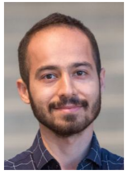
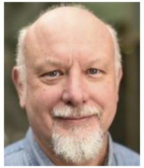
# PowerNorm: Rethinking Batch Normalization for Transformers

Sheng Shen, Zhewei Yao, Amir Gholami, Michael Mahoney, Kurt Keutzer

University of California, Berkeley







# Executive Summary

- We perform a systematic study of why **Batch Normalization** results in **poor performance in Transformers**
  - Variance in forward pass
  - Variance n backward pass

- We propose **(PN)**, a novel normalization that achieves state-of-the art result in **Machine Translation, Language Modeling**
  - Improve BLEU score on IWSLT by **0.4** and WMT14 by **0.6**
  - Decrease the PPL on PTB by **5.87** and WikiText-103 by **2.78**

- **PN** achieves this by resolving the following problems:
  - Mean estimation (recentering) in BN is better to be removed
  - Better to include running statistics during training
  - The backpropagation needs to be adjusted accordingly

# Motivation

- **Normalization is a building block of current neural networks.**

- **There is a lack of interests in NLP for studying normalization**, especially for transformer model (where the training is costly).
  - Layer Normalization. (ICLR' 17)
  - Root Mean Square Layer Normalization (NeurIPS' 19)
  - Understanding and improving layer normalization (NeurIPS' 19)
  - Improving Deep Transformer with Depth-Scaled Initialization and Merged Attention (EMNLP' 19)
  - On Layer Normalization in the Transformer Architecture (ICML' 20)

- **Implementing Batch Normalization directly on NLP does not work.**
  - Why does not it work?
  - How can we make it work on NLP?
  - What can we learn from that?

# Content

- **How Normalization is performed in NLP (vs. CV)?**
- Rethinking the variance of Batch Normalization for NLP (Transformers):
  - The variance in forward pass
  - The variance in backward pass
- Our New Normalization PN and its performance in different NLP tasks.

# How Normalization is performed in NLP

- **Weight Normalization.**

$$h' = \frac{w^T h}{\|w\|_2}$$

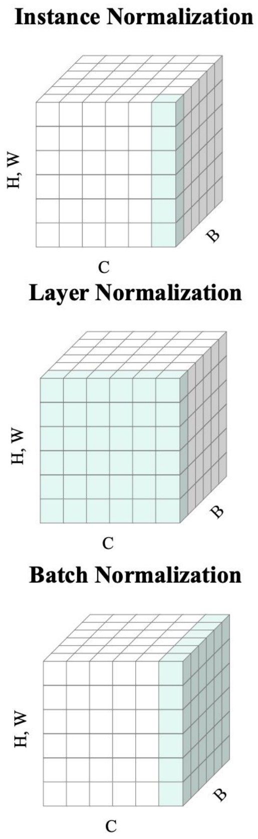
- **Activation Normalization.**

$$\hat{h}_{ncij} = \gamma \frac{h_{ncij} - \Sigma_{k\in\Omega} w_k \mu_k}{\sqrt{\Sigma_{k\in\Omega} w'_k \sigma_k^2 + \epsilon}} + \beta,$$

IN: $$\mu_{\text{in}} = \frac{1}{HW} \sum_{i,j}^{H,W} h_{ncij}, \quad \sigma_{\text{in}}^2 = \frac{1}{HW} \sum_{i,j}^{H,W} (h_{ncij} - \mu_{\text{in}})^2,$$

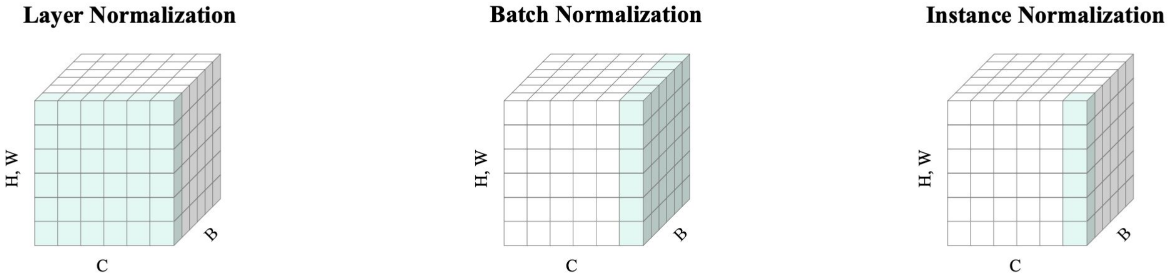
LN: $$\mu_{\text{ln}} = \frac{1}{C} \sum_{c=1}^{C} \mu_{\text{in}}, \quad \sigma_{\text{ln}}^2 = \frac{1}{C} \sum_{c=1}^{C} (\sigma_{\text{in}}^2 + \mu_{\text{in}}^2) - \mu_{\text{ln}}^2,$$

BN: $$\mu_{\text{bn}} = \frac{1}{N} \sum_{n=1}^{N} \mu_{\text{in}}, \quad \sigma_{\text{bn}}^2 = \frac{1}{N} \sum_{n=1}^{N} (\sigma_{\text{in}}^2 + \mu_{\text{in}}^2) - \mu_{\text{bn}}^2,$$



# How Normalization is performed in NLP

- **Normalization in Computer Vision**
  - **H*W (image size), C (channel/feature dim), B (batch size)**

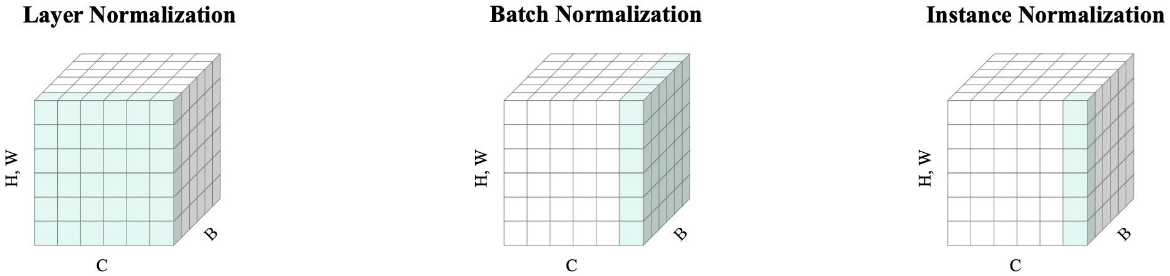



  - **LN:** reduce (C, H*W) => 2*B statistics
  - **BN:** reduce (B, H*W) => 2*C statistics
  - **IN:** reduce (H*W) => 2*B*C statistics

# How Normalization is performed in NLP

- **Normalization in NLP**
  - **L/H*W (sentence length), C (feature dim), B (batch size)**

