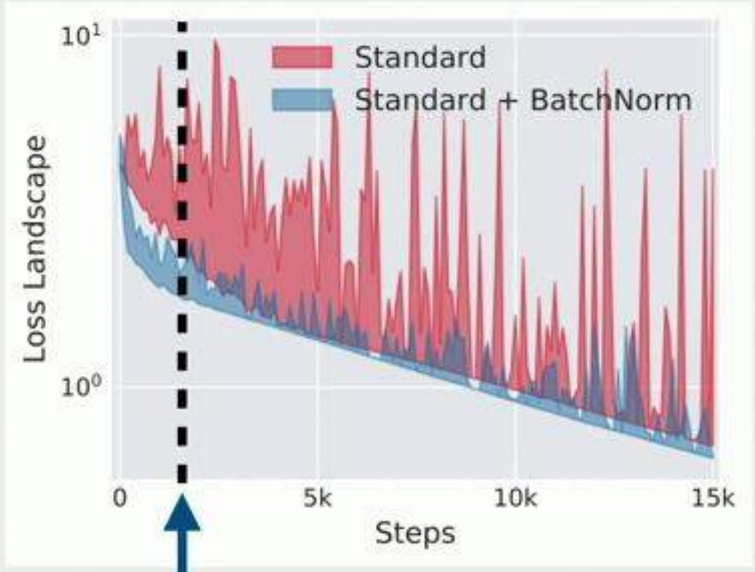
Change in Gradient $(\nabla_W L(W))$



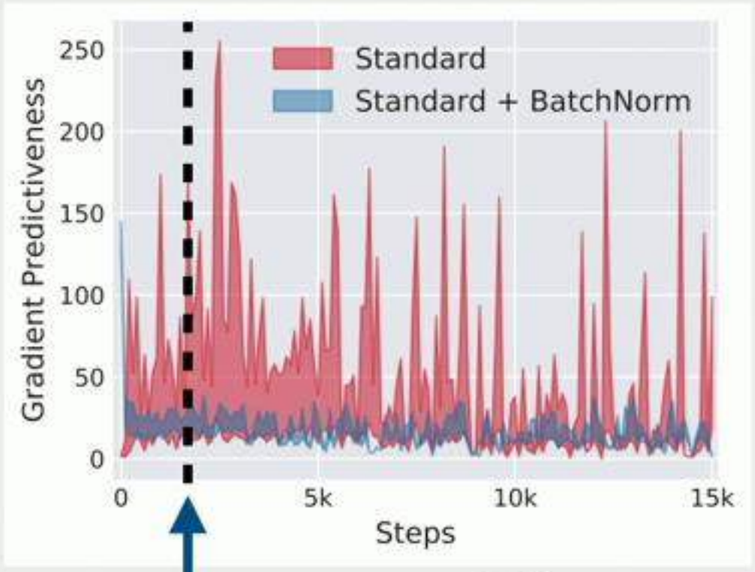
Landscape Induced by BatchNorm

Measure this variation at different points during training

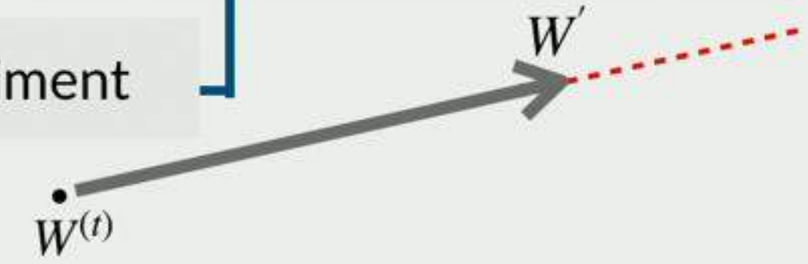
Variation in Loss $(L(W))$



Change in Gradient $(\nabla_W L(W))$

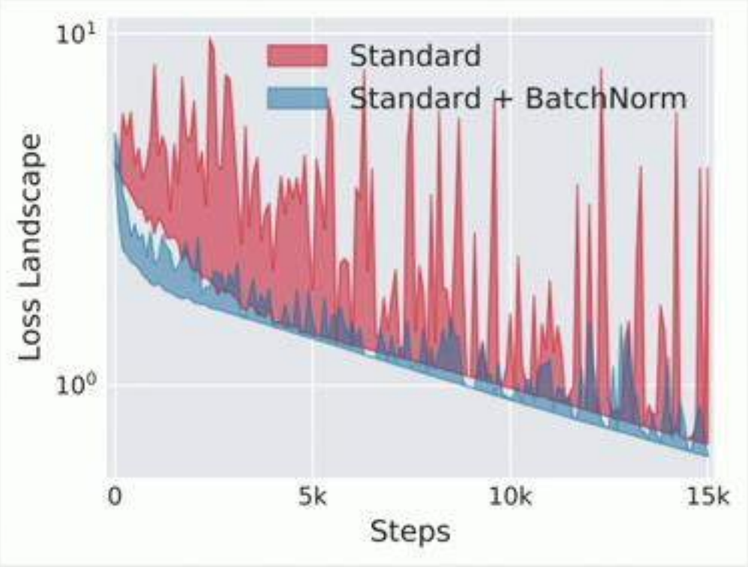


One run of the experiment

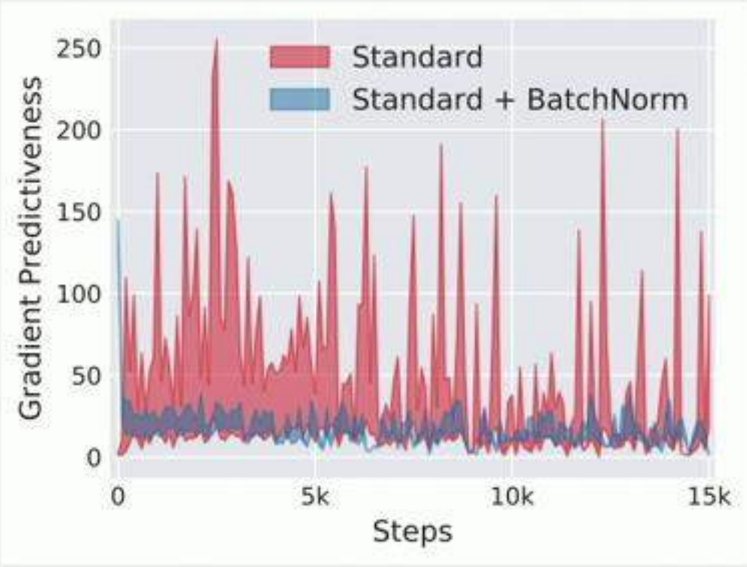


Landscape Induced by BatchNorm

Variation in Loss $(L(W))$



Change in Gradient $(\nabla_W L(W))$



Result:

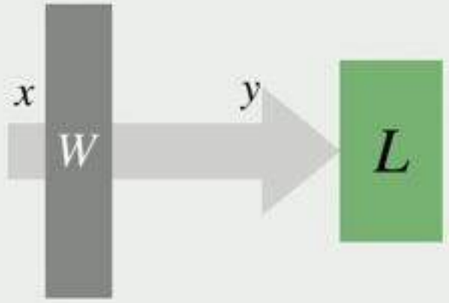
BatchNorm has **profound** effect on the landscape

