

Bug or Feature²: Weight Drift and Activation Sparsity

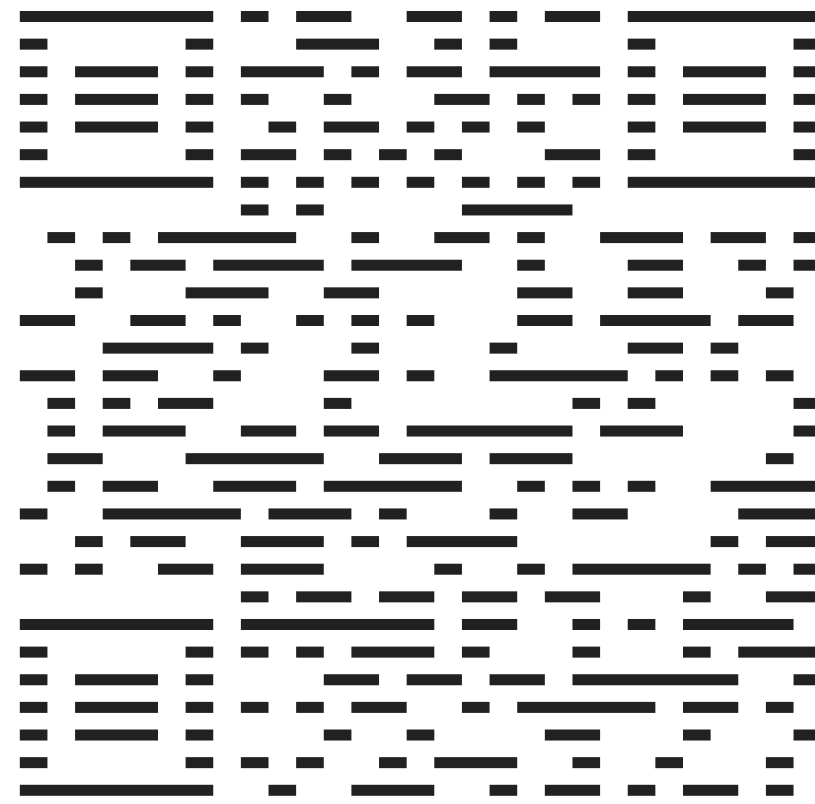
Egor Shvetsov^{1*} Aleksandr Serkov^{2*} Viacheslav Shokorov¹ Dmitry Redko¹ Vladislav Goloshchapov³ Evgeny Burnaev^{1,4}

