Extreme Model Compression with Structured Sparsity at Low Precision

Dan Liu[1,2]
daniel.liu@mail.mcgill.ca
ita Dvornik[2]
rnik.nikita@gmail.com
e Liu[1,2,3]
liu@cs.mcgill.ca

[1] McGill University
Montreal, Canada
[2] Palona AI,
Montreal, Canada
[3] MBZUAI,
Abu Dhabi, UAE

Abstract

Deep neural networks (DNNs) are used in many applications, but their large size and high computational cost make them hard to run on devices with limited resources. Two widely used techniques to address this challenge are weight quantization, which lowers the precision of all weights, and structured sparsity, which removes unimportant weights while retaining the important ones at full precision. Although both are effective individually, they are typically studied in isolation due to their compounded negative impact on model accuracy when combined. In this work, we introduce SLOPE (Structured Sparsity at Low Precision), a unified framework, to effectively combine structured sparsity and low-bit quantization in a principled way. We show that naïvely combining sparsity and quantization severely harms performance due to the compounded impact of both techniques. To address this, we propose a training-time regularization strategy that minimizes the discrepancy between full-precision weights and their sparse, quantized counterparts by promoting angular alignment rather than direct matching. On ResNet-18, SLOPE achieves $\sim 20\times$ model size reduction while retaining $\sim$99% of the original accuracy. It consistently outperforms state-of-the-art quantization and structured sparsity methods across classification, detection, and segmentation tasks on models such as ResNet-18, ViT-Small, and Mask R-CNN.

Introduction

p neural networks (DNNs) are increasingly deployed across a wide range of applications, their growing size and computational demands present major obstacles for deployment on resource-constrained hardware. Quantization has emerged as one of the most effective strategies to address these limitations, reducing the bit-width of weights and activations to lower memory usage and accelerate inference [8]. By mapping high-precision (e.g., 32-bit floating-point) weights of a trained model to lower-precision representations (e.g., 8-bit or 4-bit), quantization significantly reduces the storage and computational cost of neural networks. This process often involves a delicate trade-off: while lower precision leads to more compact models and faster execution, it also degrades model accuracy due to reduced

representational capacity. In practice, 4-bit quantization can provide up to 8× memory reduction compared to 32-bit weights while maintaining competitive performance for many tasks, making it an attractive near-lossless compression method [8]. However, in many deployment scenarios, particularly those involving large models or strict memory and latency constraints, this level of compression may still fall short [37]. Pushing the precision even further down may offer additional savings, but at what cost? Quantizing weights below 4 bits often leads to sharp drops in accuracy, making such extreme quantization levels impractical for most real-world applications[6, 8, 24, 28]. This raises a natural question: can we go beyond quantization to achieve higher compression without incurring severe accuracy loss?

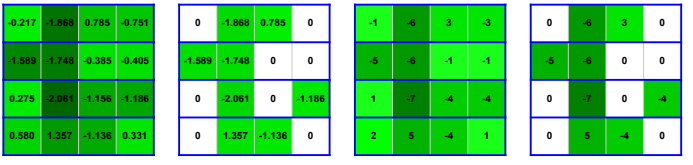

Figure 1: Weight matrix representations under different compression settings. (a) Dense, full-precision weights offer high accuracy but are computationally expensive. (b) Structured 2:4 sparsity (every 4 elements contain 2 non-zeros) in full precision reduces the number of multiplications. (c) Quantization (4-bit) compresses memory usage. (d) Structured 2:4 quantization enables much higher inference speedups and compression ratios (See the Appendix).

Another effective approach for model compression is structured N:M sparsity, which enforces that exactly M out of every N consecutive weights are zero (e.g., 2:4 sparsity ensures that in every two out of four consecutive weight elements are non-zero) [1, 2]. Beyond reducing the model's memory footprint, structured sparsity also enables significant computational speedups on modern hardware due to its compatibility with optimized sparse matrix operations [7, 27]. However, setting the sparsity ratio too high (e.g., 2:16) removes too much information from the original weights and leads to degraded model accuracy, thus limiting the extent of effective compression.

In this work, we push the limits of model compression and propose to combine structured sparsity with low-bit quantization in a unified framework (see Fig 1-d). These two techniques reduce model size through fundamentally different mechanisms, sparsity removes weights [27], while quantization reduces their precision, and are therefore expected to be complementary [16]. Yet, in our experiments, we find that applying both simultaneously often leads to a significant drop in accuracy, due to their compounded impact on model capacity. While finetuning the resulting sparse and quantized model can partially recover performance, a noticeable gap remains compared to the original full-precision model [9]. To address this challenge, we introduce SLOPE (Structured Sparsity at Low Precision), a novel training framework that integrates N:M structured sparsity with low-bit quantization in a principled way. At the core of SLOPE is a regularization strategy that minimizes the discrepancy between the original full-precision weights and their sparse, quantized counterparts. In particular, our regularizer promotes directional alignment between the original and sparse quantized weights, serving as a good prior for high-performance models while allowing for flexibility in sparse quantized weight optimization. By aligning the original weight vectors with the compressed ones, SLOPE enables the model to retain higher accuracy while enjoying the memory and computational benefits of extreme compression.

We validate SLOPE across a range datasets such as ImageNet [5] and MS-COCO [22], with ResNet18 [12], DeiT [31] and Mask-RCNN [13] models. SLOPE consistently outper-

forms state-of-the-art sparse quantization methods under 2:4 structured sparsity. On ResNet-18, it boosts the accuracy of 4-bit sparse model from 68.36% to 71.11%, exceeding the full-precision baseline. On DeiT-small, it achieves 80.7% Top-1 accuracy, surpassing all existing 2:4 sparse methods. For object detection, SLOPE improves Mask R-CNN box mAP from 37.80 to 40.83 under 4-bit 2:4 sparsity, narrowing the gap to the full-precision baseline (41.0 mAP). These results confirm that SLOPE enables highly compressed models without compromising performance. To this end, in this work we make the following contributions:

- We introduce SLOPE, a unified framework that combines structured N:M sparsity and low-bit quantization for extreme model compression.
- We propose a novel directional regularization strategy that stabilizes training under aggressive compression by aligning compressed and full-precision weight vectors.
- We demonstrate that SLOPE outperforms state-of-the-art quantization and structured sparsity baselines across different models and computer vision tasks.