(Makes it **smoother** and **easier to navigate**)

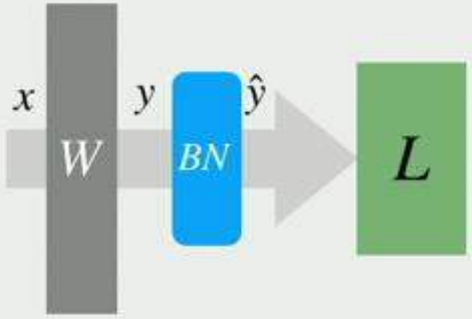
Delving Deeper

*What is the effect of a **single** BatchNorm layer on the optimization problem?*

Network **without** BatchNorm

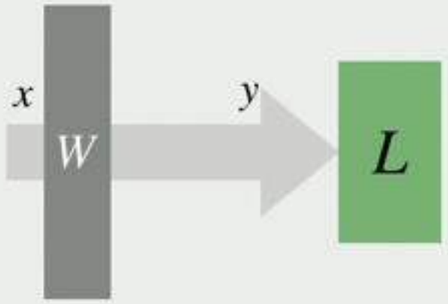


Network **with a single** BatchNorm layer

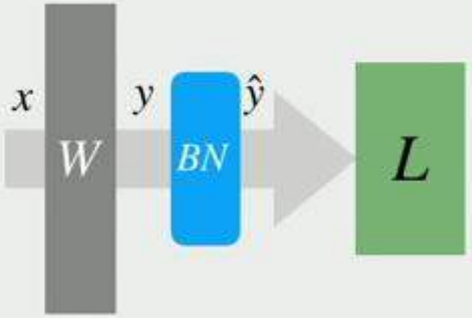


Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



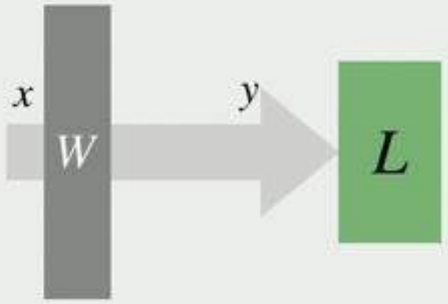
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

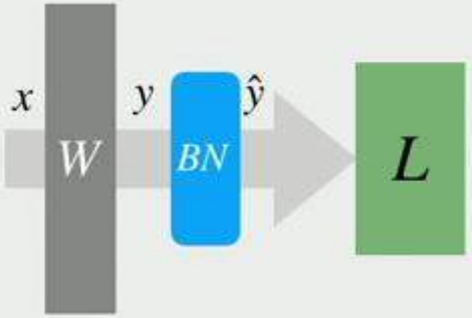
$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



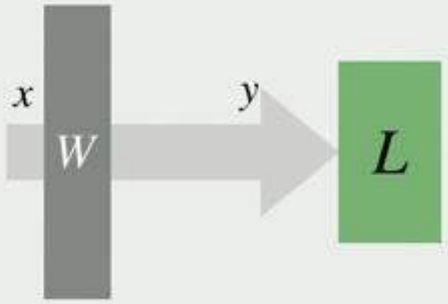
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

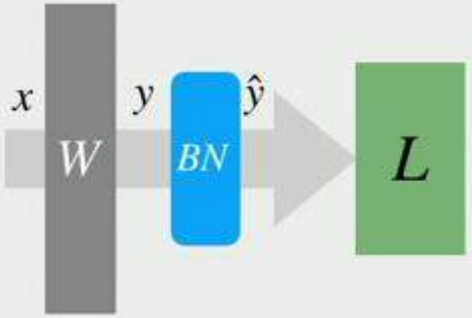
$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



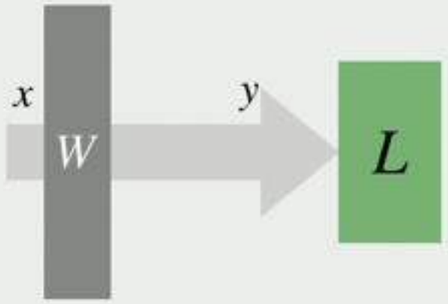
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

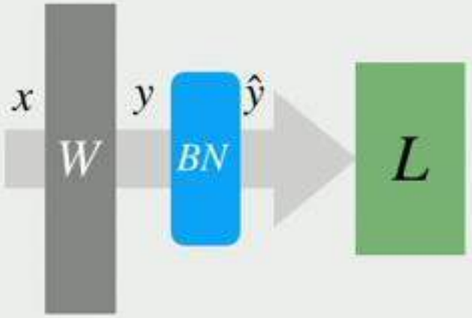
$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2 \right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



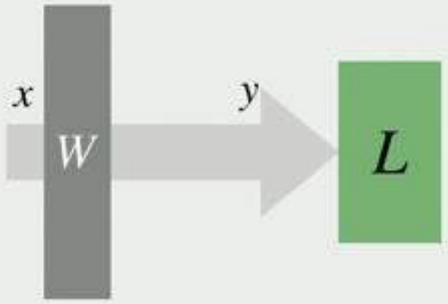
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

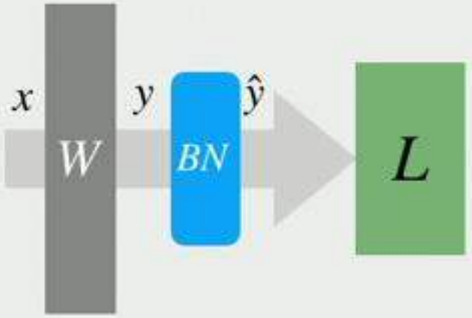
$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \Big(||\nabla_{y_j} L_{Std}||^2 - \underbrace{\mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2}_{\text{Additive} \downarrow} \Big)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



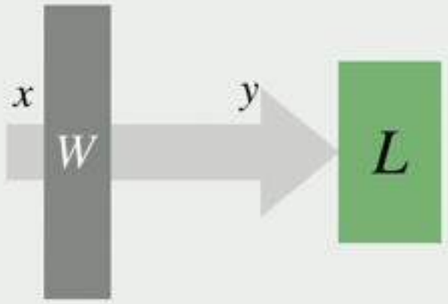
We also show:

Gradients (wrt y) become more **predictive**

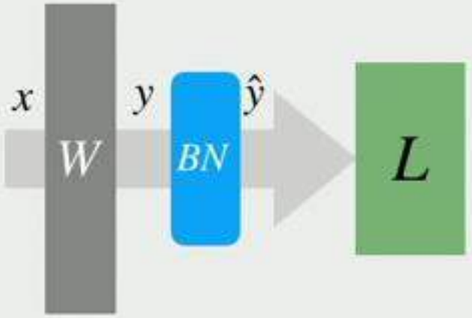
Translates into similar **worst-case** improvements

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



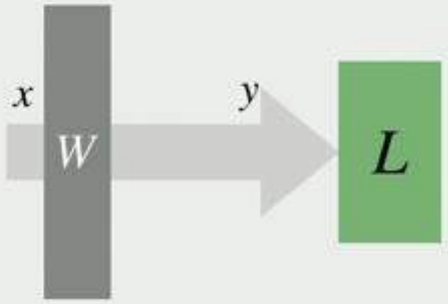
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