¹Applied AI Institute ²HSE ³Independent researcher ⁴AXXX ^{*}Equal contribution

NeurIPS 2026

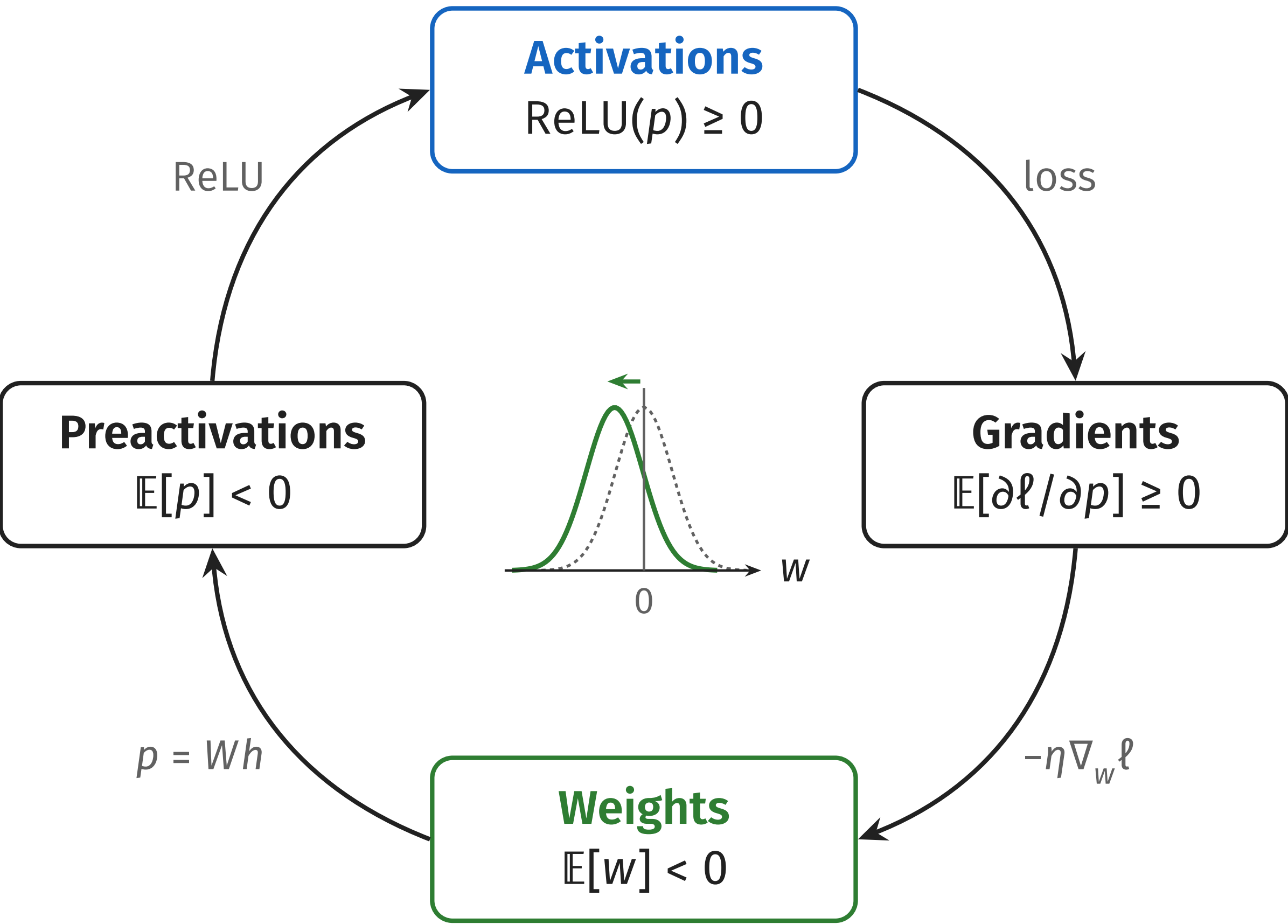
Paper
arxiv.org/pdf/2605.17659

Contact
geksut@gmail.com



Standard losses with ReLU, GELU or SiLU push weights **negative** in the first iterations. The resulting sparsity is **mostly a feature**.

1 When weights drift toward negative values



We theoretically prove that

for every neuron i in any hidden layer
 $\mathbb{E}[\partial \ell / \partial p_i] \geq 0$, strict when $p_i > 0$
so gradient descent drives weights negative.

