CAFFEINATE: RECOVERING DORMANT SUBNETWORKS IN MODERN CONVOLUTIONAL NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

One-shot pruning is understood to degrade accuracy exclusively due to the removal of critical weights from a model. Evidence from batch-normalized CNNs supports a different account. The pruned model retains a high-performing sub-network that signal collapse, which is a progressive, depth-wise decay of activation variance, renders *dormant*, and recalibrating the BatchNorm statistics recovers it (Saikumar & Varghese, 2025). Modern convolutional architectures, such as ConvNeXt, FocalNet, HorNet, and RDNet, adopt LayerNorm instead of BatchNorm, which computes statistics per sample at inference time. LayerNorm contains no stored statistics to recalibrate. We demonstrate that signal collapse exists in these networks and that damage due to pruning lies in the per-channel output moments, the subspace the LayerNorm affine parameters control. We develop CAFFEINATE, a closed-form correction of LayerNorm's affine parameters, which is applied once to a pruned network to restore the original per-channel output moments from only forward passes, without gradients or labels. Since CAFFEINATE rewrites only the affine parameters, it can be applied to any one-shot pruning mask. We validate this on magnitude pruning (MP) and CHITA, a state-of-the-art impact-based pruning method which uses second order information to select which parameters to prune and to update the unpruned parameters. CAFFEINATE recovers up to $+58.4\%$ accuracy over MP across twelve ImageNet networks. MP corrected by CAFFEINATE outperforms CHITA on ten of twelve models at $\kappa = 50\%$ and all twelve models at $\kappa = 60\%$, by up to $+47.4\%$ accuracy, with a memory footprint of 2.5–2.8 GB against CHITA's 25–64 GB. When applied to CHITA, CAFFEINATE improves the accuracy of the pruned model by up to $+54.2\%$. This demonstrates that dormant high-performing sub-networks exist after one-shot pruning, regardless of the pruning method, and that correcting per-channel output moments can recover accuracy.

1 INTRODUCTION

Neural network pruning involves removing parameters from a model to reduce the cost of inference (LeCun et al., 1989; Han et al., 2015; Blalock et al., 2020). The pruning regime with practical benefit is *one-shot* pruning—prune once with no retraining. However, it causes a sharp accuracy loss, conventionally attributed to the removal of critical weights. This accuracy loss can be mitigated by improving which weights are removed, i.e. weight *selection*. Impact-based pruning (IP) methods, such as WoodFisher (Singh & Alistarh, 2020), CBS (Yu et al., 2022c), and CHITA (Benbaki et al., 2023), use second-order Hessian-based information for weight selection, followed by an update of the surviving weights. While IP methods achieve better accuracy than magnitude pruning (MP), they require hours to prune even small models (Benbaki et al., 2023; Yu et al., 2022c).

For CNNs with Batch Normalization (BN) (Ioffe & Szegedy, 2015), the accuracy of a one-shot pruned network depends less on the choice of the pruning method (i.e. MP or IP) than on whether signal propagation is maintained. Saikumar & Varghese (2025) decoupled IP into weight selection and weight update, showing that IP-based weight selection performs comparably to MP. The accuracy gain of IP methods comes from the Hessian-based *weight update*. Saikumar & Varghese (2025) instead attributes the accuracy loss of one-shot pruned networks to *signal collapse*, the progressive decay of activation variance across layers which renders the one-shot pruned network unusable. In a pruned model, BN continues to normalize the activations using statistics inherited from the dense model, which in turn over-normalizes the activations. This reduces the activation variance

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

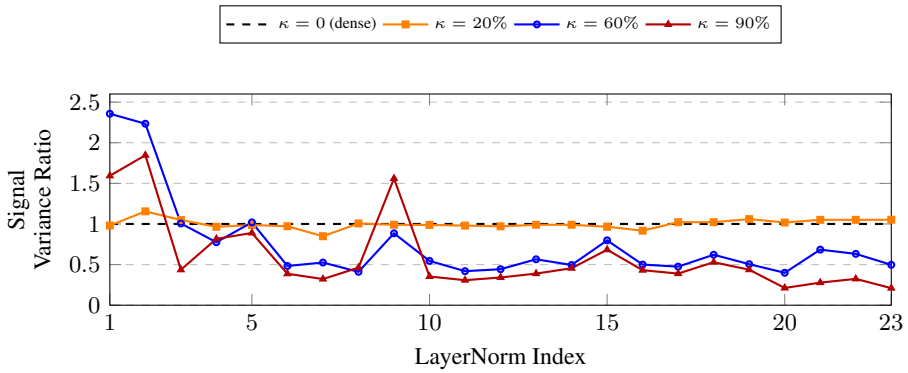


Figure 1: Measured signal variance ratios, as defined by Equation (3), after each of the LN layer of one-shot magnitude-pruned ConvNeXt-T on ImageNet, for sparsity, $\kappa = 0$, 20, 60 and 90%.

on a per-layer level, resulting in an activation variance decay across the network, causing distinct inputs to map to indistinguishable representations. Hessian-based updates improve accuracy by partially and indirectly restoring activation variance. Based on this observation REFLOW (Saikumar & Varghese, 2025) recalibrates the BN statistics of the pruned network using forward passes on a small unlabeled calibration set, thereby directly restoring the activation variance. This allows MP achieve comparable or higher accuracy than state-of-the-art Hessian-based approaches. For example, REFLOW restores ResNeXt-101 on ImageNet from 4.1% to 78.9% top-1 at 80% sparsity in a few seconds. Thus in BN-based CNNs, mitigating signal collapse closes the accuracy gap between MP and IP. One-shot pruning therefore produces high-quality yet *dormant* sparse sub-networks. Nonetheless, these *dormant* sub-networks can be recovered without modifying any weights in the convolution or linear layers and without any retraining.

While REFLOW recalibrates the BN statistics by overwriting the outdated statistics, modern CNNs, however, increasingly replace BN with Layer Normalization (LN). For example, ConvNeXt (Liu et al., 2022), FocalNet (Yang et al., 2022), HorNet (Rao et al., 2022), and RDNet (Kim et al., 2024) match or outperform the accuracy of vision transformers, such as ViT (Dosovitskiy et al., 2021) and Swin (Liu et al., 2021). These LN-based CNNs (Ba et al., 2016) are therefore a relevant setting for modern CNN pruning. However, REFLOW cannot be applied within this context, since LN computes statistics *per sample, at inference time* and stores no statistics. This raises two questions: *does signal collapse occur in LN-based CNNs, and if so, how can it be mitigated?*

Figure 1 answers the first question: the dense and 20%-sparse models maintain a near-unit signal variance ratio considered in Section 2 across all LN layers in ConvNeXt-T. At higher sparsities, signal collapse is observed. At $\kappa = 90\%$ the final LN layer transmits only 21% of the original variance. The activation variance decays with increasing layer depth and sparsity. Signal collapse is therefore not limited to BN-based CNNs, but also occurs in modern LN-based CNNs. Thus, one-shot pruning of CNNs creates dormant sub-networks, motivating the need for a mechanism to recover them.

This paper develops a mechanism to recover dormant sub-networks in modern LN-based CNNs. We first characterize the damage caused by pruning in Section 3. LN and BN reduce (or normalize) over the *orthogonal* axes of the activation tensor, but both apply the same per-channel affine transformation. For BN the affine is aligned with the reduction, so the affine *is* the output's per-channel mean and standard deviation. For LN the reduction is crosswise, and the per-channel output moments carry data-dependent factors. Pruning distorts these data-dependent factors, which results in signal collapse. The contribution to the output moments lies in the subspace controlled by the per-channel affine transformation, which makes a closed-form correction possible. We propose CAFFEINATE, considered in Section 4, which measures per-channel moments of the original and pruned network on an unlabeled calibration set and rewrites each LNs affine parameters using only forward passes and no labels. To our knowledge, this is the first method to recover dormant sub-networks in modern LN-based one-shot pruned CNNs without any Hessians or retraining.

The **contributions** of this work are as follows:

1. **Identifying signal collapse beyond BN-based CNNs**: We show that one-shot pruning of LN-based CNNs causes the signal variance ratio at most layers to fall below one, with the ratio decreasing with sparsity and layer depth (Figure 1).
2. **Developing CAFFEINATE**. We introduce a closed-form, forward-pass only, label-free correction of LN affine parameters that can be applied to any one-shot pruning method, improving the top-1 accuracy of CHITA, a state-of-the-art one shot pruning method by up to $+54.2\%$ on ImageNet.
3. **Comprehensive evaluation**: Across twelve CNNs covering ConvNeXt, FocalNet, HorNet, and RDNet at three different scales on ImageNet, CAFFEINATE applied to MP reaches 90–99% of the dense model's accuracy at $\kappa = 30\%$, recovers top-1 accuracy by up to $+58.4\%$ at $\kappa = 40\%$.

2 BACKGROUND

We first establish notation and then formalize signal collapse.