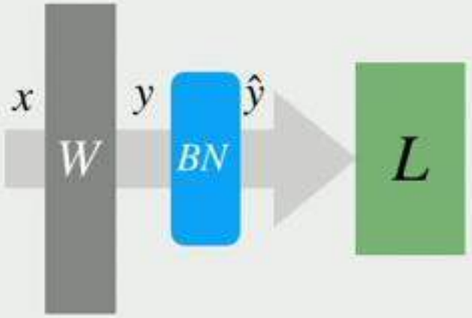
$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



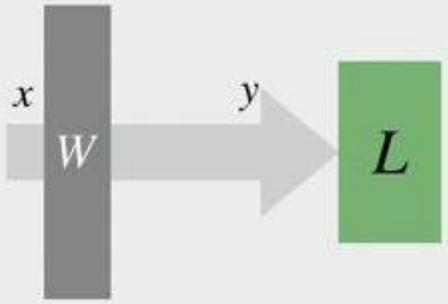
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

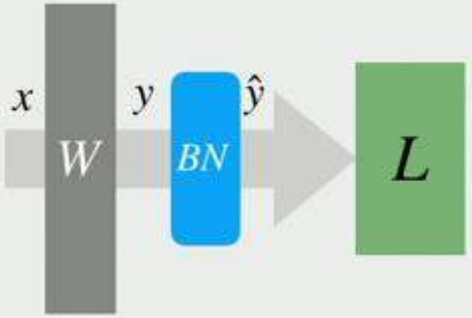
$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \left(||\nabla_{y_j} L_{Std}||^2 - \underbrace{\mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2}_{\text{Additive} \downarrow} \right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



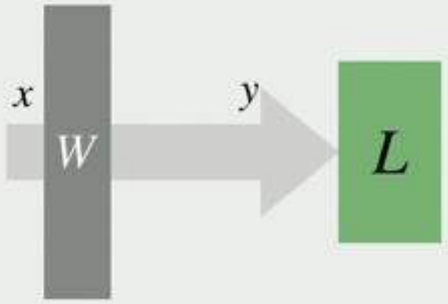
We also show:

Gradients (wrt y) become more **predictive**

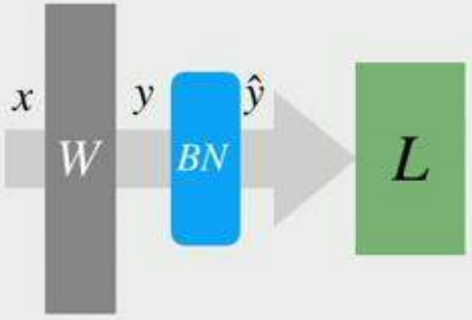
Translates into similar **worst-case** improvements

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



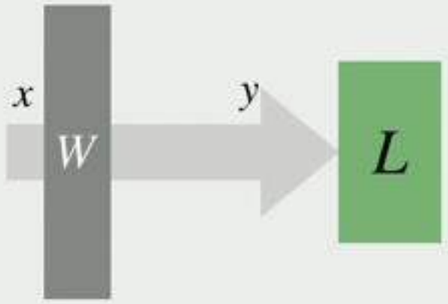
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

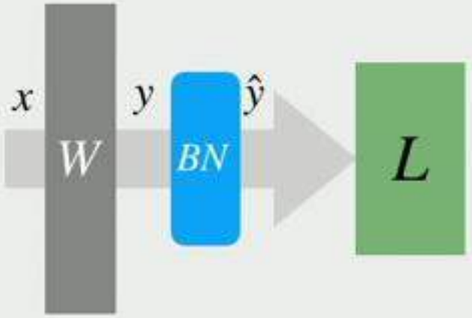
$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



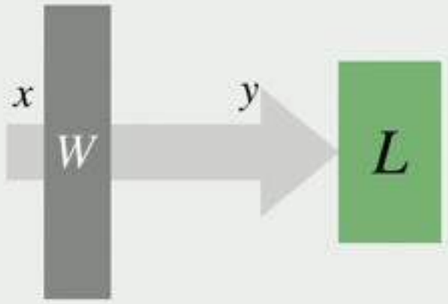
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

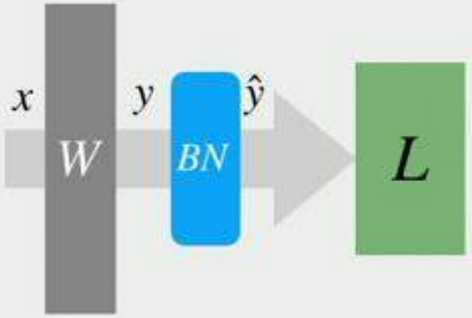
$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2 \right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



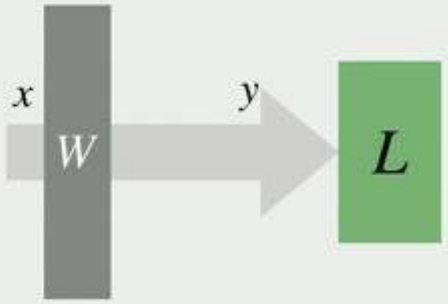
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

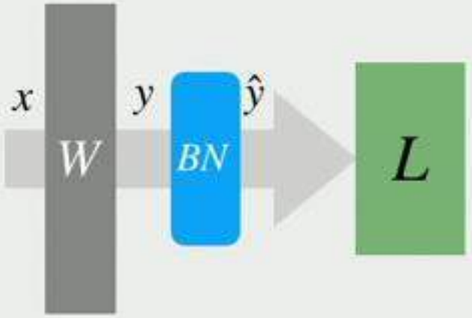
$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



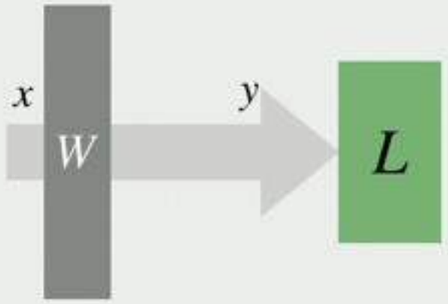
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

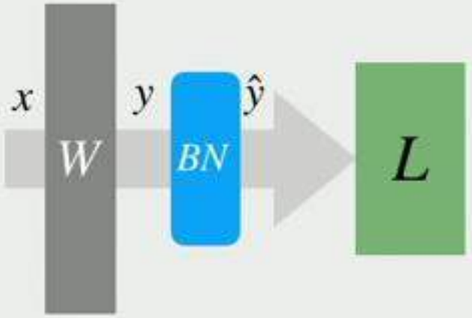
$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2 \right)$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \Big(||\nabla_{y_j} L_{Std}||^2 - \underbrace{\mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2}_{\text{Additive} \downarrow} \Big)$$

Are these effects unique to BatchNorm?

