TUCKER BOTTLENECK ATTENTION FOR MULTI-DIMENSIONAL SEQUENCE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

The quadratic cost of self-attention limits scalability to long sequences from multidimensional data. We introduce Tucker bottleneck attention (TuBA), which exploits low-rank tensor structure for efficient global token mixing. TuBA projects hidden tensors into compact Tucker cores, performs multi-head self-attention and linear projections on the cores, and writes updates back to the ambient space, enabling subquadratic computation. Its autoregressive extension combines bidirectional interactions within cores with causal attention across cores. On video prediction and global weather forecasting, TuBA achieves favorable accuracy–efficiency trade-offs over standard and efficient attention and task-specific models. Compared to standard self-attention, TuBA reduces error and computation by up to 24.7% and 66.6% for video prediction and 37.1% and 85.1% for autoregressive weather forecasting, with speedups up to $4.27\times$. Low-rank Tucker cores and multi-frame generation also outperform full-rank attention and frame-by-frame generation, respectively.

1 INTRODUCTION

Self-attention, the primary token-mixing mechanism in transformers (Vaswani et al., 2017), has driven major advances in machine learning (Devlin et al., 2019; Brown et al., 2020; Dosovitskiy et al., 2021; Nguyen et al., 2023), but its quadratic cost limits scalability to long sequences. Consequently, extensive work has sought to improve attention efficiency (Tay et al., 2022). Some methods restrict attention to local or sparse patterns (Liu et al., 2021; Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Ho et al., 2019), while others replace softmax with kernel feature maps to avoid forming the attention matrix (Katharopoulos et al., 2020; Qin et al., 2022; Choromanski et al., 2021; Peng et al., 2021), or compute attention using a smaller set of latent tokens (Lee et al., 2019; Ma et al., 2021; Jaegle et al., 2021). Other approaches exploit low-rank attention structure. Linformer (Wang et al., 2020), for example, projects keys and values into a low-dimensional space to achieve linear attention complexity, while related methods use low-rank reconstructions (Xiong et al., 2021; Chen et al., 2021b;a; Omranpour et al., 2025; Feng et al., 2024). These methods primarily rewrite or approximate self-attention itself; in contrast, exploiting low-rank tensor structure in the representations to reduce attention cost remains underexplored.

Modeling multidimensional data is central to scientific applications (Kolda & Bader, 2009), from video prediction and world modeling (Gao et al., 2022a; Bruce et al., 2024) to weather and climate forecasting (Gao et al., 2022b; Nguyen et al., 2023). Such data exhibit interactions within and across modes that models can exploit (Holmes et al., 2012; Hannachi et al., 2007; Brunton & Kutz, 2019; Solgi et al., 2025). Existing architectures, however, face tradeoffs between efficiency and global token mixing. Recurrent models limit parallelization through sequential processing (Shi et al., 2015; Wang et al., 2017; 2018b; 2019; Chang et al., 2021), while convolutional models rely on local receptive fields (Gao et al., 2022a; Tan et al., 2023a). Likewise, local and factorized attention restrict direct interactions to local regions or individual axes (Liu et al., 2021; Gao et al., 2022b; Ho et al., 2019; Bertasius et al., 2021; Arnab et al., 2021; Tang et al., 2026). Yet multidimensional data can exhibit long-range, cross-mode interactions, motivating global token mixing (Wallace & Gutzler, 1981; Wang et al., 2018a; Lam et al., 2023). Full attention provides global mixing but incurs quadratic cost and lacks structural priors (Dosovitskiy et al., 2021; d'Ascoli et al., 2021); moreover, its attention can become increasingly diffuse with sequence length (Nakanishi, 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

