Understanding the Failure of **Batch Normalization** for Transformers in NLP

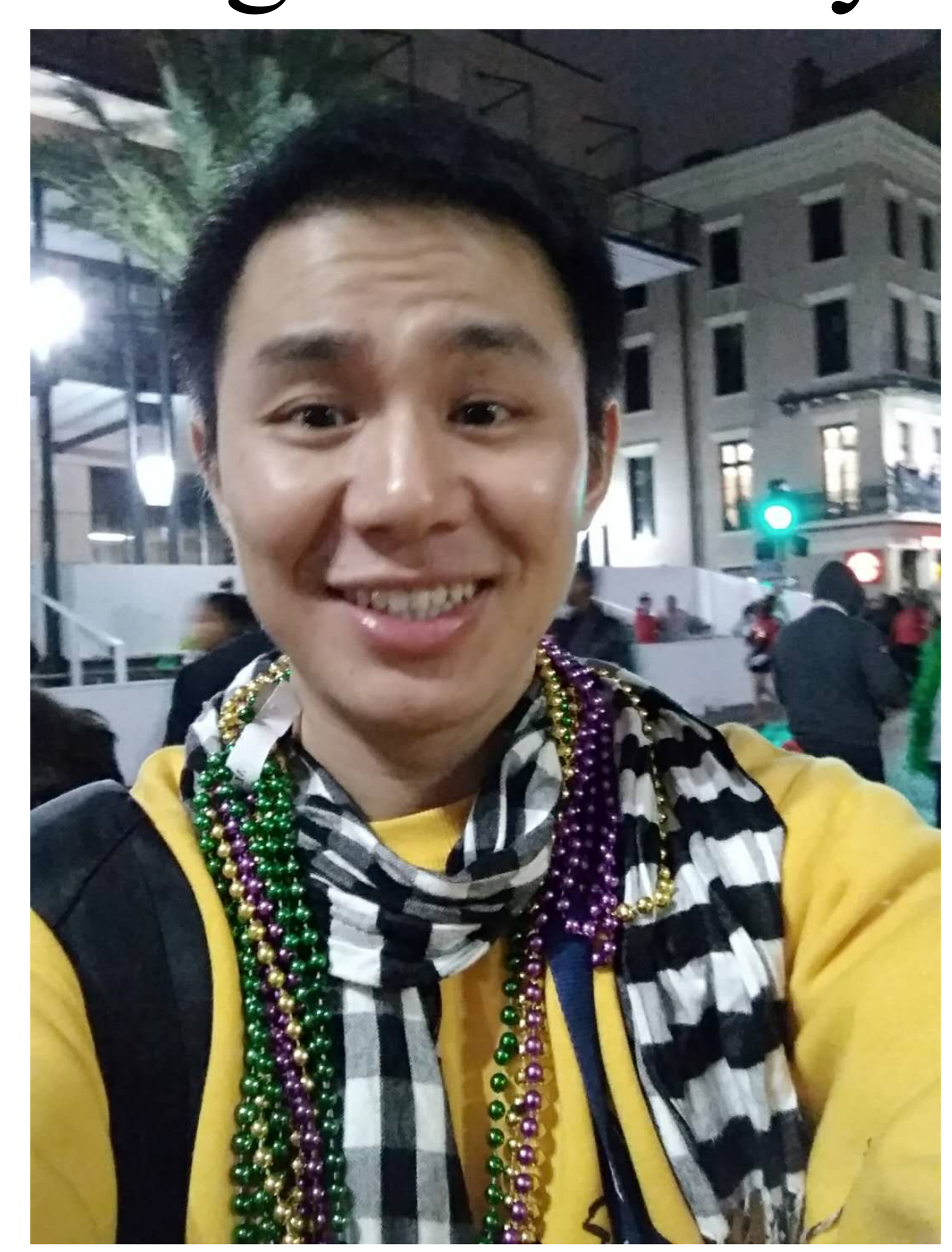
Jiaxi Wang[1], Ji Wu[1,2], Lei Huang[3]