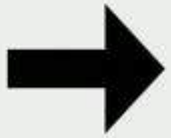
What if we normalize by a different notion of activation "scale"?

$$\hat{y} = \gamma \frac{y - \hat{\mu}}{\mathbf{C}} + \beta$$

Typical BatchNorm

$$\hat{\mu} = \frac{1}{B} \sum_{i=1}^{B} y_i$$

$$\mathbf{C} = \frac{1}{B} \| y - \hat{\mu} \|_2$$



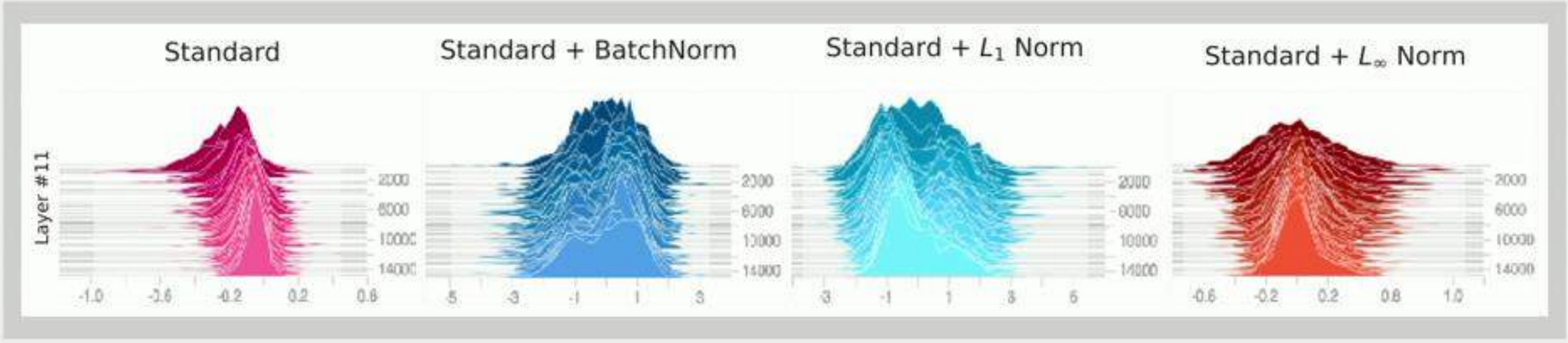
ℓ_p BatchNorm

$$\mathbf{C} = \| y \|_p = \left(\frac{1}{B} \sum_{i=1}^{B} | y_i |^p \right)^{1/p}$$

In general, **no control** over distribution moments.

Are these effects unique to BatchNorm?

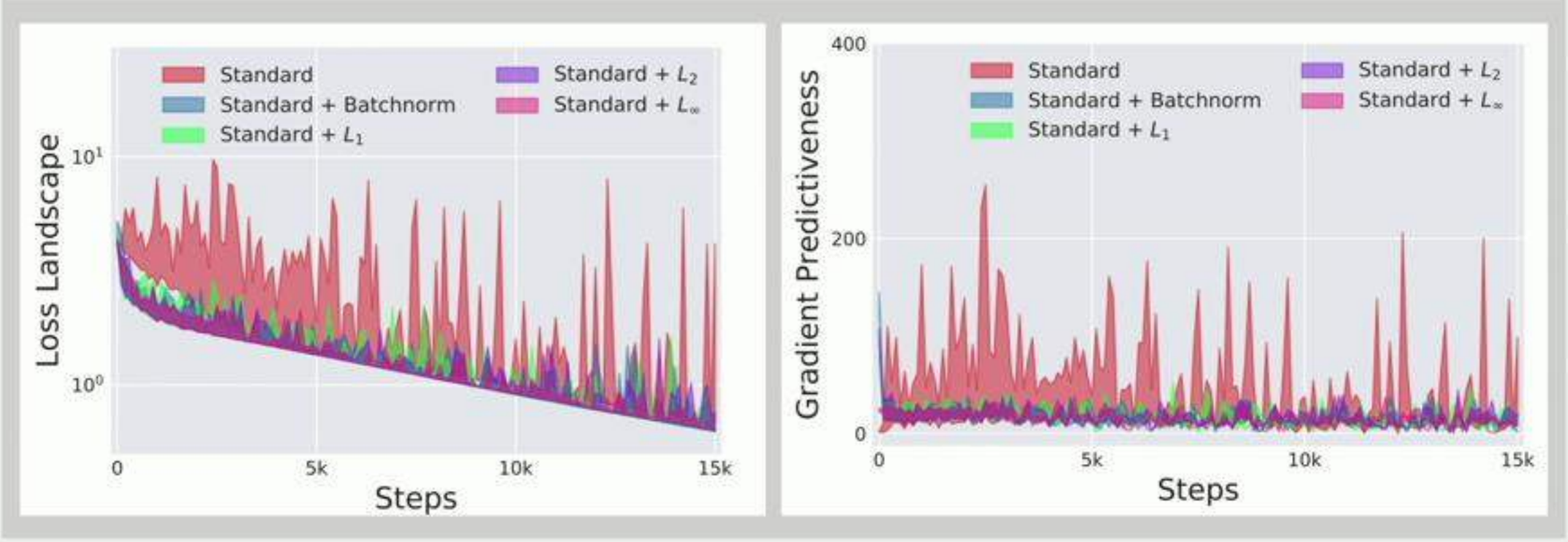
What if we normalize by a different notion of activation "scale"?



In general, **no control** over distribution moments.

Are these effects unique to BatchNorm?

Comparable landscape properties

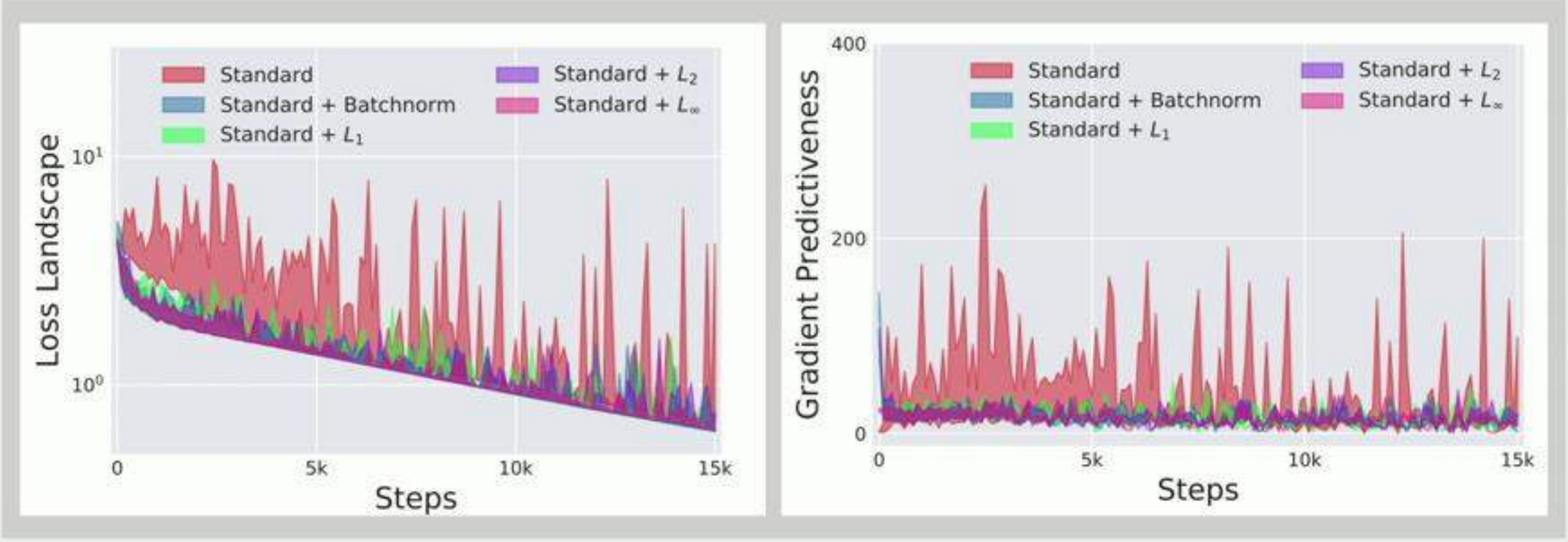


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

Are these effects unique to BatchNorm?