2 Related Work

2.1 Quantization

Quantization reduces the precision of model parameters and activations, typically from 32-bit floating point to lower-bit formats such as 4-bit or 2-bit, to decrease memory usage and accelerate inference [6, 8, 15, 30, 36]. A central goal is to ensure that the quantized weights $\widehat{\mathbf{W}}$ closely approximate the full-precision weights $\mathbf{W}$, thereby preserving model performance.

Many works aim to minimize this discrepancy. For instance, [19, 24, 28] directly minimize the L_2 distance between $\widehat{\mathbf{W}}$ and $\mathbf{W}$. Others [10, 17, 23] promote quantization-aware training by encouraging $\mathbf{W}$ to lie near quantization bin centers. Lin et al. [21] further enhance quantization by learning a set of rotation matrices that iteratively align $\mathbf{W}$ and $\widehat{\mathbf{W}}$, yielding significant improvements in binary settings. More quantization works are discussed in the work of Gholami et al. [8].

2.2 Structured N:M Sparsity

Structured N:M sparsity is a specific and increasingly popular form of structured sparsity (i.e., the type of sparsity where the weights are sparsified according to a specific pattern). It enforces a fine-grained constraint in which only N weights are retained out of every M consecutive elements. This approach strikes a balance between the flexibility of unstructured sparsity and the hardware efficiency of coarse-grained patterns. Notably, NVIDIA GPUs support 2:4 sparsity at inference time, enabling practical acceleration [2]. With careful fine-tuning, such structured sparsity can retain performance comparable to dense models.

Several techniques have been proposed to train N:M sparse models effectively. Zhou et al. [1] introduce SR-STE, a straight-through estimator adapted for N:M constraints, enabling sparse models to be trained from scratch with minimal accuracy loss and achieving up to 2× speedup on NVIDIA A100 GPUs. LBC [34] addresses the combinatorial nature of N:M sparsity via a divide-and-conquer approach that assigns learnable scores to weight subsets, enabling efficient mask learning. STEP [26] proposes an Adam-aware method for mask learning, consisting of a preconditioning phase for estimating reliable gradient variances followed by a sparsity-inducing optimization phase.

Beyond weights, Chmiel et al. [4] apply N:M sparsity to gradients during training, introducing a minimum-variance unbiased estimator (MVUE) that supports 1:2 or 2:4 sparse gradients, reducing computation without harming convergence. Finally, S-STE [14] proposes a continuous pruning framework for pretraining sparse models. It uses a projection-based pruning function and fixed rescaling of sparse weights to produce efficient 2:4 sparse models that closely match the performance of their dense counterparts.

Recent efforts have explored sparse quantization, which jointly applies sparsity and quantization to maximize model compression. Harma et al. [11] highlight the importance of aligning sparse structures with quantization to mitigate compounded degradation and propose a unified framework that jointly optimizes both constraints to maintain accuracy. Similarly, Guo et al. [9] identify key challenges in sparse quantization, such as the tendency of sparsification to preserve outliers that complicate downstream quantization. In contrast, SLOPE focuses on aligning the full-precision weights with their sparse quantized counterparts, yielding complementary improvements.

3 Preliminaries

In this section we introduce formal definitions of weight quantization, structured N:M sparsity, and cover common ways to measure discrepancy between original and compressed weights of a neural network. In this work, a linear layer is defined as: $\mathbf{y} = \mathbf{W}^\top \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^{m\times 1}$ denotes the input vector and $\mathbf{W} \in \mathbb{R}^{m\times n}$ denotes the weight matrix with $i = 1, ..., n$. $\mathbf{w}_i \in \mathbf{W}$ denotes the i-th vector of $\mathbf{W}$. $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is the layer output.

Quantization In this paper, by quantization we mean rounding float values to their nearest lower-precision counterpart. This corresponds to the LSQ [6] formalism:

$$\widehat{\mathbf{W}} = q(\mathbf{W}; s, b) = s\Big\lfloor \text{clamp}\Big(\Big\lfloor \frac{\mathbf{W}}{s} \Big\rfloor; -\text{Q}_N, \text{Q}_P\Big)\Big\rceil, \tag{1}$$

where $\mathbf{W}$ and $\widehat{\mathbf{W}}$ denote the full-precision and quantized weights, respectively. s is a learnable parameter for scaling weights, b represents the bit-width, $\lfloor\rceil$ is the rounding operator. For activation quantization, the quantization range is typically defined as $Q_N = 0, \quad Q_P = 2^{b-1}$, which corresponds to an asymmetric unsigned mapping suitable for non-negative activations (e.g., ReLU). In contrast, for weight quantization, the range is defined as $Q_N = -2^{b-1}, \quad Q_P = 2^{b-1} - 1$, which corresponds to a symmetric signed mapping centered around zero, reflecting the approximately zero-mean distribution of weights.