[1]Department of Electronic Engineering, Tsinghua University

[2]Institute for Precision Medicine, Tsinghua University

[3]SKLSDE, Institute of Artificial Intelligence, Beihang University





Motivation

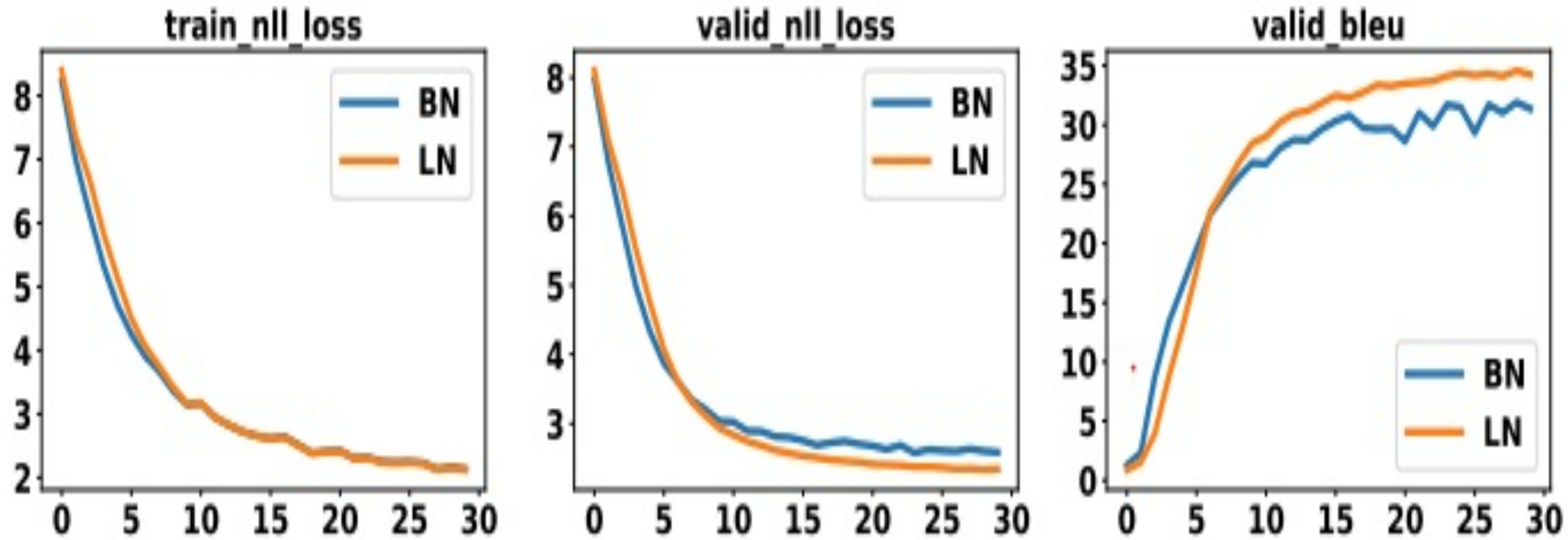
- <u>two points</u>
  - BN has better performance on CV tasks and theoretical (optimization) advantages over LN
    - **faster convergence & better test acc**(1~2%) on CIFAR10
    - **preserve numerical rank** as depth increases [1, 2]
  - BN **performs poorly in Transformer for NLP tasks** [3]

Task	NMT (+)		LM (-)		NER (+)		TextCls (+)			
Datasets	IWSLT14	WMT16	PTB	WT103	Resume	CoNLL	IMDB	Sogou	DBPedia	Yelp
Post-LN	35.5	27.3	53.2	20.9	94.8	91.3	84.1	94.6	97.5	93.3
Post-BN	34.0	25.0	45.9	17.2	94.5	90.9	84.0	94.3	97.5	93.3
Performance Gap	-1.5	-2.3	7.3	3.7	-0.3	-0.4	-0.1	-0.3	0	0
Mean TID of BN_{last}	1.5%	4.2%	0.9%	1.8%	1.7%	4.2%	1.8%	1.8%	2.2%	3.1%
Var TID of BN_{last}	10.6%	17.9%	1.1%	2.0%	3.7%	9.5%	3.9%	4.3%	3.5%	4.0%
Pre-LN	35.5	27.3	54.5	24.6	94.0	91.0	84.1	94.5	97.5	93.3
Pre-BN	34.8	25.2	45.9	17.8	93.2	89.9	84.0	94.3	97.5	93.3
Performance Gap	-0.7	-2.1	8.6	6.8	-0.8	-1.1	-0.1	-0.2	0	0
Mean TID of BN_{last}	3.4%	7.9%	1.6%	2.4%	9.6%	10.0%	2.9%	7.5%	3.9%	12.1%
Var TID of BN_{last}	4.6%	30.1%	1.7%	2.5%	6.5%	6.4%	6.2%	7.1%	3.3%	8.6%

What contributes to the failure or success of BN?

