MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Model Compression of Vision Transformer for Electric Motor Cogging Torque Prediction

Chen, Jian; Koike-Akino, Toshiaki; Wang, Ye; Yamamoto, Tatsuya; Sakamoto, Yusuke; Wang, Bingnan

TR2026-084 June 23, 2026

Abstract

Vision Transformer (ViT) models have recently demonstrated strong performance in predicting electric motor cogging torque from visual representations of motor geometry. However, the high computational cost and memory footprint of ViT architectures hinder their deployment in resource- constrained environments such as embedded motor drives and real-time digital twins. This paper investigates post-training model compression techniques for a ViT-based cogging torque prediction model. Specifically, we evaluate some activation-aware pruning methods, which leverage activation statistics to identify redundant weights, as well as low-rank factorization applied to attention and feed-forward layers. Experimental results demonstrate that substantial reductions in parameter count and inference cost can be achieved with negligible degradation in prediction performance. These findings highlight the feasibility of deploying compressed ViT models for efficient and scalable electric motor analysis.

IEEE World Congress on Computational Intelligence 2026

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Model Compression of Vision Transformer for Electric Motor Cogging Torque Prediction

Jian Chen
Department of Mechanical Engineering
University of Houston
Houston, USA
jchen216@cougarnet.uh.edu

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
koike@merl.com

Ye Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
yewang@merl.com

Tatsuya Yamamoto
AI Transformation Innovation Center
Mitsubishi Electric Corporation
Kamakura, Kanagawa 247-8501 Japan
Yamamoto.Tatsuya@ah.MitsubishiElectric.co.jp

Yusuke Sakamoto
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan
Sakamoto.Yusuke@df.MitsubishiElectric.co.jp

Bingnan Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139 USA
bwang@merl.com

***Abstract*—Vision Transformer (ViT) models have recently demonstrated strong performance in predicting electric motor cogging torque from visual representations of motor geometry. However, the high computational cost and memory footprint of ViT architectures hinder their deployment in resource-constrained environments such as embedded motor drives and real-time digital twins. This paper investigates post-training model compression techniques for a ViT-based cogging torque prediction model. Specifically, we evaluate some activation-aware pruning methods, which leverage activation statistics to identify redundant weights, as well as low-rank factorization applied to attention and feed-forward layers. Experimental results demonstrate that substantial reductions in parameter count and inference cost can be achieved with negligible degradation in prediction performance. These findings highlight the feasibility of deploying compressed ViT models for efficient and scalable electric motor analysis.**

***Index Terms*—model compression, low-rank decomposition, activation-aware pruning, electric motors, surrogate model, vision transformer.**

I. Introduction

Cogging torque [1], originating from the interaction between permanent magnets and stator slots in electric machines, is a major source of torque ripple, vibration, and acoustic noise. Accurate prediction of cogging torque is therefore essential for electric motor design, performance optimization, and noise–vibration–harshness (NVH) mitigation. Traditional approaches rely heavily on finite element analysis (FEA) [2], [3], which, although accurate, are computationally expensive and unsuitable for rapid design iteration or real-time analysis.

To address these limitations, data-driven methods have been increasingly explored as efficient alternatives to physics-based simulations. In particular, recent work [4]–[8] has demonstrated that deep learning models can learn complex nonlinear mappings between motor geometry and cogging torque characteristics directly from data. Among these approaches, Vision Transformer (ViT) models [9], [10] have demonstrated notable potential due to their strong global modeling capability and effectiveness in capturing long-range spatial dependencies in visual representations of motor geometries.

Sun et al. [10] proposed a ViT-based framework to predict electric motor cogging torque from motor geometry images (Motor ViT). By treating motor cross-sectional images as sequences of visual tokens, the ViT model effectively learns geometry–torque relationships without manual feature engineering. Experimental results showed that the proposed ViT model outperformed convolutional neural networks, like VGG [11] and ResNet [12] in cogging torque prediction accuracy, highlighting the effectiveness of transformer-based architectures for electric machine analysis.

Despite these performance gains, ViTs are known to be computationally intensive and parameter-heavy [13]. The high memory footprint and inference cost of ViT models pose significant challenges for deployment in practical engineering settings, such as embedded motor controllers, real-time digital twins, and large-scale design optimization workflows. In such scenarios, efficiency, latency, and hardware constraints are often as critical as prediction accuracy.

