Denoised Variance-Based Pruning with Optimal Brain Bias Compensation

Geon Tack Lee[1,2], Jaegul Choo[1] ✉, and Kang Eun Jeon[1] ✉

[1] KAIST [2] New York University Abu Dhabi
gtl256@nyu.edu, {jchoo, kejeon}@kaist.ac.kr
✉ Corresponding authors.

Abstract. Vision Transformers (ViTs) achieve state-of-the-art performance but carry massive computational overhead that restricts edge deployment. Although structural pruning has emerged as a key strategy to reduce these costs, existing methods often suffer from severe accuracy degradation or require expensive retraining. Recently, Variance-Based Pruning (VBP) introduced a promising paradigm by selecting neurons based on activation variance; however, it remains limited by statistical noise in finite-sample activation covariance and reliance on bias-only updates that cannot fully account for structural reconstruction error. To address these limitations, we introduce Denoised Variance-Based Pruning with Optimal Brain Bias Compensation (DVBP + OB^{2}C). We leverage random matrix theory to filter noise from the activation covariance spectrum for robust neuron selection and mathematically prove that integrating mean-shift compensation into the Optimal Brain Compression objective reduces the layer-wise Hessian exactly to the activation covariance matrix. This enables an optimal, closed-form update of the remaining weights using the same statistics gathered for selection. Extensive experiments on DeiT, Swin, and ConvNeXt architectures demonstrate that DVBP + OB^{2}C achieves state-of-the-art training-free performance; at 50% MLP pruning, it retains over 90% of the original Top-1 accuracy on Small and Base variants, outperforming VBP by up to 29.46% (ConvNeXt-T) and 7.33% (Swin-S). The code is available at: https://github.com/geontackee/DVBP_OB2C.

Keywords: Vision Transformers, Structured Pruning, Variance-Based Pruning

1 Introduction

Since the introduction of the Vision Transformer (ViT) [5], transformer-based architectures have consistently defined the state-of-the-art in computer vision, frequently outperforming traditional Convolutional Neural Networks (CNNs) [15]. However, this empirical success comes at a steep computational price. Standard ViTs demand substantial memory and computational resources for inference [27], effectively precluding their deployment on resource-constrained edge devices. For instance, ViT-B/16 requires 86 million parameters [5], contrasting sharply with

the 7.5 million of MobileNetV3 [17]. As vision models are increasingly deployed on-device for latency-critical applications, such as autonomous driving, robotics, and unmanned aerial systems, achieving smaller model footprints and higher inference throughput has become an indispensable research objective.

While pruning has emerged as a principled strategy to bridge this efficiency gap [13,16], current paradigms present a fundamental trade-off between theoretical sparsity and practical acceleration. Unstructured methods [13, 20] achieve high compression ratios by zeroing individual weights. However, these methods produce irregular sparsity patterns that demand specialized hardware accelerators [6] or custom CUDA kernels to induce practical latency reductions [11]. This severely limits their scalability and real-world adoption. On the other hand, structured pruning [21,29] physically removes entire neurons or filters, hence directly reducing model size and inference time on commodity hardware. However, existing structured methods predominantly follow the magnitude-based pruning paradigm [13, 21], which ranks parameters by the absolute magnitude of their weights under the assumption that smaller weights contribute less to the network. While this heuristic is effective for zeroing individual connections, removing entire neurons or filters based on aggregate magnitude discards complex feature interactions and leads to severely degraded accuracy [24].

In a fundamentally different direction, *Variance-Based Pruning* (VBP) [2] has demonstrated that this trade-off need not be inevitable. Rather than relying on weight magnitudes, VBP introduces an activation variance criterion that identifies neurons whose activations exhibit the lowest variance across a calibration set. The key insight is that low-variance neurons behave approximately as constants, meaning their contribution to the downstream layer can be effectively absorbed into the bias term through a simple mean-shift compensation. This allows VBP to structurally remove neurons while immediately recovering a portion of the lost performance, substantially outperforming magnitude-based approaches. However, VBP still suffers from significant accuracy degradation at higher pruning ratios, leaving considerable room for improvement.

In this work, we identify two critical limitations of VBP. First, the activation covariance matrix, from which variance scores are derived, is corrupted by statistical noise. This noise obscures the true importance ranking of neurons, leading to increasingly suboptimal pruning decisions at aggressive compression ratios (*e.g.* $\geq$50%). Second, VBP compensates for pruned neurons solely through mean-shift compensation, leaving the remaining weight matrices entirely unmodified. We observe that mean-shift compensation cannot fully account for the reconstruction error introduced by removing entire neurons, and a principled compensation through weight update is necessary to close this gap.

To address these limitations, we propose Denoised Variance-Based Pruning with Optimal Brain Bias Compensation (DVBP + OB^{2}C). Our core contributions are:

– **Denoised Variance Metric:** We leverage random matrix theory, specifically the Marchenko-Pastur distribution, to separate signal eigenvalues from noise in the activation covariance spectrum. The resulting denoised variance

scores yield a substantially more reliable neuron importance ranking, particularly at high pruning ratios.
- **OB^{2}C Framework:** We incorporate mean-shift compensation directly into the seminal Optimal Brain Compression (OBC) [8] reconstruction objective. This integration yields a remarkably elegant result: the layer-wise Hessian reduces exactly to the activation covariance matrix. Consequently, the same covariance matrix simultaneously serves as the basis for both neuron selection (via denoised variance) and optimal multi-weight recovery, unifying the two stages of our pipeline into a single, coherent statistical framework.

Extensive experiments on DeiT [27], Swin [22], and ConvNeXt [23] architectures across Tiny, Small, and Base scales demonstrate that our method consistently outperforms VBP by a widening margin as the pruning ratio increases. At a 50% structural reduction, DVBP + OB^{2}C retains approximately 80% Top-1 accuracy on Tiny-scale models and exceeds 90% on Small and Base variants, surpassing VBP by up to 7.33% on Swin-S and 29.46% on ConvNeXt-T.

