SCIENCE CHINA
Information Sciences

• LETTER •

May 2025, Vol. 68, Iss. 5, 159101:1–159101:2
https://doi.org/10.1007/s11432-023-4340-9

Low-rank Winograd transformation for 3D convolutional neural networks

Ziran QIN[1], Mingbao LIN[2], Huabin LIU[1], John SEE[3], Gui ZOU[1] & Weiyao LIN[1*]

[1]*School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*Skywork AI, Singapore 118222, Singapore*
[3]*School of Mathematical and Computer Sciences, Heriot-Watt University Malaysia, Putrajaya 62200, Malaysia*

Received 29 December 2023/Revised 10 November 2024/Accepted 1 February 2025/Published online 3 April 2025

Citation Qin Z R, Lin M B, Liu H B, et al. Low-rank Winograd transformation for 3D convolutional neural networks. Sci China Inf Sci, 2025, 68(5): 159101, https://doi.org/10.1007/s11432-023-4340-9

In recent years, 3D convolutional neural networks (CNNs) have achieved improved accuracy across various 3D data processing tasks, such as video understanding, medical image analysis, and point cloud processing. This success is largely attributed to their ability to effectively extract spatio-temporal and volumetric features.

However, 3D CNNs demand substantial computational resources, primarily because their 3D kernels are computationally intensive. Practical deployment on real-world devices is often hindered by constraints such as inference runtime, memory limitations, and cost. To address these challenges, recent studies [1–3] have explored two complementary optimization strategies. At the operator level, fast convolution algorithms such as Winograd convolution [4] replace spatial convolution operations with element-wise products, eliminating redundant multiplications and reducing the computational load of 3D convolutions. At the model level, network pruning techniques [5] simplify network complexity by removing unnecessary units, significantly decreasing parameters and computational demands. Combining Winograd convolution and network pruning holds promise for enabling deployment on resource-constrained devices. However, these methods are not inherently compatible. The sparsity introduced by pruning is often negated by the kernel transformation used in the Winograd algorithm.

To tackle this incompatibility, some studies have applied the Winograd transformation to CNNs, followed by pruning and retraining the models within the Winograd domain. These efforts, however, have primarily focused on 2D CNNs. Extending this approach to 3D CNNs presents two key challenges. First, directly applying the Winograd transformation to 3D CNNs leads to a substantial increase in parameters, resulting in model redundancy and elevated resource costs. Additionally, existing methods struggle to accelerate pruned Winograd models owing to irregular sparse weight matrices, which fail to fully utilize vector processing architectures or memory buses. Therefore, the issue of accelerating the Winograd transformation, particularly for 3D CNNs, remains unsolved. See Appendixes A and B for more elaboration.

This study introduces a novel low-rank Winograd transformation tailored for 3D CNNs, addressing the challenges associated with high parameter counts and computational complexity inherent in traditional Winograd transformations. To further enhance efficiency, we propose a low-rank-oriented sparse granularity, which enables more regular and efficient sparse patterns in computations.

We evaluate our method across various 3D CNN backbones and datasets, with experimental results demonstrating not only superior performance but also significant reductions in computational overhead. To validate the practicality of our approach, we developed a custom acceleration kernel optimized for the proposed sparse granularity on mobile ARM (advanced RISC machine) CPUs (central processing units). The experimental results indicate that our method achieves practical acceleration with minimal performance loss compared to the vanilla Winograd transformation.

Proposed method. To efficiently integrate Winograd transformation and network pruning techniques for practical acceleration, our method follows three main steps.

First, we convert 3D convolutional layers into 3D Winograd layers using the Winograd transformation. As analyzed, weight updates in the Winograd domain should focus on principal directions within the Winograd space. Consequently, we introduce the low-rank Winograd transformation (illustrated on the right part of Figure 1(a)). This approach simplifies weight updates by representing the transition from a pre-trained Winograd weight tensor to a fine-tuned one with two smaller matrices. By strategically updating weights along the principal directions during sparsity training, this method significantly reduces the number of trainable parameters in the Winograd domain and outperforms the vanilla Winograd transformation (shown in the left part of Figure 1(a)).