Observing training and valid loss

Post-Norm Transformer for IWSLT14 De-En machine translation

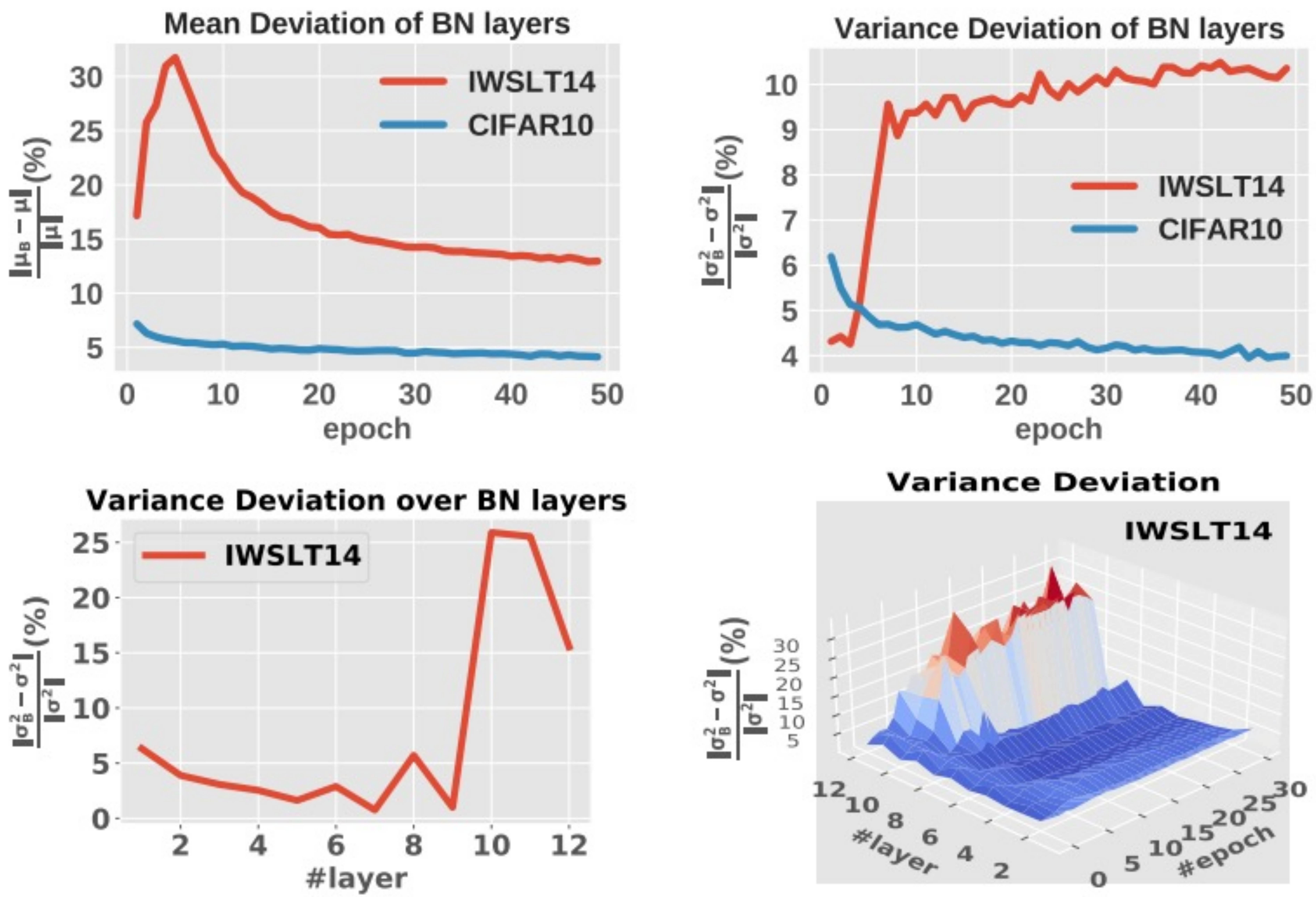


BN formula training: $$\hat{\mathbf{x}}_j = BN_{train}(\mathbf{x}_j) = \frac{\mathbf{x}_j - \mu_{B,j}}{\sqrt{\sigma^2_{B,j}}}, \;\; j = 1, 2, ..., d,$$

inference: $$\hat{\mathbf{x}}_j = BN_{inf}(\mathbf{x}_j) = \frac{\mathbf{x}_j - \mu_j}{\sqrt{\sigma^2_j}}, \;\; j = 1, 2, ..., d.$$

updating statistics: $$\begin{cases} \mu^{(t)} = (1-\alpha)\mu^{(t-1)} + \alpha\mu_B^{(t)}, \\ (\sigma^2)^{(t)} = (1-\alpha)(\sigma^2)^{(t-1)} + \alpha(\sigma^2_B)^{(t)}. \end{cases}$$