Structured N:M Sparsity Structured N:M sparsity keeps N non-zero and prunes M-N weights to zero in every M consecutive elements (e.g., N = 2, M = 4; Fig. 1b). Let $\mathsf{w}_{N:M} \subset \mathbf{W}$ be a block of M consecutive elements in $\mathbf{W}$, and let $\widetilde{\mathsf{w}}_{N:M} \subset \widetilde{\mathbf{W}}$ be the corresponding block in $\widetilde{\mathbf{W}}$. Then each element $w_i \in \mathsf{w}_{N:M}$ is mapped to:

$$\widetilde{w}_i = S(w_i; N, M) = \begin{cases} w_i, & \text{if } |w_i| \geq \xi, \\ 0, & \text{if } |w_i| < \xi, \end{cases} \quad \text{for } i = 1, 2, \ldots, M \tag{2}$$

where ξ is the N-th largest absolute value in the set $\{|w_1|, |w_2|, \ldots, |w_M|\}$. Essentially, the N:M sparsification operation zeroes out $(M-N)$ smallest elements in each M-element group, while keeping the remaining N values intact.

Weight Deviation Measures There are multiple ways to measure the deviation of the compressed weights with respect to the original ones. Some common measures include L_2, L_1 and sometimes cosine distance, which are typically used to to get the overall magnitude (or direction) of change. Another application-specific measure is Signal-to-Quantization-Noise Ratio (SQNR). It captures how much $\widehat{\mathbf{W}}$ deviates from $\mathbf{W}$ in a way that matters most in quantization. Formal definition of QSNR is as follows:

$$\text{SQNR} = 10\log_{10}\Big(\frac{\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{w}_i\|^2}{\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{w}_i-\hat{\mathbf{w}}_i\|^2}\Big). \tag{3}$$

It measures the ratio between the maximum nominal signal strength and the quantization error. Higher SQNR means lower quantization error [18, 20, 29].

4 Our Approach

In this section we introduce SLOPE, our approach for combining structured sparsity with quantization. We elaborate on the design of our regularization function and detail the training and inference procedure of SLOPE. Yet, to motivate the need for our approach, we start with a simple experiment where quantization and sparsity are combined naively.

Metric	Full-precision		4-bit Quant		2-bit Quant	
	w/o Sparse	+ 2:4 Sparse	w/o Sparse	+ 2:4 Sparse	w/o Sparse	+ 2:4 Sparse
Test Acc. (%) ↑	69.76	**69.96**	71.10	68.36	67.60	64.47
Cosine ↑	NA	**0.975 ± 0.01**	0.948 ± 0.01	0.919 ± 0.01	0.878 ± 0.03	0.831 ± 0.03
SQNR (dB) ↑	NA	**14.74 ± 0.98**	10.51 ± 1.01	8.43 ± 1.04	7.54 ± 1.08	4.84 ± 1.13

Table 1: Impact of combining structured 2:4 sparsity and quantization on top-1 ImageNet accuracy of ResNet-18. Cosine and SQNR depict weight deviations from the full-precision weights.

4.1 Our Motivation: Weight Deviation in Sparse Quantization

Structured sparsity and weight quantization achieve compression through fundamentally different mechanisms: quantization reduces the numerical precision of weights, while sparsity retains only a subset of weights. A natural idea is to combine these techniques: first applying sparsity, then quantizing the non-zero weights, followed by light fine-tuning to adapt the surviving values while preserving the sparsity pattern. This approach, known as *sparse quantization* [27], intuitively leverages the strengths of both methods. However, when applied to models such as ResNet-18, we observe a significant drop in accuracy compared to using sparsity or quantization alone, particularly at lower bit-widths (see Table 1). We hypothesize that this degradation arises from the compounded distortion induced by both compression techniques. This is supported by angular deviation and SQNR analyses, which show markedly larger weight discrepancies when sparsity and quantization are combined. These findings highlight the importance of controlling both magnitude and directional deviations of weight vectors to preserve accuracy in sparse low-bit quantization, and motivate the need for a principled solution.

4.2 SLOPE: Structured Sparsity at Low Precision

To address the performance drop from combining structured sparsity and quantization, we add a regularizer during fine-tuning that explicitly encourages the full-precision weights $\mathbf{W}$

to stay close to the sparse quantized weights $\widehat{\mathbf{W}}$. The optimization process is formulated as:

$$\min_{\mathbf{W}} J(\mathbf{W}) = L(\widehat{\mathbf{W}}) + \lambda L_{reg}(\mathbf{W}, \widehat{\mathbf{W}}), \tag{4}$$

where $\widehat{\mathbf{W}} = q\big(S(\mathbf{W};N,M);s,b\big)$, $L()$ denotes a standard objective function, and λ is the regularizer strengths empirically set based on the loss value of $L()$ to make sure L_{reg} and L are at the same scale. The L_{reg} is defined by:

$$L_{reg}(\mathbf{W}, \widehat{\mathbf{W}}) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left(1 - \cos(\mathbf{w}_i, \hat{\mathbf{w}}_i)\right), \tag{5}$$

where $\cos()$ denotes calculating the cosine similarity. Minimizing L_{reg} reduces the cosine distance (angular deviation) between $\mathbf{w}_j$ and $\hat{\mathbf{w}}_j$. The closer the distance between $\mathbf{w}_j$ and $\hat{\mathbf{w}}_j$ is, the smaller the deviation θ is during the sparse quantization.

An ideal regularizer L_{reg} should, in principle, reduce the discrepancy between $\mathbf{W}$ and $\widehat{\mathbf{W}}$ to zero given sufficient weight precision, making L_1 or L_2 penalties appealing choices [10]. However, in the presence of structured sparse quantization, perfect alignment is no longer achievable. As shown in our results, the imposed low-precision sparse pattern significantly alters the weight direction, and standard L_1/L_2 penalties fail to effectively constrain angular deviation. As demonstrated in our ablation results (Table 7), minimizing $L_2(\mathbf{W}, \widehat{\mathbf{W}})$ becomes ineffective under low-precision sparsity.