Next, we present a low-rank-oriented sparse granularity that is highly acceleration-friendly. This granularity leverages the position-sensitive characteristics of Winograd

* Corresponding author (email: wylin@sjtu,edu.cn)

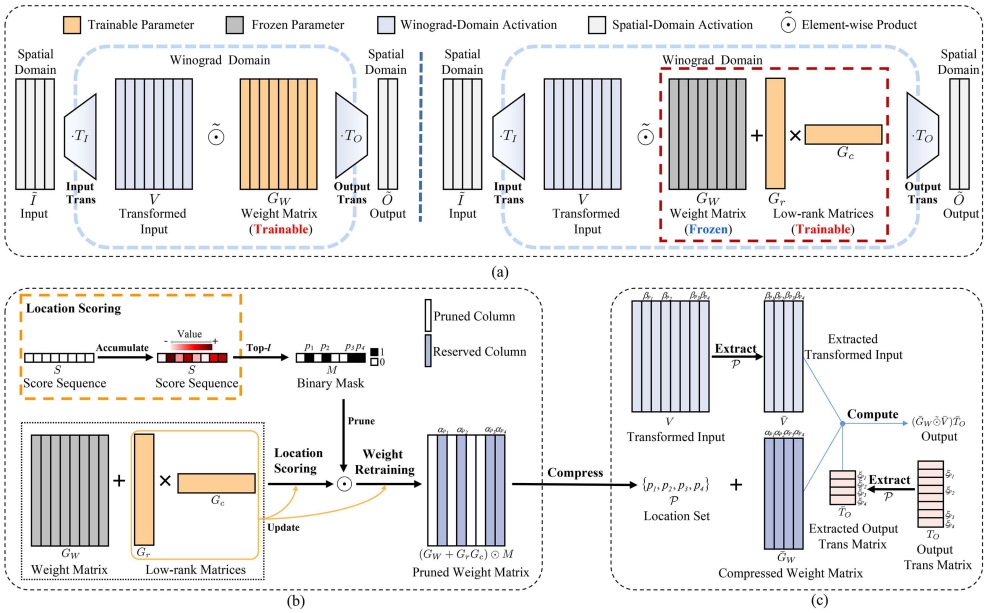

Figure 1 (Color online) (a) Comparison between the vanilla Winograd transformation (left) and our low-rank Winograd transformation (right). We freeze G_W and optimize the whole Winograd weights by two low-rank matrices G_r and G_c, leading to a significant reduction in trainable parameters. (b) Sparse training of our low-rank oriented sparse granularity. (c) Dense inference after applying our low-rank oriented sparse granularity.

weights to establish a regularized column sparsity pattern. As depicted in Figure 1(b), we introduce a scoring sequence that quantifies the significance of weight columns by accumulating the magnitude and gradient of column locations during each training iteration. After thorough evaluation, only the most important weight columns are retained, while the rest are pruned. The model is then retrained efficiently using only the updated low-rank matrices, completing the training process for the regularized sparse Winograd model.

Finally, as shown in Figure 1(c), we compress the pruned model by retaining the non-zero columns. This compression enables the model to perform dense element-wise products during inference. This strategy not only capitalizes on the inherent performance benefits of the Winograd algorithm but also effectively converts the achieved sparsity into a practical acceleration ratio. This ensures the model remains highly efficient while delivering accelerated performance.

For more detailed methods, please refer to Appendix C.

Experiments. Our methodology is applied to 3D CNN models consisting of plentiful standard 3D convolution layers. We replace all these layers with 3D Winograd layers and apply the low-rank Winograd transformation on these layers for sparse training.

We validated the effectiveness of our proposed method from two perspectives: reduction in computational load and practical speedup. First, we compared our method with state-of-the-art pruning methods designed for 3D CNNs. Experimental results indicate that our method significantly reduces computational load while achieving superior performance. Second, when tested on mobile devices against other common acceleration methods, our approach demonstrated higher acceleration ratios without severely impacting accuracy. These results demonstrate both the theoretical and practical advantages of our method. Additional experimental results are available in Appendix D.

Conclusion. This study presents a novel low-rank Winograd transformation to address the over-parameterization issue in 3D CNNs. By decoupling the original Winograd weight matrix into two space-efficient matrices, we achieve a remarkable reduction in trainable parameters. The low-rank constraint eliminates redundant parameters and directs updates toward the principal directions of the Winograd space, resulting in improved performance.

In addition, we propose low-rank-oriented sparsity to achieve effectual speedups. This approach models the column-wise importance of the Winograd weight matrix and removes the low-scoring ones. By promoting a more regular sparsity pattern, our method supports practical acceleration more effectively than irregular sparsity.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62325109, U21B2013).

Supporting information Appendixes A–F. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhang Y X, Wang H A, Luo Y, et al. Three-dimensional convolutional neural network pruning with regularization-based method. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2019. 4270–4274
2 Niu W, Sun M H, Li Z G, et al. RT3D: achieving real-time execution of 3D convolutional neural networks on mobile devices. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2021. 9179–9187
3 Wang Z L, Lan Q, He H J, et al. Winograd algorithm for 3D convolution neural networks. In: Proceedings of International Conference on Artificial Neural Networks (ICANN), 2017. 609–616
4 Lavin A, Gray S. Fast algorithms for convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 4013–4021
5 Luo J H, Wu J X, Lin W Y. ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2017. 5058–5066