

Characterizing Learning in Deep Neural Networks using Tractable Algorithmic Complexity Analysis



Pedram Bakhtiarifard¹ · Sophia N. Wilson¹ · Mahmoud Affi² · Jonathan Wenshøj¹ · Raghavendra Selvan¹

¹Department of Computer Science, University of Copenhagen · ²The American University in Cairo, Egypt

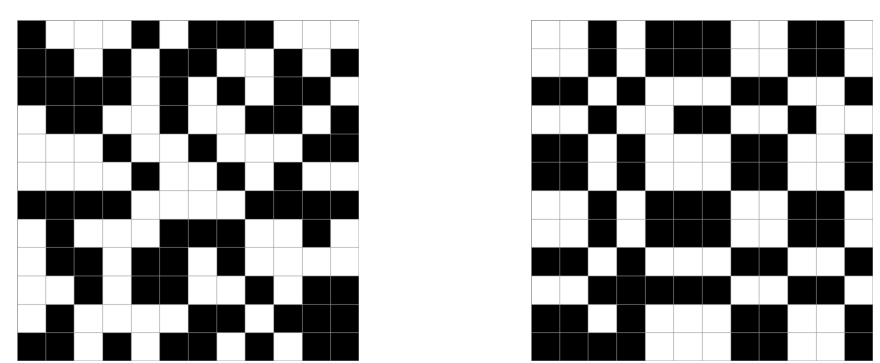


Complexity of Binary Objects

Which of the two sequences is most complex?

A = [101110010110] → print "101110010110"
B = [100100100100] → [print "100"] × 4

How about the two matrices?



Algorithmic Complexity

Algorithmic Kolomogorov–Chaitin–Solomonoff (KCS) complexity of a finite object $\mathbf{w} \in \{0, 1\}^*$ is the length of the shortest program p among all possible programs that outputs $\mathbf{w}$ and halts:

$$\mathcal{C}(\mathbf{w}) := \min\{|p| : U(p) = \mathbf{w}\}$$

Here U is a fixed universal prefix-free Turing machine and $|p|$ denotes the length of the program in bits. The KCS complexity is uncomputable by definition.

Coding Theorem Method (CTM)

The algorithmic probability

$$P(\mathbf{w}) = \sum_{p: U(p)=\mathbf{w}} 2^{-|p|}$$

relates to $\mathcal{C}$ via the coding theorem

$$\mathcal{C}(\mathbf{w}) = -\log_2 P(\mathbf{w}) + O(1)$$

CTM estimates $P(\mathbf{w})$ by running all n -state Turing machines and taking the fraction D_n that halt with output $\mathbf{w}$:

$$\mathcal{C}_{\text{CTM}}(\mathbf{w}) = -\log_2 D_n(\mathbf{w}) \approx \mathcal{C}(\mathbf{w})$$