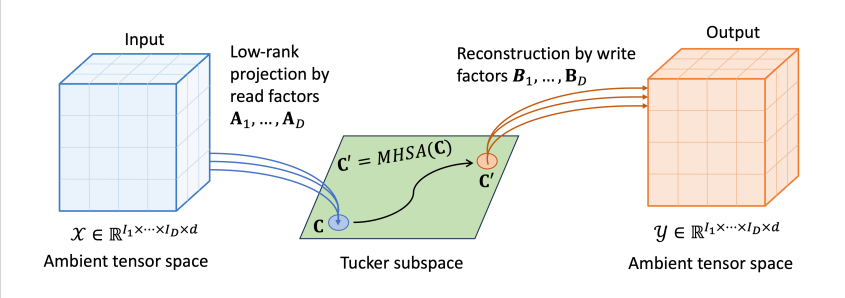

Figure 1: Overview of Tucker bottleneck attention. TuBA projects a tensor into a compact Tucker core, applies attention on the core, then reconstructs the output.

Veličković et al., 2025; Jin et al., 2023). Efficient global attention (Katharopoulos et al., 2020; Wang et al., 2020; Xiong et al., 2021) reduces computation but still lacks structural priors and can sacrifice accuracy (Tay et al., 2022). We propose that the low-rank tensor structure of hidden representations provides an efficient inductive bias for global token mixing.

Transformer hidden representations exhibit approximate low-rank structure and lie on low-dimensional manifolds (Yu & Wu, 2023; Modell et al., 2025). For high-order representations, this structure can be exploited through tensor decomposition. Using Tucker factorization, we observe that hidden representations largely share a common subspace across samples: shared mode-wise factors with compact per-sample cores achieve nearly the same reconstruction accuracy as sample-specific decompositions (Appendix A). This observation motivates *Tucker bottleneck attention (TuBA)*, which projects a multidimensional hidden tensor into a compact Tucker core using learned mode-wise read factors, applies standard multi-head self-attention (MHSA) within the core, and writes the update back to the original tokens (Figure 1). TuBA thereby enables global token mixing at reduced computational cost.

Our main contributions are summarized below:

- **A high-order tensor view for efficient attention.** We introduce TuBA, which exploits low-rank tensor representations to perform global token mixing in a compact Tucker core at subquadratic cost. We further extend TuBA to autoregressive tensor generation through causal attention.
- **Interpretation and expressivity of TuBA.** We show that TuBA's learned read projection yields a unique least-squares Tucker core for associated factors, while its write projection produces a Tucker-structured update in a potentially distinct output subspace. TuBA also contains standard self-attention as a special case.
- **Experimental results.** Across video and weather benchmarks, TuBA achieves favorable accuracy–efficiency trade-offs over standard and efficient attention methods and task-specific models. Compared with standard self-attention, TuBA reduces video prediction MSE by up to 24.7% and FLOPs by up to 66.6%, while reducing autoregressive weather normalized RMSE by 37.1% with a $4.27\times$ speedup. Ablations further show that intermediate Tucker ranks and multi-frame generation outperform full-rank attention and frame-by-frame generation, respectively.

2 Background

2.1 Multi-head self-attention

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ contain N tokens of hidden dimension d. Multi-head self-attention (Vaswani et al., 2017) with H heads of width d_h is

$$\mathrm{MHSA}(\mathbf{X}) = \sum_{i=1}^{H} \underbrace{\mathrm{softmax}\Big(\tfrac{1}{\sqrt{d_h}}\,\mathbf{X}\mathbf{W}_Q^i\big(\mathbf{W}_K^i\big)^\top\mathbf{X}^\top\Big)}_{\mathbf{S}_i(\mathbf{X}) \,\in\, \mathbb{R}^{N\times N}}\ \mathbf{X}\ \underbrace{\mathbf{W}_V^i\mathbf{W}_O^i}_{\mathbf{M}_i \,\in\, \mathbb{R}^{d\times d}}, \tag{1}$$

where softmax acts row-wise, $\mathbf{W}_Q^i, \mathbf{W}_K^i, \mathbf{W}_V^i \in \mathbb{R}^{d\times d_h}$ are the learned query, key, and value projection matrices, and $\mathbf{W}_O^i \in \mathbb{R}^{d_h\times d}$ is the output-projection block for head i. Thus, each head applies the state-dependent operator $\mathbf{S}_i(\mathbf{X})$ along the token dimension and the fixed linear operator $\mathbf{M}_i$ along the channel dimension.