2.1 ONE-SHOT PRUNING, AND WHERE ITS ACCURACY COMES FROM

Consider a pre-trained network $f(\theta; x)$ with pre-trained weights $\overline{\theta} \in \mathbb{R}^d$. Pruning produces a sparse sub-network $f(\theta \odot m; x)$, where $m \in \{0,1\}^d$ is a binary mask and the sparsity $\kappa \in [0, 100]\%$ is the percentage of parameters set to zero. Magnitude pruning (MP) scores weights by their absolute magnitude $z_i = |\overline{\theta}_i|$ (Han et al., 2015; Frankle & Carbin, 2019). Impact-based pruning (IP) scores weights by their impact on loss, following the Optimal Brain Surgeon procedure (Hassibi & Stork, 1992):

$$z_i = \frac{\overline{\theta}_i^2}{2[H^{-1}]_{ii}}, \qquad \delta\theta^* = \frac{-\overline{\theta}_i [H^{-1}] e_i}{[H^{-1}]_{ii}}, \tag{1}$$

where H is the Hessian at $\overline{\theta}$ and $\delta\theta^*$ is the update to surviving weights. WoodFisher (Singh & Alistarh, 2020), CBS (Yu et al., 2022c), and CHITA (Benbaki et al., 2023) scale this approach to modern networks by approximating the Hessian. Saikumar & Varghese (2025) decoupled the two steps of Equation (1) on BN-based CNNs, and found that weight-selection-only variants of IP improve on MP by at most $\sim 2\%$ accuracy. The Hessian weight update, on the other hand, produces the observed performance gain over MP. We identify that a similar trend holds for LN-based CNNs as shown in Figure 2. CHITA-S, the weight selection-only variant, uses the score (z_i) of Equation (1) to select which parameters to prune. CHITA-U updates the weights by $\delta\theta^*$ after weight selection. On the three models of Figure 2, CHITA-S changes accuracy over MP between -1.6% and $+0.7\%$ while CHITA-U improves accuracy up to $+35.2\%$. Therefore, model accuracy in one-shot CNN pruning can be attributed to the Hessian-based weight update, and not to which weights are selected as determined by the choice of pruning mask, i.e., MP or IP.

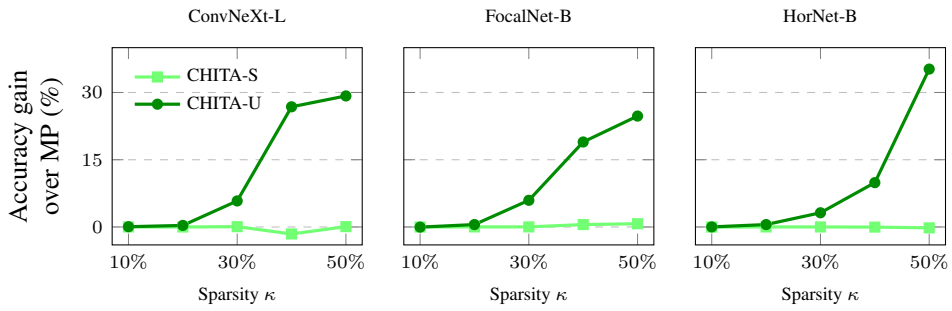


Figure 2: Top-1 accuracy gain over MP of CHITA-S (weight selection-only) and CHITA-U (selection followed by Hessian weight update, i.e., full CHITA) for one-shot pruned ConvNeXt-L, FocalNet-B and HorNet-B on ImageNet.

2.2 SIGNAL COLLAPSE

For a layer $\ell \in \{1, \ldots, L\}$ with input $\mathbf{H}_{\ell-1}$, the pre-normalization activation and the output of the BN (Ioffe & Szegedy, 2015) layer are

$$\mathbf{X}_\ell = f(\mathbf{H}_{\ell-1}; \theta_\ell), \qquad \mathbf{Z}_\ell(n) = \frac{\mathbf{X}_\ell(n) - \mu_\ell}{\sqrt{\mathrm{Var}_\ell^{(\mathrm{Orig})}(\mathbf{X}_\ell) + \epsilon}} \cdot \gamma_\ell + \beta_\ell, \tag{2}$$

where n indexes the batch, θ_ℓ are layer ℓ's parameters, μ_ℓ and $\mathrm{Var}_\ell^{(\mathrm{Orig})}(\mathbf{X}_\ell)$ are the stored running mean and variance (statistics), γ_ℓ, β_ℓ are the affine parameters, $\epsilon > 0$ is a small constant for numerical stability, and $\mathbf{H}_\ell = \mathbf{Z}_\ell$. Superscripts (Orig) and (Pruned) denote quantities measured on the original and pruned network, respectively, on identical inputs. Primes denote pruned activations. Saikumar & Varghese (2025) explain that signal collapse arises via two mechanisms. First, pruning reduces the variance of the pre-normalization signal (or activation). Each output channel's variance sums the squared-weight contributions over the set of parameters that survive $\mathcal{S}$, so removing weights removes summands, which inturn reduces the variance. Second, BN normalizes the signal by the *original* stored statistics $(\mu_\ell, \mathrm{Var}_\ell^{(\mathrm{Orig})})$, dividing the diminished signal by an overestimated denominator, which results in *over-normalization*. The extent of the reduction in activation variance at layer ℓ is quantified by the *signal variance ratio*:

$$\eta_\ell \;=\; \frac{\mathrm{Var}_\ell^{(\mathrm{Pruned})}(\mathbf{Z}'_\ell)}{\mathrm{Var}_\ell^{(\mathrm{Orig})}(\mathbf{Z}_\ell)}. \tag{3}$$

In the BN setting, pruning results in η_ℓ to be below one at nearly every layer. Since layer $\ell{+}1$'s input is layer ℓ's post-BN output, the reduction compounds across layers resulting in a decay in activation variance across the network. The activation variance at the final layer in the pruned model is:

$$\mathrm{Var}_L^{(\mathrm{Pruned})}(\mathbf{Z}'_L) = \Big(\prod_{\ell=1}^{L} \eta_\ell\Big) \mathrm{Var}_L^{(\mathrm{Orig})}(\mathbf{Z}_L). \tag{4}$$

Thus even a modest signal variance ratio $\eta_\ell = 0.8$ over $L = 25$ layers will result in $0.8^{25} \approx 0.003$. *Signal collapse* is defined as the decay of signal variance ratio with layer depth and sparsity, $\lim_{\kappa \to 1} \mathrm{Var}_L^{(\mathrm{Pruned})} / \mathrm{Var}_L^{(\mathrm{Orig})} \to 0$, because of which distinct inputs map to nearly identical outputs and the network predicts a single class. Hessian-based updates *partially* restore the activation variance, thereby improving accuracy (Saikumar & Varghese, 2025). Measured variance ratio curves shown in Figure 1 report Equation (3) evaluated at each layer.

2.3 BATCHNORM RECALIBRATION

Instead of updating weights using Hessian information, REFLOW recalibrates BN statistics. A small calibration set $\mathcal{B}$ of unlabeled training batches is passed through the pruned network. The per-channel mean and variance of $\mathbf{X}'_\ell$ over $\mathcal{B}$ are accumulated as $\widehat{\mu}_\ell^{(\mathrm{Pruned})}$ and $\widehat{\mathrm{Var}}_\ell^{(\mathrm{Pruned})}$, and the stored pair $(\mu_\ell, \mathrm{Var}_\ell^{(\mathrm{Orig})})$ is replaced by $(\widehat{\mu}_\ell^{(\mathrm{Pruned})}, \widehat{\mathrm{Var}}_\ell^{(\mathrm{Pruned})})$. γ_ℓ, β_ℓ and all weights remain unchanged:

$$\mathbf{Z}'^{(\mathrm{REFLOW})}_\ell(n) = \frac{\mathbf{X}'_\ell(n) - \widehat{\mu}_\ell^{(\mathrm{Pruned})}}{\sqrt{\widehat{\mathrm{Var}}_\ell^{(\mathrm{Pruned})}(\mathbf{X}'_\ell) + \epsilon}} \, \gamma_\ell + \beta_\ell. \tag{5}$$

Each layer then normalizes the pruned signal by its *own* statistics, so every channel of the normalized activation has mean 0 and variance 1, and the affine returns the output to the dense model's per-channel moments:

$$\mathbb{E}[z'_j] = \beta_j = \mathbb{E}[z_j^{(\mathrm{Orig})}], \qquad \mathrm{Std}[z'_j] = |\gamma_j| = \mathrm{Std}[z_j^{(\mathrm{Orig})}]. \tag{6}$$

The above restores $\eta_\ell \approx 1$ across the network and enables MP to match or exceed the accuracy of Hessian-based methods at a lower computational cost. By recalibrating the BN statistics, REFLOW enforces a per-channel **output-moment condition**, and overwriting the running statistics is the interface BN offers for it. However, that interface does not carry over to LN as it maintains no running statistics to update, as its statistics are recomputed per sample basis. Despite this Figure 1 shows signal collapse exists in LN-based CNNs. Thus, a mechanism to recover LN based networks which have undergone signal collapse is required.