2 Background and Related Works

Model Compression and Pruning. Model compression relies primarily on quantization [1,10] and pruning [16,19]. Unlike quantization, which reduces bit-width but often demands specialized low-bit hardware, pruning directly eliminates operations, natively accelerating inference. Pruning roots trace to second-order methods like OBD [18] and OBS [14]. Later approaches introduced magnitude thresholding [13], gradient-based sensitivities [20], and identified sparse, trainable subnetworks [7]. However, these *unstructured* methods create irregular sparsity requiring custom accelerators [11]. *Structured* pruning instead removes entire neurons or channels [21], yielding immediate speedups on commodity hardware at the cost of severe accuracy drops, typically requiring extensive iterative retraining [28].

Variance-Based Pruning. Variance-Based Pruning (VBP) [2] selects neurons for removal based on activation variance rather than weight magnitude. Given an input activation vector $\mathbf{x}$ with mean $\boldsymbol{\mu}$, VBP introduces a mean-shift vector to compensate for pruned neurons. The mean-shift vector $\Delta\boldsymbol{\mu}$'s j-th entry is $\boldsymbol{\mu}_j$ if neuron j is pruned and zero otherwise. The output of the pruned layer is then expressed as:

$$\mathbf{y} = \hat{\mathbf{W}}\hat{\mathbf{x}} + \mathbf{b}' \tag{1}$$

where $\hat{\mathbf{W}}$ is the pruned weight of the subsequent layer, $\hat{\mathbf{x}}$ is the pruned output, and $\mathbf{b}'$ is the mean-shift compensation term that absorbs the effect of the pruned neurons into the bias:

$$\mathbf{b}' = \mathbf{b} + \mathbf{W}\Delta\boldsymbol{\mu}. \tag{2}$$

After this compensation, the residual reconstruction error for neuron i is governed by its activation variance, σ_i^2:

$$\mathbb{E}[(x_i - \mathbb{E}[x_i])^2] = \mathbb{E}[(x_i - \mu_i)^2] = \sigma_i^2 \tag{3}$$

Therefore, pruning neurons with the lowest σ_i^2 minimizes the expected reconstruction error under bias compensation.

Optimal Brain Compression. The Optimal Brain Compression (OBC) framework [8] has served as the foundation for numerous model compression algorithms, notably extending to Large Language Models through methods such as GPTQ [10] and SparseGPT [9]. Building upon the Optimal Brain Surgeon (OBS) [14] framework, OBC minimizes the squared reconstruction error between the original and compressed layer outputs:

$$\underset{\hat{\mathbf{W}}}{\operatorname{argmin}} ||\mathbf{W}\mathbf{X} - \hat{\mathbf{W}}\mathbf{X}||_F^2, \tag{4}$$

where $\hat{\mathbf{W}}$ is the compressed weight matrix and $\mathbf{X} \in \mathbb{R}^{d \times N}$ is the input activation matrix whose columns are activation vectors collected from N calibration samples. To minimize this objective, the OBC framework derives a block-wise closed-form update rule that adjusts the remaining weights as

$$\delta_P = -\mathbf{H}^{-1}_{:,P}((\mathbf{H}^{-1})_P)^{-1}\mathbf{w}_P \tag{5}$$

where δ_P is the weight update for block P, $\mathbf{H} = 2\mathbf{X}\mathbf{X}^T$ is the layer-wise proxy-Hessian, $\mathbf{w}_P$ is the current weight vector of the row being updated, and P denotes the set of indices corresponding to the block being compressed. Recent work OBS-Diff [30] adapts second-order statistics for structured pruning of diffusion models. However, OBS-Diff predominantly utilizes complex, Hessian-based OBS calculations as the primary criterion for selecting which structures to remove. Our method decouples this pipeline: we employ an efficient denoised variance metric to identify redundant neurons, reserving the OBC framework strictly for optimal weight recovery after pruning.

3 Methodology

In this section, we present the DVBP + OB^{2}C structured pruning pipeline. We begin by augmenting the OBC [8] reconstruction objective with mean-shift compensation (§3.1), showing that the resulting layer-wise Hessian reduces to the input activation covariance matrix. We then describe an online method for computing this covariance without prohibitive memory costs (§3.2). To address the inherent noise in finite-sample covariance estimates, we apply spectral denoising via the Marchenko-Pastur distribution (§3.3), yielding robust variance scores for neuron selection (§3.4). Finally, we detail the end-to-end pruning and weight update mechanism (§3.5).

3.1 Optimal Brain Bias Compensation (OB^{2}C)

To mitigate the mean-shift error introduced during weight pruning, we introduce Optimal Brain Bias Compensation (OB^{2}C), which explicitly incorporates a bias

term into the layer-wise reconstruction objective. Given an input activation matrix $\mathbf{X} \in \mathbb{R}^{d \times N}$, original weights $\mathbf{W}$, pruned weights $\hat{\mathbf{W}}$, and a bias vector $\mathbf{b}$, the augmented objective is formulated as:

$$\arg\min_{\hat{\mathbf{W}},\mathbf{b}} \left\| \mathbf{W}\mathbf{X} - (\hat{\mathbf{W}}\mathbf{X} + \mathbf{b}\mathbf{1}^T) \right\|_F^2 \tag{6}$$

where $\mathbf{1}$ is an N-dimensional vector of ones. Setting the partial derivative of the objective with respect to $\mathbf{b}$ to zero yields the optimal bias compensation:

$$\mathbf{b}^* = (\mathbf{W} - \hat{\mathbf{W}})\boldsymbol{\mu} \tag{7}$$

where $\boldsymbol{\mu} = \frac{1}{N}\mathbf{X}\mathbf{1}$ denotes the mean vector of the input activations. Substituting $\mathbf{b}^*$ back into the objective function yields a simplified minimization problem:

$$\arg\min_{\Delta\mathbf{W}} \left\| \Delta\mathbf{W}\widetilde{\mathbf{X}} \right\|_F^2 \tag{8}$$

where $\Delta\mathbf{W} = \mathbf{W} - \hat{\mathbf{W}}$ represents the weight perturbation, and $\widetilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\mathbf{1}^T$ represents the mean-centered activations.