Comparable landscape properties

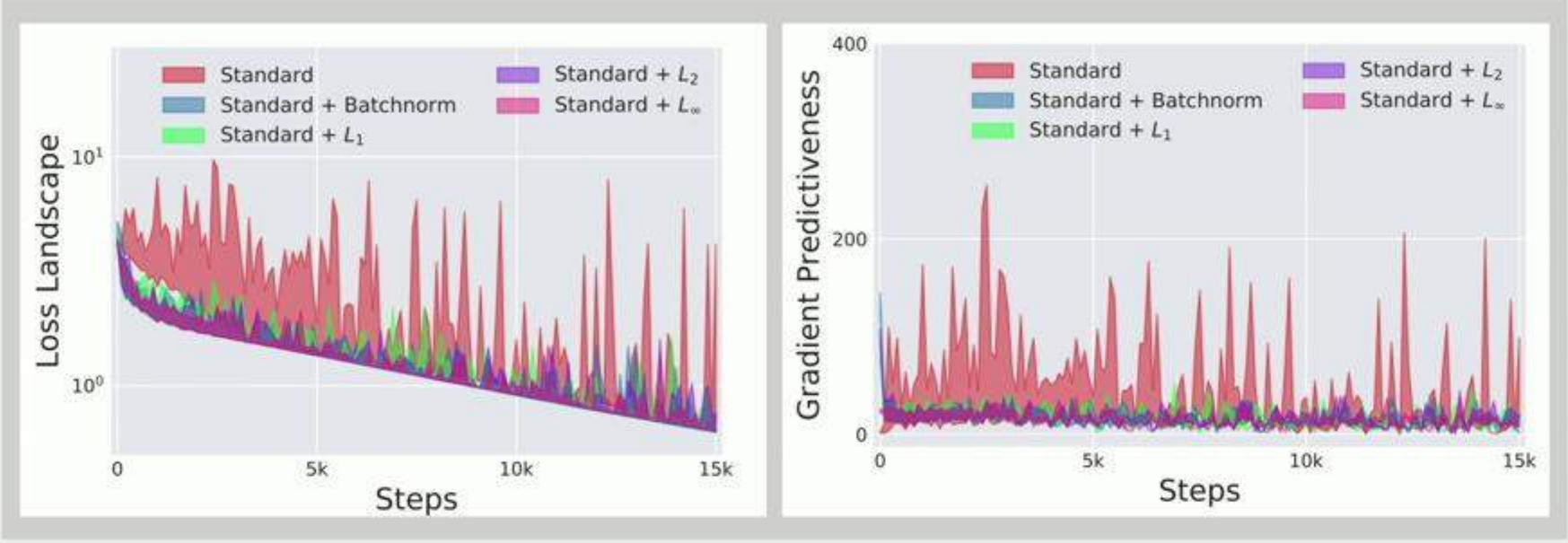


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

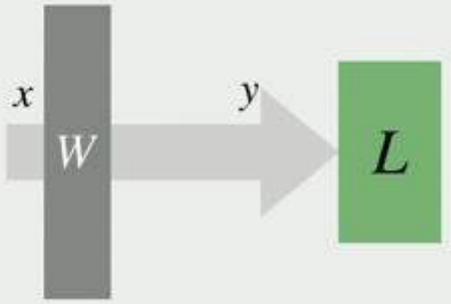
Are these effects unique to BatchNorm?

Comparable landscape properties

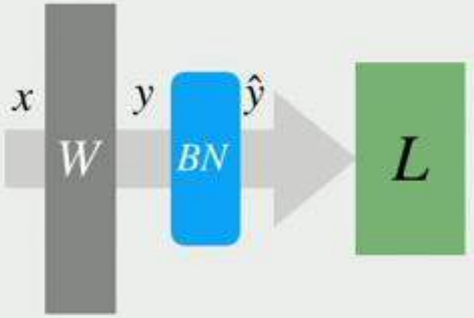


Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Are these effects unique to BatchNorm?

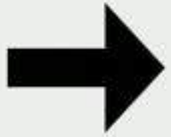
What if we normalize by a different notion of activation "scale"?

$$\hat{y} = \gamma \frac{y - \hat{\mu}}{\mathbf{C}} + \beta$$

Typical BatchNorm

$$\hat{\mu} = \frac{1}{B} \sum_{i=1}^{B} y_i$$

$$\mathbf{C} = \frac{1}{B} \|y - \hat{\mu}\|_2$$

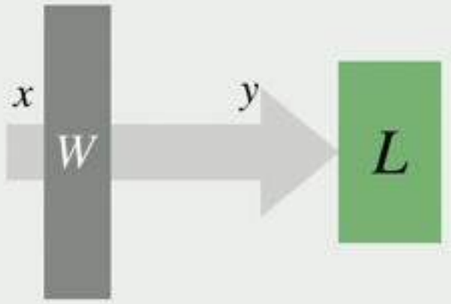


ℓ_p BatchNorm

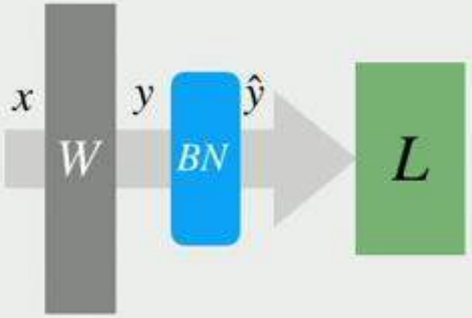
$$\mathbf{C} = \|y\|_p = \left(\frac{1}{B} \sum_{i=1}^{B} |y_i|^p \right)^{1/p}$$

Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



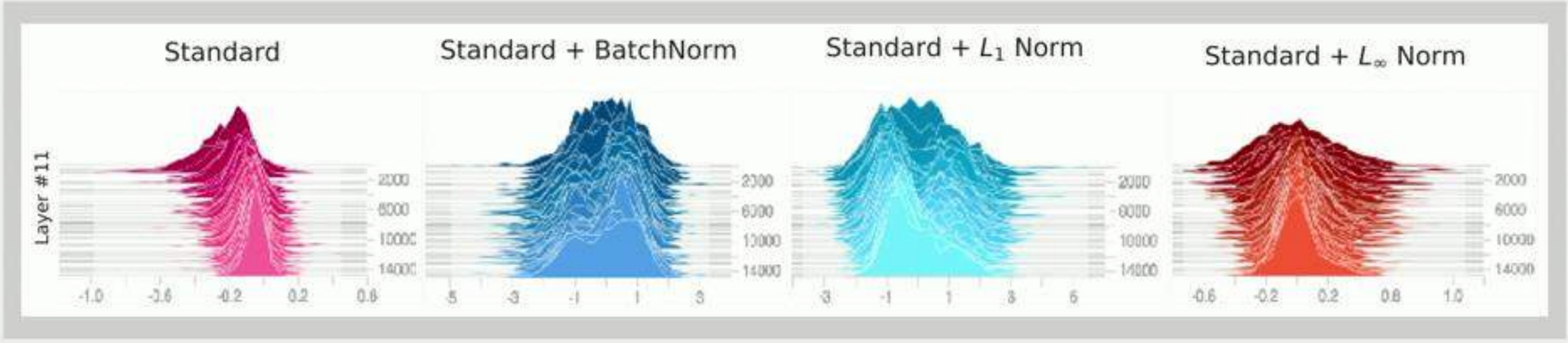
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

$$||\nabla_{y_j} L_{BN}||^2 \leq \frac{\gamma^2}{\sigma_j^2}\left(||\nabla_{y_j} L_{Std}||^2 - \mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2\right)$$

Are these effects unique to BatchNorm?

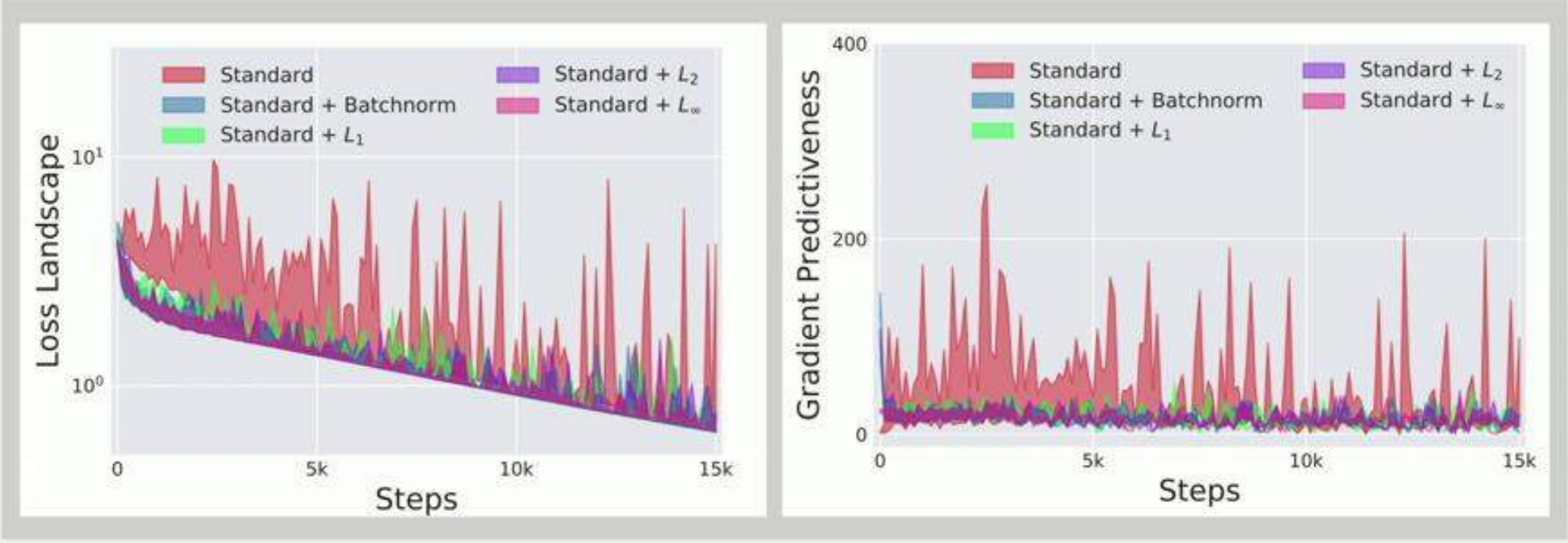
What if we normalize by a different notion of activation "scale"?



In general, **no control** over distribution moments.

Are these effects unique to BatchNorm?

Comparable landscape properties

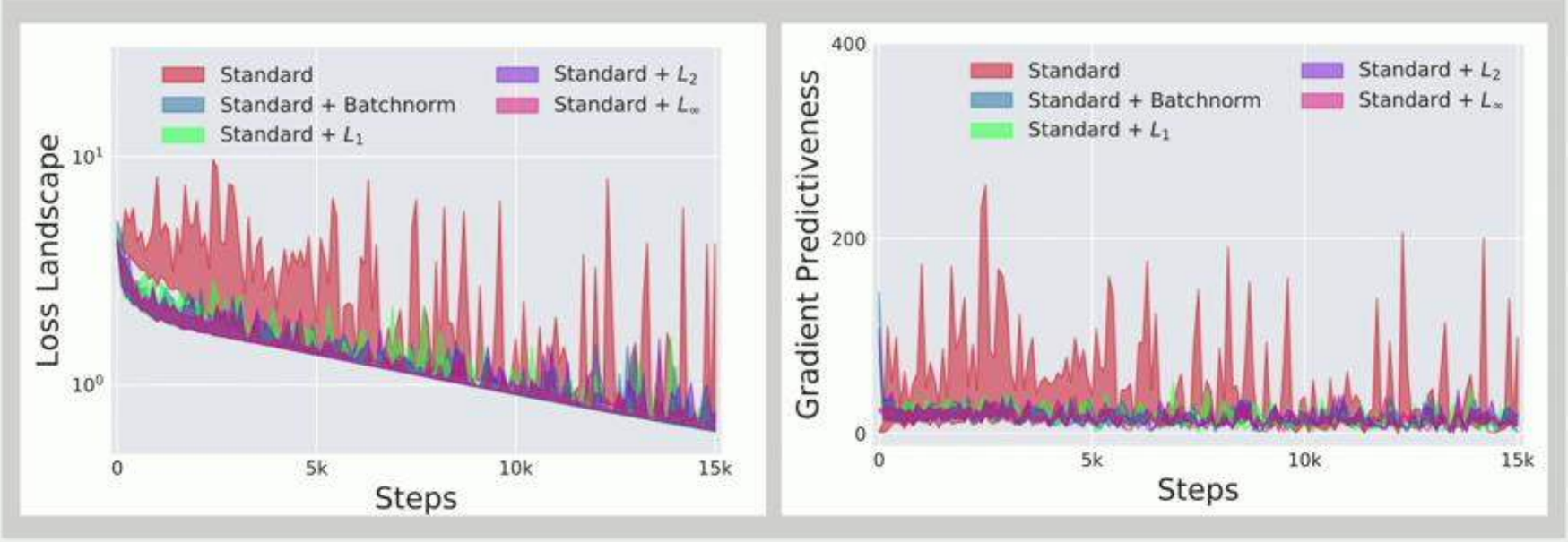


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

Are these effects unique to BatchNorm?