Model compression [14], [15] provides a promising pathway to address these challenges by reducing model size and computational complexity while preserving predictive performance. However, most existing studies on ViT compression focus on image classification tasks and generic computer vision benchmarks. The effectiveness of advanced compression techniques—particularly activation-aware pruning and activation-aware low-rank decomposition—in regression problems such as cogging torque prediction remains largely unexplored.

Motivated by these gaps and the aim to reduce model complexity, this work investigates post-training model compression techniques of the ViT-based cogging torque prediction model

This work was done when J. Chen was with Mitsubishi Electric Research Laboratories as research intern. Corresponding Author: Bingnan Wang (bwang@merl.com).

(Motor ViT) proposed by Sun et al [10]. We systematically apply activation-aware compression strategies to the model, including pruning and decomposition methods, and evaluate their impact on model performance, parameter count, and computational cost.

II. Methodology

To compress ViT-based cogging torque prediction model (we call Motor ViT hereafter), we evaluate four modern activation-aware compression strategies, namely SparseGPT [16], Pruning by Weights and Activations (Wanda) [17], Pruning with Projected Gradient Descent (PGD) [18], and rank reduction with Activation-aware Singular-Value Decomposition (ASVD) [19]. The objective of those methods is to reduce model size and computational complexity while preserving prediction accuracy.

A. SparseGPT Pruning

SparseGPT [16] is a post-training, activation-aware pruning method that formulates weight sparsification as a layer-wise reconstruction problem. Instead of directly optimizing the global task loss, SparseGPT seeks a sparse approximation of pretrained weights individually based on optimal brain surgeon (OBS) to preserve the original module outputs under input activations obtained in a small set of calibration data.

Given a linear transformation module $\mathbf{Y} = \mathbf{W}\mathbf{X}$, SparseGPT aims to find a sparse weight matrix $\widehat{\mathbf{W}}$ that minimizes the reconstruction error

$$\mathcal{L}(\widehat{\mathbf{W}}) = \|\mathbf{W}\mathbf{X} - \widehat{\mathbf{W}}\mathbf{X}\|^2 = \mathrm{tr}[(\mathbf{W}-\widehat{\mathbf{W}})^\top \mathbf{C}(\mathbf{W}-\widehat{\mathbf{W}})], \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denotes the original weight matrix and $\mathbf{X}$ represents the matrix of input activations collected from a calibration dataset. $\mathbf{C} \triangleq \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ is the empirical input correlation. This objective induces a quadratic form in the weight perturbation $\mathbf{W} - \widehat{\mathbf{W}}$, where the empirical input covariance $\mathbf{C}$ captures the sensitivity of the layer output to individual weight elements.

Pruning is performed in a greedy, sequential manner according to the triangularized sensitivity matrix based on the Cholesky decomposition of $\mathbf{C}$. At each pruning step, SparseGPT removes weights that contribute least to the reconstruction objective. Importantly, after a weight is pruned, the remaining weights are locally updated to compensate for the induced output error. This compensation is computed by projecting the output error caused by the pruned weight onto the subspace spanned by the remaining input activations, effectively redistributing the contribution of the removed parameter across the unpruned weights. By explicitly accounting for the correlations between input channels, this mechanism prevents error accumulation during successive pruning steps.

B. Wanda Pruning

Wanda [17] is a simplified yet effective pruning method, competitive to SparseGPT. Wanda approximates the covariance matrix into a diagonal matrix, which can remove the sequential operations needed for SparseGPT. Specifically, the original loss in (1) will be simplified as:

$$\mathcal{L}'(\widehat{\mathbf{W}}) = \|(\mathbf{W} - \widehat{\mathbf{W}})\mathsf{diag}[\mathbf{C}]^{\frac{1}{2}}\|^2, \quad (2)$$

where $\mathsf{diag}[\cdot]$ is a diagonal matrix, nulling out off-diagonal elements of an argument matrix. Because there is no correction, the optimal sparse matrix $\widehat{\mathbf{W}}$ can be obtained in a straightforward manner. Wanda computes an importance score for each weight element as:

$$\mathbf{Q}_{ij} = |\mathbf{W}_{ij}| \cdot |\mathbf{C}_{jj}|^{\frac{1}{2}}. \quad (3)$$

Weight elements with the smallest scores are pruned to achieve a target sparsity level. Wanda pruning is attractive for post-training scenarios due to its simplicity, low computational overhead, and strong empirical performance at moderate sparsity levels.