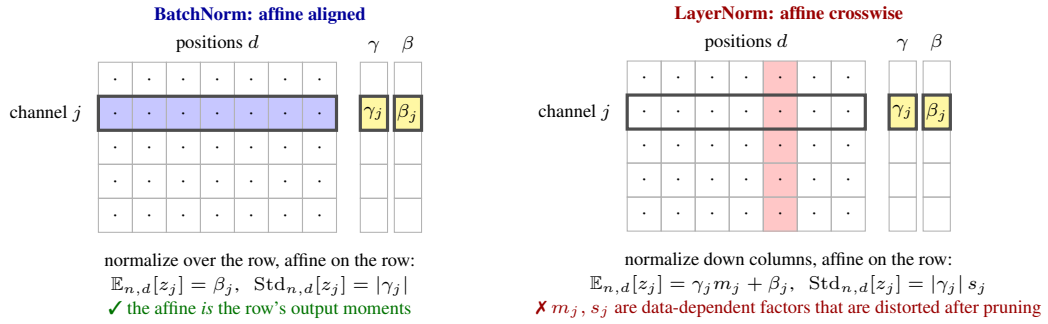


Figure 3: The crosswise-affine asymmetry on one sample of the activation tensor. Rows are channels j and columns are positions d. Shaded indicate the normalization axis, outlined is channel j's row and its affine entries. Both normalizers carry the same per-channel affine vectors $\gamma, \beta \in \mathbb{R}^C$. **Left:** BatchNorm normalizes and rescales the *same* row as shown in Equation (9). **Right:** LayerNorm normalizes columns while the affine rescales rows as shown in Equation (11).

3 Pruning Damage in LayerNorm Networks

Signal collapse exists in LN-based networks despite LN not maintaining any stored statistics. Thus, the BN-based explanation of signal collapse and its recovery mechanism do not transfer directly to LN-based networks. To locate the source of signal collapse after pruning, we identify the axes of the activation tensor BN and LN normalizations reduce over. The two normalizers reduce over orthogonal axes, yet they apply the same per-channel affine. We show that the damage due to pruning resides in per-channel moments, the subspace the affine parameters control.

3.1 The activation tensor and orthogonal reduction

An activation tensor is written $x \in \mathbb{R}^{N\times C\times D}$ with entries $x[n, j, d]$: N samples, C channels, and D positions per channel ($D = H \cdot W$ for a convolutional feature map). As Figure 3 shows, the two normalizers reduce the activation tensor orthogonally:

$$\text{BN: } \tilde{x}[n,j,d] = \frac{x[n,j,d] - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}, \qquad \text{LN: } \hat{x}[n,j,d] = \frac{x[n,j,d] - \mu_{n,d}}{\sqrt{\sigma_{n,d}^2 + \epsilon}}, \tag{7}$$

where BN's $\mu_j = \mathbb{E}_{n,d}[x[n,j,d]]$ and $\sigma_j^2 = \mathrm{Var}_{n,d}[x[n,j,d]]$ give one statistic per *channel*, from the $N \cdot D$ values in that channel across the batch, and LN's $\mu_{n,d} = \mathbb{E}_j[x[n,j,d]]$ and $\sigma_{n,d}^2 = \mathrm{Var}_j[x[n,j,d]]$ give one per *(sample, position)*, from the C values in that column. Both then apply the same per-channel affine to their normalized output, written for $\hat{x}$, identically for $\tilde{x}$:

$$z[n,j,d] = \gamma_j\, \hat{x}[n,j,d] + \beta_j, \qquad \gamma, \beta \in \mathbb{R}^C. \tag{8}$$

3.2 The crosswise-affine identity

BN normalizes each row and its affine rescales the row, so after Equation (7) (left) every row of $\tilde{x}$ has mean 0 and variance 1, and the affine sets the output's row statistics:

$$\mathbb{E}_{n,d}\big[z_j\big] = \beta_j, \qquad \mathrm{Std}_{n,d}\big[z_j\big] = |\gamma_j|. \tag{9}$$

Meanwhile, LN normalizes each *column* but its affine rescales each *row* crosswise. Equation (7) (right) guarantees every column of $\hat{x}$ has mean 0 and variance 1, not imposing any conditions on the rows. Along a row the values of the mean and variance are determined by the data:

$$m_j^\bullet = \mathbb{E}_{n,d}\big[\hat{x}_j^\bullet\big], \qquad s_j^\bullet = \sqrt{\mathrm{Var}_{n,d}\big[\hat{x}_j^\bullet\big]}, \qquad \bullet \in \{\text{(Orig)}, \text{(Pruned)}\}, \tag{10}$$

The output's row statistics are therefore

$$\mathbb{E}_{n,d}\big[z_j\big] = \gamma_j m_j + \beta_j, \qquad \mathrm{Std}_{n,d}\big[z_j\big] = |\gamma_j|\, s_j. \tag{11}$$

Figure 3 compares Equation (9) with Equation (11). **For BN the affine *is* the output's per-channel mean and standard deviation. For LN those quantities are the affine multiplied by data-dependent factors m_j, s_j that are distorted due to one-shot pruning.**

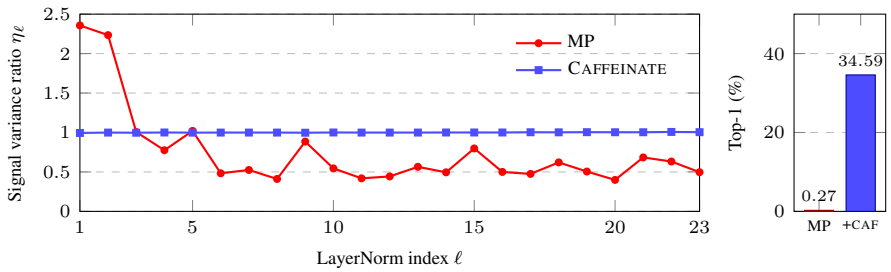


Figure 4: One-shot magnitude-pruned ConvNeXt-T at $\kappa = 60\%$. **Left:** CAFFEINATE restores the per-layer signal variance ratio η_ℓ to $\approx$1. **Right:** rewriting only γ, β raises top-1 from 0.27% to 34.59% — the pruned weights already encoded a working subnetwork; it was dormant.

3.3 WHERE DOES SIGNAL COLLAPSE RESIDE IN LN-BASED NETWORKS?

Does pruning induced distortion in $m_j^{(\text{Pruned})}, s_j^{(\text{Pruned})}$ matter, given that per-sample normalization? LN divides each sample by one scalar across its channels, so channel j of $\hat{x}$ carries the scale $s_j/\bar{s}$, which is its magnitude *relative to the cross-channel average*. Uniform attenuation would be absorbed completely by LN, but since global pruning is non-uniform, it removes a different set of parameters from each output channel. Therefore the attenuation is non-uniform, and the distorted $m_j^{(\text{Pruned})}, s_j^{(\text{Pruned})}$ are subsequently passed into the LN layer. Thus, signal collapse persists in the absence of stored statistics. In Equation (11), the post-affine signal variance of channel j is $\gamma_j^2 s_j^2$. Pruning results in $s_j^{(\text{Pruned})} < s_j^{(\text{Orig})}$, which results in a post-affine signal variance reduction on a per layer basis. This variance reduction compounds over layers and results in a decay in the signal variance ratio η_ℓ as shown in Figure 1. Since γ, β act on a per channel basis, pruning induced damage is located in the subspace the affine parameters control, making a closed-form correction possible.

4 CAFFEINATE: A CLOSED-FORM, FORWARD-ONLY CORRECTION

CAFFEINATE runs once on an already-pruned network and does not change the mask or any weights in the convolution and linear layers. Throughout, γ, β denote the *original* model's affine parameters, and $m_j^\bullet, s_j^\bullet$ are the per-channel moments of Equation (10), measured on the calibration set $\mathcal{B}$ for the original ($\bullet = $ (Orig)) and pruned ($\bullet = $ (Pruned)) network.