Necessary conditions

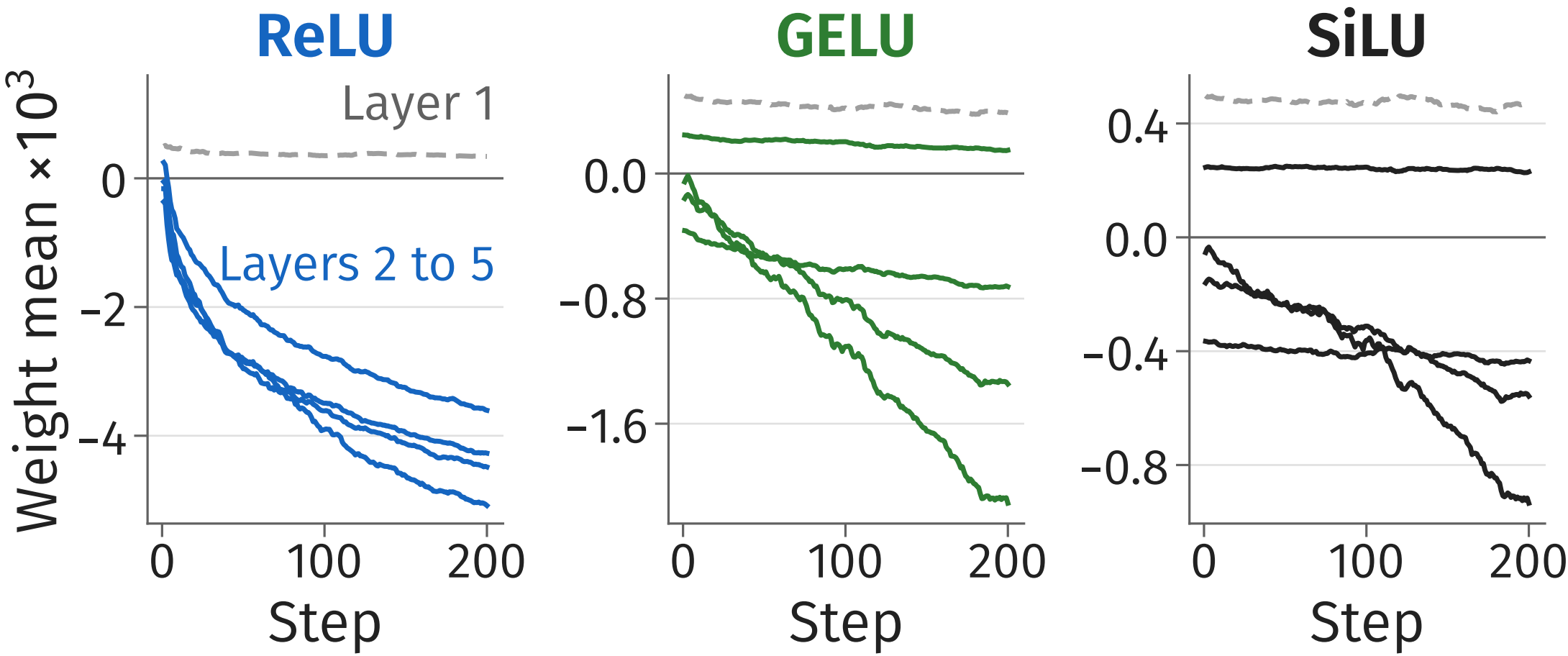
- ReLU, no mean centering normalization
- Zero mean i.i.d. weights, at initialization
- MSE, or cross entropy up to a small softmax term

After initialization

Our proof strictly applies only at initialization. However, the drift persists after the first iterations. Gradients are largest while predictions are wrong, so most drift happens early. Later, gradients decay and the offset stays.