C. AWP Pruning

AWPPGD [18] formulates post-training pruning as a compressed sensing problem, where sparsity is enforced through iterative projection rather than one-shot weight removal. Since the pruning problem in ((1)) has no closed-form solution, AWPPGD exploits the iterative hard-thresholding (IHT) method under sparsification constraint:

$$\mathcal{C} := \left\{ \widehat{\mathbf{W}} \ : \ \|\widehat{\mathbf{W}}\|_0 \le \rho\, d_{\text{in}} d_{\text{out}} \right\}, \quad (4)$$

where ρ is a sparsity value, and $\|\cdot\|_0$ is the ℓ_0 pseudo norm that counts the total number of nonzero elements. This is a well-studied sparse approximation problem.

The optimization is carried out using Projected Gradient Descent (PGD) to deal with such a constraint. The weight parameters are updated iteratively via gradient descent, followed by a projection step that enforces sparsity:

$$\mathbf{Z}^{(t)} = \widehat{\boldsymbol{W}}^{(t)} + \eta(\mathbf{W} - \widehat{\mathbf{W}}^{(t)})\mathbf{C}, \quad (5)$$

$$\widehat{\mathbf{W}}^{(t+1)} = \mathsf{Proj}_{\mathcal{C}}\left(\mathbf{Z}^{(t)}\right). \quad (6)$$

where $\widehat{\mathbf{W}}^{(t)} \in constraint$ is the compressed weight at the tth iteration, and η is the step size. In particular, to ensure $\widehat{\mathbf{W}}^{(0)}$ meets the constraint, we initialize it with Wanda- or SparseGPT-compressed weights in experiments. This iterative adaptation mitigates abrupt performance degradation and improves robustness at moderate-to-high sparsity levels. Note that AWPPGD can incorporate quantization as well as pruning into a constraint.

D. ASVD Low-Rank Decomposition

Low-rank decomposition [20] reduces model complexity by factorization. For a linear layer with weight matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, it seeks:

$$\mathbf{W} \approx \mathbf{B}\mathbf{A}, \quad \mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}, \ \mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}} \quad (7)$$

where $r \ll \min\{d_{\text{in}}, d_{\text{out}}\}$ is the rank. A natural way to realize the low-rank approximation is to use the truncated SVD, which decomposes a matrix $\mathbf{W}$ as $\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, where

$\mathbf{U}$, $\mathbf{V}$ are the left and right singular vectors, respectively, and $\mathbf{S}$ is a diagonal matrix of non-negative singular values that are sorted in descending order. By keeping only the largest r singular values and corresponding vectors, we can have the optimal rank-r approximation of $\mathbf{W} \approx \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^\top$. The low-rank matrices $\mathbf{B}$ and $\mathbf{A}$ can be formed, e.g., as follows:

$$\mathbf{B} = \mathbf{U}_r\mathbf{S}_r^{1/2}, \qquad \mathbf{A} = \mathbf{S}_r^{1/2}\mathbf{V}_r^\top \tag{8}$$

Restricting the approximation to pretrained weights alone is suboptimal, since the interaction between weight perturbations and activation distributions ultimately determines model outputs. Consequently, the optimization objective should be formulated to directly minimize the activation-aware loss in (1, which can be realized by approximating $\mathbf{W}\mathbf{C}^{\frac{1}{2}}$ [19], [20]:

$$\mathbf{W} = (\mathbf{W}\mathbf{C}^{\frac{1}{2}})\mathbf{C}^{-\frac{1}{2}} = \mathsf{svd}[\mathbf{W}\mathbf{C}^{\frac{1}{2}}]\mathbf{C}^{-\frac{1}{2}} = (\mathbf{U}\mathbf{S}\mathbf{V}^\top)\mathbf{C}^{-\frac{1}{2}}. \tag{9}$$

By keeping only the largest r singular values and corresponding vectors, we can have $\mathbf{W} \approx \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^\top\mathbf{C}^{-1/2}$, and the low-rank matrices are given, e.g., by

$$\mathbf{B} = \mathbf{U}_r\mathbf{S}_r^{1/2}, \qquad \mathbf{A} = \mathbf{S}_r^{1/2}\mathbf{V}_r^\top\mathbf{C}^{-1/2} \tag{10}$$

E. Motor ViT Compression

Activation-aware low-rank decomposition is applied primarily to linear layers, which dominate the parameter count and computational cost in Motor ViT architectures. Pruning methods introduce sparsity into the weight matrices; however, translating this sparsity into inference-time speedups and model size reduction requires dedicated sparse computation backends and storage formats. Low-rank decomposition, on the other hand, provides a structured parameterization that directly reduces the number of model parameters and inference floating-point operation per second (FLOPs), independent of sparse execution support.