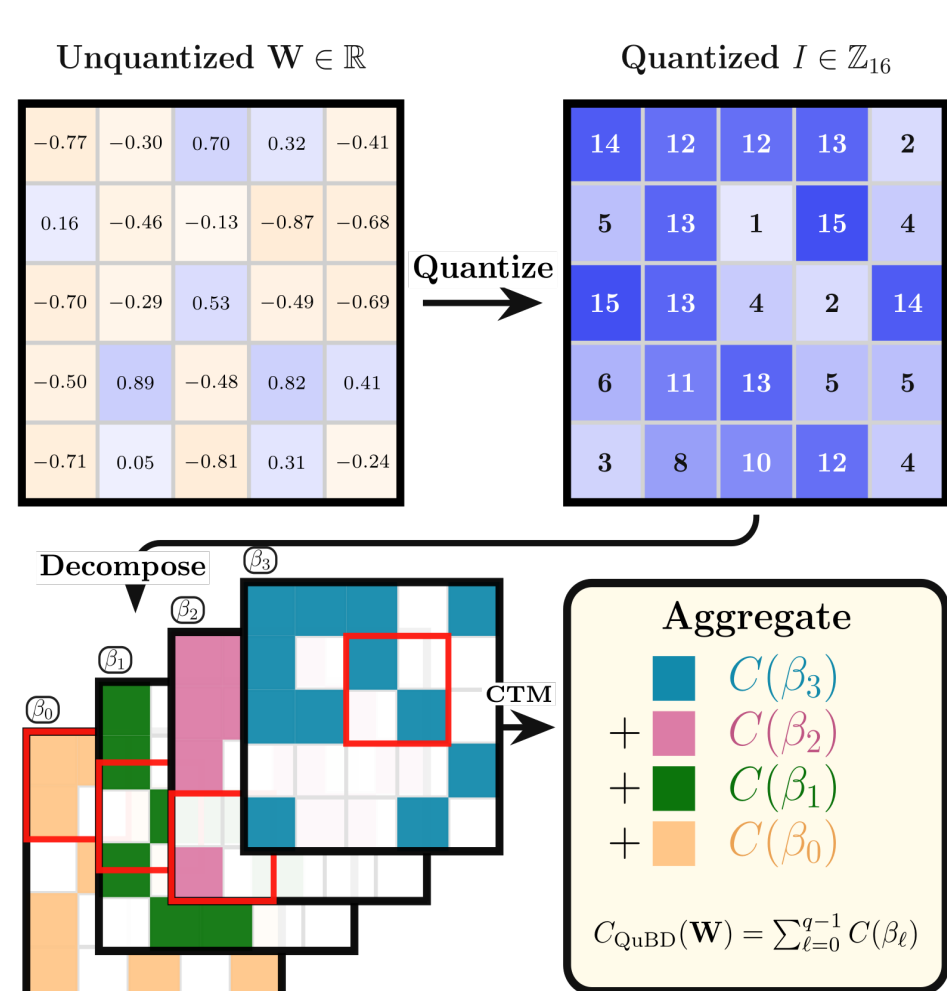
Block Decomposition Method (BDM)

CTM does not scale to large objects. BDM extends it by splitting $\mathbf{w}$ into smaller blocks $\Pi(\mathbf{w}) = (w_1, \dots, w_m)$.

Let $\Pi_u(\mathbf{w})$ be the set of distinct blocks and n_w the number of occurrences of $w \in \Pi(\mathbf{w})$. Then

$$\mathcal{C}_{\text{BDM}}(\mathbf{w}) = \sum_{w \in \Pi_u(\mathbf{w})} (\mathcal{C}_{\text{CTM}}(w) + \log_2 n_w)$$

Illustration of QuBD Method



QuBD first quantizes weights into finite codes (top), then decomposes them into aligned bit-planes to aggregate the per-plane QuBD complexities (bottom).

Quantized Block Decomposition (QuBD) Method

Let $\mathbf{w} = (w_1, \dots, w_d) \in \mathbb{R}^d$ denote a d -dimensional real object. We map $\mathbf{w}$ to a finite alphabet Σ_b using an affine uniform quantizer $Q_b: \mathbb{R}^d \rightarrow \Sigma_b^d$ yielding the discrete object

$$\mathbf{q} = Q_b(\mathbf{w}) = (q_1, \dots, q_d) \in \Sigma_b^d, \quad \text{with } \Sigma_b := \{0, \dots, 2^b - 1\}$$

where each q_i is a b -bit symbol of 2^b possible elements. The bit-plane representation uses the binary expansion of each quantized weight

