BSA: Ball Sparse Attention for Large-scale Geometries

Catalin E. Brita [*1] **Hieu Nguyen** [*1] **Lohithsai Yadala Chanchu** [*1] **Domonkos Nagy** [*1] **Maksim Zhdanov** [2]

Abstract

Self-attention scales quadratically with input size, limiting its use for large-scale physical systems. Although sparse attention mechanisms provide a viable alternative, they are primarily designed for regular structures such as text or images, making them inapplicable for irregular geometries. In this work, we present Ball Sparse Attention (BSA), which adapts Native Sparse Attention (NSA) (Yuan et al., 2025) to unordered point sets by imposing regularity using the Ball Tree structure from the Erwin Transformer (Zhdanov et al., 2025). We modify NSA's components to work with ball-based neighborhoods, yielding a global receptive field at sub-quadratic cost. On an airflow pressure prediction task, we achieve accuracy comparable to Full Attention while significantly reducing the theoretical computational complexity. We open-source our implementation [1].

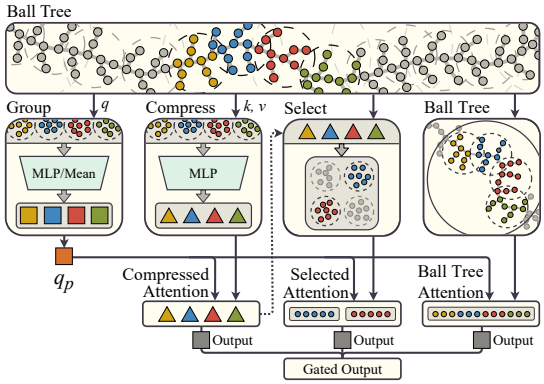

Figure 1. Ball Sparse Attention (BSA) pipeline. A ball tree imposes spatial locality, then three sparse-attention branches: grouping (clustering points for shared selection), compression (MLP-based token pooling), and selection (top-k block retrieval) operate alongside fine-grained Ball Tree Attention. A learnable gate fuses their outputs into the final attention.

1. Introduction

Scientific applications such as climate modeling (Curran et al., 2024), molecule property prediction (Pengmei et al., 2023), or fluid flow simulation (Peng et al., 2023) increasingly rely on transformer-based architectures to capture complex, long-range dependencies in irregular data (Abramson et al., 2024; Chen et al., 2025; Luo et al., 2024; Miao et al., 2024). However, standard self-attention scales quadratically with input size, making it impractical for large-scale tasks in the scientific domain. This has motivated the development of scalable strategies for large-scale physical systems.

Sparse attention mechanisms mitigate quadratic scaling by computing attention for a strategically chosen subset of token pairs. These range from predefined or random sparsity patterns (e.g., BigBird (Zaheer et al., 2020), where selection is random) to learned, data-dependent sparsity, as seen in Native Sparse Attention (NSA) (Yuan et al., 2025). NSA can select and compress tokens across the full sequence, allowing it to capture fine global dependencies.

However, NSA is designed to work with text sequences that have a regular structure, while physical systems are often defined on irregular geometries. This poses a challenge as such structures are represented as unordered sets that do not have a canonical ordering that sparse methods utilize. Many approaches (Liu et al., 2023; Sun et al., 2022; Wu et al., 2024b) induce regular structure and transform point clouds into sequences to serve as inputs for sparse attention. One such approach is Point Transformer V3 (Wu et al., 2024b), which serializes points along a space-filling curve and partitions the resulting sequence into chunks for attention computation. Another solution is Octformer (Cui et al., 2023), which uses octrees to divide point clouds into local windows, each containing a fixed number of points.

Recently proposed Erwin (Zhdanov et al., 2025) organizes points into a ball tree: the leaf level represents the full sequence, and balls in higher levels of the tree represent

[*]Equal contribution [1]Informatics Institute, University of Amsterdam [2]AMLab, University of Amsterdam. Correspondence to: <{catalin.brita, hieu.nguyen, lohithsai.yadala.chanchu, domonkos.nagy2}@student.uva.nl, m.zhdanov@uva.nl>.