**suppose:**

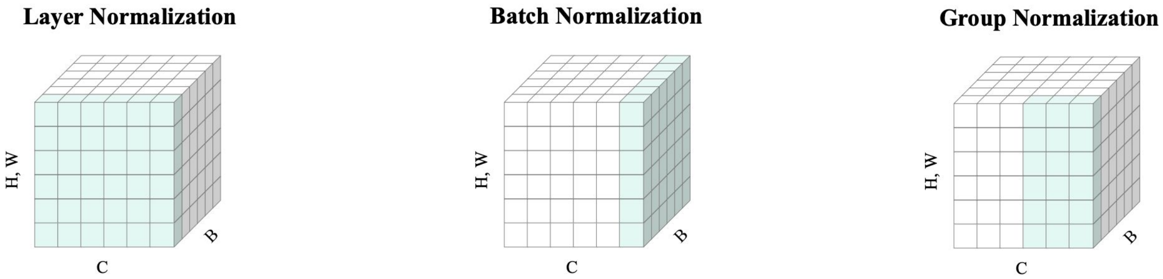



- **LN:** reduce (C, H*W) => 2*B statistics
- **BN:** reduce (B, H*W) => 2*C statistics
- **IN:** reduce  (H*W) => 2*B*C statistics

# How Normalization is performed in NLP

- **Normalization in NLP**
  - **L/H*W (sentence length), C (feature dim), B (batch size)**

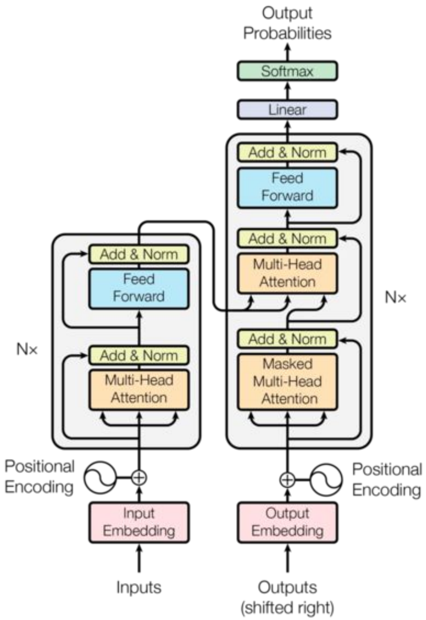
**reality:**




- **LN:** reduce (C, ~~H*W~~) => 2*B*L statistics
- **BN:** reduce (B, L) => 2*C statistics (padding will destroy the statistics)
- **GN:** reduce (C/g) => 2*B*L*g statistics

# How Normalization is performed in NLP

- **Transformer Architecture**




- Encoder: self-attention + point-wise feed-forward

  1 + 2 * N layer norm before/after each block.

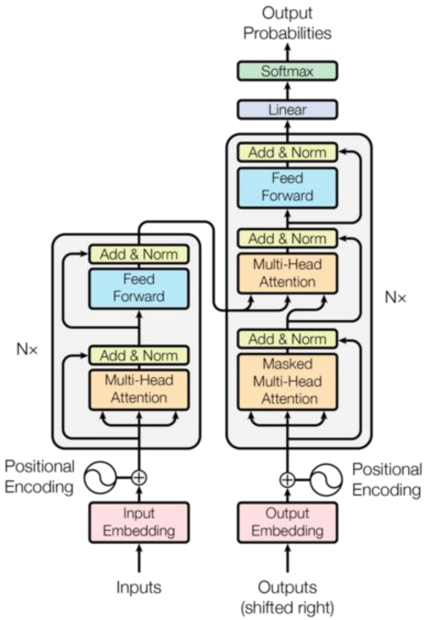
- Decoder: self-attention + enc-decoder attention + point-wise feed-forward

  1 + 3 * N layer norm before/after each block.

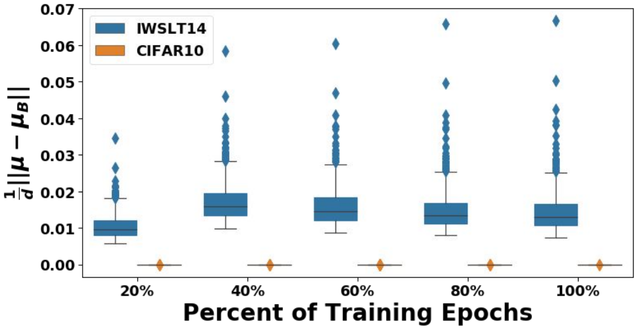
[1] Vaswani et al., “Attention is all you need”, NIPS 2017

# How Normalization is performed in NLP

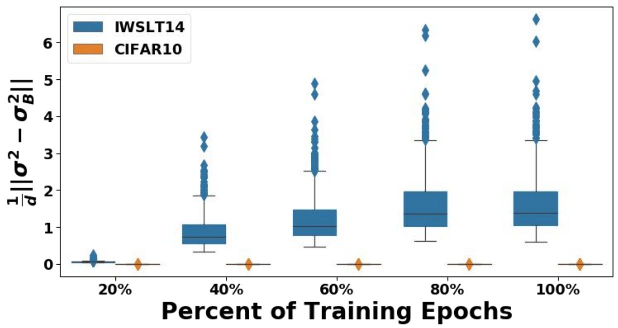
- **Difference btw BN's usage in CNN and Transformer (CV, NLP)**




- **Train/Test Statistical Discrepancy**: the mismatch of running batch-statistics and real batch-statistics for ***NLP***.



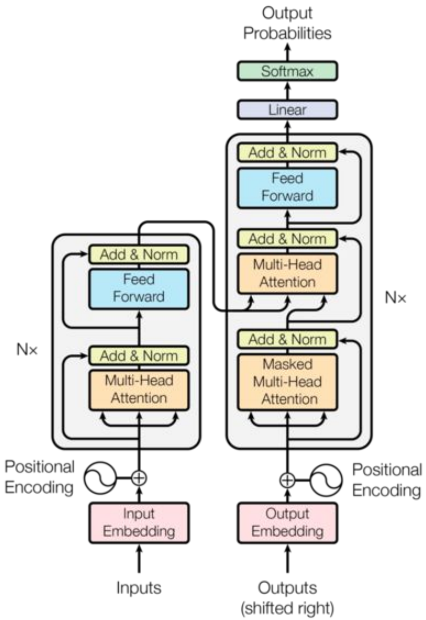

(a) $\|\mu - \mu_B\|^2$ at bn.1 for CIFAR10 vs **IWSLT** (training)



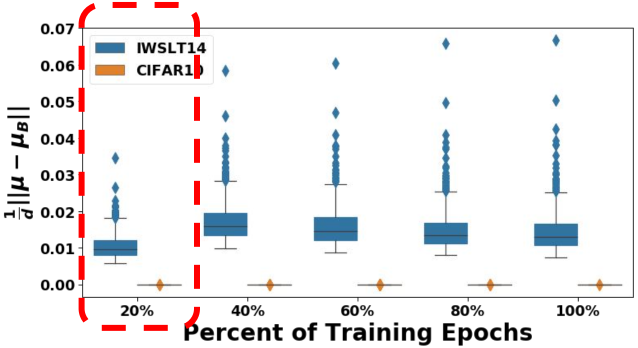

(b) $\|\sigma^2 - \sigma_B^2\|^2$at bn.1 for CIFAR10 vs **IWSLT** (training)