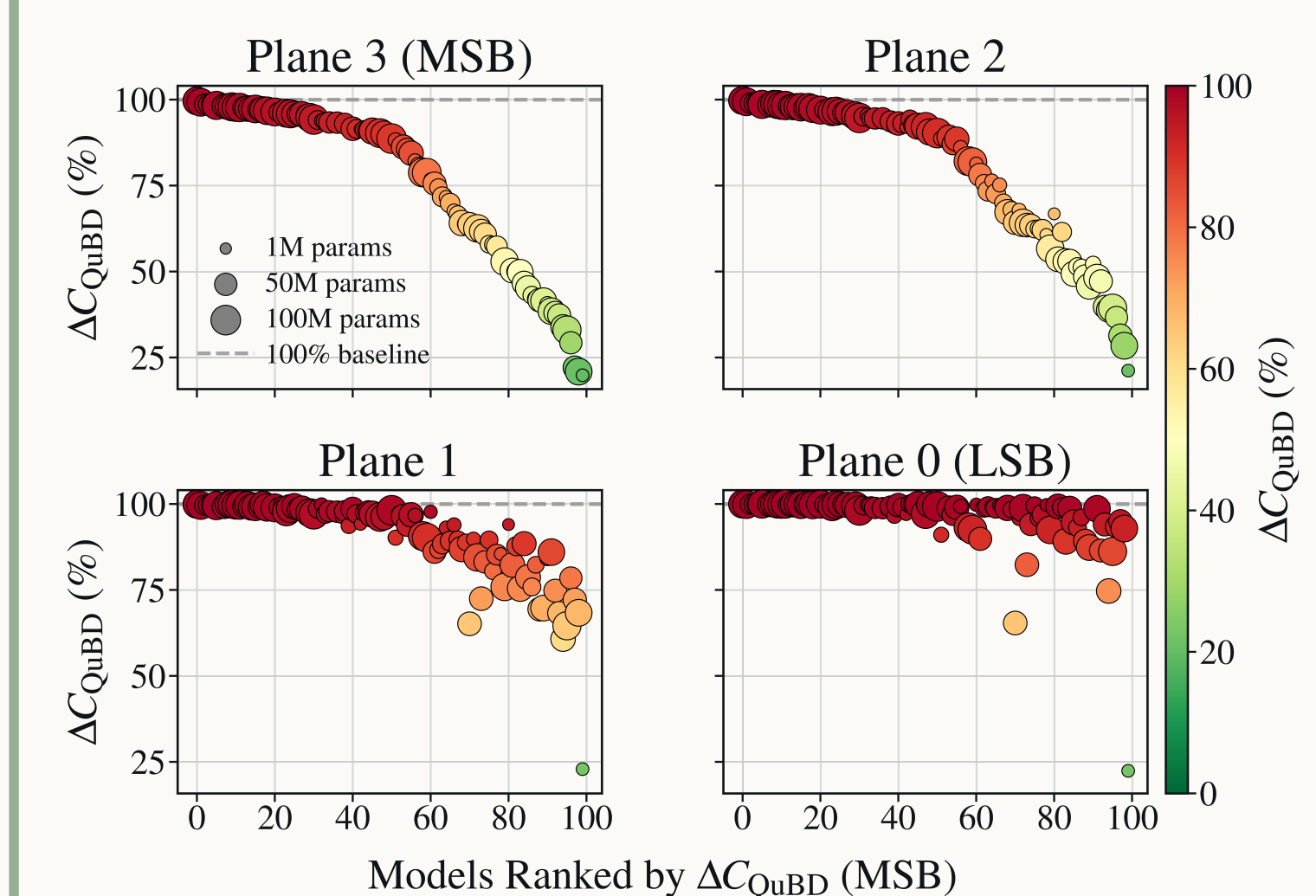
$$q_i = \sum_{\ell=0}^{b-1} 2^\ell \beta_{i,\ell}, \quad \beta_{i,\ell} \in \{0, 1\}, \quad \beta_\ell := (\beta_{1,\ell}, \dots, \beta_{d,\ell}) \in \{0, 1\}^d$$

where $\beta_{i,b-1}$ denotes the most significant bit (MSB) and $\beta_{i,0}$ denotes the least significant bit (LSB) of q_i . The vector β_ℓ collects the ℓ -th bit of each of the d symbols, forming the ℓ -th bit-plane.