Proceedings of the 2nd Workshop on Long-Context Foundation Models, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

[1]https://github.com/britacatalin/bsa

larger neighborhoods. Erwin enables linear-time attention by processing nodes in parallel within local neighborhoods of fixed size. It combines fine-grained local attention with progressive pooling for capturing global interactions. This approach performs well at local interactions but may require several steps for distant ones, as accumulating global information requires multiple layers. The progressive coarsening of such hierarchical methods also results in a loss of fidelity, as coarsened features cannot be processed at finer scales.

To address these issues, we introduce Ball Sparse Attention (BSA), which incorporates Ball Tree Attention within NSA's framework to achieve a global receptive field at a sub-quadratic cost without sacrificing accuracy.

Our contributions are:

1. A hybrid architecture integrating Ball Tree Attention into NSA's framework for scalable scientific modeling.
2. A locality-based sparsification strategy for attention that preserves modeling capacity while reducing computation compared to Full Attention.
3. A comprehensive validation of Ball Sparse Attention on airflow pressure modeling and stress field prediction in hyperelastic materials.

2. Methodology

2.1. Background

The **self-attention mechanism** builds on the scaled dot-product attention (Vaswani et al., 2017). For an input matrix $X \in \mathbb{R}^{N\times C}$, X is projected into queries, keys, and values:

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v, \tag{1}$$

where $W_q,\ W_k\ W_v \in \mathbb{R}^{C\times d_k}$ are learnable weight matrices. Then, the attention output is computed as:

$$\mathrm{Attn}(Q,K,V) = \mathrm{softmax}\Big(\frac{QK^T}{\sqrt{d_k}} + \mathcal{B}\Big)\,V, \tag{2}$$

with $\mathcal{B} \in \mathbb{R}^{N\times N}$ being the bias term. Even with highly optimized implementations (e.g., Dao et al., 2022), self-attention scales quadratically with the sequence length, making it hard to process sequences longer than tens of thousands of tokens. Recent studies have attempted to address this by imposing geometric or sparsity-based relaxations.

Ball Tree Attention: Zhdanov et al. (2025) partition the input sequence into disjoint balls B of size m, each having m feature vectors $X_B \in \mathbb{R}^{m\times C}$. Then, Ball Tree Attention (BTA) applies scaled-dot-product attention *within* each ball:

$$X'_B = \mathrm{Attn}^{\mathrm{ball}}(X_B) := Attn\left(X_BW_q, X_BW_k, X_BW_v\right), \tag{3}$$

where the weights are shared across balls and maintain row correspondence with X_B.

Native Sparse Attention (NSA): Yuan et al. (2025) preserves full sequence resolution by sparsifying attention in three branches: (1) *compression*, (2) *selection*, and (3) *sliding window*. These branches combine via gated attention:

$$\mathrm{Attn} = \sum_{b\in\{\mathrm{sld},\mathrm{cmp},\mathrm{slc}\}} \sigma(\gamma_b)\odot Attn^b \tag{4}$$

where each gate γ_b is passed through a sigmoid function $\sigma(\cdot)$ and used to modulate its branch output Attn^b.

Compression: NSA splits K and V into non-overlapping blocks of length ℓ (stride $=\ell$) and maps each block to a single coarse token via an MLP ϕ (or via mean pooling):

$$\begin{aligned} K^{\mathrm{cmp}} &= \left\{\phi\big(K_{(i-1)\ell:i\ell-1}\big)\right\}_{i=1}^{\lceil N/\ell\rceil}, \\ V^{\mathrm{cmp}} &= \left\{\phi\big(V_{(i-1)\ell:i\ell-1}\big)\right\}_{i=1}^{\lceil N/\ell\rceil}, \end{aligned} \tag{5}$$

where $K_{(i-1)\ell:i\ell-1}, V_{(i-1)\ell:i\ell-1} \in \mathbb{R}^{\ell\times d_k}, \forall i \le \lceil N/\ell\rceil$ (0-padded), and $\phi(\cdot)$ outputs a vector in $\mathbb{R}^{d_k}$. The *compressed attention* is $\mathrm{Attn}^{\mathrm{cmp}} = \mathrm{Attn}(Q, K^{\mathrm{cmp}}, V^{\mathrm{cmp}})$.