# How Normalization is performed in NLP

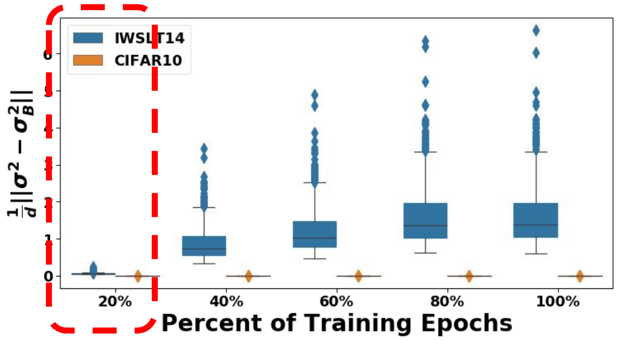
- **Difference btw BN's usage in CNN and Transformer (CV, NLP)**




- **Train/Test Statistical Discrepancy**: the mismatch of running batch-statistics and real batch-statistics for ***NLP***.



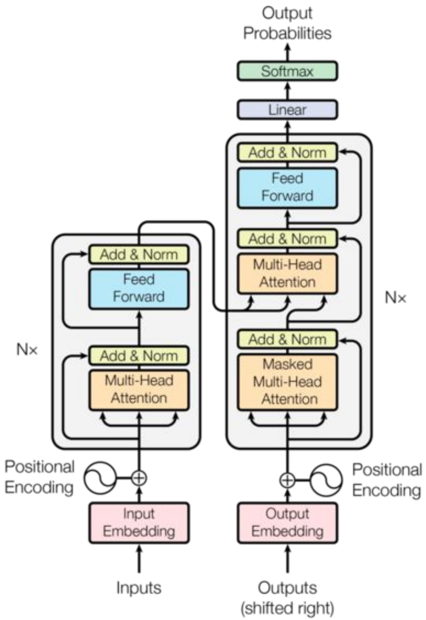

(a) $\|\mu - \mu_B\|^2$ at bn.1 for CIFAR10 vs **IWSLT** (training)



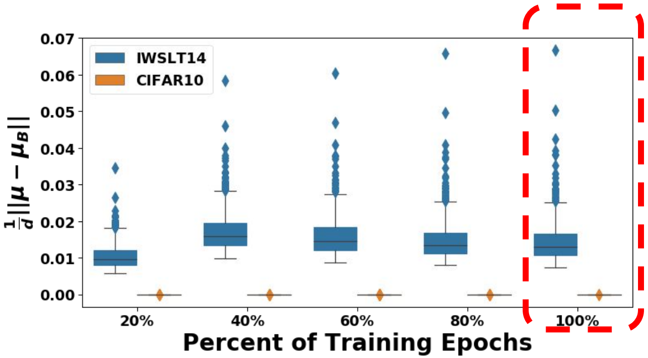

(b) $\|\sigma^2 - \sigma_B^2\|^2$at bn.1 for CIFAR10 vs **IWSLT** (training)

# How Normalization is performed in NLP

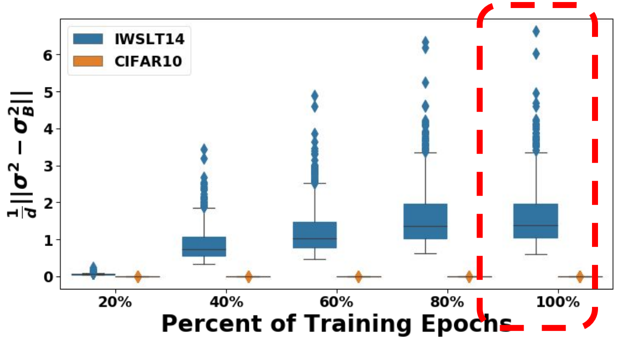
- **Difference btw BN's usage in CNN and Transformer (CV, NLP)**




- **Train/Test Statistical Discrepancy**: the mismatch of running batch-statistics and real batch-statistics for ***NLP***.




(a) $\|\mu - \mu_B\|^2$ at bn.1 for CIFAR10 vs **IWSLT** (training)




(b) $\|\sigma^2 - \sigma_B^2\|^2$at bn.1 for CIFAR10 vs **IWSLT** (training)

# Content

- How Normalization is performed in NLP (vs. CV)?
- **Rethinking the variance of Batch Normalization for NLP (Transformers):**
    - The variance in forward pass
    - The variance in backward pass
- Our New Normalization PN and its performance in different NLP tasks.

# Rethinking the variance of Batch Normalization for NLP

- **The variance in the Forward Pass:**
  - The variance btw the real mean/variance and running mean/variance.
    - $\|\mu - \mu_B\|^2$ and $\|\sigma^2 - \sigma_B^2\|^2$

**Input:** Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: $\gamma$, $\beta$
**Output:** $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

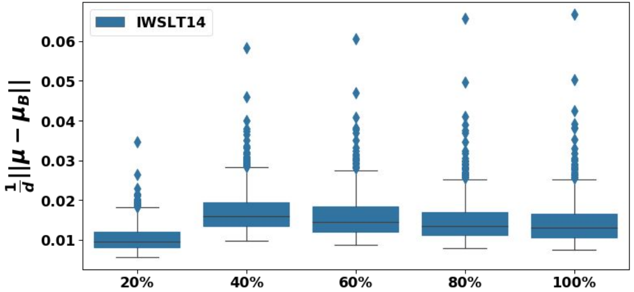
$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

$$\mu := \mu + \alpha(\mu_\mathcal{B} - \mu) \qquad \text{// Update moving averages}$$

$$\sigma := \sigma + \alpha(\sigma_\mathcal{B} - \sigma)$$




(a) $\|\mu - \mu_B\|^2$ at bn.1 for **IWSLT** (training)

# Rethinking the variance of Batch Normalization for NLP

- **The variance in the Forward Pass:**
  - The variance btw the real mean/variance and running mean/variance.
    - $\|\mu - \mu_B\|^2$ and $\|\sigma^2 - \sigma_B^2\|^2$

**Input:** Values of $x$ over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: $\gamma$, $\beta$
**Output:** $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

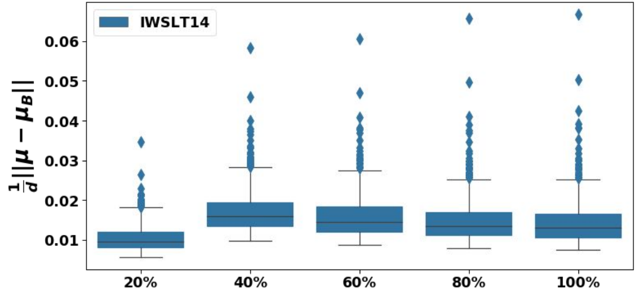
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