The QuBD complexity and the normalized QuBD complexity are then given by

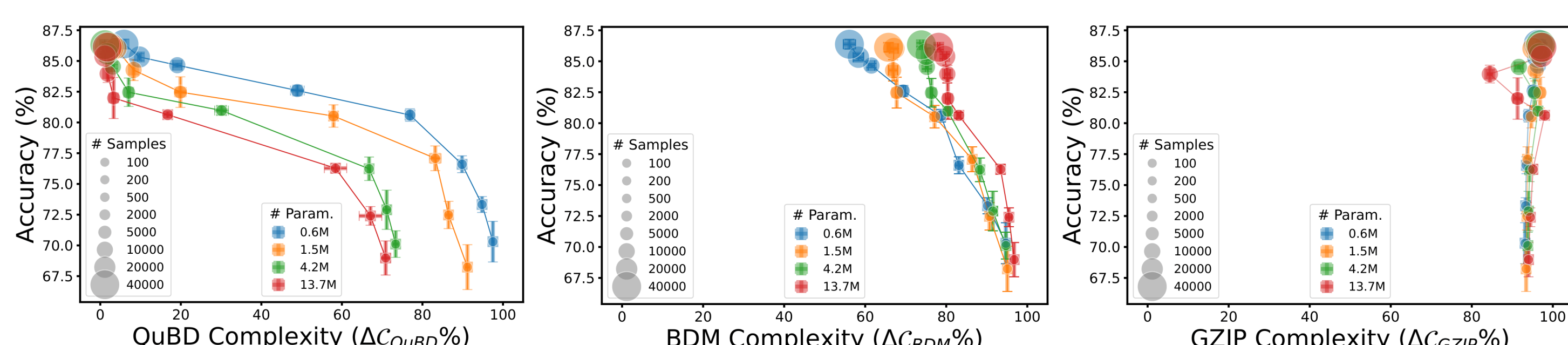
$$\mathcal{C}_{\text{QuBD}} = \sum_{\ell=0}^{b-1} \mathcal{C}_{\text{BDM}}(\beta_\ell) \quad \text{and} \quad \Delta \mathcal{C}_{\text{QuBD}} = \frac{\mathcal{C}_{\text{QuBD}}(f_{\text{PT}})}{\mathcal{C}_{\text{QuBD}}(f_{\text{RND}})}.$$

Complexity across Bit-planes



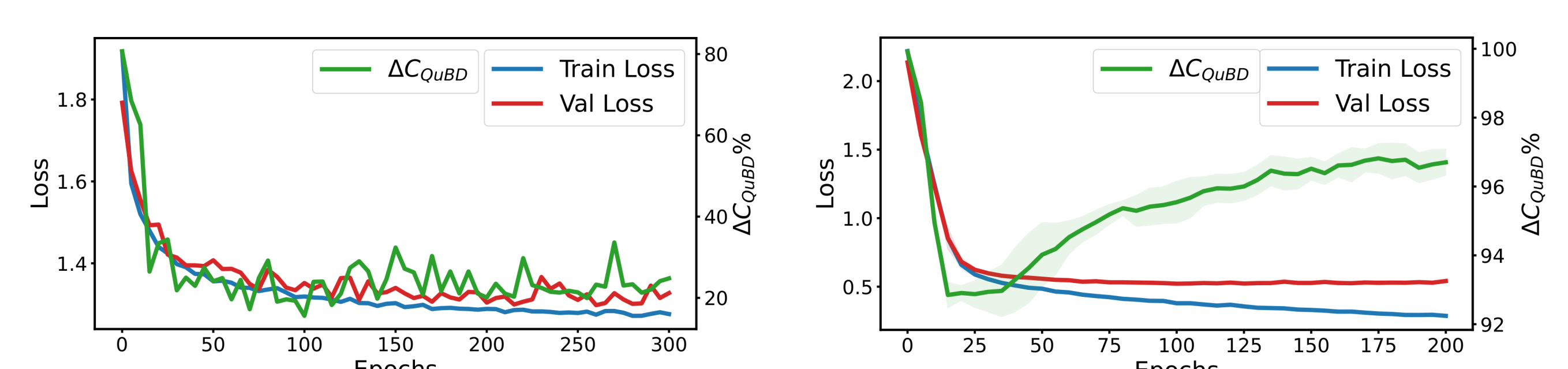
Normalized QuBD complexity for 100 pre-trained models with up to 100M parameters with four bit-planes.