After training of the Motor ViT model [10], these four compression methods are applied to the linear layers within the multi-head self-attention blocks and the feed-forward network blocks of the transformer encoders. Activation statistics are collected using a calibration dataset to guide pruning and low-rank decomposition decisions. The compressed model is evaluated with and without fine-tuning.

III. Experimental Setup

Sun et al. [10] developed the Motor ViT to evaluate the cogging torque of 4-pole, 24-slot interior permanent magnet synchronous motor (IPMSM). To prepare the dataset, a total of 13 geometric design parameters on the rotor and stator are varied to generate 19,373 distinct design candidates, and finite-element analysis (FEA) is conducted under no-load conditions to obtain the torque waveform and induced voltage (EMF) for each design candidate. Instead of learning a high-dimensional waveform directly, the torque profile is decomposed into the dominant Fourier terms,

$$\begin{aligned} T(\theta) =& A_{12}\cos(12\theta) + B_{12}\sin(12\theta) \\ &+ A_{24}\cos(24\theta) + B_{24}\sin(24\theta), \end{aligned} \tag{11}$$

TABLE I Model Description

Model Name	Motor ViT-B-32
Model Input	2D image and 13 design parameters
Model Output	4 cogging torque Fourier coefficients and electromotive force(EMF)
Number of trainable parameters	88M

such that the waveform can be reconstructed from these coefficients $(A_{12}, B_{12}, A_{24}, B_{24})$. The learning target therefore consists of predicting the Fourier coefficients (for cogging torque reconstruction) and EMF given the motor design input, which includes both the 2D cross-section image and the corresponding design-parameter vector. Finally, the cogging torque is computed from the reconstructed waveform as the peak-to-peak difference

$$T_c = \max(T(\theta)) - \min(T(\theta)), \tag{12}$$

Truncation error of the Fourier method is negligible compared with computation directly from original torque waveform on the dataset. The cogging torque prediction error is computed as:

$$\text{Error} = \text{RMSE}\left(T_c^{\text{target}}, T_c^{\text{predicted}}\right). \tag{13}$$

We evaluate the model compression for Motor ViT [10] for motor cogging torque prediction. Among the four model variants in [10], we use the trained Motor ViT-B-32 model (TABLE I). The dataset contains 19,373 samples, split into training, validation, and test sets with a ratio of 70%, 10%, and 20%, respectively. Next, the model is trained for 300 epochs with a batch size of 128 and evaluated on the test set. More details of Motor ViT are provided in [10].

The four compression methods described in Section II are implemented and evaluated at three different sparsity levels (0.4, 0.5, 0.6), which indicates that (40%, 50%, 60%) of the model weights are non-zero after compression respectively. We apply these compression methods to most of the linear modules in the Motor ViT. In particular, for ASVD, the sparsity is used to compute the decomposition rank such that the resulting compressed layer achieves a parameter reduction level comparable to that of the pruning-based methods under the same reported sparsity setting. In addition, we randomly selected 200 samples from the training set as the calibration set.

Next, we consider two experimental settings: compression without fine-tuning and compression followed by fine-tuning. For fair comparison, all compressed models use the same fine-tuning configuration. Specifically, fine-tuning is performed on the training set for 300 epochs with a learning rate of 0.01 and a batch size of 128. During the fine-tuning stage, we first initialized the model with compressed weights and only updated the non-zero elements in these weights to maintain the weight sparsity through masked gradient update. The mask matrices were saved during the model compression stage and element-wise multiplications were conducted between mask matrices and gradient matrices.

IV. Results and Discussion

A. Model Compression w/o Fine-Tuning

In this experiment, we apply compression methods to trained Motor ViT without fine-tuning under three sparsity values. Fig. 1 illustrates the performance comparison between the original model and four compressed models obtained using different compression techniques. In addition, TABLE II-IV present the corresponding numeric values as a reference. We also provided the standard deviation for RMSE in these tables and it shows the stability of model output. It can be observed that across all compression methods, when we change the sparsity levels from 0.6 (40% weights to 0) to 0.4 (60% weights to 0), the model weights become sparser, the model size decreases, and the performance declines. The most noticeable drops occur when the sparsity value changes from 0.5 to 0.4.