Evaluating the error induced by this weight perturbation, we find that the layer-wise proxy-Hessian matrix with respect to the rows of $\Delta\mathbf{W}$ is given by:

$$\mathbf{H} = 2\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top = 2N\mathbf{C} \tag{9}$$

where N is the calibration set size and $\mathbf{C}$ is the activation covariance matrix. Consequently, the layer-wise Hessian is directly proportional to the covariance of the centered activations, rather than the uncentered second moment used in standard formulations. Integrating this into the OBC multi-weight update, we replace the standard Hessian with $2N\mathbf{C}$. Let P denote the set of indices corresponding to the pruned neurons. The update formula for the weights is rewritten as:

$$\delta_{\mathbf{W}} = -\mathbf{W}_{:,P} \left([\mathbf{C}^{-1}]_{P,P} \right)^{-1} [\mathbf{C}^{-1}]_{P,:} \tag{10}$$

This yields an update expression that is structurally identical to the standard OBC rule, but inherently minimizes the mean-shift error by operating on the centered covariance matrix $\mathbf{C}$.

3.2 Covariance Matrix Computation

To apply OB^{2}C, we first require the exact centered covariance matrix of the activations for each layer. Storing full-precision activations for a naive computation demands an $\mathcal{O}(N)$ memory footprint, rendering the approach computationally infeasible on large calibration sets. To ensure our approach remains computationally tractable, we track each layer's covariance matrix in an online, batched manner during calibration. To track the activation statistics of neurons, the VBP [2] paper employs Welford's algorithm to iteratively update their mean and variance. In a similar vein, Chan et al. [3] introduced a robust formula for

updating the mean and second central moment across partitioned dataset $\mathbf{A}$ partitioned into $\mathbf{B}$ and $\mathbf{C}$ where $\mathbf{A} = \mathbf{B} \cup \mathbf{C}$, using the expressions from Pébay [26]:

$$\boldsymbol{\mu_A} = \boldsymbol{\mu_B} + n_{\mathbf{C}} \frac{\delta_{\mathbf{C},\mathbf{B}}}{n}, \qquad \mathbf{M}_{2,\mathbf{A}} = \mathbf{M}_{2,\mathbf{B}} + \mathbf{M}_{2,\mathbf{C}} + n_{\mathbf{B}} n_{\mathbf{C}} \frac{\delta^2_{\mathbf{C},\mathbf{B}}}{n_A} \tag{11}$$

where $\mathbf{M}_2$ denotes the second central moment matrix,$n_{\mathbf{A}}$, $n_{\mathbf{B}}$, $n_{\mathbf{C}}$ are the respective cardinalities, and $\delta_{\mathbf{C},\mathbf{B}} = \boldsymbol{\mu}_{\mathbf{C}} - \boldsymbol{\mu}_{\mathbf{B}}$ is the difference between the partition means.

Let $\mathcal{P}$ denote the set of previously processed activations containing $n_{\mathcal{P}}$ samples, and let $\mathcal{N}$ denote an incoming new batch containing $n_{\mathcal{N}}$ samples. The combined set is given by $\mathcal{C} = \mathcal{P} \cup \mathcal{N}$, with a total of $n_{\mathcal{C}} = n_{\mathcal{P}} + n_{\mathcal{N}}$ samples. When a new batch arrives, we first update the running mean of the activations using Eq. 11.

To update the sample covariance matrix, we define the weighting factor $\alpha = \frac{n_{\mathcal{P}}}{n_C}$, which implies $1 - \alpha = \frac{n_{\mathcal{N}}}{n_C}$. By decomposing the second moment across the disjoint sets $\mathcal{P}$ and $\mathcal{N}$, we derive the following online update formula for the covariance matrix:

$$\mathbf{C}_{\mathcal{C}} = \alpha \mathbf{C}_{\mathcal{P}} + (1 - \alpha)\mathbf{C}_{\mathcal{N}} + \alpha(1 - \alpha)\boldsymbol{\delta}\boldsymbol{\delta}^T \tag{12}$$

where $\boldsymbol{\delta} = \boldsymbol{\mu}_{\mathcal{N}} - \boldsymbol{\mu}_{\mathcal{P}}$ represents the difference between the mean vectors of the new batch and the previously accumulated data. Using this recursive formulation, we can accurately construct the sample covariance matrix of each layer for an arbitrarily large calibration set without requiring the full activation history to be held in memory.

3.3 Denoising the Covariance Matrix with Marchenko-Pastur

Marchenko-Pastur Distribution. We observe that the spectral density of the activation covariance matrix $\mathbf{C}$ exhibits a bulk profile characteristic of the MP [25] distribution, strongly suggesting that the underlying signal subspace is corrupted by noise. To mitigate this, we denoise $\mathbf{C}$ by conducting eigendecomposition on $\mathbf{C}$ and by fitting its eigenvalue spectrum to the MP distribution. The MP distribution describes the asymptotic eigenvalue behavior of large random matrices. Specifically, for a large random matrix $\mathbf{X}$ of size $m \times n$ whose entries are independent and identically distributed (i.i.d.) with mean 0 and a variance of σ^2, the probability density function of the eigenvalues of its corresponding sample covariance $\mathbf{Y}_n = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ is expressed as:

$$f(x) = \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{2\pi\sigma^2\lambda x} \tag{13}$$

for $\lambda_- < x < \lambda_+$ and 0 otherwise. Here, $\lambda = \frac{m}{n}$ represents the aspect ratio of the matrix dimensions. According to the MP Law, if $\mathbf{X}$ satisfies the conditions, all of its non-zero eigenvalues will asymptotically fall within the theoretical bounds of λ_+ and λ_-. These bounds can be obtained by the following equation

$$\lambda_\pm = \sigma^2(1 \pm \sqrt{\lambda})^2. \tag{14}$$

Noise Variance of Noisy Matrices. Gavish and Donoho [12] focuses on the analysis of noisy matrices under the additive model $\mathbf{Y} = \mathbf{X} + \sigma\mathbf{Z}$, where $\mathbf{Y}$ is the observed data matrix, $\mathbf{X}$ is the underlying low-rank signal, and $\mathbf{Z}$ represents random noise. A primary challenge in practical application is that the true noise scale, σ, is typically unknown. To address this, the authors formulate a robust estimator $\hat{\sigma}(\mathbf{Y})$ using the ratio of the median singular value of the data matrix to the theoretical median of the MP distribution:

$$\hat{\sigma}(\mathbf{Y}) \stackrel{\text{def}}{=} \frac{y_{med}}{\sqrt{n\mu_\lambda}}, \tag{15}$$

where y_{med} is the median singular value of $\mathbf{Y}$. μ_λ is the median of MP distribution obtainable by solving for $\lambda_- < x < \lambda_+$

$$\int_{\lambda_-}^{x} \frac{\sqrt{(\lambda_+ - t)(t - \lambda_-)}}{2\pi t} dt = \frac{1}{2}. \tag{16}$$

Following our observation where eigenvalue distribution contains MP-like noise, visualized in Fig. 2. We assume the centered activation data matrix $\widetilde{\mathbf{X}}$ follows an additive noise model, $\widetilde{\mathbf{X}} = \widetilde{\mathbf{X}}_{\text{signal}} + \sigma\mathbf{Z}$, where the entries of the noise matrix $\mathbf{Z}$ adhere to the Marchenko-Pastur conditions.