Learning Reduces Algorithmic Complexity



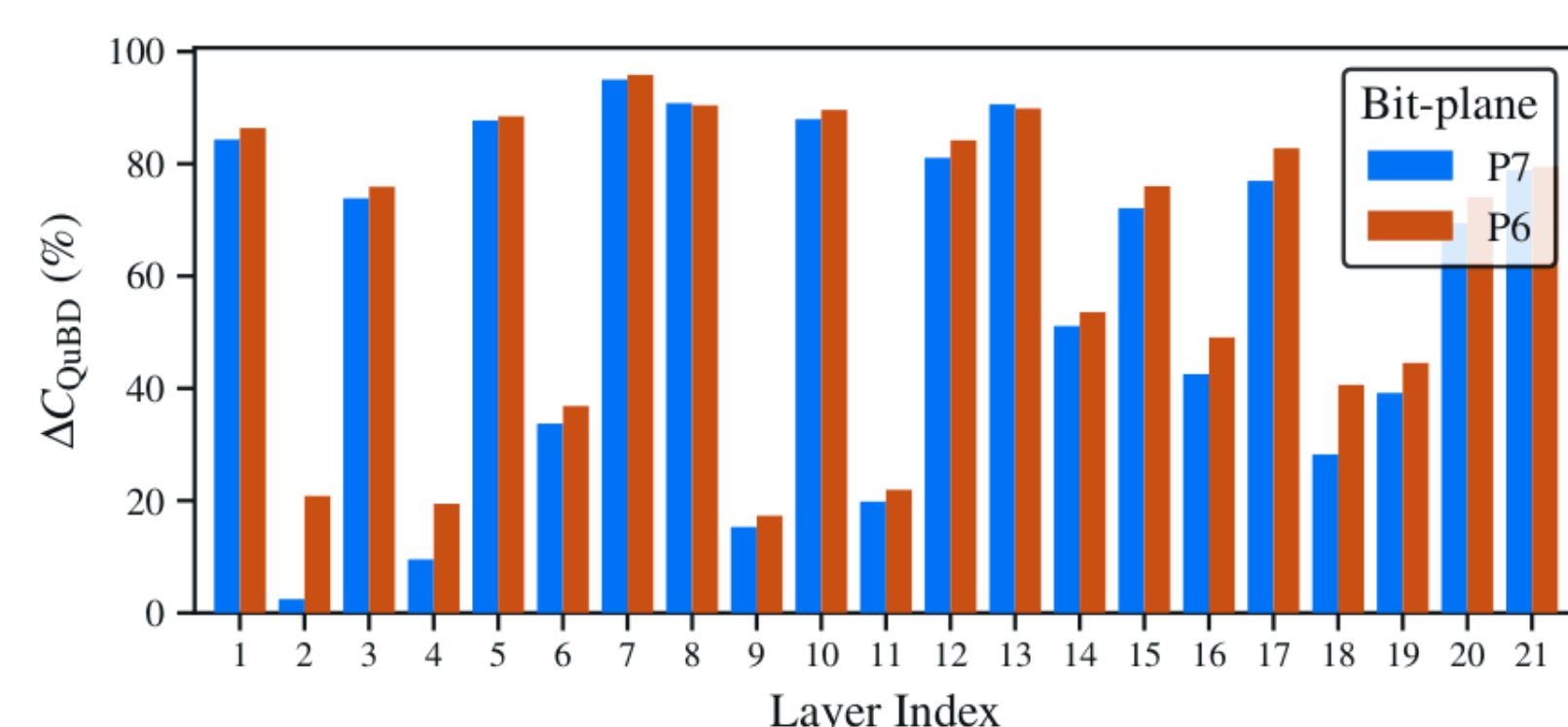
Accuracy vs. normalized complexities using QuBD, BDM, and GZIP. Four MLPs (0.6M–13.7M parameters) are trained with an increasing number of training data samples (Fashion–MNIST). QuBD complexity tracks the connection between learning and complexity best.