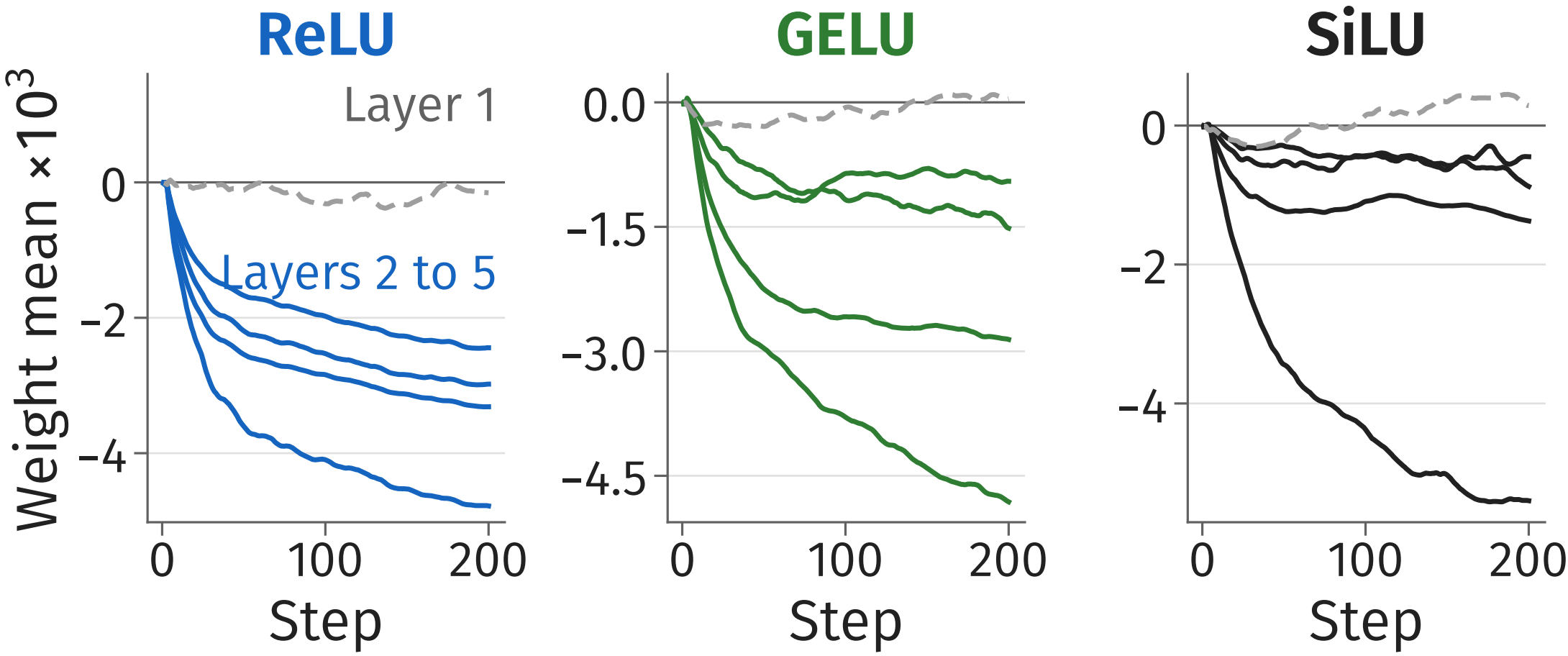
2 Experimental evidence

Random inputs and targets



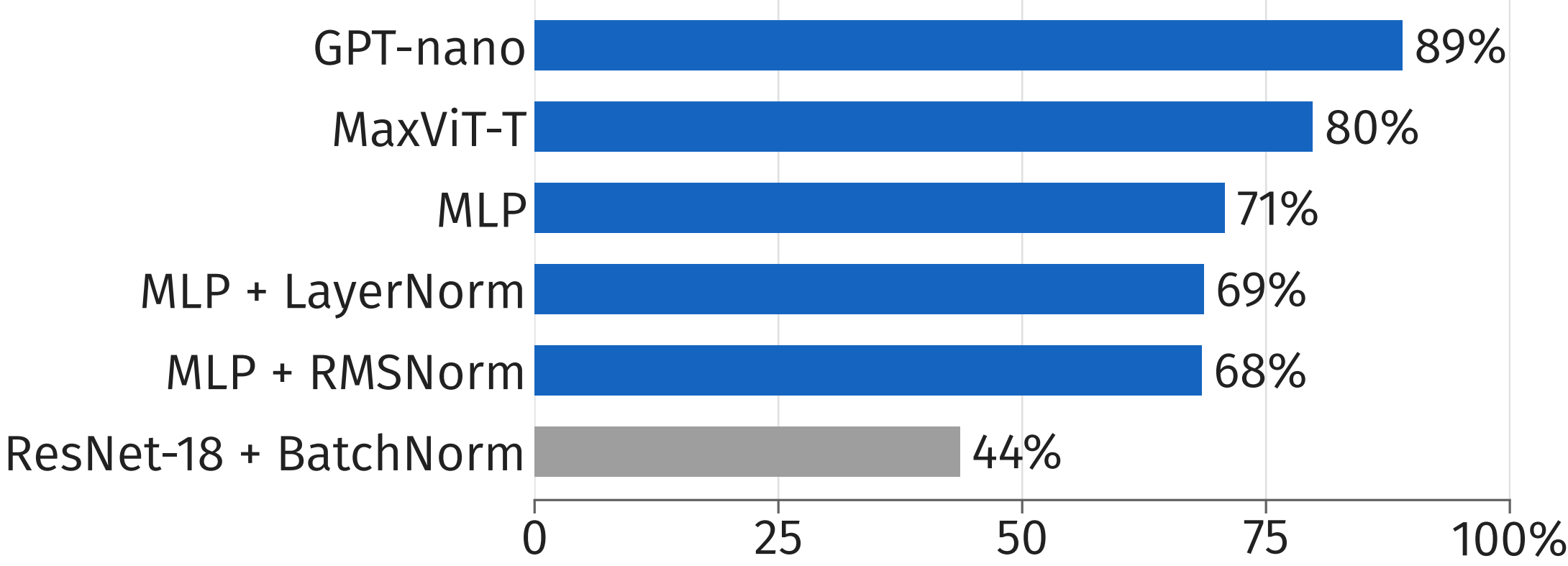
MLP trained with SGD on noise. Layer 1 sees raw inputs and does not drift. Most later layers do.