Selection: For each query position t, we reuse the coarse keys to build the dot-product similarity (importance) matrix:

$$S = Q(K^{\mathrm{cmp}})^T \in \mathbb{R}^{N\times\lceil N/\ell\rceil}, \quad S_{tj} = \langle q_t, k_j^{\mathrm{cmp}}\rangle \tag{6}$$

The indices of the top $k^\star$ blocks are then selected for each t:

$$\mathcal{I}_t = \text{top-k}(S_{t,:}, k^\star) \subset \{1,\cdots,\lceil N/\ell\rceil\} \tag{7}$$

The selected blocks are then converted back to token-level representations by concatenating their respective KV tokens:

$$\begin{aligned} K_t^{\mathrm{slc}} &= \mathrm{Cat}\Big\{K_{(i-1)\ell:i\ell-1} \mid i\in\mathcal{I}_t\Big\} \\ V_t^{\mathrm{slc}} &= \mathrm{Cat}\Big\{V_{(i-1)\ell:i\ell-1} \mid i\in\mathcal{I}_t\Big\} \end{aligned} \tag{8}$$

The *selection attention* is $\mathrm{Attn}^{\mathrm{slc}} = \mathrm{Attn}(Q, K^{\mathrm{slc}}, V^{\mathrm{slc}})$.

Despite attending to only a fraction of possible pairs, NSA matches or outperforms Full Attention on NLP tasks and achieves an $11\times$ speedup in computation (Yuan et al., 2025).

2.2. Ball Sparse Attention

Ball Sparse Attention (BSA): We propose BSA, an attention mechanism that inherits NSA's three-branch design: *compression*, *selection*, and *local attention*. However, we replace the conventional sliding window with Ball Tree Attention (BTA) (Zhdanov et al., 2025) (see BTA in Figure 1 when $q_p = q_t$), allowing the model to attend within continuous geometric regions in $\mathbb{R}^D$ and avoid artificial discontinuities. The compression and selection branches remain as in NSA, and we combine all three branches:

$$\mathrm{Attn} = \sum_{b\in\{\mathrm{ball},\mathrm{cmp},\mathrm{slc}\}} \sigma(\gamma_b)\odot Attn^b. \tag{9}$$

Group selection: Beyond locality in BTA, we also exploit locality during *selection*, in the top-k computation. To achieve this, we group query positions t into contiguous groups of size g (see Groping in Figure 1), and enforce one set of selected blocks across all queries in the same group:

$$G_p = \{(p-1)g, \ldots, pg-1\}, \quad p = 1, \ldots, \lceil N/g \rceil, \tag{10}$$

where p indexes each group and G_p is the set of indices in the p-th group. We also pool the queries in each group:

$$q_p = \frac{1}{|G_p|} \sum_{t \in G_p} Q_t \tag{11}$$

We then average the similarity scores within a group and perform top-k selection on these averages:

$$\begin{aligned} \bar{S}_{pj} &= \frac{1}{|G_p|} \sum_{t \in G_p} S_{tj}, \quad j = 1, \ldots, \lceil N/\ell \rceil, \\ \mathcal{I}_p &= \text{top-k}(\bar{S}_{p,\cdot}, k^\star), \quad \mathcal{I}_t \equiv \mathcal{I}_p \;\; \forall t \in G_p. \end{aligned} \tag{12}$$

This reduces top-k calls by a factor of g and allows KV blocks to be fetched in contiguous chunks, improving GPU cache utilisation and lowering memory-access latency.