Algorithmic Complexity Tracks Learning and Overfitting



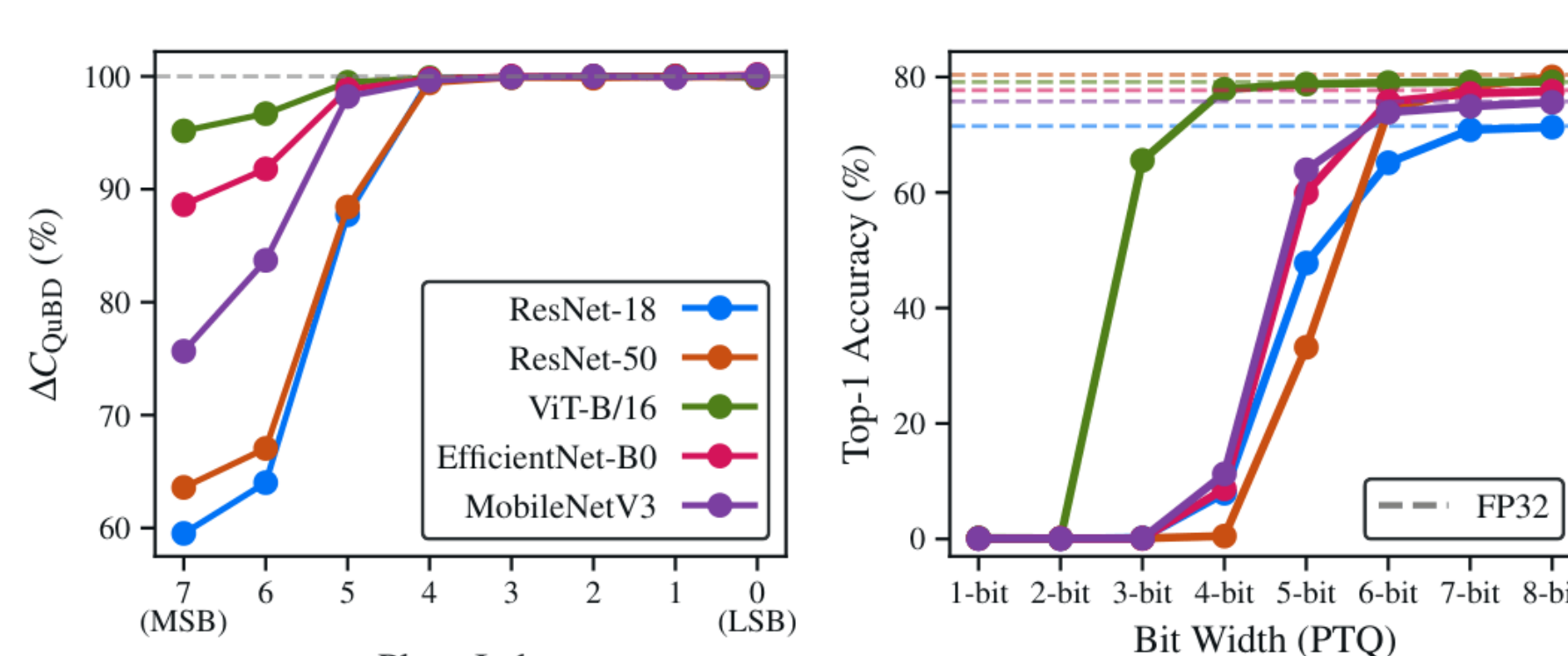
Learning curves along with the evolution of the normalized QuBD complexity for ViT (left) and MLP (right) (CIFAR–10). In both cases, QuBD complexity tracks the generalization performance and for MLP it even tracks overfitting.