To this end, we propose SLOPE, which relaxes the regularization objective and instead optimize an upper bound that does not impose per-element constraints on the weight matrix. In the structured sparsity setting, we can show that the L_2 distance is upper bounded by:

$$\|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 \leq \underbrace{2\|\mathbf{w}\|_2^2(1 - \cos\theta)}_{\text{Upper bound}}. \tag{6}$$

While the upper bound scales with the weight norm, its minimization is driven by the angular term only. This regularizer promotes directional alignment without forcing sub-optimal individual weight values. SLOPE leverages this property to recover most of the lost accuracy under compression, making angular regularization an effective strategy for sparse quantization. (See Appendix for derivation and geometric interpretation.)

SLOPE Implementation: Going Forward and Backwards Here we detail how the forward and backward passes are performed, in the presence of quantization and sparsification. During the forward pass, following Harma et al. [11], we first sparsify the full-precision weights and then quantize them, $\widehat{\mathbf{W}} = q\big(S(\mathbf{W};N,M);s,b\big)$.The sparse quantized weights are optimized with the straight-through estimator (STE) [1, 3, 6, 28, 33]:

$$\begin{cases} \widehat{\mathbf{W}}_t = q\big(S(\mathbf{W}_t;N,M);s,b\big), \\ \mathbf{W}_{t+1} = \mathbf{W}_t - \dfrac{\partial L(\widehat{\mathbf{W}}_t)}{\partial \widehat{\mathbf{W}}_t}, \end{cases} \tag{7}$$

where t represents the training iteration and $\dfrac{\partial L(\widehat{\mathbf{W}}_t)}{\partial \widehat{\mathbf{W}}_t}$ is the approximated gradient.

5 Experiments

In this section, we conduct experiments across a diverse set of models and tasks to demonstrate the generalization ability of our method. We evaluate ResNet-18 [12] and DeiT-small [31] on ImageNet [5] for classification, and Mask R-CNN [32] on MS-COCO [22]

for detection and segmentation. Our study covers both weight-only and weight-activation quantization, combined with 2:4 structured sparsity. We also explore extreme sparse configurations such as 2:8 and 2:16. Evaluation metrics include Top-1 accuracy on ImageNet and mean Average Precision (mAP) [32] on MS-COCO. Additional training details (e.g., overhead and training time) and sparse-only results are provided in the Appendix.

5.1 Image Classification

We evaluate SLOPE against SoTA compression methods that use quantization and structured sparsity (i.e., ASP [27], SR-STE [1], and MVUE [4]), or quantization alone (i.e., Quant [10]). We can see that quantization generally degrades performance when combined with structured sparsity. However, for ResNet18, SLOPE achieves 71.11% Top-1 accuracy under 4-bit quantization, exceeding the full-precision baseline (69.76%), while using memory close to 2-bit precision. While it may seem surprising, sparse models are generally known to outperform their dense counterparts thanks to the sparsity-induced regularization, yet, we are the first to show this result under quantization! For ResNet50, SLOPE also consistently outperforms other methods. It reaches 75.93% Top-1 accuracy in the 4-bit setting and 72.30% with 2-bit quantization, both of which are significantly better than other baselines. Notably, even under aggressive 2-bit compression, SLOPE maintains competitive accuracy compared to 4-bit results of prior work. This demonstrates SLOPE's strong generalization ability and robustness under extreme quantization.

Method	FP		4-bit		2-bit	
	ResNet-18	ResNet-50	ResNet-18	ResNet-50	ResNet-18	ResNet-50
Quant	69.76	76.13	71.10	76.70	67.60	73.70
ASP	69.90†	76.80†	68.36	74.70	64.47	71.20
SR-STE	71.20†	77.00†	69.23	-	63.71	-
MVUE	70.6	77.12	67.22	-	-	-
SLOPE	**71.23**†	**77.24**†	**71.11**	**75.93**	67.59	72.34

Table 2: Comparison of different structured 2:4 sparse quantization methods on ResNet models under various bit-widths. † denotes training-from-scratch. "Quant" denotes quantization only.

In Fig. 2, we plot SLOPE's performance in comparison to other baselines under different quantization bit-widths and compression levels. It is clear that SLOPE consistently outperforms (or performs on par) all sparsity- and/or quantization methods across 8-bit, 4-bit, and 2-bit settings, with particularly large gains at lower precisions. Fig. 2 shows top-1 accuracy and compression savings against FP32, where SLOPE achieves the best accuracy under aggressive compression (up to 93.75%).

DeiT-small For DeiT-small on ImageNet, we evaluate 2:4 structured sparsity under 2-bit and 4-bit quantization. Table 3 shows results with quantized activations (A16/A4/A2), denoted as A16/W4, A4/W4, and A2/W2. A16 denotes activation quantization with bfloat16 precision. SLOPE consistently outperforms the baseline, especially in low-bit settings. Under the challenging 2-,4- and 8-bit configurations, SLOPE achieves consistently outperforms the baseline, by up to 1.8%.

Notably, SLOPE shows clear advantages under similar memory budgets: sparse A16/W4 outperforms dense A16/W2 (80.41% vs. 77.34%), and sparse A8/W8 exceeds dense A4/W4 (80.59% vs. 78.15%) (Table 9). In addition, SLOPE's A4/W4 accuracy (78.15%) exceeds that of dense A2/W2 (75.72%). These results highlight SLOPE's superior balance between compression and accuracy.

Method	A16/W4	A16/W2	A8/W8	A4/W4	A2/W2
Dense	80.87	77.34	79.56	80.33	75.72
ASP	79.28	75.76	78.75	77.81	60.77
SLOPE	**80.41**	**76.81**	**80.59**	**78.15**	**61.32**

Table 3: Results of sparse 2:4 quantization on DeiT-small with ImageNet.

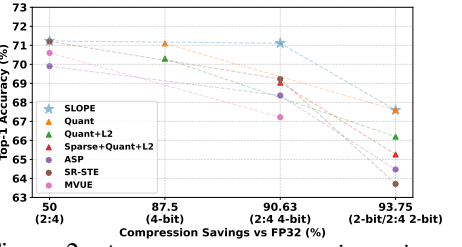

Figure 2: Accuracy vs. compression ratio on ResNet-18 models.