Following Gavish and Donoho's work [12], we estimate the noise scale σ by comparing the median eigenvalue of $\mathbf{C}$ to the theoretical median of the MP distribution. The median singular value of the data matrix, denoted as s_{med}, relates to the median eigenvalue of the covariance matrix, $\mathit{\Lambda}_{\text{med}}$, obtained through eigendecomposition as follows:

$$s_{\text{med}} = \sqrt{N\mathit{\Lambda}_{\text{med}}}. \tag{17}$$

Substituting this relationship into the estimator Eq. 15 yields:

$$\hat{\sigma}(\widetilde{\mathbf{X}}) = \sqrt{\frac{\mathit{\Lambda}_{\text{med}}}{\mu_\lambda}} \tag{18}$$

where μ_λ is the theoretical median of the MP distribution.

To obtain μ_λ, we require matrix $\widetilde{\mathbf{X}}$'s aspect ratio λ. In this context, we define the aspect ratio as $\lambda = \frac{d}{N}$, where d is the number of input neurons and N is the calibration set size. We use calibration set size (number of independent images) rather than the total number of patches for the denominator because patches within a single image are highly correlated, using the total patch count would violate the i.i.d. noise assumption. By defining our sample size strictly as the calibration set size, we ensure the i.i.d. assumption holds. With the aspect ratio λ, we compute the theoretical median eigenvalue μ_λ via Eq. 16, which subsequently allows us to estimate the noise scale σ.

Using the estimated noise variance, we calculate the theoretical MP upper bound λ_+ using Eq. 14. After computing the eigendecomposition of the covariance matrix $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ contains the eigenvalues

and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_d]$ the corresponding eigenvectors, we apply hard thresholding to retain only the signal components:

$$\mathbf{\Lambda}_{\text{signal}} = \text{diag}(\lambda_i) \quad \forall \lambda_i > \lambda_+ \tag{19}$$

$$\mathbf{V}_{\text{signal}} = [\mathbf{v}_i] \quad \forall \lambda_i > \lambda_+ \tag{20}$$

3.4 Denoised Variance Score

Using the retained signal components, we reconstruct the denoised covariance matrix:

$$\mathbf{C}_{\text{denoised}} = \mathbf{V}_{\text{signal}} \mathbf{\Lambda}_{\text{signal}} \mathbf{V}_{\text{signal}}^T \tag{21}$$

Finally, we extract the denoised variance scores for each neuron directly from the diagonal entries of the reconstructed matrix:

$$\mathbf{s}_{\text{dvar}} = \text{diag}(\mathbf{C}_{\text{denoised}}) \tag{22}$$

Empirically, we observe that the denoised variance scores $\mathbf{s}_{\text{dvar}}$ excel at high pruning ratios, whereas the raw variance scores $\mathbf{s}_{\text{var}} = \text{diag}(\mathbf{C})$, extracted from the original covariance matrix, remain highly effective at low pruning ratios. Thus, we store the raw scores prior to eigendecomposition and formulate a hybrid scoring metric, controlled by the target pruning ratio p:

$$\mathbf{s}_{\text{hybrid}} = (1 - p)\mathbf{s}_{\text{var}} + p\mathbf{s}_{\text{dvar}} \tag{23}$$

3.5 Pruning and Update Mechanism

Following the structural framework of Variance-Based Pruning (VBP), we target MLP blocks in Vision Transformers (ViTs) [5] and ConvNeXt [23] models. While we adopt this standard layer selection protocol, our approach introduces a novel neuron selection criterion and weight update mechanism.

Specifically, we globally rank the hidden dimensions based on our proposed hybrid score, and select the lowest-ranking $p\%$ of neurons across the entire network to consist the pruned indices set P. Rather than enforcing a uniform per-layer pruning ratio, the network dynamically allocates capacity, preserving critical layers while heavily penalizing redundant ones.

We apply pruning and compensation in a two-stage process. First, we apply our novel OB^{2}C update to the outgoing weight matrix of the block (the second linear layer). We do this by by adding an update term, $\delta_{\mathbf{W}}$, computed using the pruned indices set P, to the original weights $\mathbf{W}$. This effectively prunes the neurons in P while compensating the surviving neurons. Then, we apply mean-shift compensation to the subsequent layer, which is the optimal bias term, $\mathbf{b}^*$, to its existing bias.

For a standard two-layer block, pruning reduces the intermediate hidden dimension from d_{hidden} to d_{pruned}. Consequently, the weight matrix of the first layer is reduced from $d_{\text{in}} \times d_{\text{hidden}}$ to $d_{\text{in}} \times d_{\text{pruned}}$, and the weight matrix of the second layer is reduced from $d_{\text{hidden}} \times d_{\text{out}}$ to $d_{\text{pruned}} \times d_{\text{out}}$. Because the external input (d_{in}) and output (d_{out}) dimensions remain strictly preserved, the pruned block seamlessly integrates back into the overarching network architecture.

Table 1: Comparison of zero-shot Top-1 accuracy (%) on ImageNet-1K across different pruning methods on DeiT and Swin. All models are evaluated immediately after compression without any post-pruning fine-tuning.