Layers Exhibit Different Complexity Reductions



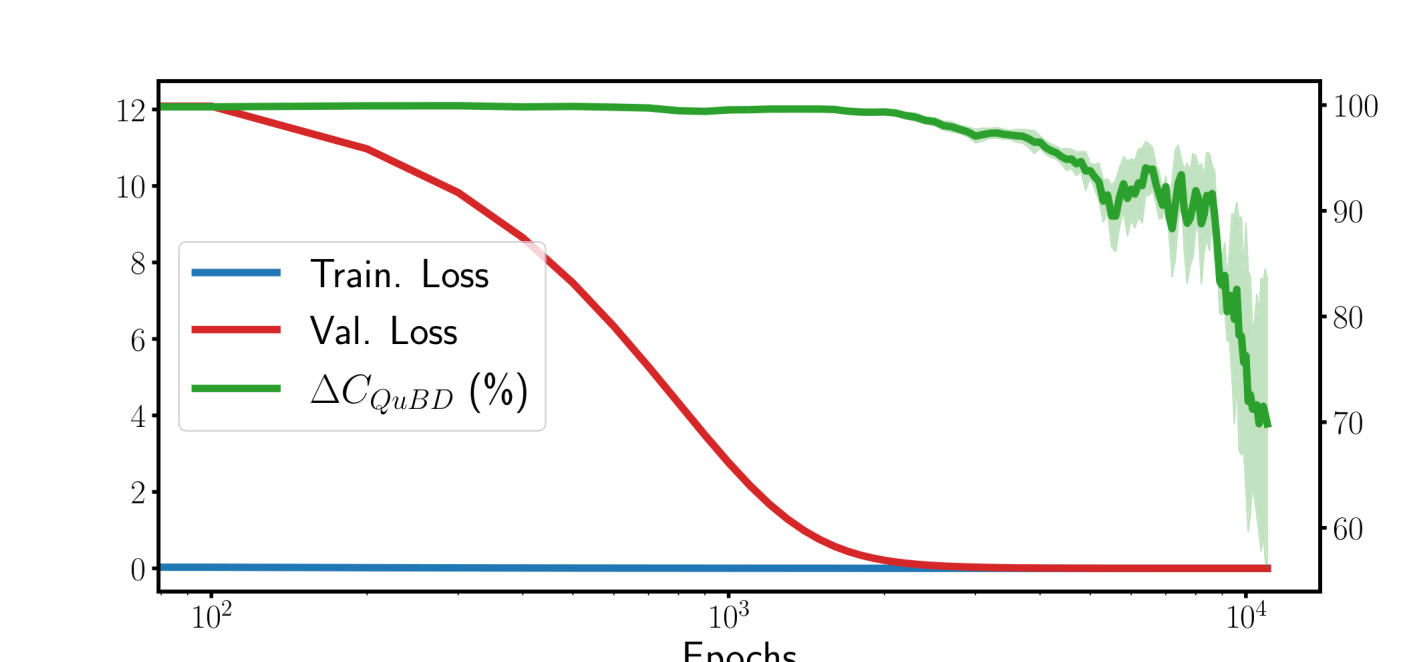
Normalized QuBD complexity per layer for ResNet–18 (ImageNet–1k), showing variation across layers.

QuBD Complexity as a Diagnostic for Compression



Normalized QuBD complexity per bit-plane (left) and top–1 PTQ accuracy (ImageNet–1k).

Algorithmic Complexity during Grokking



Learning curves and the evolution of the normalized QuBD complexity for MLP (P2 Cartesian Product dataset). QuBD complexity reduces after the validation loss has reached zero.

References

- Kolmogorov. Three approaches to the quantitative definition of information. Problems of information transmission, 1965.
- Chaitin. On the length of programs for computing finite binary sequences. Journal of the ACM, 1966.
- Solomonoff. A preliminary report on a general theory of inductive inference. Zator Company Cambridge, 1960.
- E. Y. Sakabe et al. Binarized Neural Networks Converge Toward Algorithmic Simplicity: Empirical Support for the Learning-as–Compression Hypothesis. arXiv, 2025.
- P. Bakhtiarifard, T. Chen, J. Wenshøj, E. B. Dam, and R. Selvan. Algorithmic Simplification of Neural Networks with Mosaic–of–Motifs. arXiv, 2026.

QR Codes for Paper and Code