In addition, different compression methods exhibit varying degrees of performance degradation. Wanda exhibits the largest performance degradation for cogging torque and EMF predictions among all compression methods across all sparsity values, while SparseGPT consistently achieves the best performance. At sparsity 0.4 (60% weights to 0), SparseGPT yields substantially lower RMSE and higher R-squared value than the other three compression methods, especially in cogging torque prediction, indicating stronger robustness under aggressive pruning. This advantage is expected because SparseGPT minimizes layer-wise reconstruction error under calibration activations and applies a compensation update after pruning, which helps preserve the functional behavior of transformer linear layers.

ASVD and AWPPGD show competitive performance at moderate-to-high sparsity (0.5–0.6). Wanda shows the highest degradation at low sparsity values (high sparsity level). However, it improves rapidly as the sparsity level decreases (from 0.4 to 0.6) and catches up with the other methods at sparsity 0.6. This behavior may imply that simple activation-weight scoring is sensitive to aggressive pruning but becomes more reliable when the target sparsity is less challenging.

For EMF prediction, all methods achieve very high R-squared values close to 1.0 (Fig. 1(D)) and relatively small RMSE even at sparsity 0.4, indicating that EMF is an easier target for the model to learn and preserve under compression. Nevertheless, clear differences still can be observed.

Finally, we can tell that in compression-only case, SparseGPT is more suitable for the compression of Motor ViT, followed by ASVD and AWPPGD, while Wanda exhibits the poorest performance.

B. Model Compression with Fine-Tuning

In this experiment, Motor ViT is first compressed at three sparsity levels and then fine-tuned on the training set. Fig. 2 compares the performance of the fine-tuned compressed models, while TABLE II-IV report the corresponding numerical results.

A key observation from Fig. 2 is that fine-tuning consistently improves the compressed models, leading to lower RMSE and higher R-squared values across almost all methods and sparsity levels. More importantly, after fine-tuning, the performance of several compressed models match or even surpass the dense baseline model (no compression) in cogging torque prediction. This demonstrates that the dense Motor ViT may be over-parameterized and the sparsification can act as an effective form of regularization when followed by brief adaptation. This behavior contrasts with the compression-only results in Fig. 1, where aggressive sparsity—especially at 0.4—causes notable degradation for methods such as Wanda. Fine-tuning substantially reduces these gaps, indicating that the remaining non-zero weight elements can be reconfigured to compensate for the pruning-induced performance loss.

Similarly, The compressed-and-fine-tuned models show monotonic degradation as the sparsity level increases from 0.6 (40%) to 0.4 (60%), with RMSE increasing and R-squared values decreasing for all methods. The ASVD + fine-tuning (ft) achieves the highest prediction accuracy for both the cogging torque and the EMF at sparsity 0.5. At sparsity 0.4 and 0.6, ASVD + ft attains the lowest RMSE (highest R-squared values) among different approaches in EMF prediction and cogging torque prediction, respectively. In addition, AWPPGD + ft and SparseGPT + ft show competitive performance for both EMF and cogging torque prediction at all sparsity levels. In contrast, Wanda + ft, while significantly improved over its compression-only counterpart, remains more sensitive at sparsity 0.4, exhibiting the largest RMSE and lowest R-squared value at that setting. However, its performance improves steadily with the reduced sparsity level (0.4→0.6) and becomes comparable to the other approaches at sparsity 0.6, implying that fine-tuning partially mitigates suboptimal one-shot pruning decisions.

V. Conclusions

This work investigated post-training compression of the Motor ViT model [10], with the goal of reducing model complexity while maintaining prediction fidelity for both cogging torque and electromotive force (EMF). Four activation-aware compression strategies—SparseGPT, Wanda, AWPPGD, and activation-aware SVD-based low-rank decomposition (ASVD)—were applied to the linear modules of Motor ViT-B-32 (88M parameters) and evaluated in three sparsity settings (0.4/0.5/0.6).

In the case of compression only without fine-tuning, predictive performance decreased as the sparsity level increased (0.6→0.4), with the largest drops observed in the most aggressive sparsity setting (0.4, 60%). SparseGPT consistently delivered the best accuracy among the evaluated methods, particularly under aggressive pruning. When fine-tuning was introduced after compression, compressed models showed systematic performance recovery and multiple configurations matched or exceeded the dense baseline in cogging torque prediction. Notably, ASVD with fine-tuning achieved the strongest overall performance in several settings.