Model	Prune Ratio (%)	Top-1 Accuracy (%)					Model	Prune Ratio (%)	Top-1 Accuracy (%)				
		Full	Mag.	SNIP	VBP	Ours			Full	Mag.	SNIP	VBP	Ours
DeiT-T	40	72.16	14.64	55.80	57.56	**62.56**	Swin-T	40	81.38	47.39	37.99	64.25	**72.23**
	50		3.73	39.49	42.89	**56.40**		50		26.40	28.87	49.98	**66.12**
	60		0.81	23.59	26.97	**47.58**		60		7.91	17.87	26.03	**56.42**
DeiT-S	40	79.86	20.68	55.37	72.35	**73.71**	Swin-S	40	83.32	2.19	62.08	78.41	**80.69**
	50		4.24	50.65	66.03	**70.25**		50		0.22	45.36	69.91	**77.24**
	60		1.17	42.17	54.73	**65.25**		60		0.14	25.72	45.30	**70.79**
DeiT-B	40	81.98	1.95	65.03	76.49	**79.00**	Swin-B	40	85.27	35.62	57.02	78.70	**80.56**
	50		0.38	62.78	68.92	**75.85**		50		6.44	38.55	70.86	**76.16**
	60		0.16	52.82	50.63	**69.68**		60		1.10	23.81	55.88	**69.00**

Table 2: Comparison of parameter counts (M), MACs (G), and Top-1 accuracy (%) for DeiT models at a 50% pruning ratio.

Model	Parameters (M)		MACs (G)		Top-1 Accuracy (%)				
	Full	Pruned	Full	Pruned	Full	Mag.	SNIP	VBP	Ours
DeiT-T	5.72	3.94	1.26	0.91	72.16	3.73	39.49	42.89	**56.40**
DeiT-S	22.05	14.96	4.61	3.21	79.86	4.24	50.65	66.03	**70.25**
DeiT-B	86.57	58.24	17.59	12.01	81.98	0.38	62.78	68.92	**75.85**

4 Experiments and Test Results

We evaluate our DVBP+OB^{2}C structured pruning framework on the Tiny, Small, and Base variants of the DeiT [27], Swin [22], and ConvNeXt [23] architectures, all pre-trained on the ImageNet-1K [4] dataset. To compute the activation covariance matrices, we construct a calibration dataset of 8,192 images, randomly sampled from the ImageNet-1K training split using a fixed seed. Notably, we omit data augmentation during calibration to accurately capture the true, unperturbed neuron activation statistics. Our evaluation primarily focuses on direct post-pruning accuracy retention, aiming to maximize zero-shot, off-the-shelf performance without relying on expensive retraining or fine-tuning pipelines. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU. To guarantee a rigorous and fair comparison, we locally compute both the uncompressed baseline accuracies and the post-pruning performance of all evaluated methods under the exact same setup.

4.1 Results

Our results demonstrate that the proposed approach consistently outperforms all baseline pruning methods across all evaluated models and pruning ratios. Notably, when compared to our closest competitor, VBP [2], the accuracy gap widens as the pruning ratio increases. For instance, on the Swin-S model, the

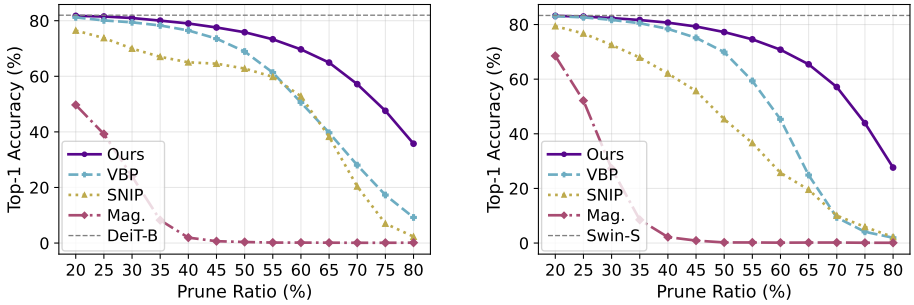
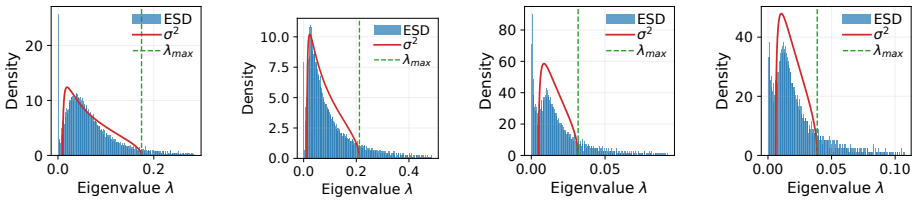


Fig. 1: Pruning accuracy across diverse pruning ratios for DeiT-B (left) and Swin-S (right). SNIP and Magnitude pruning are adjusted to structured pruning.



(a) DeiT-B Layer 9 (b) DeiT-B Layer 10 (c) Swin-S Layer 11 (d) Swin-S Layer 12

Fig. 2: Empirical spectral density fitted with the Marchenko-Pastur distribution. The largest 5% of eigenvalues are excluded for visual clarity.

post-pruning accuracy gap between our method and VBP is 2.28% at a 40% pruning ratio. This margin increases substantially to 7.33% at a 50% ratio, and expands even further to 25.49% at a 60% ratio.

As detailed in Table 1, at a 50% pruning ratio, our method successfully retains approximately 80% of the original accuracy for the Tiny models, and roughly 90% for the Small and Base models, representing a significant improvement over VBP. This diverging performance trend is further illustrated in Fig. 1, which shows the performance gap between our approach and VBP becoming increasingly pronounced from the 40% pruning threshold onward.

Finally, because our approach targets the same layers and utilizes the same fundamental pruning mechanism as the baselines, the overall parameter counts and multiply-accumulate operations (MACs) remain identical. However, despite this equivalent footprint, our method yields a superior top-1 accuracy (Table 2).

As illustrated in Fig. 2, the empirical eigenvalue distributions of the models exhibit a noise structure characteristic of the Marchenko-Pastur (MP) [25] distribution. Furthermore, the fitted MP [25] curve captures these noise components.