5.2 Object Detection and Segmentation

In this section we measure the effect of SLOPE's compression on object detection and segmentation, and compare it to other 2:4 sparse quantization methods on the COCO dataset. In Table 4, besides reporting the numbers achieved with finetuning, we also show the performance when training the models from scratch. In both settings, SLOPE achieves significantly better performance than the baseline [32], even under structured 2:4 sparsity. Specifically, under the A32/W4 setting, SLOPE yields 40.83 box mAP and 37.13 mask mAP, outperforming the baseline (which applies quantization only) by +3.03 and +2.21, respectively. A similar trend holds for the more challenging A4/W4 configuration, where SLOPE achieves 38.79/35.28 compared to the baseline's 29.64/27.59, highlighting the robustness of SLOPE even under both quantization and sparsity constraints.

Method	Bits	Box$_{mAP}$	Mask$_{mAP}$
Baseline		41.0	37.2
Dense	W4	37.80	34.92
SLOPE	W4	**40.83**	**37.13**
Dense	A4/W4	29.64	27.59
SLOPE	A4/W4	**38.79**	**35.28**

Method	Bits	Box$_{mAP}$	Mask$_{mAP}$
Baseline		41.0	37.2
SR-STE	FP.	39.0	35.3
LBC	FP.	39.3	35.4
SLOPE	W4	**39.3**	**36.2**

Table 4: Results of object detection with structured 2:4 sparsity on Mask R-CNN models. Left: finetuning from pre-trained models. Right: training-from-scratch models. "Dense" denotes without applying 2:4 sparsity.

5.3 Analysis

In this section, we perform additional analysis of SLOPE, studying the effect of weight discrepancy on the performance, and perform the ablation study on the regularization term.

5.3.1 The Effect of Weight Discrepancy on SLOPE's performance

Table 5 demonstrates that reducing weight discrepancy with SLOPE is crucial for achieving strong performance under high compression rates. Under the 4-bit 2:4 setting, SLOPE improves accuracy from 68.36% to 71.11%, surpassing even the full-precision baseline. In the more extreme 2-bit 2:4 case, it improves from 64.47% to 67.59%. These gains are supported by alignment metrics: cosine similarity increases from 0.919 to 0.953 (4-bit) and from 0.831 to 0.922 (2-bit); SQNR improves from 8.43 dB to 11.59 dB (4-bit) and from 4.84 dB to 8.92 dB (2-bit). These results demonstrate that SLOPE effectively reduces both directional and magnitude deviations introduced by sparse low-bit quantization.

Settings	Acc.	Cos ± std.	SQNR ± std.	Compression
2:4	69.96	0.975 ± 0.01	14.74 ± 0.98	50%
4-bit	71.10	0.948 ± 0.01	10.51 ± 1.01	87.5%
2:4, 4-bit	68.36	0.919 ± 0.01	8.43 ± 1.04	90.63%
2:4, 4-bit, SLOPE	**71.11**	0.953 ± 0.01	11.59 ± 1.09	90.63%
2-bit	67.60	0.878 ± 0.03	7.54 ± 1.08	93.75%
2:4, 2-bit	64.47	0.831 ± 0.03	4.84 ± 1.13	93.75%
2:4, 2-bit, SLOPE	**67.59**	0.922 ± 0.02	8.92 ± 1.04	93.75%

Table 5: ResNet-18 performance across compression settings with and without SLOPE.

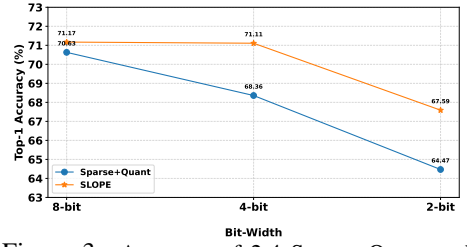

Figure 3: Accuracy of 2:4 Sparse+Quant and SLOPE under varying bit-widths on ResNet-18.

5.3.2 Ablation Study

Here we study the role of our cosine weight regularizer, and compare it to L_2 penalty on weight discrepancy. We compare SLOPE to the baseline without regularization under varying sparsity patterns (2:4, 2:8, 2:16) and bit-widths (Table 6, Table 7) on ResNet-18 with ImageNet. The results indicate that higher sparsity ratios (e.g., 2:16) which can offer greater computational savings, lead to a more pronounced drop in accuracy. However, the proposed SLOPE method consistently mitigates this degradation, particularly in low-bit quantization settings, demonstrating its ability to preserve accuracy across varying sparsity levels (Table 6, Fig. 3). These improvements highlight the importance of angular (and not absolute) weight alignment for maintaining representational fidelity [25] and justify our cos regularizer.

	FP		A8/W8			A4/W4			A2/W2		
N:M	Baseline	SLOPE	Baseline	L_2	SLOPE	Baseline	L_2	SLOPE	Baseline	L_2	SLOPE
2:4	70.70	71.25	70.63	70.42	71.17	68.36	69.04	71.11	64.47	65.26	67.59
2:8	69.62	70.36	69.71	69.33	70.11	66.83	67.53	**68.77**	59.76	60.65	61.87
2:16	66.73	67.94	66.85	66.81	67.93	65.38	66.07	66.59	-	-	-

Table 6: Different settings of sparse N:M n-bit quantization on ResNet-18. "-" denotes not converged models. "Baseline" does not use any additional loss term. **The 4-bit 2:8 quantization scheme achieves a** 19.7× **compression ratio while preserving nearly 99% of the original accuracy (68.77 vs. 69.76).** Please find the compression ratio in Appendix.

Bits	Quant	Quant+L_2 [10]	Sparse$_{2:4}$+Quant	Sparse$_{2:4}$+Quant+L_2	SLOPE$_{2:4}$
2-bit	67.60	66.20	64.47	65.26	**67.59**
4-bit	71.10	70.30	68.36	69.04	**71.11**

Table 7: Comparison of different quantization and regularization strategies under 2-bit and 4-bit settings with ResNet-18 models.