Overall, the findings demonstrate the feasibility of deploying compressed ViT-based surrogate models for efficient

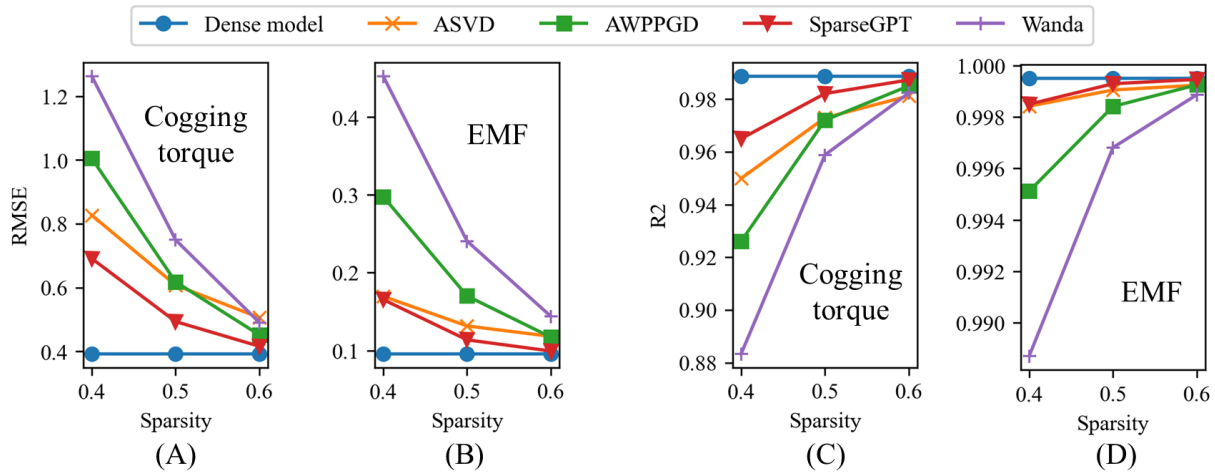

Fig. 1: Performance comparison of compressed models (Motor ViT) under different compression methods. (A) and (B) show the root mean square error (RMSE) of cogging torque and EMF predictions, respectively. (C) and (D) present the R-squared values of cogging torque and EMF predictions, respectively. The sparsity value of 0.4 means that 60% ($(1-0.4)\times 100\%$) of weight elements in linear modules are reduced to 0. The horizontal line of Dense model represent the prediction performance of uncompressed/original Motor ViT.

TABLE II PERFORMANCE COMPARISON OF COMPRESSION METHODS WITH THE SPARSITY VALUE = 0.4

Method	**RMSE of Cogging Torque**	R^2 **of Cogging Torque**	**RMSE of EMF**	R^2 **of EMF**
Dense Model (no compression)	0.3924 ± 0.3542	0.9887	0.0957 ± 0.0620	0.9995
ASVD	0.8270 ± 0.7186	0.9499	0.1696 ± 0.1186	0.9984
AWPPGD	1.0051 ± 0.8170	0.9261	0.2978 ± 0.2073	0.9951
SparseGPT	0.6912 ± 0.5890	0.9650	0.1654 ± 0.1102	0.9985
Wanda	1.2628 ± 1.0224	0.8834	0.4526 ± 0.2811	0.9887
ASVD + ft	0.3894 ± 0.3296	0.9889	0.1241 ± 0.0801	0.9992
AWPPGD + ft	0.3874 ± 0.3239	0.9890	0.1333 ± 0.0865	0.9990
SparseGPT + ft	0.3833 ± 0.3248	0.9893	0.1319 ± 0.0854	0.9990
Wanda + ft	0.4201 ± 0.3463	0.9871	0.1467 ± 0.0950	0.9988