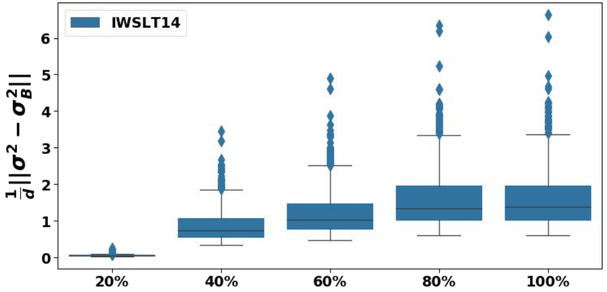
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

$$\mu := \mu + \alpha(\mu_{\mathcal{B}} - \mu) \qquad \text{// Update moving averages}$$

$$\sigma := \sigma + \alpha(\sigma_{\mathcal{B}} - \sigma)$$




(a) $\|\mu - \mu_B\|^2$ at bn.1 for **IWSLT** (training)




(b) $\|\sigma^2 - \sigma_B^2\|^2$ at bn.1 for **IWSLT** (training)

# Rethinking the variance of Batch Normalization for NLP

- **The variance in the Backward Pass:**
  - The variance of gradient w.r.t different portion of the data.
    - $g_{mean}$ and $g_{var}$ depend on the statistics of the current mini-batch, which will introduce variance.

$$\mu = \frac{1}{B}\sum_{i=1}^{N} x_i$$

$$\sigma^2 = \frac{1}{B}\sum_{i=1}^{N}(x_i - \mu)^2$$

$$\boxed{\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}}$$

$$y_i = \gamma \hat{x}_i + \beta$$

(1) **Forward Pass of BN**

$$\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2 + \epsilon}}\Big(B\frac{\partial L}{\partial \hat{x}_i} - \underbrace{\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}}_{g_{mean}} - \underbrace{\hat{x}_i \sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot \hat{x}_j}_{g_{var}}\Big)$$

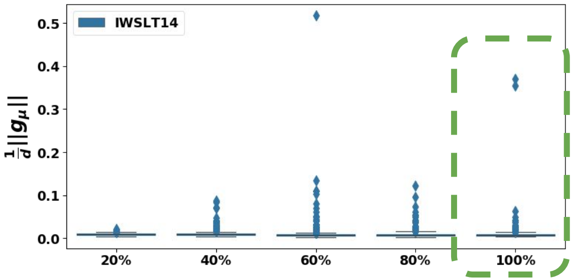
(2) **Backward Pass of BN**

# Rethinking the variance of Batch Normalization for NLP

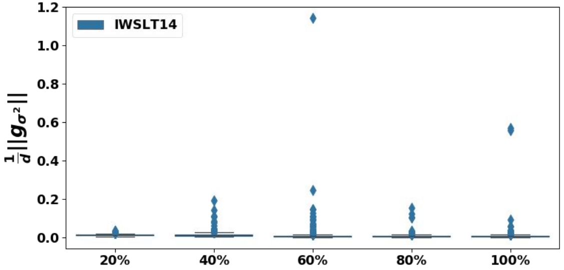
- **The variance in the Backward Pass:**
  - The variance of gradient w.r.t different portion of the data.
    - $g_{mean}$ and $g_{var}$ depend on the statistics of the current mini-batch, which will introduce variance.

$$\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2+\epsilon}}\Big(B\frac{\partial L}{\partial \hat{x}_i} - \underbrace{\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}}_{g_{mean}} - \underbrace{\hat{x}_i\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot\hat{x}_j}_{g_{var}}\Big)$$

**Backward Pass of BN**



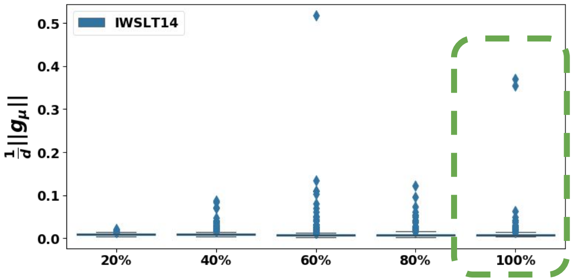

(a) The variance of $g_{mean}$ w.r.t different **IWSLT** batches




(b) The variance of $g_{var}$ w.r.t different **IWSLT** batches

# Rethinking the variance of Batch Normalization for NLP

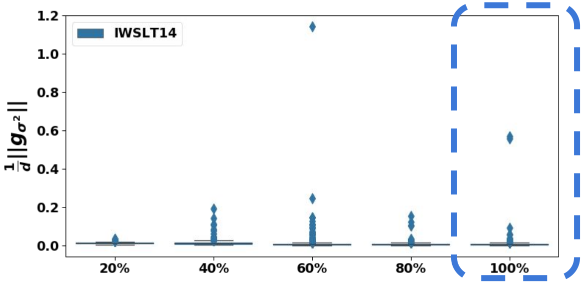
- **The variance in the Backward Pass:**
  - The variance of gradient w.r.t different portion of the data.
    - $g_{mean}$ and $g_{var}$ depend on the statistics of the current mini-batch, which will introduce variance.




(a) The variance of $g_{mean}$ w.r.t different **IWSLT** batches

$$\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2+\epsilon}}\Big(B\frac{\partial L}{\partial \hat{x}_i} - \underbrace{\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}}_{g_{mean}} - \underbrace{\hat{x}_i\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot\hat{x}_j\Big)}_{g_{var}}$$

**Backward Pass of BN**



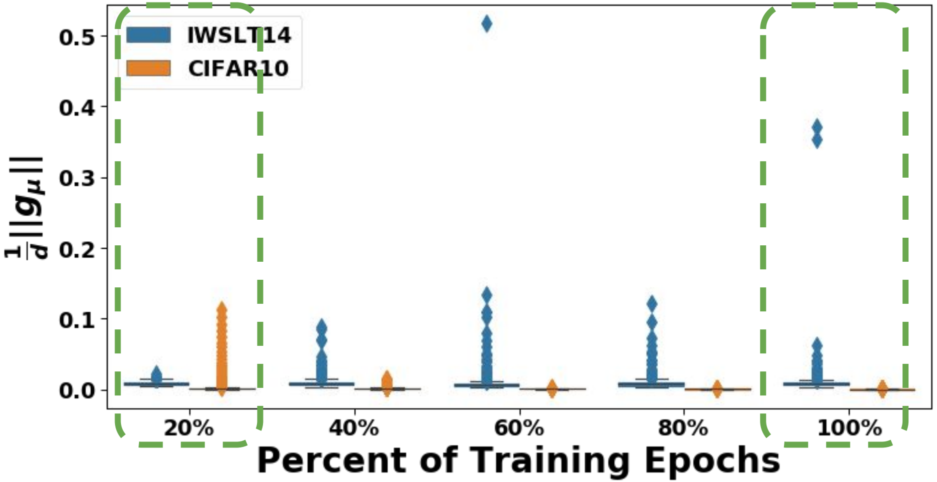

(b) The variance of $g_{var}$ w.r.t different **IWSLT** batches

# Rethinking the variance of Batch Normalization for NLP

- **The variance in the Backward Pass:**
  - The variance of gradient w.r.t different portion of the data.
    - CV vs. NLP.