Compare Transformer with ResNet18



Training Inference Discrepancy (TID)

$$\frac{x-\mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} = \left(\frac{x-\hat{\mu}}{\hat{\sigma}} + \boxed{\frac{\hat{\mu}-\mu_{\mathcal{B}}}{\hat{\sigma}}}\right)\boxed{\frac{\hat{\sigma}}{\sigma_{\mathcal{B}}}}.$$

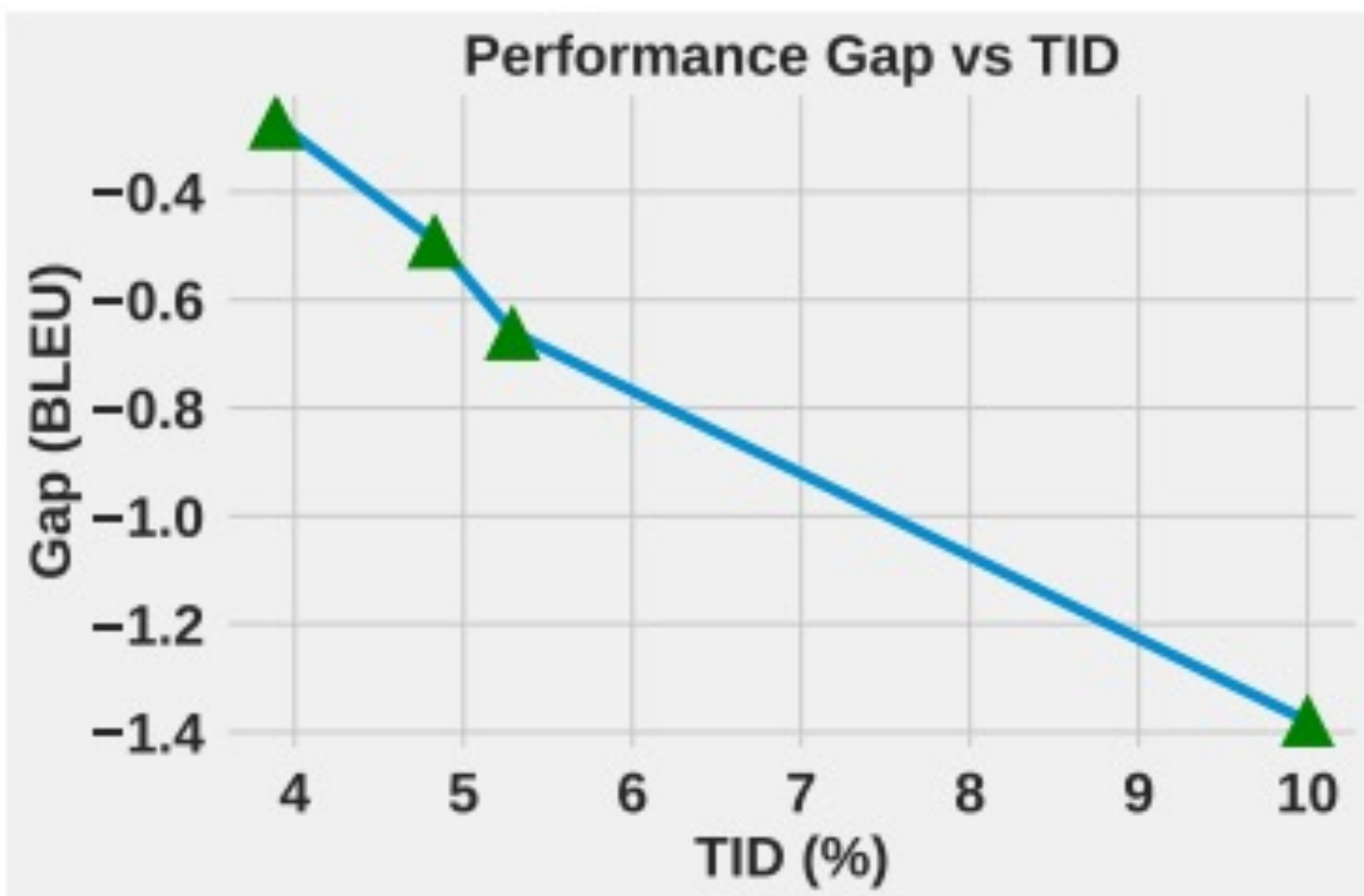
Their magnitude can **characterize the diversity of mini-batch examples** during training and **indicate how hard the estimation of population statistics** is.