Under both 4-bit and 2-bit configurations (Table 7), SLOPE significantly outperforms L_2, indicating that aligning weight directions (as SLOPE encourages) is more effective than minimizing Euclidean distance alone. Moreover, SLOPE surpasses the baseline in all settings, validating the effectiveness of our loss design for sparse low-bit quantization.

6 Conclusion

A key contribution of this work is identifying the significant performance degradation when combining weight quantization and structured sparsity for model compression. We then show how identifying the root cause: high weight discrepancy between compressed and original weights, and fixing it with a novel regularizer, can recover most of the lost performance.

Through extensive experiments on diverse models and datasets, such as ResNet-18, ViT-small, and Mask R-CNN, we demonstrate that our proposed method (SLOPE) significantly enhances the performance of quantized sparse models. We show that SLOPE is capable of aggressive compression at low bit rates with induced 2:4 structured sparsity, while maintaining most of the original performance. We believe that this research uncovers the fundamental mechanisms of how sparsity and quantization interact, and proposes new tools and methods for the research community.

References

[1] Zhou Aojun, Ma Yukun, Zhu Junnan, Liu Jianbo, Zhang Zhijie, Yuan Kun, Sun Wenxiu, and Li Hongsheng. Learning n:m fine-grained structured sparse neural networks from scratch. In *International Conference on Learning Representations*, 2021.

[2] Hongxiao Bai. Structured sparsity in the NVIDIA Ampere architecture and applications in search engines | NVIDIA Technical blog, 7 2023. URL https://t.co/LV0wtght40.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[4] Brian Chmiel, Itay Hubara, Ron Banner, and Daniel Soudry. Minimum variance unbiased n: M sparsity for the neural gradients. In *The Eleventh International Conference on Learning Representations*, 2023.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[7] Elias Frantar, Roberto L Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. Marlin: Mixed-precision auto-regressive parallel inference on large language models. In *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, pages 239–251, 2025.

[8] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021.

[9] Jinyang Guo, Jianyu Wu, Zining Wang, Jiaheng Liu, Ge Yang, Yifu Ding, Ruihao Gong, Haotong Qin, and Xianglong Liu. Compressing large language models by joint sparsification and quantization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=sCGRhnuMUJ.

[10] Tiantian Han, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Improving low-precision network quantization via bin regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5261–5270, 2021.

[11] Simla Burcu Harma, Ayan Chakraborty, Elizaveta Kostenok, Danila Mishin, Dongho Ha, Babak Falsafi, Martin Jaggi, Ming Liu, Yunho Oh, Suvinay Subramanian, et al. Effective interplay between sparsity and quantization: From theory to practice. *arXiv preprint arXiv:2405.20935*, 2024.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE ICCV*, pages 2961–2969, 2017.

[14] Yuezhou Hu, Jun Zhu, and Jianfei Chen. S-STE: Continuous pruning function for efficient 2:4 sparse pre-training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8abNCVJs2j.

[15] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

[16] Itay Hubara, Brian Chmiel, Moshe Island, Ron Banner, Joseph Naor, and Daniel Soudry. Accelerated sparse neural training: A provable and efficient method to find n: m transposable masks. *Advances in neural information processing systems*, 34: 21099–21111, 2021.

[17] Jangho Kim, KiYoon Yoo, and Nojun Kwak. Position-based scaled gradient for model quantization and pruning. *Advances in neural information processing systems*, 33: 20415–20426, 2020.

[18] Andrey Kuzmin, Markus Nagel, Mart Van Baalen, Arash Behboodi, and Tijmen Blankevoort. Pruning vs quantization: Which is better? 2023.

[19] Fengfu Li, Bo Zhang, and Bin Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

[20] Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pages 2849–2858. PMLR, 2016.

[21] Mingbao Lin, Rongrong Ji, Zihan Xu, Baochang Zhang, Yan Wang, Yongjian Wu, Feiyue Huang, and Chia-Wen Lin. Rotated binary neural network. *NeurIPS*, 33:7474–7485, 2020.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Dan Liu, Xi Chen, Chen Ma, and Xue Liu. Hyperspherical quantization: Toward smaller and more accurate models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5262–5272, 2023.

[24] Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *40th International Conference on Machine Learning*, volume 202, pages 21813–21824. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23w.html.

[25] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018.

[26] Yucheng Lu, Shivani Agrawal, Suvinay Subramanian, Oleg Rybakov, Christopher De Sa, and Amir Yazdanbakhsh. Step: learning n: M structured sparsity masks from scratch with precondition. In *International Conference on Machine Learning*, pages 22812–22824. PMLR, 2023.

[27] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*, 2021.

[28] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *39th International Conference on Machine Learning*, volume 162, pages 16318–16330. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nagel22a.html.

[29] Pytorch. PyTorch Numeric Suite Tutorial, 1 2024. URL https://pytorch.org/tutorials/prototype/numeric_suite_tutorial.html.

[30] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. arxiv 2020. *arXiv preprint arXiv:2012.12877*, 2020.

[32] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[33] Kohei Yamamoto. Learnable companding quantization for accurate low-bit neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5029–5038, 2021.

[34] Yuxin Zhang, Mingbao Lin, Zhihang Lin, Yiting Luo, Ke Li, Fei Chao, Yongjian Wu, and Rongrong Ji. Learning best combination for efficient n: M sparsity. *Advances in Neural Information Processing Systems*, 35:941–953, 2022.

[35] Yuxin Zhang, Yiting Luo, Mingbao Lin, Yunshan Zhong, Jingjing Xie, Fei Chao, and Rongrong Ji. Bi-directional masks for efficient n: M sparse training. In *International conference on machine learning*, pages 41488–41497. PMLR, 2023.