To further optimize the runtime, we shrink the dimensionality of the query before the *selection* stage by coarsening Q with an operation ϕ before computing the similarity matrix. ϕ can either be mean pooling or an MLP:

$$Q^{\text{cmp}} = \{\, \phi\big(Q_{(i-1)\ell:i\ell-1}\big) \}_{i=1}^{\lceil N/\ell \rceil}, \tag{13}$$

with $Q_{(i-1)\ell:i\ell-1}, \forall i \le \lceil N/\ell \rceil$ (zero-padded). The smaller similarity matrix and its selection step are then:

$$\begin{aligned} \widetilde{S} &= Q^{\text{cmp}} (K^{\text{cmp}})^T \in \mathbb{R}^{\lceil N/\ell \rceil \times \lceil N/\ell \rceil}, \\ \widetilde{\mathcal{I}}_p &= \text{top-k}(\widetilde{S}_{p,\cdot}, k^\star) \subset \{1, \cdots, \lceil N/\ell \rceil\}, \end{aligned} \tag{14}$$

after which we proceed exactly as before. This yields a small loss in resolution but improves similarity computation.

Group compression: Finally, we down-sample queries not only for *selection* but also for *compression* itself:

$$\text{Attn}^{\text{cmp}} = \underbrace{(I_{\lceil N/\ell \rceil} \otimes \mathbf{1}_\ell)}_{\text{repeat}} \text{Attn}\Big(Q^{\text{cmp}}, K^{\text{cmp}}, V^{\text{cmp}}\Big), \tag{15}$$

where $I_{\lceil N/\ell \rceil} \otimes \mathbf{1}_\ell$ repeats each attention output ℓ times. By compressing the queries in both *selection* and *compression*, this variant offers the greatest speed-up at the price of a coarser attention map and a drop in downstream accuracy.

3. Experiments and Results

3.1. Experiments setup

Task We evaluate the proposed architecture on an airflow pressure modeling task that requires predicting pressure values at discrete points on three-dimensional geometric structures. We use the ShapeNet-Car dataset (Umetani & Bickel, 2018) with 889 car models, each modeled with 3586 surface points in 3D space. The ground truth airflow pressure is generated through simulations with the Reynolds number = 5×10^6. The data is split into 700 training and 189 testing samples. To test the generalization of our method, we further evaluate with the stress field prediction task on Elasticity benchmark (Li et al., 2021).

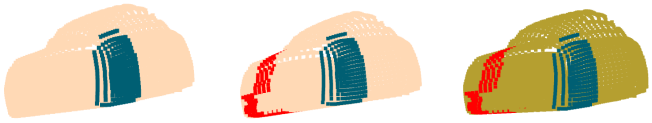

Figure 2. Receptive field visualization of a car in the Shapenet dataset with different components: ball attention; ball attention and selection; ball attention, selection and compression. The receptive field increases with more components.

Training details The model consists of 18 transformer blocks, each containing an RMSNorm layer (Zhang & Sennrich, 2019), a Ball Sparse Attention layer (BSA), and a SwiGLU (Shazeer, 2020) for non-linear transformation. The BSA parameters and training hyperparameters are detailed in Appendix A. For the query coarsening operation, mean pooling was used for regular BSA, and MLP was used when group compression is used with BSA. To measure FLOPs, we use Deepspeed FLOPs profiler [2].

3.2. Experimental results

Table 1. ShapeNet MSE test performance compared with previous methods.

Model	MSE
PointNet (2016)	43.36
GINO (2023a)	35.24
UPT (2024)	31.66
Transolver (2024a)	19.88
PTv3 (2024c)	19.09
GP-UPT (2025)	17.02
Erwin (2025)	15.85
BSA (Ours)	14.31
Full Attention (2017)	13.29

Table 2. Elasticity RMSE compared to previous works. All results are multiplied by 10^2.

Model	RMSE
LSM (2023)	2.18
LNO (2024)	0.69
Oformer (2023b)	1.83
Gnot (2023)	0.86
Ono (2024)	1.18
Transolver (2024a)	0.64
Erwin (2025)	0.34
BSA (Ours)	0.38
Full Attention (2017)	0.30

BSA Comparison with previous methods: Table 1 shows that BSA outperforms all previous methods, and has

[2] https://github.com/deepspeedai/DeepSpeed

1.02 MSE worse than Full Attention on ShapeNet. On the stress field prediction task, Table 2 shows that BSA achieves approximately the same performance as Erwin (Zhdanov et al., 2025). This outcome can be attributed to the small-scale nature of the Elasticity dataset, with sequence lengths of 972, where BSA fails to demonstrate a clear advantage.