4.1 OBJECTIVE AND CLOSED-FORM SOLUTION

CAFFEINATE imposes on LayerNorm the per-channel output-moment condition. By the crosswise-affine identity in Equation (11), the original layer's output on channel j has moments $\mathbb{E}[z_j^{(\text{Orig})}] = \gamma_j m_j^{(\text{Orig})} + \beta_j$ and $\text{Std}[z_j^{(\text{Orig})}] = |\gamma_j|\, s_j^{(\text{Orig})}$. CAFFEINATE therefore solves for affine parameters γ_j', β_j' for the pruned model whose output $z_j' = \gamma_j' \hat{x}_j^{(\text{Pruned})} + \beta_j'$ reproduces both:

$$\mathbb{E}[z_j'] = \mathbb{E}[z_j^{(\text{Orig})}], \qquad \text{Std}[z_j'] = \text{Std}[z_j^{(\text{Orig})}]. \tag{12}$$

The standard-deviation condition gives $|\gamma_j'|\, s_j^{(\text{Pruned})} = |\gamma_j|\, s_j^{(\text{Orig})}$. The sign-preserving root and the mean condition yield

$$\boxed{\gamma_j' = \gamma_j \cdot \frac{s_j^{(\text{Orig})}}{s_j^{(\text{Pruned})}}} \qquad \boxed{\beta_j' = \gamma_j m_j^{(\text{Orig})} + \beta_j - \gamma_j' m_j^{(\text{Pruned})}} \tag{13}$$

Equation (12) is satisfied *exactly*, with two equations, and two unknowns per channel. The γ_j, β_j on the right-hand sides are the original model's values. Channels with $s_j^{(\text{Pruned})} \approx 0$ retain the original parameters (Appendix A), and Appendix G provides an overview of updating only γ_j or β_j.

4.2 SEQUENTIAL EXECUTION

The measurement of $s_j^{(\text{Pruned})}$ at layer ℓ is meaningful only if layers $1, \ldots, \ell{-}1$ are already corrected. Otherwise it conflates layer ℓ's damage with attenuation inherited from the preceding layers, resulting

Table 1: Accuracy (%) on ImageNet after one-shot pruning. +CAF applies CAFFEINATE to the preceding column's pruned network (MP or CHITA). CH. refers to CHITA. *Weight update* denotes whether any weights in the convolution and linear layers are modified.

			$\kappa = 30\%$				$\kappa = 40\%$				$\kappa = 50\%$				$\kappa = 60\%$			
Family	Model	Dense	MP	+CAF	CH.	+CAF	MP	+CAF	CH.	+CAF	MP	+CAF	CH.	+CAF	MP	+CAF	CH.	+CAF
CN	T (29M)	82.13	66.79	79.55	80.53	**81.18**	15.08	73.49	77.73	**79.67**	1.93	59.80	18.27	**69.94**	0.27	34.59	2.72	**56.38**
	B (88M)	83.70	72.60	81.20	82.00	**82.89**	36.80	74.91	69.61	**80.36**	7.96	63.16	42.04	**68.60**	1.24	44.11	24.42	**46.46**
	L (198M)	84.12	77.76	83.11	83.57	**83.73**	50.45	79.14	77.26	**82.37**	18.81	72.60	48.04	**78.69**	2.89	41.43	6.96	**47.28**
FN	T (28M)	81.88	75.76	78.16	80.95	**80.97**	58.68	72.82	78.64	**79.36**	13.32	58.96	63.84	**72.17**	0.39	19.91	7.22	**33.40**
	S (50M)	83.18	79.49	80.99	**82.61**	82.45	63.22	72.71	79.96	**80.82**	13.41	41.93	41.42	**57.12**	0.24	12.10	7.73	**28.92**
	B (88M)	83.52	77.03	80.27	**82.98**	82.97	60.66	73.02	79.62	**80.51**	13.65	44.00	38.39	**61.62**	0.44	7.19	2.54	**13.65**
HN	T (22M)	82.58	75.88	77.47	81.02	**81.38**	60.19	70.53	78.22	**79.57**	4.62	56.36	51.52	**66.35**	0.18	8.26	1.61	**39.68**
	S (49M)	83.66	77.16	80.41	82.92	**83.02**	65.10	76.25	80.63	**81.50**	25.40	67.02	64.98	**76.40**	0.39	26.99	9.98	**50.49**
	B (87M)	84.05	80.31	81.12	83.48	**83.61**	72.61	78.63	82.50	**82.99**	41.12	68.16	76.34	**80.44**	1.25	43.49	28.10	**63.79**
RD	T (24M)	82.78	63.45	76.41	81.28	**81.88**	19.25	58.85	43.39	**71.87**	2.15	40.92	18.66	**59.95**	0.17	19.36	2.64	**25.39**
	S (50M)	83.54	59.03	75.46	76.62	**79.94**	26.21	56.69	47.60	**60.81**	8.62	39.32	15.66	**43.71**	0.31	21.01	2.35	**29.54**
	B (87M)	84.24	68.93	79.65	82.67	**83.34**	44.04	72.83	73.27	**80.11**	12.36	60.46	13.06	**67.28**	1.04	37.73	5.08	**50.64**
Weight update		—	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓

in an over-correction. Layers are therefore corrected one at a time in *forward-execution order* (Algorithm 1, Appendix B).

5 EXPERIMENTS

Models. The experiments are conducted across twelve LN-based networks spanning four modern CNN families and three scales each are considered: ConvNeXt-{Tiny (T), Base (B), Large (L)} (Liu et al., 2022), FocalNet-{Tiny (T), Small (S), Base (B)} (Yang et al., 2022), HorNet-{T, S, B} (Rao et al., 2022), and RDNet-{T, S, B} (Kim et al., 2024), all with public ImageNet-1k (Deng et al., 2009) checkpoints.

Sparsity range. Modern LN-based CNNs are less resilient to one-shot pruning than BN-based CNNs. At $\kappa = 50\%$, BN-based MobileNet that is pruned with MP retains 62.6% top-1 accuracy (Saikumar & Varghese, 2025) while ConvNeXt-T falls to 1.9% when pruned with MP, as shown in Table 1. The accuracy drop of BN-based CNNs at 70–80% sparsity occurs at 30–60% for LN-based CNNs. Therefore results here cover sparsities ranging from 10% to 60%.

Protocol. CAFFEINATE which is abbreviated as CAF is applied *after* pruning. It takes a pruned network and corrects only the LN affine parameters. CAFFEINATE uses $1{,}000$ unlabeled training images as its calibration set. We report top-1 accuracy on the full ImageNet validation set. All reported measurements are averages of three independent runs.

Baselines. (i) MP, (ii) CHITA (Benbaki et al., 2023), a state-of-the-art impact-based method for CNN pruning, (iii) CAFFEINATE applied to MP, (iv) CAFFEINATE applied to CHITA, and (v) the original dense model, representing the upper bound of accuracy that can be recovered.

Hardware. A single NVIDIA RTX A6000 (48 GB) coupled with an AMD EPYC 7713P CPU, 256 GB RAM, Python 3.9.21, PyTorch 2.5.1.

Observation 1: Moment correction determines accuracy where signal collapse occurs.

In Table 1 and Figure 5, CAFFEINATE offers accuracy gains at sparsities where signal collapse occurs. MP corrected by CAFFEINATE reaches 90–99% of dense model's accuracy at $\kappa = 30\%$, of which the correction contributes $+0.8$ to $+16.4$ points. MP corrected by CAFFEINATE reaches 68–94% of the dense accuracy at $\kappa = 40\%$ and 47–86% of the dense accuracy at $\kappa = 50\%$, where the uncorrected models achieve as low as 1.9%. The largest recovery is $+58.4\%$ accuracy on ConvNeXt-T at $\kappa = 40\%$, where the accuracy improves from 15.08% to 73.49%. Two patterns are observed. First, at lower sparsities, where signal collapse has not occurred, CAFFEINATE offers limited improvements over MP, and Figure 1 shows that ConvNeXt-T maintains a near-unit signal variance ratio at lower sparsities. As an example, at $\kappa \leq 20\%$ CAFFEINATE changes accuracy between -0.1 and $+2.1\%$ (Tables 4 to 7). Second, at higher sparsities, where signal collapse has occurred CAFFEINATE offers significant improvements. For example, the mean accuracy gain across the twelve models rises

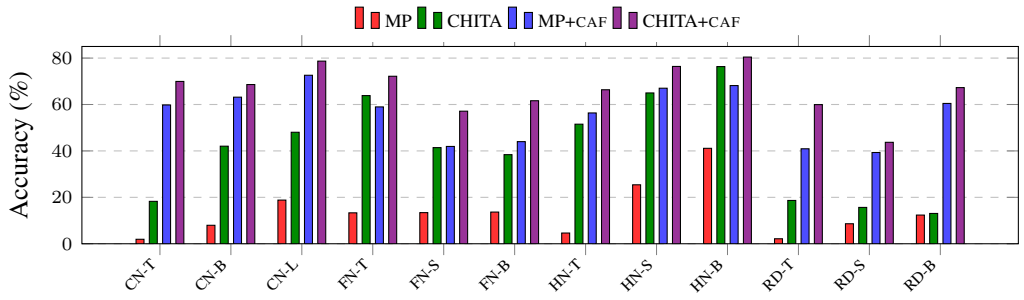


Figure 5: Accuracy at $\kappa = 50\%$ on ImageNet. On every model CAFFEINATE improves the accuracy of whichever pruner it follows: MP+CAF (blue) offers better accuracy than CHITA (green) on ten of twelve models without labels or gradients, and CHITA+CAF (purple) is best overall. MP in red. CN = ConvNeXt, FN = FocalNet, HN = HorNet, and RD = RDNet.

from $+0.4\%$ at $\kappa \leq 20\%$ to $+6.6\%$, $+24.0\%$, and $+42.4\%$ at $\kappa = 30\%$, 40%, and 50% sparsity respectively.