[36] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016. URL http://arxiv.org/abs/1606.06160.

[37] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

Appendix

Sparse-only Results

Table 8 compares our proposed L_{reg} with other leading sparse 2:4 training methods such as Bi-Mask [35], SR-STE[1], T-Mask[16], LBC [34], and S-STE [14]. Our method achieves the best Top-1 accuracy of 80.7%, outperforming all baselines.

ASP[27]	Bi-Mask	SR-STE	T-Mask	LBC	S-STE	L_{reg}
79.9	77.6	79.6	71.5	78.0	78.5	**80.7**

Table 8: Results of 2:4 sparse-only on DeiT-small models with ImageNet dataset.

Efficiency of n-bit Structured 2:4 Sparsity

Implementing n-bit structured 2:4 sparsity yields significant memory savings and computational speedups [27]. With 8-bit quantization, this yields up to 2× speedup with negligible accuracy loss [1, 14, 27], while 4-bit quantization achieves up to 4× speedup [7]. Table 9 summarizes the storage compression ratio relative to 32-bit dense weights.

Bitwidth (n)	2:4 Sparse (bits)	Savings vs FP32	Formula	Compression Ratio
8-bit	$2\times 8+4=20$	84.38%	$\frac{128-20}{128}$	6.4×
4-bit	$2\times 4+4=12$	90.63%	$\frac{128-12}{128}$	10.7×
2-bit	$2\times 2+4=8$	93.75%	$\frac{128-8}{128}$	16×

Table 9: Storage per 4-weight block and compression savings under structured 2:4 sparsity [27], normalized against 32-bit dense baseline (128 bits per 4 weights).

Bitwidth (n)	2:8 Sparse (bits)	Savings vs FP32	Formula	Compression Ratio
8-bit	$2\times 8+5=21$	91.80%	$\frac{256-21}{256}$	12.2×
4-bit	$2\times 4+5=13$	94.92%	$\frac{256-13}{256}$	19.7×
2-bit	$2\times 2+5=9$	96.48%	$\frac{256-9}{256}$	28.4×

Table 10: Storage per 8-weight block and compression savings under structured 2:8 sparsity, normalized against 32-bit dense baseline (256 bits per 8 weights).

Training Details

The proposed method is in Algorithm 1. The overall process can be summarized as follows: Taking pre-trained model weights as initialization, training the model with $L_{reg}(\mathbf{W}, \widehat{\mathbf{W}})$ (Eq. (4)) to reduce the weight discrepancy and updating the weights and other parameters through STE. The scaling factors of s_x and s_w are initialized and updated by using the LSQ method. When training the ResNet-18 model with 8×V100, each epoch takes about 6 minutes. It takes about $MAX\_EPOCH = 120$ epochs to obtain a sparse quantized models. We follow Nvidia's hyper-parameter settings and training code [1]. For the DeiT vision transformer, we apply our method to the original training code [32] and follow its settings.

[1] https://github.com/NVIDIA/DeepLearningExamples

Algorithm 1 Sparse quantization training approach

while *epoch* $<$ *MAX_EPOCH* **do**
  $\hat{\mathbf{x}} = q(\mathbf{x}; s_x, b)$
  $\widehat{\mathbf{W}} = q(S(\mathbf{W}; N, M); s_w, b)$
  $\mathbf{y} = \widehat{\mathbf{W}}^\top \hat{\mathbf{x}}$
  $J(\mathbf{W}) = L(\widehat{\mathbf{W}}) + \lambda L_{reg}(\mathbf{W}, \widehat{\mathbf{W}})$
  Get $\frac{\partial J}{\partial \widehat{\mathbf{W}}}$ via STE to update $\mathbf{W}$, s_x, and s_w.
end while

Error Bound for Structured Sparse Quantization

Theorem 1 (Structured 2:4 Sparse Quantization Bounds)**.** *Let* $\mathbf{w} \in \mathbb{R}^d$ *and obtain* $\hat{\mathbf{w}}$ *by structured* 2:4 *sparsification, i.e. in every 4-element block we keep the two largest-magnitude entries and set the other two to zero. Let the angle between* $\mathbf{w}$ *and* $\hat{\mathbf{w}}$ *be* $\theta \in [0, \frac{\pi}{2}]$ *(so* $\cos\theta = \frac{\mathbf{w}^\top \hat{\mathbf{w}}}{\|\mathbf{w}\|_2 \|\hat{\mathbf{w}}\|_2}$*). Then*

$$\boxed{\|\mathbf{w}\|_2^2 \sin^2\theta \;\le\; \|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 \;\le\; 2\|\mathbf{w}\|_2^2(1-\cos\theta)} \tag{A}$$

and, necessarily,

$$\boxed{\cos\theta \;\ge\; \frac{1}{\sqrt{2}} \quad\Longleftrightarrow\quad \theta \;\le\; 45^\circ} \tag{B}$$

Proof. The proof has two short ingredients.

1. 2:4 blocks preserve $\ge \frac{1}{2}$ of the energy. Fix a 4-vector $\mathbf{z} = (z_1, z_2, z_3, z_4)$ and, w.l.o.g., relabel so that $|z_1| \ge |z_2| \ge |z_3| \ge |z_4|$. Averaging tells us

$$\frac{z_1^2 + z_2^2}{2} \;\ge\; \frac{z_1^2 + z_2^2 + z_3^2 + z_4^2}{4} \;=\; \frac{1}{4}\|\mathbf{z}\|_2^2,$$

so $z_1^2 + z_2^2 \ge \frac{1}{2}\|\mathbf{z}\|_2^2$. Applying this block-by-block and summing yields

$$\boxed{\|\hat{\mathbf{w}}\|_2^2 \;\ge\; \tfrac{1}{2}\|\mathbf{w}\|_2^2} \quad\Longrightarrow\quad \|\hat{\mathbf{w}}\|_2 \;\ge\; \tfrac{1}{\sqrt{2}}\|\mathbf{w}\|_2. \tag{1}$$