As shown in Fig. 3, our hybrid scoring mechanism yields a distinct pruning profile compared to VBP [2]. Specifically, our approach prunes the earlier layers less aggressively while increasing the pruning rate in the deeper layers. However, because both methods rely on a variance-based metric, their overall macroscopic

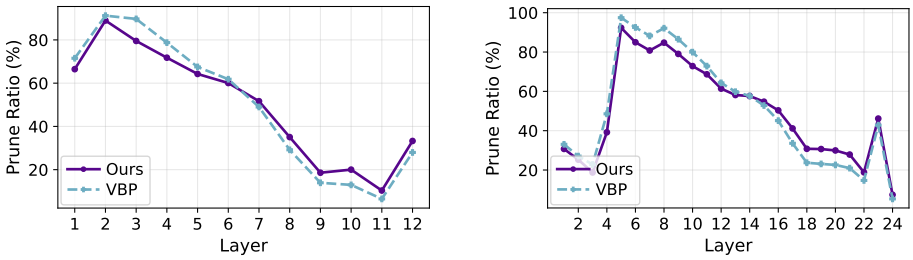


Fig. 3: Layerwise pruning ratio for DeiT-B (left) and Swin-S (right), for model pruning rate 50%.

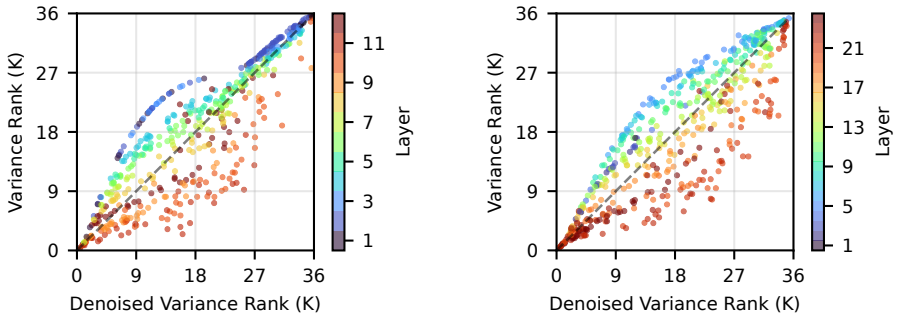


Fig. 4: Spearman correlation plot of DeiT-B (left) and Swin-S (right) between denoised and raw variance score rank.

trends remain similar. This relationship is further quantified by the Spearman rank correlation plot (Fig. 4), which reveals a general linear correlation between the ranks of the standard variance and our denoised scores, albeit with considerable deviations occurring within the middle portion of the rankings.

As demonstrated in Table 3 and Fig. 5, our pruning methodology generalizes effectively to the ConvNeXt [23] Tiny, Small, and Base models. SNIP [20] and Magnitude [13] pruning cause severe performance degradation on ConvNeXt architectures, reducing the accuracy to chance-level performance on the ImageNet-1K dataset. While VBP does achieve some accuracy retention, it is considerably lower than the baseline accuracies. Our method improves upon VBP by 29.46% for ConvNeXt-T, 21.88% for ConvNeXt-S, and 14.64% for ConvNeXt-B, representing a substantial improvement. This demonstrates that our method is highly applicable to diverse model architectures.

4.2 Analysis and Ablation Studies

Despite the significant accuracy gains, our method introduces minimal computational and memory overhead, ensuring its practical deployment on consumer-

Table 3: Results of ConvNeXt Pruning at 50%.

Model	Top-1 Accuracy (%)				
	Full	SNIP	Mag.	VBP	Ours
ConvNeXt-T	84.17	1.90	0.10	15.44	**44.90**
ConvNeXt-S	85.17	1.76	0.10	28.13	**50.01**
ConvNeXt-B	85.81	29.91	0.13	56.12	**70.76**
Model	**Parameters (M)**				
	Full	Mag.	SNIP	VBP	Ours
ConvNeXt-T	28.58	16.98	15.94	12.61	12.96
ConvNeXt-S	50.22	27.32	28.50	23.37	24.04
ConvNeXt-B	88.59	48.80	46.92	41.18	42.63
Model	**MACs (G)**				
	Full	Mag.	SNIP	VBP	Ours
ConvNeXt-T	4.46	2.09	2.48	2.95	2.94
ConvNeXt-S	8.68	4.34	4.67	5.11	5.08
ConvNeXt-B	15.35	7.37	8.44	8.90	8.85

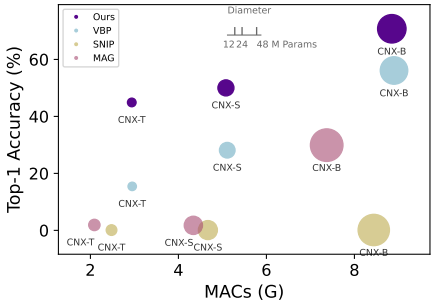


Fig. 5: Overall Diagram for ConvNeXt pruning results.

Table 4: Memory and VRAM usage comparison between VBP.

Model	Peak VRAM (G)		Time (s)	
	VBP	Ours	VBP	Ours
DeiT-T	0.52	1.25	14.44	19.25
DeiT-S	0.98	2.48	13.16	21.79
DeiT-B	2.01	5.25	26.66	51.15
Swin-T	3.31	3.91	15.03	24.5
Swin-S	3.40	5.96	23.88	44.14
Swin-B	4.58	8.11	34.89	66.92

Table 5: Ablation study: denoising and OB^{2}C.

Model	Top-1 Accuracy (%)			
	Full	VBP	+D	+D +O
DeiT-T	72.16	42.90	44.32	**56.40**
DeiT-S	79.86	66.03	65.37	**70.25**
DeiT-B	81.98	68.92	69.90	**75.85**
Swin-T	81.38	49.98	52.48	**66.12**
Swin-S	83.32	69.91	71.21	**77.24**
Swin-B	85.27	70.86	70.11	**76.16**

grade GPUs. Because we utilize a block-wise approach in OB^{2}C and precompute the pruning selections via our hybrid scoring mechanism, peak VRAM consumption is strictly bounded under 10 GB. Furthermore, the pruning process is highly time-efficient; as detailed in Table 4, while the execution time is approximately 2× that of VBP [2], it completes in under one minute for most architectures, requiring only 66.92 seconds for the larger Swin-B [22] model. These constrained resource requirements demonstrate that our approach can readily scale to larger models without necessitating specialized, high-memory hardware.