Real data, cross entropy



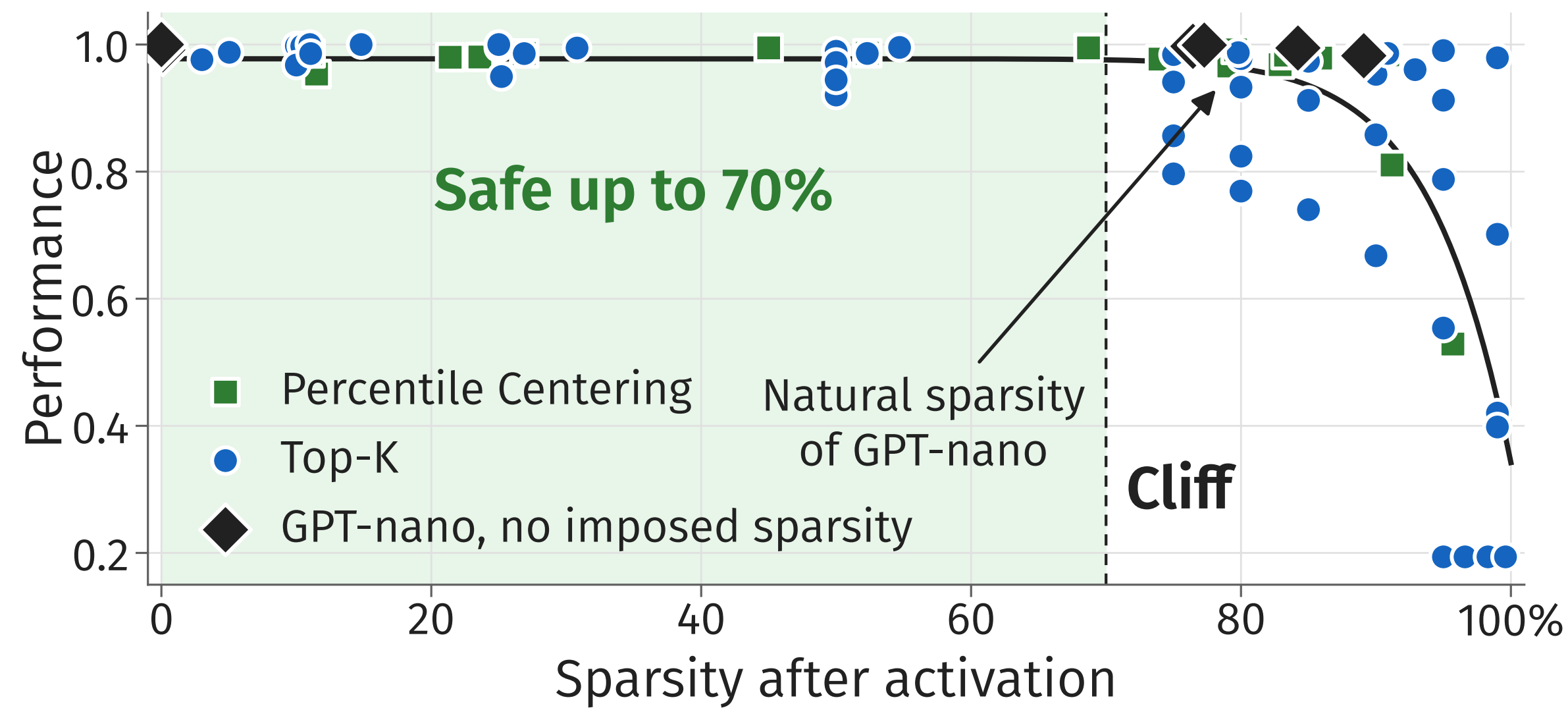
MLP on CIFAR-10 with Adam, 5 runs. Same pattern.