Table 3. Comparison of the Full Attention, BSA, and Erwin in terms of MSE, runtime (ms), and FLOPS on ShapeNet.

Attention type	MSE	Runtime (ms)	GFLOPS
Erwin	16.12	19.35	14.60
Full Attention	13.29	37.82	87.08
BSA	14.31	36.53	27.91
BSA w/o group selection	14.44	66.92	32.67
BSA w group compression	14.80	23.42	20.82

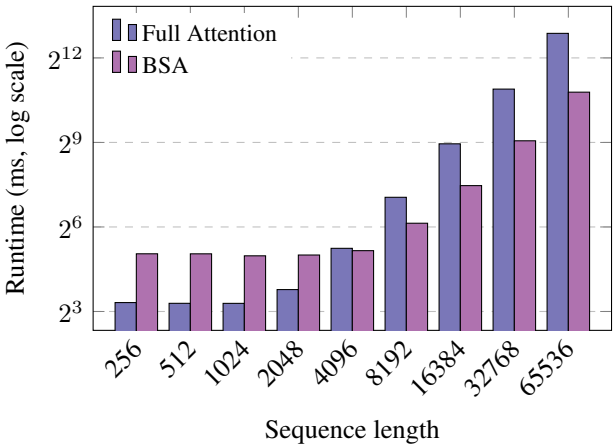

Figure 3. Runtime of BSA and Full attention with increasing sequence length (from 256 to 65536).

Accuracy-Efficiency trade-off: Table 3 shows that all BSA variants achieve significantly better results than Erwin (Zhdanov et al., 2025) and approach the accuracy of Full Attention (Vaswani et al., 2017). Although these variants have higher GFLOPs than Erwin, they require fewer GFLOPs than Full Attention. However, when measuring the runtime on ShapeNet with a sequence length of 4096, Full Attention outperforms BSA without group selection. This is because, unlike NSA (Yuan et al., 2025), we do not implement a Triton kernel for efficient selection.

Table 3 also illustrates the trade-off between accuracy and computational efficiency across different BSA variants. While BSA without group selection has higher GFLOPS and runtime, the difference in MSE is marginal (only 0.13). This supports our assumption about the importance of local structures in physical tasks, showing that efficiency can be improved without a significant change in accuracy. Furthermore, a preliminary analysis of how the compressed block size ℓ and the group selection size $|G_p|$ influence ShapeNet performance is provided in Appendix B.

BSA scaling Figure 3 shows the runtime scaling of our BSA with sequence length compared to Full Attention. At smaller sequence lengths, full attention exhibits faster runtime due to the computational overhead at the MLP of BSA. However, as the sequence length increases, the gap narrows, with BSA outperforming Full Attention at 4096 sequence length. At the longest sequence length of 65536, BSA is 5× faster than full attention. Appendix C provides a detailed scaling comparison of the BSA variants. This highlights the superior computational efficiency of our BSA at scale.

BSA receptive field Figure 2 demonstrates the receptive field of each individual component in BSA. Initially, Ball Tree Attention (BTA) only attends to points inside a local ball. With selection, it can attend to different blocks far away from the current point. Additionally, to prevent the model from over-fitting to local patterns already captured by the BTA, we mask all blocks within each ball, encouraging it to attend to more distant regions. Finally, with compression, it achieves a global receptive field by attending to all coarse key–value representations of each local block.

4. Conclusion

We introduce BSA, a novel sparse attention mechanism suited for large scale physical systems. Our approach integrates Ball Tree Attention within the NSA framework and introduces a locality-based sparsification strategy that preserves the performance of high-quality long-range modeling while drastically reducing computational efficiency. Through grouped selection and hybrid attention, we preserve local and global context with minimal overhead.