$$\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2+\epsilon}}\left(B\frac{\partial L}{\partial \hat{x}_i} - \underbrace{\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}}_{g_{mean}} - \hat{x}_i\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot\hat{x}_j\right)$$

**Backward Pass of BN**



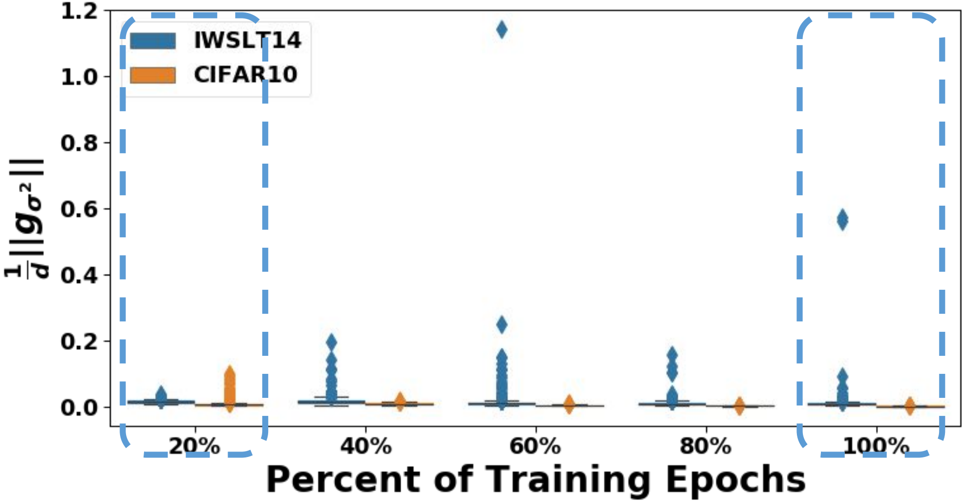

(a) The variance of $g_{mean}$ w.r.t different CIFAR10 vs IWSLT batches

# Rethinking the variance of Batch Normalization for NLP

- **The variance in the Backward Pass:**
  - The variance of gradient w.r.t different portion of the data.
    - CV vs. **NLP**.

$$\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2+\epsilon}}\left(B\frac{\partial L}{\partial \hat{x}_i} - \sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j} - \underbrace{\hat{x}_i \sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot \hat{x}_j}_{g_{var}}\right)$$

**Backward Pass of BN**




(b) The variance of $g_{var}$ w.r.t different CIFAR10 vs **IWSLT** batches

# Content

- How Normalization is performed in NLP (vs. CV)?
- Rethinking Batch Normalization for NLP (Transformers):
    - The variance in forward pass
    - The variance in backward pass
- **Our New Normalization PN and its performance in different NLP tasks.**

# PN (Power Normalization)

1. **Forward Pass: PN-V (Variance-reduced Batch Normalization)**

   1.1 Remove the recentering $(x - \mu_B)$ in the batch normalization.
   1.2 Replace the running variance $(x - \mu_B)^2$ with more accurate running $(x)^2$ while training.

2. **Backward Pass: PN**

   2.1 Replace real $(x_B)^2$ with running $(x)^2$ while training
   2.2 Stabilizing the backward pass with accumulating gradient by Exponential Moving Average approximation.

# PN (Power Normalization)

## 1. Forward Pass: PN-V (Variance-reduced Batch Normalization)

1.1 Remove the recentering $(x - \mu_B)$ in the batch normalization.

1.2 Replace the running variance $(x - \mu_B)^2$ with more accurate running $(x)^2$ while training.

$$\psi_B^2 = \frac{1}{B}\sum_{i=1}^{B} x_i^2$$

$$\hat{x}_i = \frac{x_i}{\psi_B}$$

$$y_i = \gamma \hat{x}_i + \beta$$

$$\psi^2 = \psi_B^2 + \alpha(\psi_B^2 - \psi^2)$$

// update moving averages

(1.1) **Forward Pass of PN-V**

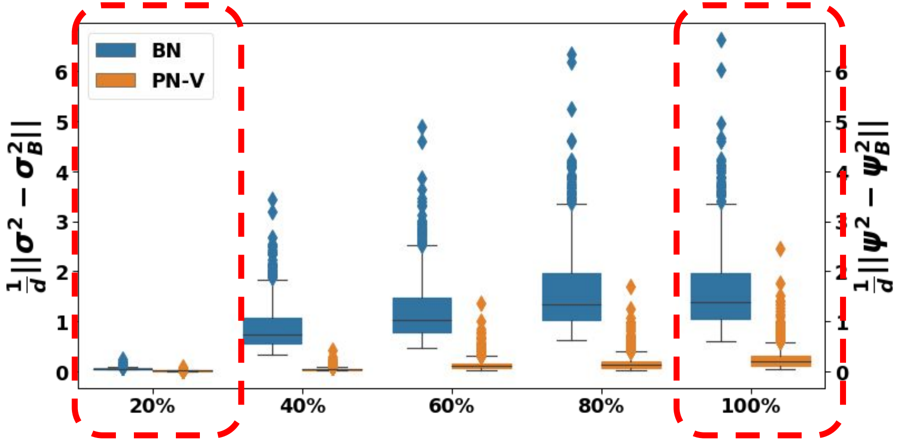
$$\mu = \frac{1}{B}\sum_{i=1}^{N} x_i$$

$$\sigma^2 = \frac{1}{B}\sum_{i=1}^{N} (x_i - \mu)^2$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta$$

(1.2) **Forward Pass of BN**




(a) $\|\sigma^2 - \sigma_B^2\|^2$ vs. $\|\phi^2 - \phi_B^2\|^2$ at bn.1 w.r.t **IWSLT** (training)