At lower sparsities MP+CAF performs comparably to CHITA without labels or gradients. For example, MP+CAF is within 0.9% accuracy of CHITA on average at $\kappa = 20\%$ and within 2.2% at $\kappa = 30\%$. At $\kappa = 40\%$, the trend swaps as CHITA outperforms MP+CAF on most models. However, at higher sparsities where signal collapse has occurred, mitigating it improves accuracy. MP+CAF outperforms CHITA on ten of twelve models at $\kappa = 50\%$ for example MP+CAF outperforms CHITA by up to $+47.4\%$ in accuracy. At $\kappa = 60\%$, MP+CAF outperforms CHITA across all twelve models and by up to $+34.5\%$ in accuracy.

Observation 2: CAFFEINATE is agnostic of the pruning method.

CAFFEINATE rewrites only the LN affines, so it can be applied on top of any pruning method. Table 1 reports it on both MP and CHITA. Tables 4 to 7 in the Appendix decouples CHITA into its two phases, weight selection (CHITA-S) and weight update (CHITA-U / CHITA) shows where the accuracy gains arise relative to MP. CHITA-S+CAF offers comparable accuracy to MP+CAF on all twelve models, within 1.5% accuracy at $\kappa = 50\%$. However, CHITA+CAF exceeds MP+CAF on every model at $\kappa \geq 30\%$, by up to $+31.4\%$ in accuracy at $\kappa = 60\%$. Using CAF improves CHITA by up to $+54.2\%$ in accuracy at $\kappa = 50\%$. This indicates that signal collapse masks the value of the Hessian based weight update, visible only once the channel moments are restored. The Hessian update weight update corrects the pruning induced damage outside the affine subspace.

Observation 3: The memory footprint of CAFFEINATE is lower than CHITA.

CAFFEINATE updates two scalars per channel. CHITA builds an empirical-Fisher matrix from labeled forward–backward passes. The Fisher matrices require upto 64 GB, higher than the available 48 GB GPU memory, and therefore is held in CPU memory (Table 3, Appendix B). MP+CAF uses up to $4.3\times$ less GPU and $25.6\times$ less CPU memory, and its host footprint stays at 2.5–2.8 GB against CHITA's 25–64 GB. The savings pertain to MP+CAF; CHITA+CAF inherits CHITA's resource footprint.

Scope and limitations. CAFFEINATE corrects damage spread over $C \times C_{\text{in}}$ weights using the $2C$ parameters of a LN, so the correction is rank-deficient. It restores how strongly each channel responds, not what the channel responds to. That suffices when pruning mainly attenuates channels, the regime of the CNN backbones studied here. It fails when pruning changes which feature a channel represents. Self-attention is where this distinction matters. For example, a ViT block mixes tokens through projections that set which directions in channel space are compared, and that comparison is recomputed for every input. Pruning those projections changes what a channel represents rather than how strongly it responds. For example, on a one-shot pruned ViT-B/16 (Dosovitskiy et al., 2021), CAFFEINATE does not improve over MP. We therefore limit CAFFEINATE to LN-based CNNs.

6 Related Work

One-shot pruning. Traditionally, pruning involves removing parameters, followed by retraining (Hanson & Pratt, 1988; LeCun et al., 1989; Han et al., 2015; Frankle & Carbin, 2019). This is referred to as gradual pruning, which can be computationally expensive due to the prune and retrain cycles.

One-shot pruning requires no retraining and therefore is an computationally cheaper alternative to gradual pruning. Impact-based methods use the Hessian to select which weights to prune, followed by an Hessian-based weight update to the surviving weights. Following Optimal Brain Surgeon (Hassibi & Stork, 1992; Hassibi et al., 1993), WoodFisher (Singh & Alistarh, 2020), CBS (Yu et al., 2022c), and CHITA (Benbaki et al., 2023) differ in how they approximate the inverse Hessian. Saikumar & Varghese (2025) separated weight selection from weight update, indicating that the accuracy gain of IP methods over MP stem from Hessian-based weight update, which indirectly mitigates signal collapse.

Recalibrating normalization statistics. Post-training quantization methods recalibrate BN statistics after quantizing weights (Banner et al., 2019; Nagel et al., 2019; Hubara et al., 2021), while reconstruction-based schemes optimize the rounding instead (Nagel et al., 2020; Li et al., 2021; Wei et al., 2022). EagleEye (Li et al., 2020) recalibrates BN to *select* a subnetwork to train rather than correct, and Wanda (Sun et al., 2024) uses activation norms in the pruning criterion. However, none of these methods update the LN parameters.

Statistical correction in adjacent settings. REPAIR (Jordan et al., 2023) mitigates variance collapse when interpolating trained networks and restores the variance using BN recalibration. Yu et al. (2026) prune language models by rescaling surviving weights to the dense model's energy. Comparisons to both of these approaches are further provided in Appendix F.

LayerNorm architectures and signal propagation. ConvNeXt (Liu et al., 2022), FocalNet (Yang et al., 2022), HorNet (Rao et al., 2022), and RDNet (Kim et al., 2024) are LN-based CNNs. Modern vision-model pruning primarily targets Vision Transformers with structured pruning or retraining-based pruning (Yu & Xiang, 2023; Yu et al., 2022b;a). Meanwhile retraining-free pruning targets vision–language models (Sung et al., 2024). One-shot, retraining-free pruning of modern LN-based CNNs was previously unaddressed.

7 Conclusion

One-shot pruning damages modern LayerNorm-based CNNs more than BatchNorm-based CNNs that have been previously studied. At $\kappa = 50\%$ using magnitude pruning (MP) on ConvNeXt-T retains only 1.9% accuracy. We demonstrate that this is due to signal collapse despite per-sample normalization, and the damage lies in the per-channel output moments. The per-channel output moments, which CAFFEINATE restores in closed form from $1{,}000$ unlabeled images, updating only γ and β, no weights in the linear or convolution layers are modified. Across twelve ImageNet models, MP corrected by CAFFEINATE restores 90–99% of dense accuracy at $\kappa = 30\%$ and recovers up to $+58.4\%$. Label-free MP+CAF outperforms CHITA in terms of accuracy at high sparsity while using a fraction of its memory, and, when applied on top of CHITA, it improves accuracy by up to $+54.2\%$.

Reproducibility statement

All LN corrections studied in this paper are deterministic given the pruning mask and the calibration set. CAFFEINATE has no random initialization, no labels, and no optimizer state, and its only setting is the calibration size n, so reproducing any number in Section 5 requires only the checkpoint, the mask seed, and the calibration indices, all of which will be released with our code.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv*, abs/1607.06450, 2016.

Ron Banner, Yury Nahshan, and Daniel Soudry. Post Training 4-bit Quantization of Convolutional Networks for Rapid-Deployment. In *Neural Information Processing Systems*, 2019.

Riade Benbaki, Wenyu Chen, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. Fast as CHITA: Neural Network Pruning with Combinatorial Optimization. In *International Conference on Machine Learning*, 2023.

Davis W. Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John V. Guttag. What is the State of Neural Network Pruning? In *Machine Learning and Systems*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021.

Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*, 2019.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning Both Weights and Connections for Efficient Neural Networks. In *Neural Information Processing Systems*, 2015.

Stephen Hanson and Lorien Pratt. Comparing Biases for Minimal Network Construction with Back-Propagation. In *Neural Information Processing Systems*, 1988.

Babak Hassibi and David Stork. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In *Neural Information Processing Systems*, 1992.

Babak Hassibi, David Stork, and Gregory Wolff. Optimal Brain Surgeon: Extensions and Performance Comparisons. In *Neural Information Processing Systems*, 1993.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv*, abs/1606.08415, 2016.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate Post Training Quantization with Small Calibration Sets. In *International Conference on Machine Learning*, 2021.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, 2015.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: REnormalizing Permuted Activations for Interpolation Repair. In *International Conference on Learning Representations*, 2023.

Donghyun Kim, Byeongho Heo, and Dongyoon Han. DenseNets Reloaded: Paradigm Shift Beyond ResNets and ViTs. In *European Conference on Computer Vision*, 2024.

Yann LeCun, John Denker, and Sara Solla. Optimal Brain Damage. In *Neural Information Processing Systems*, 1989.

Bailin Li, Bowen Wu, Jiang Su, and Guangrun Wang. EagleEye: Fast Sub-net Evaluation for Efficient Neural Network Pruning. In *European Conference on Computer Vision*, 2020.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. In *International Conference on Learning Representations*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *IEEE International Conference on Computer Vision*, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-Free Quantization Through Weight Equalization and Bias Correction. In *IEEE International Conference on Computer Vision*, 2019.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or Down? Adaptive Rounding for Post-Training Quantization. In *International Conference on Machine Learning*, 2020.

Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Nam Lim, and Jiwen Lu. HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions. In *Neural Information Processing Systems*, 2022.

Dhananjay Saikumar and Blesson Varghese. Signal collapse in one-shot pruning: When sparse models fail to distinguish neural representations, 2025. URL https://arxiv.org/abs/2502.15790.

Sidak Pal Singh and Dan Alistarh. WoodFisher: Efficient Second-Order Approximation for Neural Network Compression. In *Neural Information Processing Systems*, 2020.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A Simple and Effective Pruning Approach for Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. ECoFLaP: Efficient Coarse-to-Fine Layer-Wise Pruning for Vision-Language Models. In *International Conference on Learning Representations*, 2024.

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly Dropping Quantization for Extremely Low-bit Post-Training Quantization. In *International Conference on Learning Representations*, 2022.

Jianwei Yang, Chunyuan Li, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Focal Modulation Networks. In *Neural Information Processing Systems*, 2022.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width & Depth Pruning for Vision Transformers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 2022a.

Lu Yu and Wei Xiang. X-Pruner: eXplainable Pruning for Vision Transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Peiqi Yu, Jinhao Wang, Xinyi Sui, Nam Ling, Wei Wang, and Wei Jiang. Efficient Post-Training Pruning of Large Language Models with Statistical Correction. *arXiv*, abs/2602.07375, 2026.

Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified Visual Transformer Compression. In *International Conference on Learning Representations*, 2022b.

Xin Yu, Thiago Serra, Srikumar Ramalingam, and Shandian Zhe. The Combinatorial Brain Surgeon: Pruning Weights That Cancel One Another in Neural Networks. In *International Conference on Machine Learning*, 2022c.

A Derivations and Proofs

A.1 The closed-form solution

For a layer and a channel j, let $\hat{x}_j$ be the normalized-but-unscaled pruned activation with measured moments $m_j^{(\text{Pruned})} = \mathbb{E}_{n,d}[\hat{x}_j]$ and $(s_j^{(\text{Pruned})})^2 = \text{Var}_{n,d}[\hat{x}_j]$. The corrected output is $z'_j = \gamma'_j \hat{x}_j + \beta'_j$ with γ'_j, β'_j given by Equation (13). The moments $m_j^{(\text{Pruned})}, s_j^{(\text{Pruned})}$ are measured with all the preceding layers already corrected, as shown in Algorithm 1. By linearity of expectation, $\text{Var}(aX + b) = a^2\text{Var}(X)$, and Equation (11),

$$\begin{aligned}\mathbb{E}[z'_j] &= \gamma'_j m_j^{(\text{Pruned})} + \beta'_j = \gamma'_j m_j^{(\text{Pruned})} + \gamma_j m_j^{(\text{Orig})} + \beta_j - \gamma'_j m_j^{(\text{Pruned})} \\ &= \gamma_j m_j^{(\text{Orig})} + \beta_j = \mathbb{E}[z_j^{(\text{Orig})}], \end{aligned} \tag{14}$$

$$\text{Std}[z'_j] = |\gamma'_j|\, s_j^{(\text{Pruned})} = |\gamma_j| \frac{s_j^{(\text{Orig})}}{s_j^{(\text{Pruned})}}\, s_j^{(\text{Pruned})} = |\gamma_j|\, s_j^{(\text{Orig})} = \text{Std}[z_j^{(\text{Orig})}], \tag{15}$$

using $s_j^{(\text{Pruned})} > 0$ (channels with $s_j^{(\text{Pruned})} \approx 0$ are excluded and retain the original parameters).

A.2 Why activations, and not weights

CAFFEINATE is a data-dependent correction for three reasons.

(i) Even a single linear layer needs input statistics: the pruned pre-normalization variance is $\sum_{i \in \mathcal{S}} W'^2_{\ell,i} \text{Var}(H_{\ell-1,i})$ under the independence assumptions of Saikumar & Varghese (2025). The weights supply $W'_{\ell,i}$, but $\text{Var}(H_{\ell-1,i})$ is a property of the data.

(ii) Per-sample normalization couples channels nonlinearly: $\hat{x}_j$ centers and scales x_j by statistics taken over all C channels of that sample. Thus, even exact $\text{Var}[x_j]$ does not determine $\text{Var}[\hat{x}_j]$ without distributional assumptions: two inputs agreeing in $\text{Var}[x_j]$ but differing in the other channels' scale give different $\text{Var}[\hat{x}_j]$.

(iii) Pruning induced damage compounds through nonlinearities: layer ℓ's input depends on the output of the preceding layers which have been pruned, e.g., depthwise convolutions, GELU (Hendrycks & Gimpel, 2016), and residual mixing on real inputs. Propagating L pruned layers analytically through such maps is intractable, which motivates layer-wise calibration on data.

B Overheads

Table 2: Cost and requirements of signal-collapse recovery methods. B is calibration batches; L is affine LayerNorms.

Method	Forward-equivalents	Passes	Labels	Gradients
ReFlow (Saikumar & Varghese, 2025)	B	1	✗	✗
CAFFEINATE (this work)	$(L{+}1)B$	$L{+}1$	✗	✗

The procedure for CAFFEINATE is presented in Algorithm 1. CAFFEINATE requires one pass over the original model, then one full forward pass of the pruned model per LayerNorm layer. CAFFEINATE requires no labels and no gradients, and uses n calibration samples to correct γ, β per layer. No other model weights are modified.

C Contribution of the Calibration-Set Size

Accuracy recovery saturates at $n = 1{,}000$ as shown in Figure 6. **We therefore use $n = 1{,}000$ unlabeled images in all experiments**, which corrects ConvNeXt-T in 24 seconds on the hardware utilized in Section 5. Using $n = 1{,}000$ corresponds to 0.08% of the 1.28M-image ImageNet-1k training set. CAFFEINATE (with MP) is $6.3\times$ more sample-efficient than CHITA, which requires 6,300 images per run and, unlike CAFFEINATE, requires each image to be labeled.

Algorithm 1 CAFFEINATE. f: original model; f': the pruned model; $\mathcal{B}$: unlabeled calibration batches, drawn once and replayed. Only γ, β are written; f''s mask and weights are unchanged.

1: $\mathcal{L} \leftarrow$ affine LayerNorms of f' in forward-execution order
2: sweep $\mathcal{B}$ once through f, accumulating $m_\ell^{(\text{Orig})}, s_\ell^{(\text{Orig})}$ from $\hat{x}$ at every $\ell \in \mathcal{L}$ simultaneously
3: **for** each $\ell \in \mathcal{L}$ in forward order **do**
4: sweep $\mathcal{B}$ through f', accumulating $m_\ell^{(\text{Pruned})}, s_\ell^{(\text{Pruned})}$ from $\hat{x}$ at ℓ
5: $\gamma' \leftarrow \gamma\, s_\ell^{(\text{Orig})}/s_\ell^{(\text{Pruned})}, \quad \beta' \leftarrow \gamma\, m_\ell^{(\text{Orig})} + \beta - \gamma' m_\ell^{(\text{Pruned})}$ (original γ, β; retain them where $s_\ell^{(\text{Pruned})} \approx 0$)
6: write γ', β' into ℓ *before* proceeding, so later layers are measured through the already-corrected prefix
7: **end for**

Table 3: **Peak memory** (GB) on GPU and CPU per network for CHITA and CAFFEINATE.

	ConvNeXt			FocalNet			HorNet			RDNet		
	T	B	L	T	S	B	T	S	B	T	S	B
Params (M)	29	88	198	28	50	88	22	49	87	24	50	87
MP+CAF GPU	2.26	4.68	9.17	2.49	3.21	4.96	1.64	2.97	4.63	3.42	4.43	7.16
MP+CAF CPU	2.53	2.52	2.52	2.59	2.59	2.60	2.63	2.62	2.62	2.72	2.78	2.80
CHITA GPU	7.3	8.9	12.0	7.3	8.0	8.9	7.1	8.0	8.9	7.1	8.0	8.9
CHITA CPU	30.5	62.7	64.4	30.4	50.4	63.8	24.8	50.0	63.1	26.5	51.4	62.5

D FULL RESULTS

This section reports the complete measurements, including the accuracy recovery curves as shown in Figure 7 and the complete results for each model family as shown in Tables 4 to 7.

Table 4: Full results of top-1 (%) accuracy for ConvNeXt on ImageNet.