Comparable landscape properties

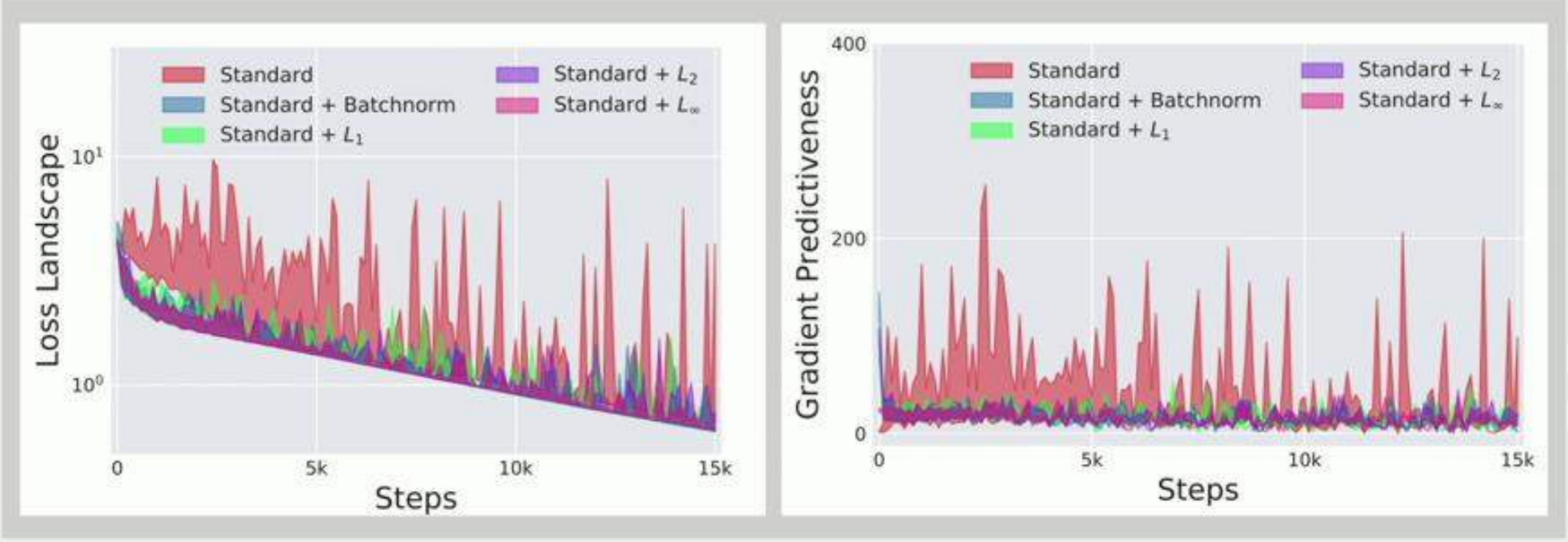


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

Are these effects unique to BatchNorm?

Comparable landscape properties

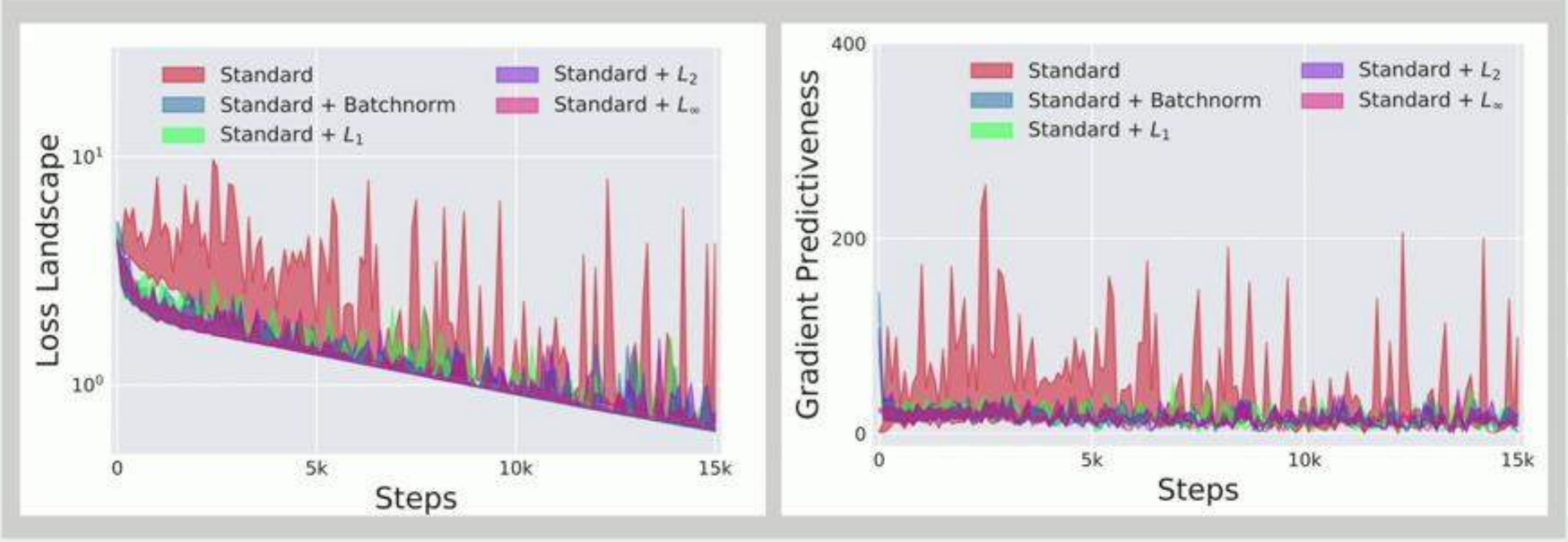


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

Are these effects unique to BatchNorm?