# PN (Power Normalization)

## 1. Forward Pass: PN-V (Variance-reduced Batch Normalization)

1.1 Remove the recentering $(x-\mu_B)$ in the batch normalization.

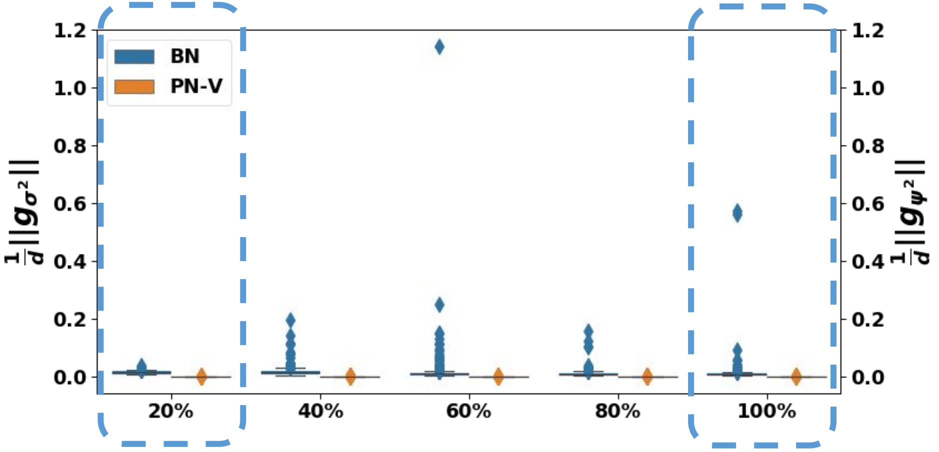
1.2 Replace the running variance $(x-\mu_B)^2$ with more accurate running $(x)^2$ while training.

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial x_i} &= \frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_i}\frac{\partial \hat{\mathrm{x}}_i}{\partial x_i} + \sum_{j\in B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_j}\frac{\partial \hat{\mathrm{x}}_j}{\partial {\psi_B}^2}\frac{\partial {\psi_B}^2}{\partial x_i} \\
&= \frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_i}\frac{\partial \hat{\mathrm{x}}_i}{\partial x_i} + \sum_{j\in B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_j}\left(-\frac{1}{2}\frac{x_j}{{\psi_B}^3}\right)\frac{2x_i}{B} \\
&= \frac{1}{\psi_B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_i} - \underbrace{\boxed{\frac{1}{B\psi_B}\sum_{j\in B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_j}\hat{\mathrm{x}}_j\hat{\mathrm{x}}_i.}}_{g_{phi}}
\end{aligned}
$$

(2.1) **Backward Pass of PN-V**

$$
\frac{\partial L}{\partial x_i} = \frac{1}{B\sqrt{\sigma^2+\epsilon}}\left(B\frac{\partial L}{\partial \hat{x}_i} - \underbrace{\boxed{\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}}}_{g_{mean}} - \underbrace{\boxed{\hat{x}_i\sum_{j=1}^{N}\frac{\partial L}{\partial \hat{x}_j}\cdot \hat{x}_j}}_{g_{var}}\right)
$$

(2.2) **Backward Pass of BN**




(b) The variance of $g_{var}$ vs $g_{phi}$ w.r.t different **IWSLT** batches

# PN (Power Normalization)

## 1. Forward Pass: PN-V (Variance-reduced Batch Normalization)

1.2 Replace the running variance $(x-\mu_B)^2$ with more accurate running $(x)^2$ while training.

$$\psi_B^2 = \frac{1}{B}\sum_{i=1}^{B} x_i^2$$

$$\psi^2 = \psi^2 + \alpha(\psi_B^2 - \psi^2) \quad \text{// update moving averages}$$

(1) **Running Statistics of PN-V**

$$\mu_B = \frac{1}{B}\sum_{i=1}^{B} x_i$$

$$\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(x_i - \mu)^2$$

$$\mu = \mu + \alpha(\mu_B - \mu) \quad \text{update moving averages}$$

$$\sigma = \sigma + \alpha(\sigma_B - \sigma) \quad \text{update moving averages}$$

(2) **Running Statistics of BN**

- The EMA (Exponential Moving Average) estimation of variance is not a desirable estimation. And it is hard to make it correct[1];

$$\begin{aligned}\sigma_t^2 &= \alpha\sigma_{t-1}^2 + (1-\alpha)(x_t-\mu_t)(x_t-\mu_{t-1}) \\ &= (1-\alpha)(\sigma_{t-1}^2 + \alpha(x_t-\mu_{t-1})^2)\end{aligned}$$

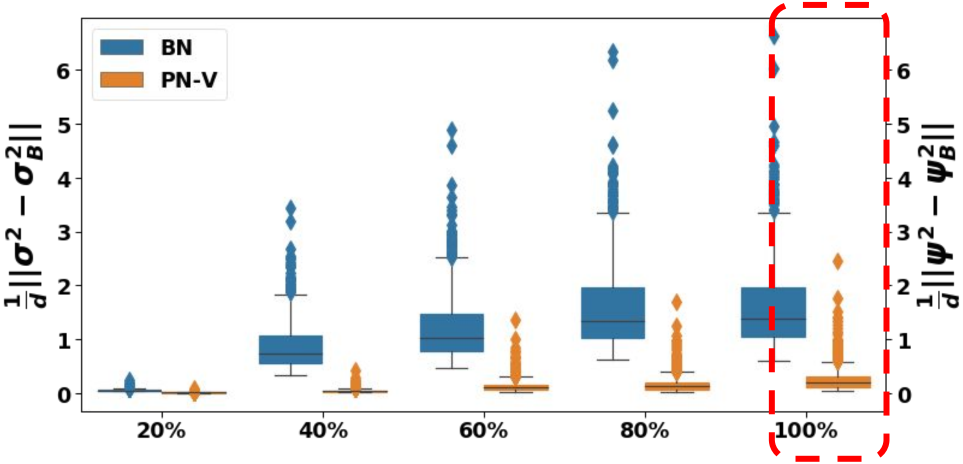
[1] Tony Finch, Incremental calculation of weighted mean and variance, Lecture Note;

# PN (Power Normalization)

## 2. Further correct the forward and backward: PN

2.1 Replace real $(x_B)^2$ with running $(x)^2$ while training (forward pass)

2.2 Stabilizing the backward pass with accumulating gradient by Exponential Moving Average approximation.




$\|\phi^2 - \phi_B^2\|^2$ at bn.1 for IWSLT (training)