Model	Method	Dense	κ=10%	20%	30%	40%	50%	60%
ConvNeXt-T	MP	82.13	82.02	80.00	66.79	15.08	1.93	0.27
	MP+CAF		82.01	81.28	79.55	73.49	59.80	34.59
	CHITA-S		82.03	79.99	66.82	15.44	1.98	0.26
	CHITA-S+CAF		82.02	81.32	79.53	73.40	59.60	34.30
	CHITA		82.12	81.81	80.53	77.73	18.27	2.72
	CHITA+CAF		**82.14**	**81.94**	**81.18**	**79.67**	**69.94**	**56.38**
ConvNeXt-B	MP	83.70	83.55	81.69	72.60	36.80	7.96	1.24
	MP+CAF		83.63	82.84	81.20	74.91	63.16	44.11
	CHITA-S		83.57	81.70	72.68	37.43	8.06	1.27
	CHITA-S+CAF		83.62	82.80	81.19	74.78	62.89	43.85
	CHITA		83.64	83.51	82.00	69.61	42.04	24.42
	CHITA+CAF		**83.70**	**83.63**	**82.89**	**80.36**	**68.60**	**46.46**
ConvNeXt-L	MP	84.12	84.07	83.59	77.76	50.45	18.81	2.89
	MP+CAF		84.11	83.87	83.11	79.14	72.60	41.43
	CHITA-S		84.08	83.59	77.83	48.89	18.89	3.10
	CHITA-S+CAF		84.10	83.84	83.11	79.37	73.92	47.00
	CHITA		**84.13**	83.94	83.57	77.26	48.04	6.96
	CHITA+CAF		**84.13**	**84.03**	**83.73**	**82.37**	**78.69**	**47.28**

E COMPARISON WITH A PER-CHANNEL LEAST-SQUARES FIT

CAFFEINATE restores the output moments of every LayerNorm channel. A natural alternative in the same parameter subspace is to fit the affine so that the pruned layer *reconstructs* the original layer's output. This section derives the reconstruction, and shows that the reconstruction is a scaled down version of Equation (13).

Objective. Let $z_j^{(\text{Orig})} = \gamma_j \hat{x}_j^{(\text{Orig})} + \beta_j$ be the original layer's output on channel j. The least-squares (LS) correction fits independently per channel the affine that best predicts the output from the pruned

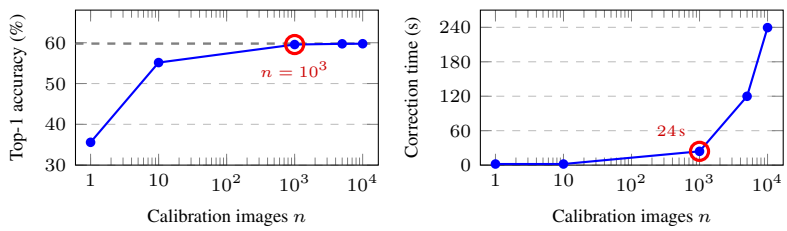


Figure 6: **Calibration-set size**, ConvNeXt-T at $\kappa = 50\%$. **Left:** top-1 saturates by $n = 10^3$. The bold dashed line marks the $n = 10^4$ value. **Right:** correction time grows linearly with n. We use $n = 10^3$ (red).

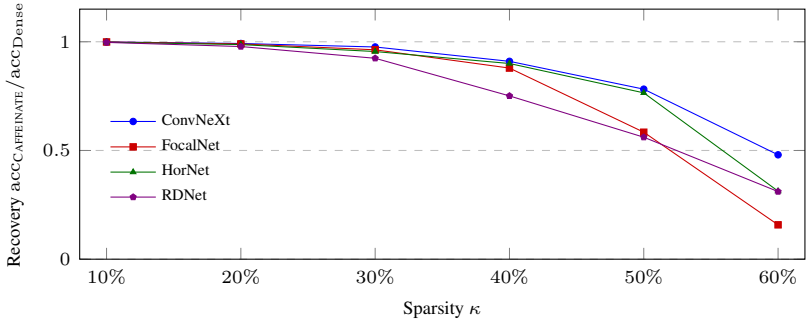


Figure 7: Accuracy Recovery vs. sparsity by model family.

normalized activation:

$$\min_{\gamma'_j, \beta'_j} \mathbb{E}\left[\left(\gamma'_j \, \hat{x}_j^{(\text{Pruned})} + \beta'_j - z_j^{(\text{Orig})}\right)^2\right], \tag{16}$$

whose solution is the one-variable ordinary least squares

$$\gamma'_j = \frac{\mathrm{Cov}\left(\hat{x}_j^{(\text{Pruned})}, z_j^{(\text{Orig})}\right)}{\mathrm{Var}[\hat{x}_j^{(\text{Pruned})}]}, \qquad \beta'_j = \mathbb{E}[z_j^{(\text{Orig})}] - \gamma'_j \, \mathbb{E}[\hat{x}_j^{(\text{Pruned})}]. \tag{17}$$

Implementation. The implementation of CAFFEINATE uses $n = 1{,}000$ calibration images, corrects layers sequentially in forward-execution order. Per layer, one pass measures four statistics $(\sum \hat{x}', \sum \hat{x}'^2, \sum z, \sum \hat{x}' z)$, from which Equation (17) follows algebraically. The cross-moment couples the two models on the same batch, so every batch runs through both the pruned and the original network.

Least squares is CAFFEINATE scaled down by the correlation. Substituting $z_j^{(\text{Orig})} = \gamma_j \hat{x}_j^{(\text{Orig})} + \beta_j$ into Equation (17) gives $\mathrm{Cov}(\hat{x}_j^{(\text{Pruned})}, z_j^{(\text{Orig})}) = \gamma_j \, \rho_j \, s_j^{(\text{Pruned})} s_j^{(\text{Orig})}$, where ρ_j is the correlation between the pruned and original normalized activations on channel j.

$$\gamma'^{\,\text{LS}}_j = \rho_j \cdot \gamma_j \frac{s_j^{(\text{Orig})}}{s_j^{(\text{Pruned})}} = \rho_j \cdot \gamma'_j, \tag{18}$$

Hence, the CAFFEINATE scale of Equation (13) multiplied by ρ_j, with the shift taking the same form around the shrunk scale. The two corrections coincide if and only if $\rho_j = 1$: the MSE-optimal affine restores a channel's full variance only when the pruned activation remains perfectly correlated with the dense activation. When pruning decorrelates a channel, the best predictor of $z_j^{(\text{Orig})}$ in the squared-error sense regresses toward the mean $\mathbb{E}[z_j^{(\text{Orig})}]$, and the scale decays with ρ_j.

Results. Table 8 reports MP + LS on all twelve models. At $\kappa \leq 20\%$, where surviving channels remain strongly correlated with their original counterparts, LS offers comparable accuracy to CAFFEINATE (Table 1). LS deteriorates from $\kappa = 30\%$, where on some models it falls below even uncorrected MP. For example on RDNet-B, LS achieves 30.10% against 68.93% for MP only. LS further deteriorates from $\kappa = 40\%$, and at $\kappa = 50\%$ no accuracy is recovered. For example, for

Table 5: Full results of top-1 (%) accuracy for FocalNet on ImageNet.

Model	Method	Dense	κ=10%	20%	30%	40%	50%	60%
FocalNet-T	MP	81.88	81.74	80.47	75.76	58.68	13.32	0.39
	MP+CAF		81.71	80.74	78.16	72.82	58.96	19.91
	CHITA-S		81.79	80.51	75.76	60.72	16.46	0.52
	CHITA-S+CAF		81.71	80.67	78.18	72.78	58.80	20.55
	CHITA		**81.83**	**81.72**	80.95	78.64	63.84	7.22
	CHITA+CAF		81.81	81.62	**80.97**	**79.36**	**72.17**	**33.40**
FocalNet-S	MP	83.18	**83.13**	82.55	79.49	63.22	13.41	0.24
	MP+CAF		83.09	82.43	80.99	72.71	41.93	12.10
	CHITA-S		83.10	82.52	79.48	63.88	14.08	0.28
	CHITA-S+CAF		83.09	82.42	80.99	72.62	41.48	11.95
	CHITA		83.11	**83.01**	**82.61**	79.96	41.42	7.73
	CHITA+CAF		83.09	82.95	82.45	**80.82**	**57.12**	**28.92**
FocalNet-B	MP	83.52	**83.56**	82.86	77.03	60.66	13.65	0.44
	MP+CAF		83.52	82.92	80.27	73.02	44.00	7.19
	CHITA-S		**83.56**	82.88	77.05	61.20	14.37	0.46
	CHITA-S+CAF		83.51	82.90	80.24	72.81	43.55	12.25
	CHITA		83.53	**83.40**	**82.98**	79.62	38.39	2.54
	CHITA+CAF		83.46	83.36	82.97	**80.51**	**61.62**	**13.65**

Table 6: Full results of top-1 (%) accuracy for HorNet on ImageNet.