$$\text{Mean TID} = \mathbb{E}_{X\sim p_B}\frac{\|\mu_B-\mu\|_2}{\|\sigma\|_2}$$

$$\text{Variance TID} = \mathbb{E}_{X\sim p_B}\frac{\|\sigma_B-\sigma\|_2}{\|\sigma\|_2}$$

TID indicates BN's performance

Dataset	WMT16 (BLEU %)	CONLL (F1 %)	IMDB (ACC %)	WT103 (PPL)
Total TID_{last}	38%	16%	9%	5%
Performance Gap	−2.1	−1.1	−0.1	6.8



- Left: Variance TID and BLEU gap between Transformer_BN and Transformer_LN when replacing different numbers of LN layers with BN
- Right: Variance TID and valid loss gap of Post-Norm Transformer through training

Penalize Discrepancy

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L}(\theta) \\ s.t. \quad & \mathbb{E}_{p_B} d_\mu(\mu_B^i, \mu^i) \le \epsilon_i, \ i = 1, \ldots, L \\ & \mathbb{E}_{p_B} d_\sigma(\sigma_B^i, \sigma^i) \le \eta_i, \ i = 1, \ldots, L \end{aligned} \longrightarrow \mathcal{L}_B(\theta) + \sum_{i=1}^{H} \lambda d_\mu(\mu_B^i, \mu^i) + \nu d_\sigma(\sigma_B^i, \sigma^i)$$

we choose $d_\mu(\mu_B, \mu) = \|\mu_B - \mu\|_2^2$ and $d_\sigma(\sigma_B, \sigma) = \|\sigma_B - \sigma\|_2^2$.