2.2 TUCKER REPRESENTATION

Let a rank-$(r_1, \ldots, r_D)$ least-squares Tucker approximation (Tucker, 1966) of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ be

$$\boldsymbol{\mathcal{X}} \approx \hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_D \mathbf{U}_D, \tag{2}$$

where $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{r_1 \times \cdots \times r_D}$ is the core, $\mathbf{U}_n \in \mathbb{R}^{I_n \times r_n}$, $n = 1, \ldots, D$, are full-column-rank factor matrices, and $\times_n$ denotes multiplication along mode n. For the given factor matrices, the core is uniquely obtained as

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{X}} \times_1 \bar{\mathbf{U}}_1^\top \times_2 \cdots \times_D \bar{\mathbf{U}}_D^\top, \tag{3}$$

where $\bar{\mathbf{U}}_n^\top = \mathbf{U}_n^\dagger$ is the Moore–Penrose pseudoinverse of $\mathbf{U}_n$, and $\bar{\mathbf{U}}_n = \mathbf{U}_n$ for orthonormal factors.

3 METHOD

We first introduce the TuBA layer and establish its connection to Tucker representations. We then examine its induced token mixing, extend it to autoregressive tensor generation, and describe its computational cost and integration into a transformer.

3.1 TUCKER BOTTLENECK ATTENTION

Tucker bottleneck attention (TuBA) performs global token mixing using a Tucker core rather than across all input tokens. It projects a multidimensional hidden tensor into a compact core using learned mode-wise factor matrices, applies standard multi-head self-attention to the core, and maps the resulting update back to the original tensor. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D \times d}$ be a hidden tensor, where the first D modes index the native token axes and d is the hidden dimension. For example, in spatiotemporal prediction, after the inputs are tokenized, the hidden representation is $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{F \times P \times Q \times d}$ with $D = 3$, where F indexes frames, P and Q index the spatial token grid, and each token carries a d-dimensional feature vector. For ranks $(r_1, \ldots, r_D)$ with $r_n \leq I_n$, TuBA learns read and write factors $\mathbf{A}_n, \mathbf{B}_n \in \mathbb{R}^{I_n \times r_n}$ for each token axis $n = 1, \ldots, D$. It first reads the hidden representation into a compact Tucker core:

$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{A}_1^\top \times_2 \cdots \times_D \mathbf{A}_D^\top \in \mathbb{R}^{r_1 \times \cdots \times r_D \times d}. \tag{4}$$

Because the projection acts mode-wise, the core is a compact representation that preserves the multidimensional structure of the input, with each mode's resolution set by its own rank. Flattening the first D modes of $\boldsymbol{\mathcal{C}}$ gives $\mathbf{C} \in \mathbb{R}^{R \times d}$, where $R = \prod_{n=1}^{D} r_n$ is the number of core slots. TuBA applies standard multi-head self-attention to these slots:

$$\mathbf{C}' = \mathrm{MHSA}(\mathbf{C}) \in \mathbb{R}^{R \times d}. \tag{5}$$

Since non-causal MHSA is permutation equivariant, and any core positional encoding is indexed by core coordinates, the flattening order does not affect the result; hence, reshaping $\mathbf{C}'$ gives $\boldsymbol{\mathcal{C}}' \in \mathbb{R}^{r_1 \times \cdots \times r_D \times d}$. The output of the TuBA layer, $\boldsymbol{\mathcal{Y}}$, is obtained by mapping this tensor back to the ambient tensor space through the learned write factors:

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{C}}' \times_1 \mathbf{B}_1 \times_2 \cdots \times_D \mathbf{B}_D \in \mathbb{R}^{I_1 \times \cdots \times I_D \times d}. \tag{6}$$

TuBA contains standard MHSA as a full-rank special case: when $r_n = I_n$ and $\mathbf{A}_n = \mathbf{B}_n = \mathbf{I}$ for every n, it reduces exactly to MHSA over the N flattened tokens. The following proposition connects the projection to a least-squares Tucker core and characterizes the structure of the attended update.