Comparable landscape properties

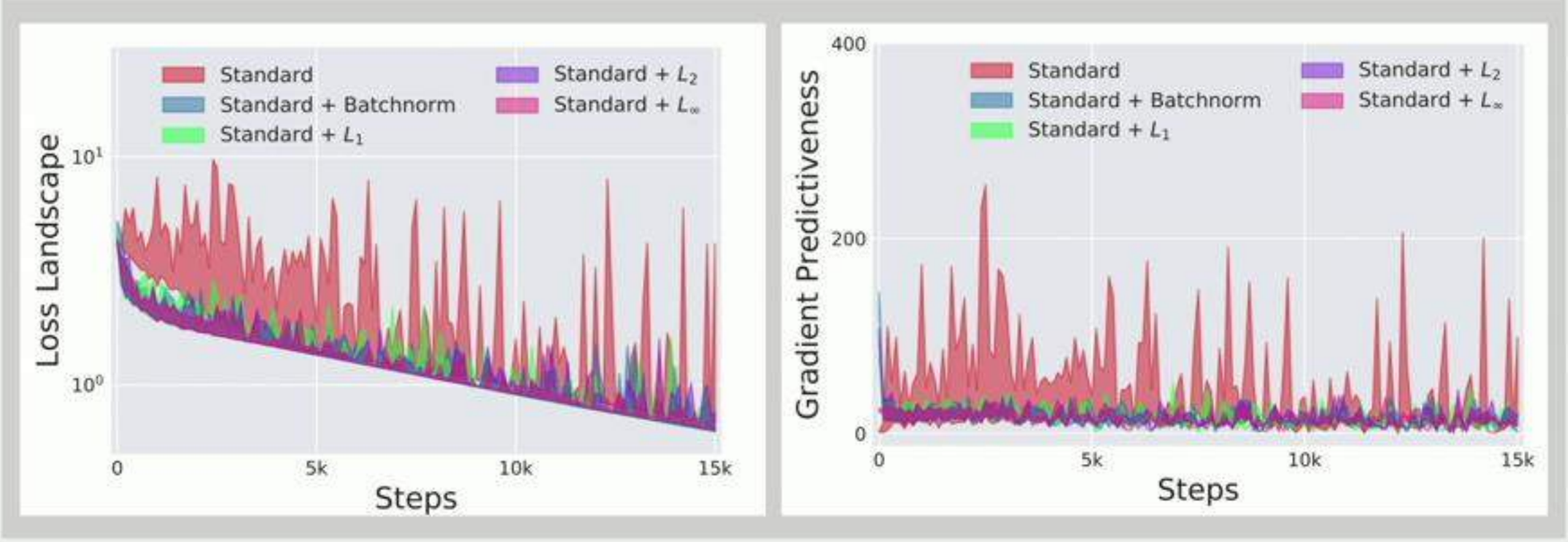


Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

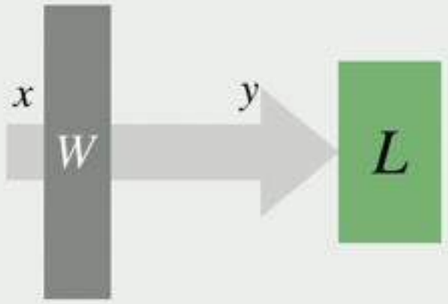
Are these effects unique to BatchNorm?

Comparable landscape properties

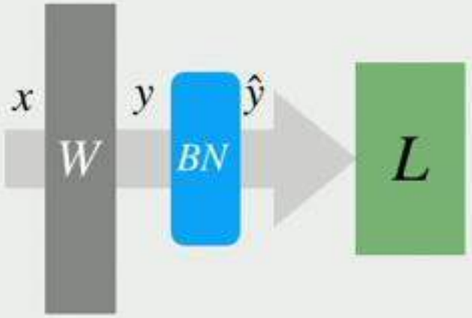


Delving Deeper

Network **without** BatchNorm



Network **with a single** BatchNorm layer



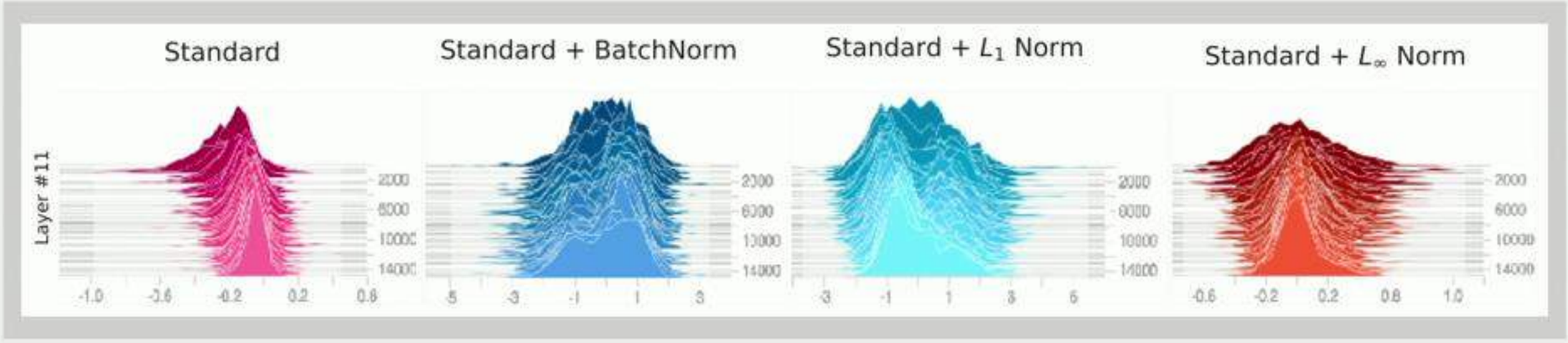
Theorem (*Effect of BatchNorm on the Lipschitzness of the loss*)

For any weights W and loss function L, *we have:*

$$||\nabla_{y_j} L_{BN}||^2 \leq \underbrace{\frac{\gamma^2}{\sigma_j^2}}_{\text{Multiplicative} \downarrow} \Big(||\nabla_{y_j} L_{Std}||^2 - \underbrace{\mu(\nabla_{y_j} L_{Std})^2 - \frac{1}{m}(\hat{y}_j^\top \nabla_{y_j} L_{Std})^2}_{\text{Additive} \downarrow} \Big)$$

Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?



In general, **no control** over distribution moments.

Are these effects unique to BatchNorm?

What if we normalize by a different notion of activation "scale"?

Are these effects unique to BatchNorm?

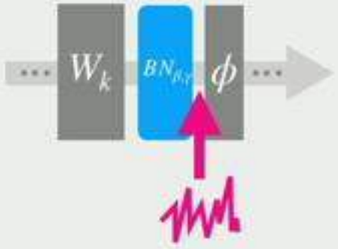
What if we normalize by a different notion of activation "scale"?

Performance improvement comparable to BatchNorm!

Goal: Understand the **exact** role of BatchNorm in optimization

BatchNorm ↔ ICS ↔ Optimization relationships tenuous

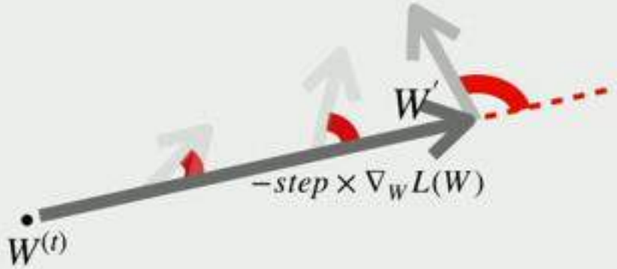
Noisy BatchNorm



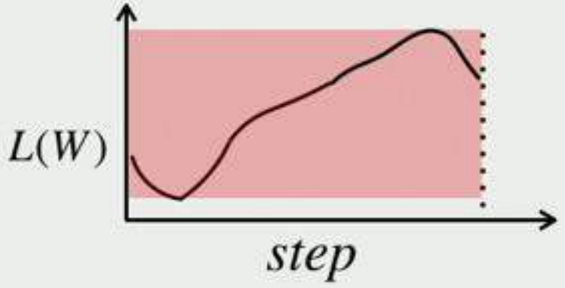
Optimization-based ICS

Identify a fundamental smoothing effect of BatchNorm

Gradient variation



Loss variation



Goal: Understand the **exact** role of BatchNorm in optimization

Moving forward:

Direct approaches to landscape smoothing

BatchNorm and generalization

Other normalization methods

Goal: Understand the **exact** role of BatchNorm in optimization

Moving forward:

Direct approaches to landscape smoothing

BatchNorm and generalization

Other normalization methods

More broadly: understand other elements of our DL toolkit

arXiv:1805.11604

See our blog post at
gradsci.org/batchnorm