To evaluate the individual contributions of our proposed components, we analyze the isolated performance of the denoising mechanism and the OB^{2}C compensation framework, as detailed in Table 5. Integrating the denoised scoring yields marginal but consistent accuracy gains of 1-2% for the majority of the evaluated models, with the exception of DeiT-S [27] and Swin-B [22], which experience a negligible decrease of less than 1%. In contrast, the application of the OB^{2}C framework serves as the primary driver of performance recovery.

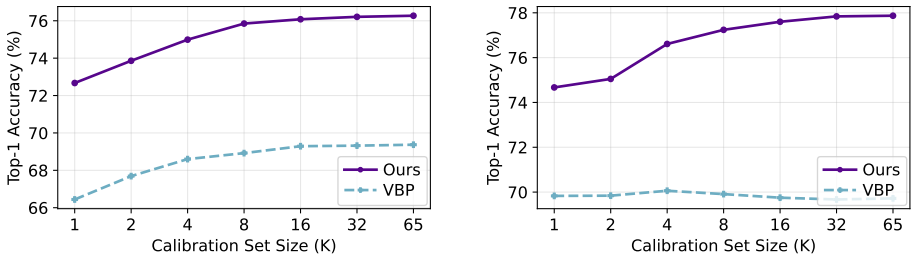


Fig. 6: Top1-Accuracy (%) across various calibration set sizes for DeiT-B (left) and Swin-S (right)

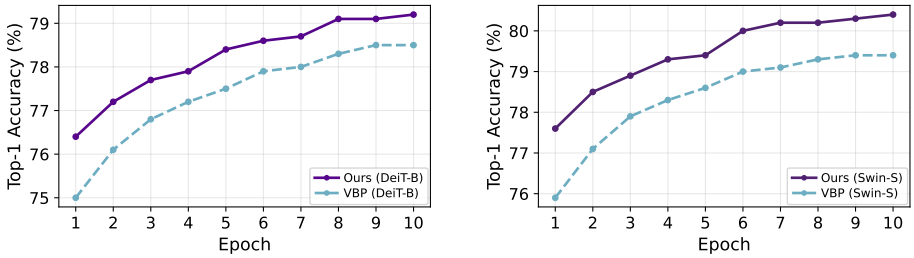


Fig. 7: Top1-Accuracy (%) across 10 epochs of finetuning for DeiT-B (left) and Swin-S (right).

Notably, it delivers a substantial accuracy improvement of nearly 14% for DeiT-T [27], while boosting the performance of the remaining models by 4-7%.

Because both VBP [2] and our proposed method can accommodate an arbitrary calibration set size, we investigate how varying this size impacts the final top-1 accuracy of each approach. While our primary experiments utilize a calibration dataset of 8,192 samples, our proposed method demonstrates robust scalability with respect to calibration size. As illustrated in Figure 6, increasing the calibration set initially yields top-1 accuracy improvements for DeiT-B pruning for both our method and VBP, with both approaches reaching convergence at approximately 8K samples. However, for Swin-S [22], our method exhibits continuous performance gains as the calibration set expands, achieving convergence at roughly 32K samples. In stark contrast, the performance of VBP shows no measurable improvement across a sweep from 1K to 65K samples.

To verify whether a high post-pruning accuracy yields higher finetuning accuracy, we finetune DeiT-B and Swin-S models with MLPs pruned by 70% via VBP and our framework. We use VBP's official finetuning script, which incorporates the unpruned model as the teacher model with AdamW optimizer, learning rate of 1.5e-5, cosine-annealing scheduler, and batch size of 32. Figure 7 shows that our proposed method achieves higher accuracy than the VBP baseline when finetuned for the same number of epochs.

Table 6: Ablation study: comparing various denoising methods.

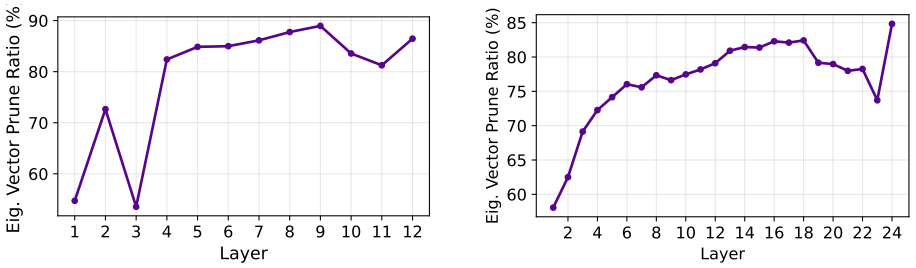
Model	Prune Ratio (%)	Top-1 Accuracy (%)						
		No Denoise	Top 50%	Top 60%	Top 70%	Top 80%	$<\lambda->$	$\lambda+>$
DeiT-B	50	75.76	**75.89**	75.71	75.73	75.76	0.14	75.85
	60	65.74	68.76	67.87	67.14	66.19	0.11	**69.68**
	70	45.49	51.81	49.84	47.49	45.89	0.12	**57.18**
Swin-S	50	76.67	**77.76**	77.64	77.41	76.95	0.28	77.24
	60	66.04	70.71	69.42	68.07	67.00	0.16	**70.79**
	70	37.26	48.89	43.30	40.65	38.12	0.12	**57.13**

Table 7: Ablation study: raw variance score vs. denoised variance score.

Prune Ratio (%)	Method	Top-1 Accuracy (%)					
		DeiT-T	DeiT-S	DeiT-B	Swin-T	Swin-S	Swin-B
30	Raw Var.	**65.45**	**75.32**	**79.41**	**71.97**	**81.70**	**82.56**
	Denoised Var.	63.55	73.38	77.45	71.35	81.05	80.11
70	Raw Var.	**16.64**	34.42	28.04	9.92	9.37	32.40
	Denoised Var.	14.19	**38.34**	**42.07**	**16.89**	**35.37**	**39.79**

We systematically compare various denoising strategies to validate our approach of fitting the MP [25] distribution and retaining only the signal eigenvectors associated with eigenvalues larger than λ_+. As demonstrated in Table 6, isolating eigenvectors with eigenvalues exceeding λ_+ consistently yields the highest top-1 accuracy among the evaluated denoising methods. Furthermore, this analysis confirms that eigenvectors with eigenvalues below λ_- predominantly capture noise, as reconstructing representations solely from these components degrades performance to chance-level accuracy. To quantify the extent of eigenvector pruning performed by our MP fitting method, we visualize the layer-wise pruning ratios in Figure 7. The results indicate that our method dynamically adjusts the pruning severity according to network depth; it discards approximately 60-70% of eigenvectors in the early layers and increases this ratio to roughly 80-90% in the deeper layers, a trend consistently observed in both the DeiT-B [27] and Swin-S [22] models. This adaptive behavior supports the hypothesis that representations become increasingly low-rank as they propagate deeper into the network, thereby requiring fewer signal eigenvectors to accurately capture the underlying information.