# PN (Power Normalization)

## 2. Further correct the forward and backward: PN

2.1 Replace real $(x_B)^2$ with running $(x)^2$ while training (forward pass)

2.2 Stabilizing the backward pass with accumulating gradient by Exponential Moving Average approximation.

$$\psi_B^2 = \frac{1}{B}\sum_{i=1}^{B} x_i^2$$

$$\hat{x}_i = \boxed{\frac{x_i}{\psi}}$$

$$y_i = \gamma \hat{x}_i + \beta$$

$$\psi^2 = \psi_B^2 + \alpha(\psi_B^2 - \psi^2)$$

$$\text{// update moving averages}$$

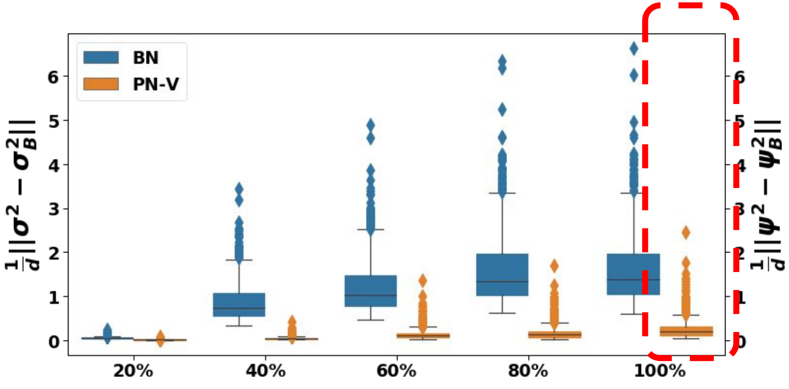
(1) **Forward Pass of PN**

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{1}{\psi_B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_i} - \underbrace{\boxed{\frac{1}{B\psi_B}\sum_{j\in B}\frac{\partial \mathcal{L}}{\partial \hat{\mathrm{x}}_j}\hat{\mathrm{x}}_j\hat{\mathrm{x}}_i.}}_{\tilde{x}'_B}$$

$$\tilde{x}'_B = \hat{x}'_B - \tau_T^{\hat{x}} \odot \hat{x}_B$$

$$\tau_T^{\hat{x}} = \tau_{T-1}^{\hat{x}} \odot [1 - (1-\alpha)\cdot \hat{x}_B \odot \hat{x}_B] + (1-\alpha)\hat{x}'_B \odot \hat{x}_B$$

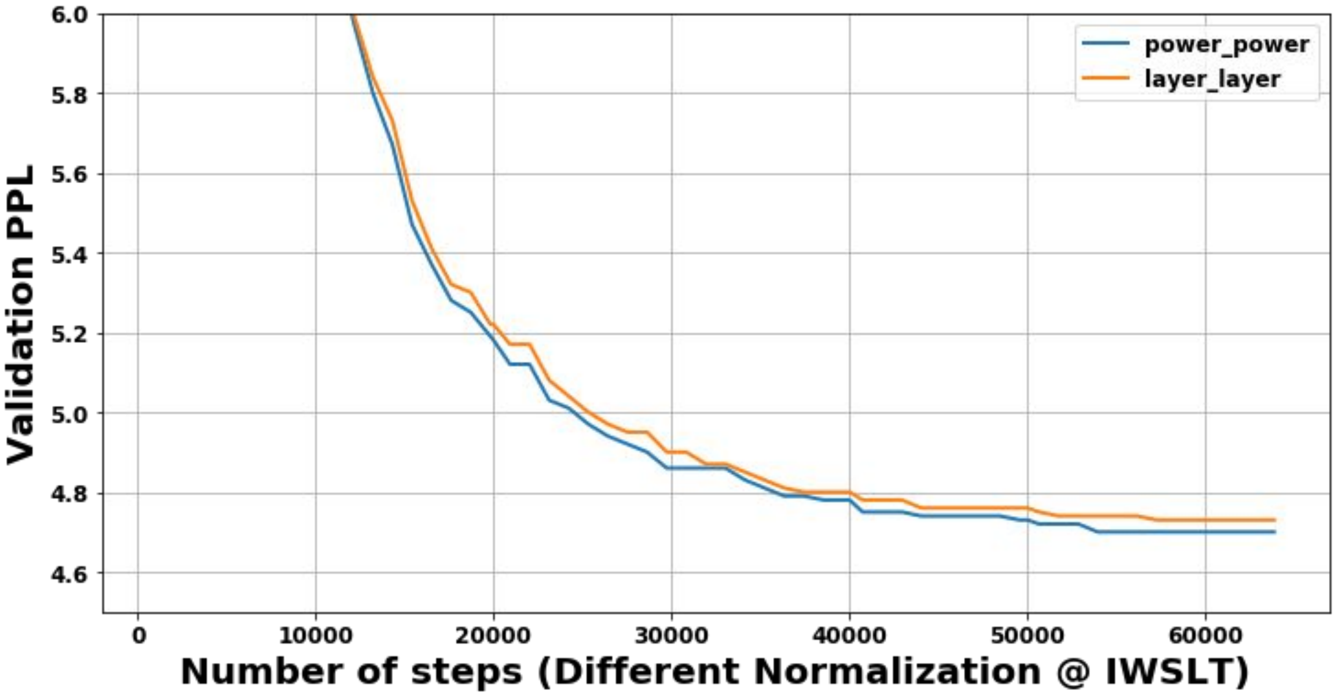
(2) **Backward Pass of PN**




$\|\phi^2 - \phi_B^2\|^2$ at bn.1 for IWSLT (training)

# PN (Power Normalization) for MT

- IWSLT14 De-En (0.16M translation pairs) Training Curve. (PN vs. LN)



# PN (Power Normalization) for MT

- IWSLT14 De-En (0.16M translation pairs)

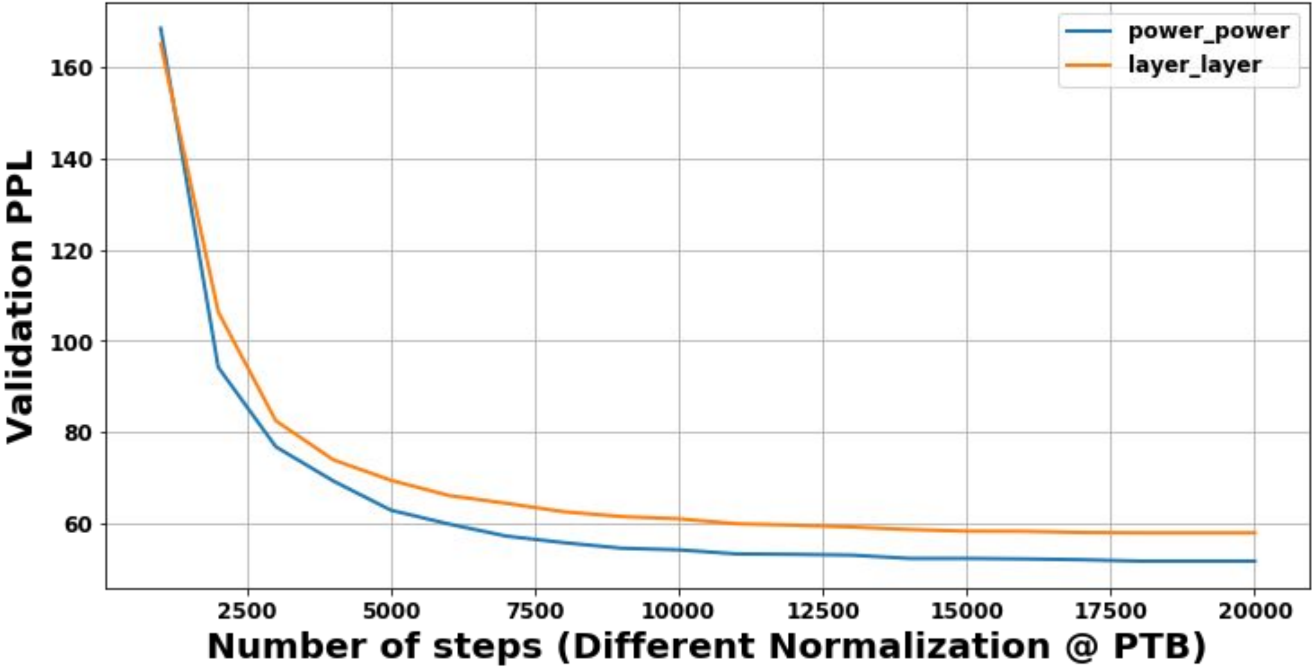
| **Small (36.7M Param)**<br>**L6+L6 D512 H4** | **BLEU** |
| --- | --- |
| Layer Norm | 35.5 |
| **Batch Norm** | **34.4** |
| PN-V (Ours) | 35.4 |
| **PN (Ours)** | **35.9** |