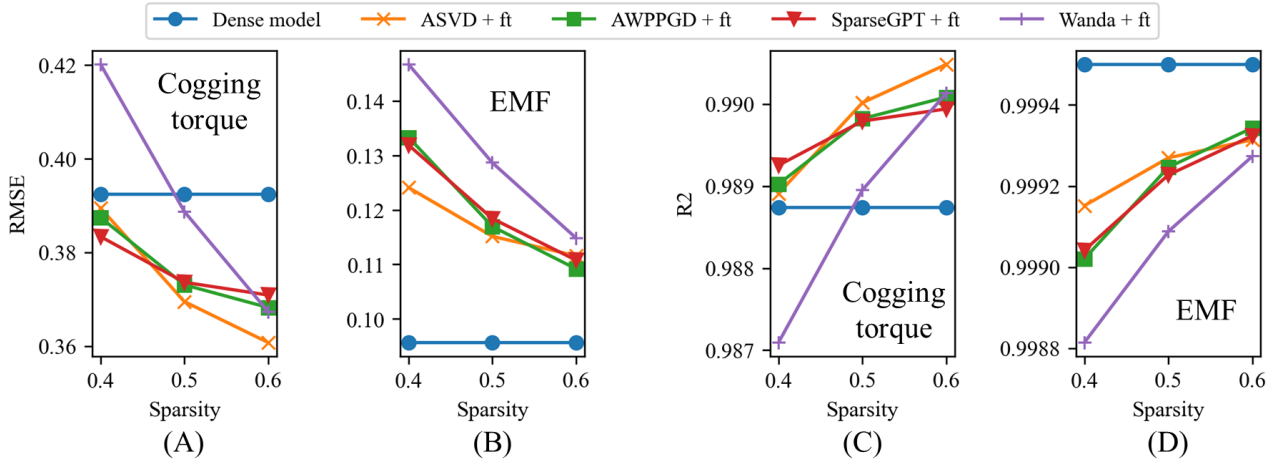

Fig. 2: Performance comparison of compressed models (Motor ViT) under different compression methods and the same fine-tuning (ft) strategy. (A) and (B) show the root mean square error (RMSE) of cogging torque and EMF predictions, respectively. (C) and (D) present the R-squared values of cogging torque and EMF predictions, respectively. The sparsity value of 0.4 means that 60% ($(1-0.4)\times 100\%$) of weight elements in linear modules are reduced to 0. The horizontal line of Dense model represent the prediction performance of uncompressed/original Motor ViT.

TABLE III PERFORMANCE COMPARISON OF COMPRESSION METHODS WITH THE SPARSITY VALUE = 0.5)

Method	RMSE of Cogging Torque	R^2 of Cogging Torque	RMSE of EMF	R^2 of EMF
Dense Model (no compression)	0.3924 ± 0.3542	0.9887	0.0957 ± 0.0620	0.9995
ASVD	0.6079 ± 0.5305	0.9730	0.1316 ± 0.0894	0.9990
AWPPGD	0.6183 ± 0.5183	0.9720	0.1701 ± 0.1159	0.9984
SparseGPT	0.4937 ± 0.4325	0.9822	0.1137 ± 0.0747	0.9993
Wanda	0.7498 ± 0.6070	0.9589	0.2407 ± 0.1480	0.9968
ASVD + ft	0.3695 ± 0.3148	0.9900	0.1152 ± 0.0740	0.9993
AWPPGD + ft	0.3731 ± 0.3199	0.9898	0.1170 ± 0.0759	0.9992
SparseGPT + ft	0.3736 ± 0.3225	0.9898	0.1184 ± 0.0766	0.9992
Wanda + ft	0.3887 ± 0.3292	0.9890	0.1287 ± 0.0836	0.9991

TABLE IV PERFORMANCE COMPARISON OF COMPRESSION METHODS WITH THE SPARSITY VALUE = 0.6

Method	RMSE of Cogging Torque	R^2 of Cogging Torque	RMSE of EMF	R^2 of EMF
Dense Model (no compression)	0.3924 ± 0.3542	0.9887	0.0957 ± 0.0620	0.9995
ASVD	0.5059 ± 0.4470	0.9813	0.1181 ± 0.0788	0.9992
AWPPGD	0.4517 ± 0.4007	0.9851	0.1171 ± 0.0769	0.9992
SparseGPT	0.4161 ± 0.3739	0.9873	0.0993 ± 0.0658	0.9995
Wanda	0.4905 ± 0.4225	0.9824	0.1439 ± 0.0896	0.9989
ASVD + ft	0.3608 ± 0.3099	0.9905	0.1116 ± 0.0718	0.9993
AWPPGD + ft	0.3682 ± 0.3204	0.9901	0.1092 ± 0.0706	0.9993
SparseGPT + ft	0.3709 ± 0.3233	0.9899	0.1108 ± 0.0717	0.9993
Wanda + ft	0.3673 ± 0.3167	0.9901	0.1148 ± 0.0739	0.9993

electric machine analysis in resource-constrained environments.However, the robustness of the conclusions across other motor types, geometries, and operating conditions has not yet been fully verified. Future work will focus on: (1) the development of sparse computation techniques to translate model sparsity into inference-time speedups; (2) trying other recent model compression methods, like hybrid compression strategies that combines structured sparsity with low-rank factorization; (3) the development of advanced compression methods that account for weight sharing; and (4) broader compression validation on other models or motor types.