Model	Method	Dense	κ=10%	20%	30%	40%	50%	60%
HorNet-T	MP	82.58	82.49	80.90	75.88	60.19	4.62	0.18
	MP+CAF		82.48	81.37	77.47	70.53	56.36	8.26
	CHITA-S		82.52	80.92	75.90	60.28	6.91	0.19
	CHITA-S+CAF		82.49	81.36	77.43	70.41	57.79	9.14
	CHITA		**82.57**	82.30	81.02	78.22	51.52	1.61
	CHITA+CAF		82.54	**82.40**	**81.38**	**79.57**	**66.35**	**39.68**
HorNet-S	MP	83.66	83.55	81.37	77.16	65.10	25.40	0.39
	MP+CAF		83.48	82.24	80.41	76.25	67.02	26.99
	CHITA-S		83.55	81.33	77.33	65.43	26.74	0.39
	CHITA-S+CAF		83.48	82.22	80.46	76.23	67.24	26.92
	CHITA		83.60	**83.48**	82.92	80.63	64.98	9.98
	CHITA+CAF		**83.61**	**83.48**	**83.02**	**81.50**	**76.40**	**50.49**
HorNet-B	MP	84.05	84.01	83.34	80.31	72.61	41.12	1.25
	MP+CAF		83.95	83.46	81.12	78.63	68.16	43.49
	CHITA-S		83.99	83.35	80.32	72.58	40.93	1.35
	CHITA-S+CAF		83.95	83.42	81.15	78.62	68.14	45.16
	CHITA		**84.03**	83.88	83.48	82.50	76.34	28.10
	CHITA+CAF		84.01	**83.91**	**83.61**	**82.99**	**80.44**	**63.79**

ConvNeXt-T, LS after MP achieves an accuracy of 1.89% against 1.93% for MP and 59.80% for CAFFEINATE. Scaling each channel by ρ_j minimizes the least square error but reinstates the variance deficit CAFFEINATE removes, i.e., regression toward the mean mutes channels whose scale CAFFEINATE restores. Within the affine subspace, recovery therefore depends on restoring the moments, not on reconstructing the original output.

F Relation to Statistical Corrections in Adjacent Settings

This section presents how CAFFEINATE differs from two lines of work that correct activation statistics after a model has been modified.

REPAIR (Jordan et al., 2023). The context is weight-space *interpolation* and not pruning. Two independently trained networks are linearly interpolated. The failure REPAIR identifies is that the activation variance of the interpolated network decays with depth.

Table 7: Full results of top-1 (%) accuracy for RDNet on ImageNet.

Model	Method	Dense	κ=10%	20%	30%	40%	50%	60%
RDNet-T	MP	82.78	82.52	80.01	63.45	19.25	2.15	0.17
	MP+CAF		82.49	80.90	76.41	58.85	40.92	19.36
	CHITA-S		82.53	80.02	63.49	19.71	3.05	0.23
	CHITA-S+CAF		82.51	80.80	76.35	58.67	41.32	19.45
	CHITA		**82.74**	82.43	81.28	43.39	18.66	2.64
	CHITA+CAF		82.73	**82.51**	**81.88**	**71.87**	**59.95**	**25.39**
RDNet-S	MP	83.54	83.12	79.14	59.03	26.21	8.62	0.31
	MP+CAF		83.38	81.22	75.46	56.69	39.32	21.01
	CHITA-S		83.12	79.18	59.08	27.13	9.45	0.34
	CHITA-S+CAF		83.35	81.13	75.38	56.43	39.49	20.75
	CHITA		**83.48**	83.07	76.62	47.60	15.66	2.35
	CHITA+CAF		**83.48**	**83.33**	**79.94**	**60.81**	**43.71**	**29.54**
RDNet-B	MP	84.24	84.00	81.90	68.93	44.04	12.36	1.04
	MP+CAF		84.00	83.00	79.65	72.83	60.46	37.73
	CHITA-S		84.00	81.93	69.12	45.59	15.02	1.30
	CHITA-S+CAF		83.98	82.93	79.54	72.70	60.28	38.53
	CHITA		84.21	**84.02**	82.67	73.27	13.06	5.08
	CHITA+CAF		**84.22**	84.01	**83.34**	**80.11**	**67.28**	**50.64**

Table 8: Top-1 (%) accuracy of MP + per-channel least squares (Equation (17)) on ImageNet, $n = 1{,}000$. MP and CAF under the identical protocol appear in Tables 4 to 7.

Model	κ=10%	20%	30%	40%	50%	60%
ConvNeXt-T	81.93	80.33	73.90	17.58	1.89	0.67
ConvNeXt-B	83.54	82.19	71.92	6.98	1.90	0.66
ConvNeXt-L	84.07	83.57	79.59	14.37	1.58	0.65
FocalNet-T	81.70	80.28	72.51	14.86	0.56	0.14
FocalNet-S	83.06	81.69	68.77	1.23	0.22	0.18
FocalNet-B	83.48	82.56	30.65	0.98	0.20	0.12
HorNet-T	82.43	80.60	60.10	2.15	0.40	0.12
HorNet-S	83.48	81.24	66.13	3.42	0.36	0.16
HorNet-B	83.91	83.15	72.97	14.51	0.34	0.16
RDNet-T	82.52	79.94	44.70	3.27	0.94	0.39
RDNet-S	83.38	78.63	22.25	2.66	1.01	0.50
RDNet-B	84.01	81.88	30.10	4.18	1.01	0.44

The correction is *carried* by BatchNorm. REPAIR inserts BatchNorm layers, sets their affine parameters to the target moments, and estimates the running statistics from a calibration pass. For architectures with BatchNorm after every convolution this reduces to recalibrating those running statistics. LayerNorm holds no such stored state, which is the premise of the problem addressed in this paper.

Energy compensation (Yu et al., 2026). In post-training pruning of large language models, Yu et al. (2026) combine a variance-calibrated importance score with an *energy compensation* step. After the pruning mask is fixed, the surviving weights are rescaled, column-wise and then row-wise, until their centered ℓ_2 energy matches that of the dense weight matrix. Like CAFFEINATE, this is closed-form and retraining-free. It differs in where it is applied. The correction is computed from, and applied to, the *weights*. Every surviving weight in every layer is updated. Restoring the retained squared-weight energy reproduces the pruned activation variance only under *assumed* homogeneous input variances, because the pruned variance $\sum_{i\in\mathcal{S}} W'^2_{\ell,i}\mathrm{Var}(H_{\ell-1,i})$ depends on input statistics that the weights do not determine. CAFFEINATE measures the statistics instead of assuming it, and updates only the LayerNorm affine parameters, leaving every convolution and linear weight unchanged.

G CONTRIBUTION OF THE SCALE AND THE SHIFT

Equation (13) corrects two values per channel, namely the scale γ'_j, which restores the channel's standard deviation, and the shift β'_j, which restores its mean. Figure 8 compares correcting only the scale, only the shift, and both (CAFFEINATE) on the ConvNeXt family at $\kappa = 60\%$.

Only correcting the scale or the shift is not sufficient. For example, comparing the accuracy gain over MP, the scale adds $+1.0\%$ to $+2.9\%$ and the shift $+6.1\%$ to $+18.3\%$, against $+34.3\%$ to $+42.9\%$ for correcting both the scale and the shift (CAFFEINATE).

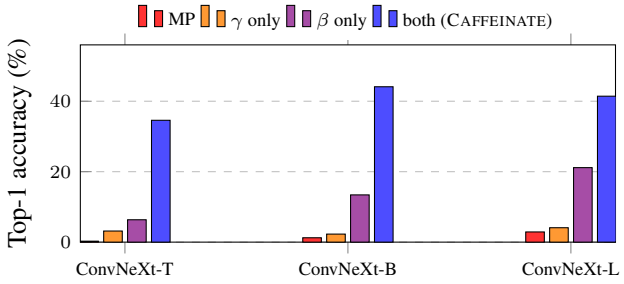


Figure 8: **Scale versus shift** on the ConvNeXt family at $\kappa = 60\%$.

H CALIBRATION UNDER DISTRIBUTION SHIFT

CAFFEINATE corrects the affine from calibration images, so its sensitivity to a mismatch between the calibration images and the evaluation data needs to be quantified. Figure 9 adds Gaussian noise of standard deviation σ to the calibration images *after* ImageNet normalization.

CAFFEINATE tolerates can tolerate some noise. For example, at $\sigma = 0.3$ it is within 0.7% of the accuracy obtained with uncorrupted calibration images, and even at $\sigma = 1.0$ it still recovers $+9.3\%$ over MP. Beyond $\sigma = 0.3$, accuracy falls monotonically. However, at $\sigma = 0.1$ the top-1 accuracy reaches 39.65%, 5% *above* the value obtained with uncorrupted calibration images.

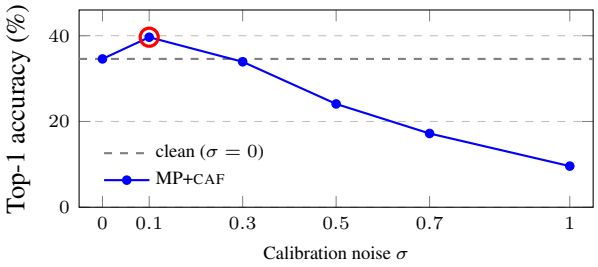


Figure 9: **Calibration under distribution shift**, ConvNeXt-T at $\kappa = 60\%$. Dashed line shows the accuracy when calibrating with samples with no noise.