- WMT14 En-De (4.5M translation pairs)

| **Big (68.2M Param)**<br>**L6+L6 D1024 H16** | **BLEU** |
| --- | --- |
| Layer Norm | 29.5 |
| **Batch Norm** | **28.1** |
| PN-V (Ours) | 28.5 |
| **PN (Ours)** | **30.1** |

# PN (Power Normalization) for LM

- **PennTree Bank (0.09M tokens)**
  - Training Curve. (PN vs. LN)



# PN (Power Normalization) for LM

- **PennTree Bank (0.09M tokens)**

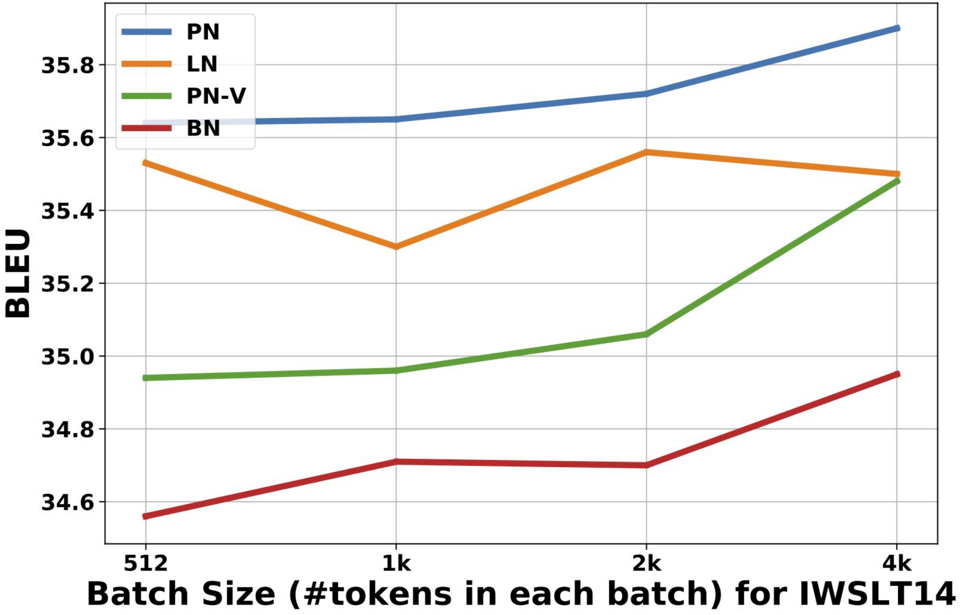
| **BDP-1Core (12.0M Param)** | **PPL (Lower is better)** |
|---|---|
| Layer Norm | 53.19 |
| **Batch Norm** | **64.72** |
| PN (Ours) | **47.32** |

- **Wikitext-103 (103M tokens)**

| **BDP-1Core (85.3M Param)** | **PPL (Lower is better)** |
|---|---|
| Layer Norm | 20.90 |
| **Batch Norm** | **27.01** |
| PN (Ours) | **18.12** |

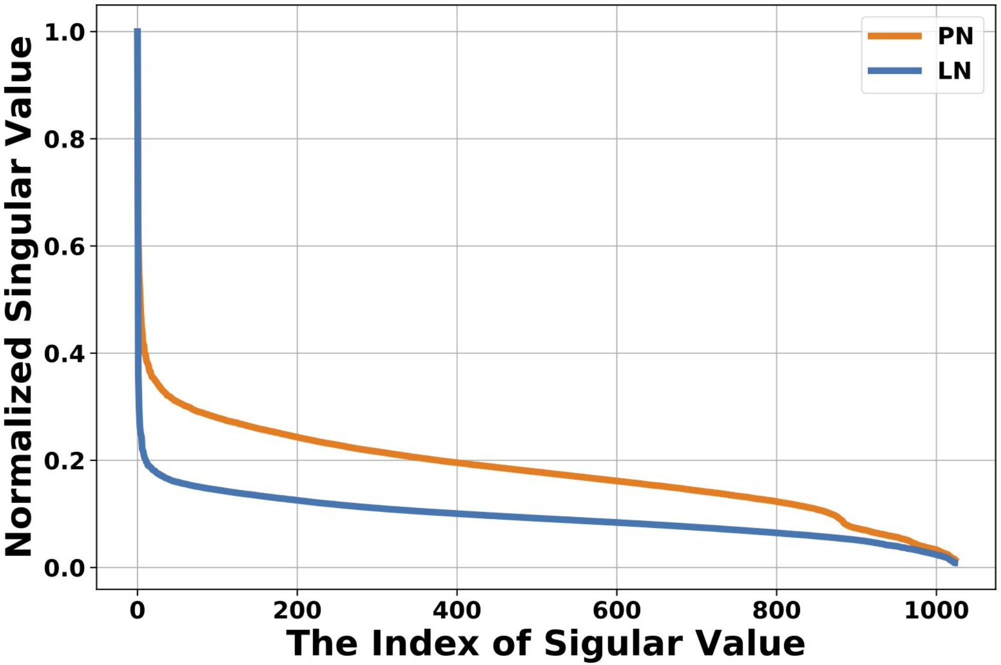
# Analysis (various batch sizes)

- PN is the most robust in various batch-size setting.



# Analysis (embedding layer)

- PN leads to a more well-conditioned embedding layer v.s. LN.



# Conclusion

- The variance between batch statistics/running statistics results in the poor performance of **Batch Normalization in Transformers**

- Reduce the variance through **PN** can achieves significantly better result in **Machine Translation, Language Modelling**
  - Improve BLEU score on IWSLT by **0.4** and WMT14 by **0.6**
  - Decrease the PPL on PTB by **5.87** and Wiki-Text by **2.78**

# Reference

**Thank you for you attention**

**Paper available at** https://arxiv.org/abs/2003.07845.

**Code available at** https://github.com/sIncerass/powernorm/