Our results show that BSA achieves accuracy competitive with both Full Attention and the Erwin Transformer (Zhdanov et al., 2025). Notably, at larger sequence lengths, BSA is up to five times faster than Full Attention. Its hybrid attention mechanism expands the receptive field compared to Erwin, enabling better long-range information propagation. Given its efficiency and performance at scale, BSA is ideal for simulating large-scale physical systems.

Future work: In the future, we will evaluate our fixed-group query partitioning scheme on a broad spectrum of point-cloud datasets to assess its robustness across domains, conduct comprehensive hyperparameter sweeps and ablation studies to quantify BSA's sensitivity to tuning. Moreover, we plan to develop a GPU kernel for BSA operations to fully make use of NSA's hardware alignment for improved computational efficiency and reduced memory footprint.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL http://dx.doi.org/10.1038/s41586-024-07487-w.

Alkin, B., Fürst, A., Schmid, S., Gruber, L., Holzleitner, M., and Brandstetter, J. Universal physics transformers. *arXiv preprint arXiv:2402.12365*, 2024.

Bleeker, M., Dorfer, M., Kronlachner, T., Sonnleitner, R., Alkin, B., and Brandstetter, J. Neuralcfd: Deep learning on high-fidelity automotive aerodynamics simulations. *CoRR*, abs/2502.09692, 2025. URL https://doi.org/10.48550/arXiv.2502.09692.

Chen, S., Fang, Z., Wan, S., Zhou, T., Chen, C., Wang, M., and Li, Q. Geometrically aware transformer for point cloud analysis. *Scientific Reports*, 15(1), May 2025. ISSN 2045-2322. doi: 10.1038/s41598-025-00789-7. URL http://dx.doi.org/10.1038/s41598-025-00789-7.

Cui, M., Long, J., Feng, M., Li, B., and Kai, H. Octformer: Efficient octree-based transformer for point cloud compression with local enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:470–478, 06 2023. doi: 10.1609/aaai.v37i1.25121.

Curran, D., Saleem, H., Hobeichi, S., and Salim, F. Resolution-agnostic transformer-based climate downscaling. In *NeurIPS 2024 Workshop on Tackling Climate Change with Machine Learning*, 2024. URL https://www.climatechange.ai/papers/neurips2024/74.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with i/o-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.

Hao, Z., Ying, C., Wang, Z., Su, H., Dong, Y., Liu, S., Cheng, Z., Zhu, J., and Song, J. Gnot: A general neural operator transformer for operator learning. *arXiv preprint arXiv:2302.14376*, 2023.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations, 2021. URL https://arxiv.org/abs/2010.08895.

Li, Z., Kovachki, N. B., Choy, C. B., Li, B., Kossaifi, J., Otta, S. P., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., and Anandkumar, A. Geometry-informed neural operator for large-scale 3d pdes. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a.

Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023b. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

Liu, Z., Yang, X., Tang, H., Yang, S., and Han, S. Flatformer: Flattened window attention for efficient point cloud transformer. pp. 1200–1211, 06 2023. doi: 10.1109/CVPR52729.2023.00122.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Luo, Z., Zeng, Z., Wan, J., Tang, W., Jin, Z., Xie, Z., and Xu, Y. D2t-net: A dual-domain transformer network exploiting spatial and channel dimensions for semantic segmentation of urban mobile laser scanning point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 132:104039, 2024. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2024.104039. URL https://www.sciencedirect.com/science/article/pii/S1569843224003935.

Miao, S., Lu, Z., Liu, M., Duarte, J., and Li, P. Locality-sensitive hashing-based efficient point transformer with applications in high-energy physics. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Peng, W., Yuan, Z., Li, Z., and Wang, J. Linear attention coupled fourier neural operator for simulation of three-dimensional turbulence. *Physics of Fluids*, 35(1):015106, 01 2023.

Pengmei, Z., Li, Z., chan Tien, C., Kondor, R., and Dinner, A. R. Transformers are efficient hierarchical chemical graph learners, 2023. URL https://arxiv.org/abs/2310.01704.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

Shazeer, N. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020. URL https://arxiv.org/abs/2002.05202.