Negative preactivations with ReLU



Consider models without BatchNorm or centering since it subtracts the mean and cancels drift.

3 Is natural sparsity damaging?



79 configurations, four architectures. Fit $A - B s^N$ with $N = 16.7$.

Skip connections help

At 95% sparsity an MLP keeps 27% with a skip, 10% without.

Activation functions and drift

	Accuracy	Negative values %	GPT loss
ReLU	59.1	66.2	3.290
GELU	61.2	61.8	3.260
SiLU	61.2	60.5	n/a
NoisyReLU	60.3	65.2	3.287
SUGARBSiLU	54.8	68.0	3.291
ReLU ²	51.9	68.3	3.251

Average over five vision settings. ReLU² wins on GPT-nano but needs LayerNorm or RMSNorm.

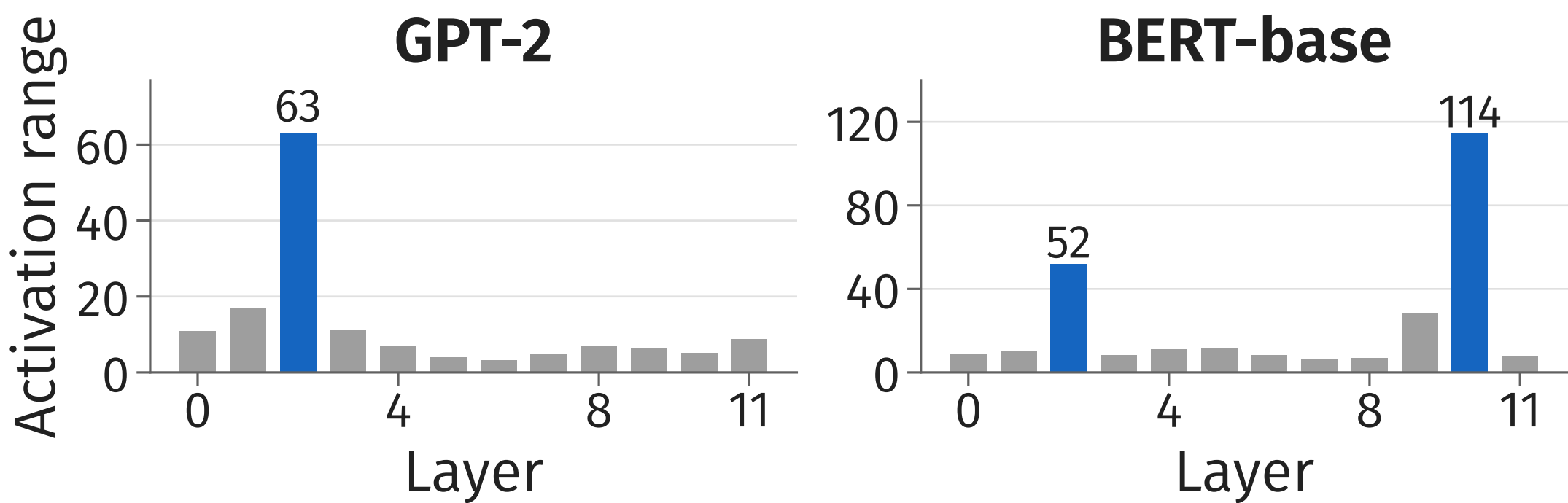
ReLUfication, one epoch GELU to ReLU

Accuracy	GELU	ReLU	Sparsity
MLP	51.5	51.6	70%
ResNet-18	94.0	93.1	58%
MaxViT-T	70.3	70.2	74%

Bug or Feature?

Feature, up to 70% sparsity

4 Spikes



Pretrained language models have a few very large activations in early layers. Vision models do not.