We call it **Regularized BN (RBN)**

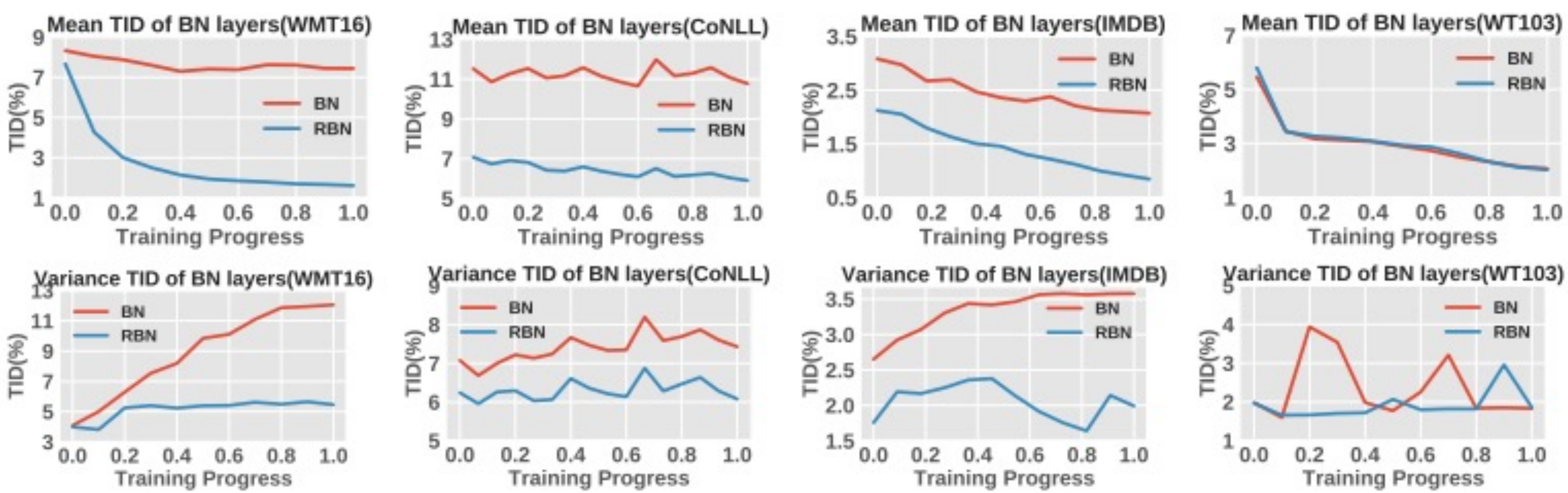
Performance of RBN

Task	NMT (+)		LM (-)		NER (+)		TextCls (+)			
Datasets	IWSLT14	WMT16	PTB	WT103	Resume	CoNLL	IMDB	Sogou	DBPedia	Yelp
Post-LN	**35.5**	**27.3**	53.2	20.9	**94.8**	91.3	84.1	94.6	97.5	93.3
Post-BN	34.0	25.0	45.9	17.2	94.5	90.9	84.0	94.3	97.5	93.3
Post-RBN	**35.5**	26.5	**44.6**	**17.1**	**94.8**	**91.4**	**84.5**	**94.7**	**97.6**	**93.6**
Pre-LN	35.5	**27.3**	54.5	24.6	**94.0**	**91.0**	84.1	94.5	**97.5**	93.3
Pre-BN	34.8	25.2	45.9	17.8	93.2	89.9	84.0	94.3	**97.5**	93.3
Pre-RBN	**35.6**	26.2	**43.2**	**17.1**	**94.0**	90.6	**84.4**	**94.7**	**97.5**	**93.5**

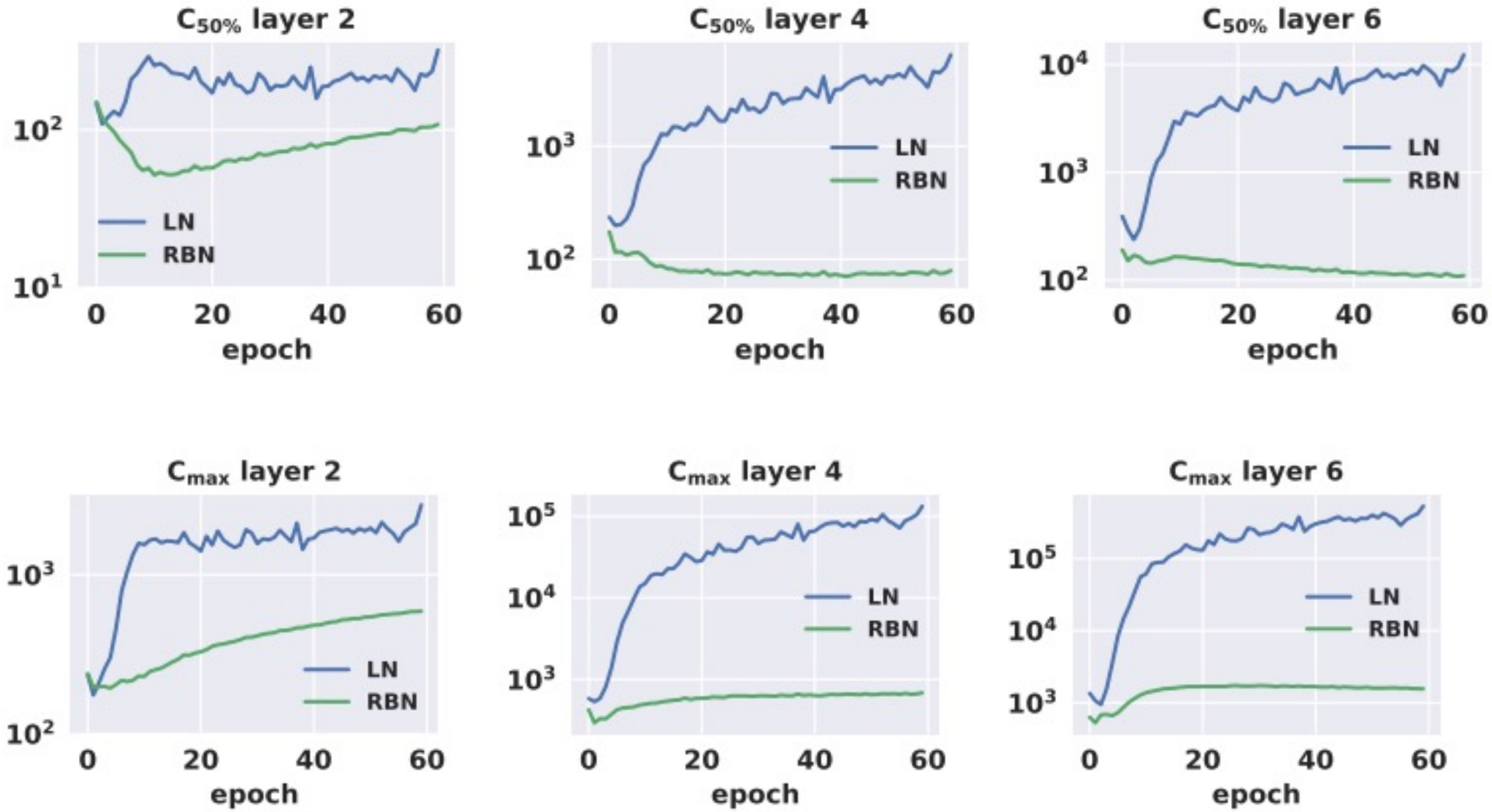
Performance of RBN compared to BN variants