2. Standard law-of-cosines decomposition. For any pair of vectors with angle θ,

$$\|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 = \|\mathbf{w}\|_2^2 + \|\hat{\mathbf{w}}\|_2^2 - 2\|\mathbf{w}\|_2\|\hat{\mathbf{w}}\|_2\cos\theta = \big(\|\mathbf{w}\|_2 \sin\theta\big)^2 + \big(\|\hat{\mathbf{w}}\|_2 - \|\mathbf{w}\|_2\cos\theta\big)^2. \tag{2}$$

Lower bound. Since the second square in (2) is non-negative,

$$\|\mathbf{w} - \hat{\mathbf{w}}\|_2^2 \;\ge\; \|\mathbf{w}\|_2^2 \sin^2\theta,$$

which is exactly the left-hand inequality in (A). (The $\frac{1}{2}$ factor that appeared in the draft is unnecessary—it only weakens the bound.)

Upper bound. Use $\|\hat{\mathbf{w}}\|_2 \le \|\mathbf{w}\|_2$ (sparsification never increases the norm) in the first form of (2):

$$\|\mathbf{w}-\hat{\mathbf{w}}\|_2^2 \le \|\mathbf{w}\|_2^2 + \|\mathbf{w}\|_2^2 - 2\|\mathbf{w}\|_2^2\cos\theta = 2\|\mathbf{w}\|_2^2(1-\cos\theta),$$

giving the right-hand inequality in (A).

Angular constraint. Combine the Cauchy lower bound $\mathbf{w}^\top\hat{\mathbf{w}} \ge \frac{1}{2}\|\mathbf{w}\|_2^2$ (sum of preserved squares per block) with (1):

$$\cos\theta = \frac{\mathbf{w}^\top\hat{\mathbf{w}}}{\|\mathbf{w}\|_2\|\hat{\mathbf{w}}\|_2} \ge \frac{\frac{1}{2}\|\mathbf{w}\|_2^2}{\|\mathbf{w}\|_2\cdot(\|\mathbf{w}\|_2/\sqrt{2})} = \frac{1}{\sqrt{2}},$$

establishing (B). Equality occurs when each 4-block looks like (a,a,a,a) up to sign, matching the intuitive "worst-case" example.

□

Tightness

Proposition 1 (Both bounds coalesce as $\theta \to 0$)**.** *For the setting of Theorem 1, denote the quantization error by* $E(\theta) = \|\mathbf{w}-\hat{\mathbf{w}}\|_2^2$ *and recall the bounds*

$$L(\theta) := \|\mathbf{w}\|_2^2\sin^2\theta \;\le\; E(\theta) \;\le\; U(\theta) := 2\|\mathbf{w}\|_2^2\big(1-\cos\theta\big).$$

Then, as $\theta \to 0$,

$$L(\theta) = \|\mathbf{w}\|_2^2\theta^2 + \mathcal{O}\big(\theta^4\big), \qquad U(\theta) = \|\mathbf{w}\|_2^2\theta^2 + \mathcal{O}\big(\theta^4\big),$$

and hence

$$U(\theta) - L(\theta) \;=\; \mathcal{O}\big(\theta^4\big).$$

Consequently, minimising the surrogate loss $1-\cos\theta$ *(equivalently,* $\frac{1}{2}\theta^2 + o(\theta^2)$*) drives the* true *quantisation error* $E(\theta)$ *down quadratically in* θ*, while simultaneously squeezing the gap between the lower and upper analytical bounds at the even faster quartic rate* θ^4 *(Fig. 4).*

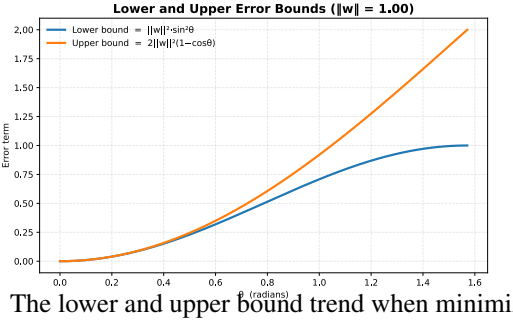

Figure 4: The lower and upper bound trend when minimizing θ.

Proof. A second-order Maclaurin expansion of the elementary trigonometric functions yields, for $\theta \to 0$,

$$\sin\theta = \theta - \frac{\theta^3}{6} + \mathcal{O}\Big(\theta^5\Big), \qquad \cos\theta = 1 - \frac{\theta^2}{2} + \frac{\theta^4}{24} + \mathcal{O}\Big(\theta^6\Big).$$

Lower bound.

$$L(\theta) = \|\mathbf{w}\|_2^2 \sin^2\theta = \|\mathbf{w}\|_2^2\big(\theta - \tfrac{\theta^3}{6} + \mathcal{O}(\theta^5)\big)^2 = \|\mathbf{w}\|_2^2\theta^2 + \mathcal{O}\big(\theta^4\big)\,.$$

Upper bound.

$$U(\theta) = 2\|\mathbf{w}\|_2^2\big(1-\cos\theta\big) = 2\|\mathbf{w}\|_2^2\big(\tfrac{\theta^2}{2} - \tfrac{\theta^4}{24} + \mathcal{O}(\theta^6)\big) = \|\mathbf{w}\|_2^2\theta^2 + \mathcal{O}\big(\theta^4\big)\,.$$

Gap between bounds. Subtracting the two expansions shows $U(\theta) - L(\theta) = \mathcal{O}(\theta^4)$. Thus both analytical bounds converge to the same leading-order term $\|\mathbf{w}\|_2^2\theta^2$, while their separation shrinks *two orders faster* than the error itself .

□