Sun, P., Tan, M., Wang, W., Liu, C., Xia, F., and Leng, Z. *SWFormer: Sparse Window Transformer for 3D Object Detection in Point Clouds*, pp. 426–442. 11 2022. ISBN 978-3-031-20079-3. doi: 10.1007/978-3-031-20080-9_25.

Umetani, N. and Bickel, B. Learning three-dimensional flow for interactive aerodynamic design. *ACM Trans. Graph.*, 37(4), July 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6000–6010, 2017. URL https://proceedings.neurips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, T. and Wang, C. Latent neural operator for solving forward and inverse pde problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wu, H., Hu, T., Luo, H., Wang, J., and Long, M. Solving high-dimensional pdes with latent spectral models. In *International Conference on Machine Learning*, 2023.

Wu, H., Luo, H., Wang, H., Wang, J., and Long, M. Transolver: A fast transformer solver for pdes on general geometries. In *International Conference on Machine Learning*, 2024a.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., and Zhao, H. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024b.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., and Zhao, H. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024c.

Xiao, Z., Hao, Z., Lin, B., Deng, Z., and Su, H. Improved operator learning by orthogonal attention. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

Yuan, J., Gao, H., Dai, D., Luo, J., Zhao, L., Zhang, Z., Xie, Z., Wei, Y. X., Wang, L., Xiao, Z., Wang, Y., Ruan, C., Zhang, M., Liang, W., and Zeng, W. Native sparse attention: Hardware-aligned and natively trainable sparse attention, 2025. URL https://arxiv.org/abs/2502.11089.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhang, B. and Sennrich, R. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019. URL https://openreview.net/references/pdf?id=S1qBAf6rr.

Zhdanov, M., Welling, M., and van de Meent, J.-W. Erwin: A tree-based hierarchical transformer for large-scale physical systems, 2025. URL https://arxiv.org/abs/2502.17019.

A. Training hyperparameters

Table 4 details the sparse attention hyperparameters. The model is trained with mean squared error loss for the regression task, and it is trained for 100000 iterations with AdamW optimizer (Loshchilov & Hutter, 2019) with a cosine learning rate scheduler, learning rate 0.001, and weight decay 0.01.

Table 4. Sparse attention parameters

Parameter	Value
Ball size	256
Compression block size	8
Compression block sliding stride	8
Selection block size	8
Number of blocks selected	4

B. Block size ablations on ShapeNet

Table 5. BSA test MSE for various compression and group selection sizes on ShapeNet. We use k=4 (for top-k) with mean pooling.

Compr. Block Size	Group Selection Size	Val. MSE
4	4	15.43
8	8	14.31
16	16	14.97
32	32	132.14
4	8	14.81
16	8	14.88
8	4	14.88
8	16	14.84

C. Runtime scaling analysis

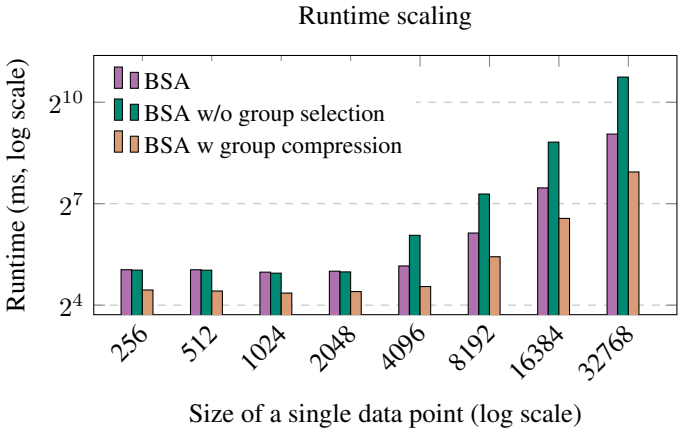

Figure 4. Runtime of BSA and its variants with increasing sequence length (from 256 to 32768).