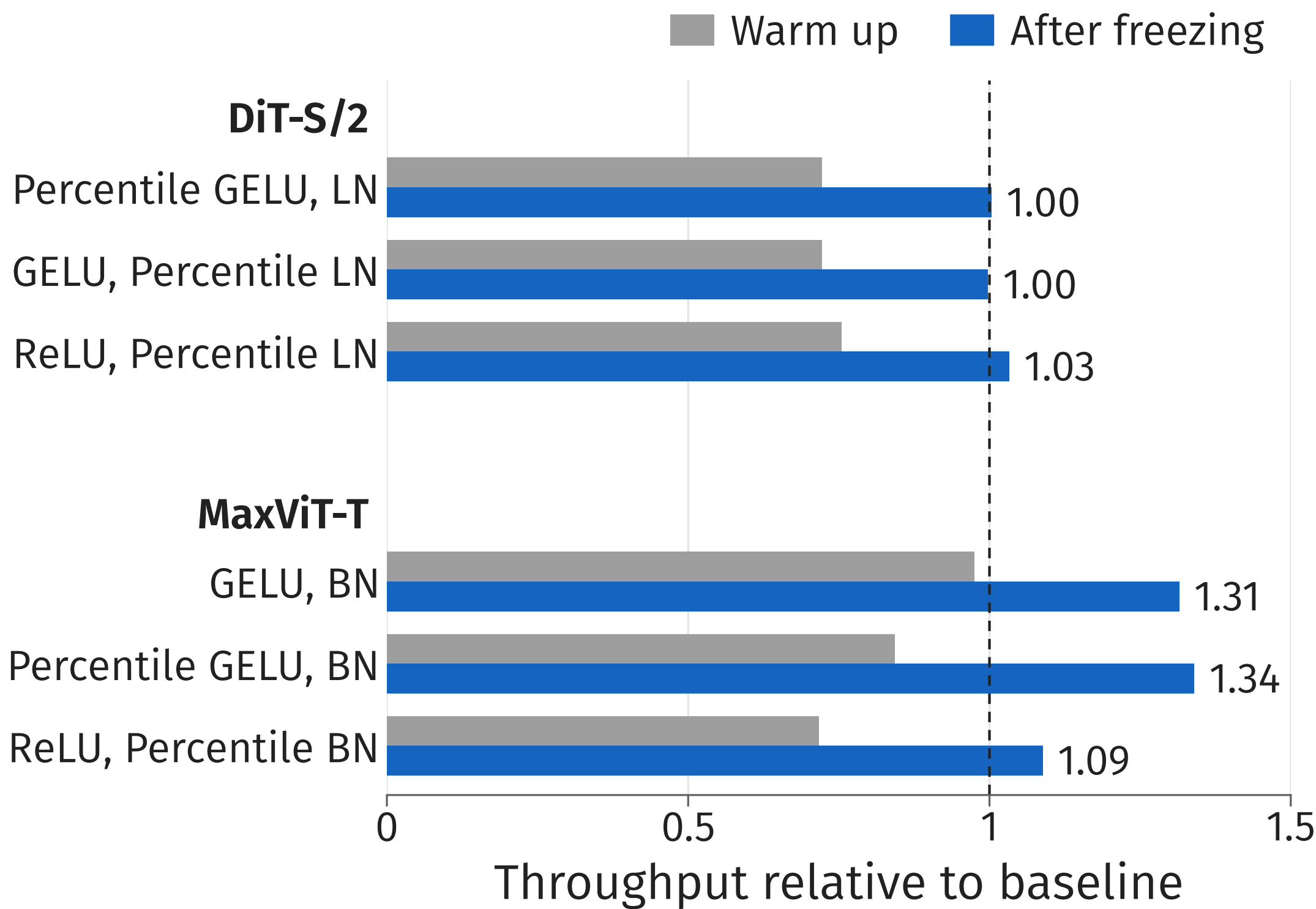
ReLU² amplifies such spikes in GPT-nano. Clipping at 50 removes them and keeps the loss.

Bug or Feature?

Amplified spikes may be a bug

5 Drift and efficiency

Drift settles early, so normalization statistics can be frozen after warm up.



Warm up is slower because our percentile code is plain PyTorch.