We also experimented with the affects of raw variance score and denoised variance score. Table 7 shows that denoised variance score is better than the raw score at high pruning ratios, but worse at low pruning ratios, where denoising inadvertently removes small signals. To address this, we propose the hybrid score, but this remains a highly heuristic approach. Developing a theoretically grounded hybrid score remains an important direction for future work.



(a) DeiT-B (b) Swin-S

Fig. 8: Eigenvector prune ratio based on Marchenko-Pastur denoising.

4.3 Additional Details and Experiments

We provide the complete algorithmic details of our pipeline and the mathematical proof for OB^{2}C in the supplementary material. The supplement also contains comprehensive results for the calibration set experiments, an analysis of both layerwise prune ratio and eigenvector layerwise pruning ratios, and additional evaluations demonstrating the scalability and applicability of our approach.

5 Conclusion

In this paper, we introduce DVBP + OB^{2}C, a novel structured pruning methodology that integrates statistical denoising techniques—specifically, Marchenko-Pastur (MP) distribution fitting within the VBP framework—with a highly efficient compensation mechanism. Our OB^{2}C framework advances the standard Optimal Brain Compression (OBC) approach by incorporating a multi-weight update strategy for pruned weights, further refined by mean-shift compensation. Extensive evaluations demonstrate that our method achieves substantial improvements in zero-shot, post-pruning accuracy without the need for subsequent fine-tuning or retraining. Because this approach is fundamentally applicable to any linear layer, it generalizes seamlessly across a wide range of modern deep neural network architectures. We hope this work encourages further exploration into structured pruning within the computer vision community, paving the way for near-lossless compression combined with significant inference acceleration.

Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST)). This work was supported by the InnoCORE program of the Ministry of Science and ICT(N10250156). This work was supported by the National Supercomputing Center with supercomputing resources including technical support (KSC-2025-CRE-0482).

References

1. Ashkboos, S., Mohtashami, A., Croci, M.L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., Hensman, J.: QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs. arXiv preprint arXiv:2404.00429 (2024)
2. Berisha, U., Mehnert, J., Condurache, A.P.: Variance-Based Pruning for Accelerating and Compressing Trained Networks. In: ICCV (2025)
3. Chan, T.F., Golub, G.H., LeVeque, R.J.: Updating Formulae and a Pairwise Algorithm for Computing Sample Variances. In: COMPSTAT (1982)
4. Deng, J., Socher, R., Fei-Fei, L., Dong, W., Li, K., Li, L.J.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021)
6. Emani, M., Foreman, S., Sastry, V., Xie, Z., Raskar, S., Arnold, W., Thakur, R., Vishwanath, V., Papka, M.E.: A Comprehensive Performance Study of Large Language Models on Novel AI Accelerators. arXiv preprint arXiv:2310.04607 (2023)
7. Frankle, J., Carlin, M.: The Lottery Ticket Hypothesis: Finding Sparse, Trainable Networks. In: ICLR (2019)
8. Frantar, E., Alistarh, D.: Optimal Brain Compression: A Framework for Accurate Post-Training Quantization and Pruning. In: NeurIPS (2022)
9. Frantar, E., Alistarh, D.: SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot. In: ICML (2023)
10. Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D.: GPTQ: Accurate Quantization for Generative Pre-trained Transformers. In: ICLR (2023)
11. Gale, T., Elsen, E., Hooker, S.: The State of Sparsity in Deep Neural Networks. arXiv preprint arXiv:1902.09574 (2019)
12. Gavish, M., Donoho, D.L.: The Optimal Hard Threshold for Singular Values is $4/\sqrt{3}$. IEEE TIT (2014)
13. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both Weights and Connections for Efficient Neural Network. In: NeurIPS (2015)
14. Hassibi, B., Stork, D.: Second order derivatives for network pruning: Optimal Brain Surgeon. In: NeurIPS (1992)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
16. Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., Peste, A.: Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. JMLR (2021)
17. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for MobileNetV3. In: ICCV (2019)
18. LeCun, Y., Denker, J.S., Solla, S.A.: Optimal Brain Damage. In: NeurIPS (1989)
19. Lee, K., Jang, H., Lee, D., Alistarh, D., Lee, N.: The Unseen Frontier: Pushing the Limits of LLM Sparsity with Surrogate-Free ADMM. arXiv preprint arXiv:2510.01650 (2025)
20. Lee, N., Ajanthan, T., Torr, P.H.S.: SNIP: Single-shot Network Pruning based on Connection Sensitivity. In: ICLR (2019)
21. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning Filters for Efficient ConvNets. In: ICLR (2017)

22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In: ICCV (2021)
23. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: CVPR (2022)
24. Luo, J.H., Wu, J.: An Entropy-based Pruning Method for CNN Compression. arXiv preprint arXiv:1706.05791 (2017)
25. Marchenko, V.A., Pastur, L.A.: Distribution of eigenvalues for some sets of random matrices. Mathematics of the USSR-Sbornik (1967)
26. Pébay, P.: Formulas for Robust, One-Pass Parallel Computation of Covariances and Arbitrary-Order Statistical Moments. Tech. rep., Sandia National Laboratories (2008)
27. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
28. Yang, H., Yin, H., Shen, M., Molchanov, P., Li, H., Kautz, J.: Global Vision Transformer Pruning with Hessian-Aware Saliency. In: CVPR (2023)
29. Yu, L., Xiang, W.: X-Pruner: eXplainable Pruning for Vision Transformers. In: CVPR (2023)
30. Zhu, J., Wang, H., Su, M., Wang, Z., Wang, H.: OBS-Diff: Accurate Pruning for Diffusion Models in One-Shot. In: ICLR (2026)