Task	NMT (+)		LM (-)		NER (+)		TextCls (+)			
Datasets	IWSLT14	WMT16	PTB	WT103	Resume	CoNLL	IMDB	Sogou	DBPedia	Yelp
Post-PN-only	0	0	254.6	inf	94.4	67.1	84.2	90.6	97.1	89.6
Post-PN+LS	**35.6**	0	49.8	21.0	94.3	90.9	84.0	94.6	97.4	93.2
Post-BRN	35.3	25.8	45.1	17.3	93.6	89.9	83.6	94.5	97.5	93.3
Post-MABN	0	0	47.4	33.6	94.4	90.8	84.1	94.5	97.5	93.5
Post-RBN	35.5	**26.5**	**44.6**	**17.1**	**94.8**	**91.4**	**84.5**	**94.7**	**97.6**	**93.6**
Pre-PN-only	34.5	26.0	48.6	inf	5.0	11.1	84.2	94.4	97.4	93.3
Pre-PN+LS	**35.6**	**27.2**	59.8	20.9	93.3	54.1	83.3	94.4	97.3	93.4
Pre-BRN	35.2	25.3	45.7	17.5	94.1	**91.1**	84.3	94.5	97.4	93.4
Pre-MABN	35.0	26.8	48.7	inf	**94.8**	90.9	**84.4**	94.6	97.5	93.3
Pre-RBN	**35.6**	26.2	**43.2**	**17.1**	94.0	90.6	**84.4**	**94.7**	**97.5**	**93.5**

Ablation study of RBN: RBN reduces the TID of BN



Layer-wise Training Dynamics [4]



$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{BT}] \in \mathbb{R}^{BT \times d}$$

$$C_p(\mathbf{X}) = \frac{\sigma_1}{\sigma_{\lceil pd \rceil}}, 0 < p \leq 1. \qquad C_{max}(\mathbf{X}) = \lambda_{max}((\mathbf{X}^T\mathbf{X})^{\frac{1}{2}})$$

Conclusion and Limitation

https://github.com/wjxts/RegularizedBN

✓ We defined Training Inference Discrepancy (TID) and showed that **TID is a good indicator of BN's performance for Transformers**, supported by comprehensive experiments.

✓ We observed **BN performs much better than LN when TID is negligible** and proposed Regularized BN (RBN) to alleviate TID when TID is large.

✓ Our RBN has theoretical advantages in optimization and works empirically better by controlling the TID of BN when compared with LN.

Limitations:

☐ Still worse than LN on WMT16 (large dataset, large data diversity)

☐ It is better to further model the geometric distribution of word embedding, evolving along with the training dynamics and information propagation, with theoretical derivation under mild assumptions

- We welcome questions and discussions!

References

[1] Hadi Daneshmand, Jonas Moritz Kohler, Francis Bach, Thomas Hofmann, and Aurélien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In NeurIPS, 2020.

[2] Hadi Daneshmand, Amir Joudaki, and Francis Bach. Batch normalization orthogonalizes representations in deep random networks. In NeurIPS, 2021.

[3] Sheng Shen, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. Powernorm: Rethinking batch normalization in transformers. In ICML, 2020.

[4] Lei Huang, Jie Qin, Li Liu, Fan Zhu, and Ling Shao. Layer-wise conditioning analysis in exploring the learning dynamics of dnns. In ECCV, 2020.