REFERENCES

[1] D. R. McIntosh, "Identification and analysis of low-frequency cogging torque component in permanent magnet machines," in *Comsol Multiphysics Conference Proceedings*, 2008.

[2] A. Dalcalı, "Cogging torque analysis in permanent magnet synchronous generators using finite elements analysis," *International Transactions on Electrical Energy Systems*, vol. 30, no. 10, p. e12588, 2020.

[3] X. Zhu, W. Hua, and G. Zhang, "Analysis and reduction of cogging torque for flux-switching permanent magnet machines," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 5854–5864, 2019.

[4] Y. Sakamoto, Y. Xu, B. Wang, T. Yamamoto, and Y. Nishimura, "Electric motor surrogate model combining subdomain method and neural network," in *2023 24th International Conference on the Computation of Electromagnetic Fields (COMPUMAG)*. IEEE, 2023, pp. 1–4.

[5] Y. Sakamoto, B. Wang, T. Yamamoto, and Y. Nishimura, "Permanent magnet motor torque waveform prediction using learned gap flux," in *2024 IEEE 21st Biennial Conference on Electromagnetic Field Computation (CEFC)*. IEEE, 2024, pp. 1–2.

[6] M. Tahkola, J. Keränen, D. Sedov, M. F. Far, and J. Kortelainen, "Surrogate modeling of electrical machine torque using artificial neural networks," *IEEE Access*, vol. 8, pp. 220 027–220 045, 2020.

[7] H. Ren, D. Wang, C. Zhang, W. Yan, and C. Liu, "Prediction method and analysis of cogging torque based on adaboost-brnn," in *2021 2nd International Conference on Artificial Intelligence and Information Systems*, 2021, pp. 1–6.

[8] T. Aoyagi, Y. Otomo, H. Igarashi, H. Sasaki, Y. Hidaka, and H. Arita, "Prediction of current-dependent motor torque characteristics using deep learning for topology optimization," *IEEE Transactions on Magnetics*, vol. 58, no. 9, pp. 1–4, 2022.

[9] Y. Shimizu and K. Akatsu, "Deep learning-based automatic design system for ipmsm with variable magnet properties," in *2024 International Conference on Electrical Machines (ICEM)*. IEEE, 2024, pp. 1–6.

[10] S. Sun, Y. Wang, T. Koike-Akino, T. Yamamoto, Y. Sakamoto, and B. Wang, "Electric motor cogging torque prediction with vision transformer models," in *2025 IEEE International Electric Machines & Drives Conference (IEMDC)*. IEEE, 2025, pp. 782–788.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] L. Papa, P. Russo, I. Amerini, and L. Zhou, "A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 12, pp. 7682–7700, 2024.

[14] D. Liu, Y. Zhu, Z. Liu, Y. Liu, C. Han, J. Tian, R. Li, and W. Yi, "A survey of model compression techniques: Past, present, and future," *Frontiers in Robotics and AI*, vol. 12, p. 1518965, 2025.

[15] S. Saha and L. Xu, "Vision transformers on the edge: A comprehensive survey of model compression and acceleration strategies," *Neurocomputing*, p. 130417, 2025.

[16] E. Frantar and D. Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *International conference on machine learning*. PMLR, 2023, pp. 10 323–10 337.

[17] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter, "A simple and effective pruning approach for large language models," *arXiv preprint arXiv:2306.11695*, 2023.

[18] J. Liu, T. Koike-Akino, Y. Wang, H. Mansour, and M. Brand, "Awp: Activation-aware weight pruning and quantization with projected gradient descent," *arXiv preprint arXiv:2506.10205*, 2025.

[19] X. Wang, Y. Zheng, Z. Wan, and M. Zhang, "Svd-llm: Truncation-aware singular value decomposition for large language model compression," *arXiv preprint arXiv:2403.07378*, 2024.

[20] X. Chen, J. Liu, Y. Wang, M. Brand, T. Koike-Akino *et al.*, "Tunecomp: Joint fine-tuning and compression for large foundation models," *arXiv preprint arXiv:2505.21835*, 2025.