Proposition 1 (Interpretation of TuBA)**.** *For full-column-rank read factors $\mathbf{A}_n$, define*

$$\mathbf{U}_n = \mathbf{A}_n (\mathbf{A}_n^\top \mathbf{A}_n)^{-1}.$$

The TuBA read core in Eq. (4) *is the unique least-squares Tucker core associated with the factors $\mathbf{U}_1, \ldots, \mathbf{U}_D$. Reconstruction through these factors gives the orthogonal projection of $\boldsymbol{\mathcal{X}}$ onto the tensor-product subspace determined by the read factors. For any transformed core, the TuBA update in Eq.* (6) *has an exact Tucker representation with mode-n rank at most r_n for each token mode.*

See Appendix B for proof.

3.2 TuBA TOKEN MIXING

To clarify how core attention induces interactions among the original tokens and the distinct roles of the read and write factors, we express the read, attend, and write operations as a single token-mixing map. Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ denote the unfolding of the first D modes of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_D\times d}$, with the mode-1 index varying fastest and $N = \prod_{n=1}^{D} I_n$. Collecting the mode-wise factors gives the Kronecker-product read and write matrices

$$\mathbf{A} = \mathbf{A}_D \otimes \cdots \otimes \mathbf{A}_1 \in \mathbb{R}^{N\times R}, \qquad \mathbf{B} = \mathbf{B}_D \otimes \cdots \otimes \mathbf{B}_1 \in \mathbb{R}^{N\times R}. \tag{7}$$

Equations (4)–(6) then take the matrix form

$$\mathrm{TuBA}(\mathbf{X}) = \mathbf{B}\,\mathrm{MHSA}(\mathbf{A}^\top\mathbf{X}). \tag{8}$$

Expanding headwise gives

$$\mathrm{TuBA}(\mathbf{X}) = \sum_{i=1}^{H} \mathbf{B}\,\widetilde{\mathbf{S}}_i(\mathbf{A}^\top\mathbf{X})\mathbf{A}^\top\mathbf{X}\mathbf{M}_i, \tag{9}$$

where $\widetilde{\mathbf{S}}_i(\mathbf{A}^\top\mathbf{X}) \in \mathbb{R}^{R\times R}$ is the attention matrix of head i on the Tucker core. Thus, TuBA replaces each head's $N \times N$ attention computation with an $R \times R$ core attention computation, retaining the standard channel-side maps $\mathbf{M}_i$. The induced token-mixing map $\mathbf{B}\widetilde{\mathbf{S}}_i(\mathbf{A}^\top\mathbf{X})\mathbf{A}^\top$ has rank at most R, with column and row spaces contained in $\mathrm{range}(\mathbf{B})$ and $\mathrm{range}(\mathbf{A})$, respectively. It is generally nonsymmetric, and its input- and output-side low-rank subspaces need not coincide. TuBA therefore learns separate read and write factors (the untied configuration). The tied special case, $\mathbf{A}_n = \mathbf{B}_n$ for all n, assumes one shared Tucker subspace suffices for reading the input and writing the attended update. Denote the corresponding single-layer function classes by $\mathcal{F}_{\mathrm{untied}}$ and $\mathcal{F}_{\mathrm{tied}}$, respectively.

Theorem 1 (Expressivity of untied read and write factors)**.** *For $N > 1$ and the same Tucker ranks, hidden dimension, and heads, with MHSA as in Eq.* (1)*, $\mathcal{F}_{\mathrm{tied}} \subsetneq \mathcal{F}_{\mathrm{untied}}$.*

See Appendix C for proof.

Theorem 1 shows that untied factors represent mappings that tied factors cannot at the same core size. Untying the factors increases expressivity without increasing TuBA's computational cost.

3.3 CAUSAL TuBA FOR AUTOREGRESSIVE TENSOR GENERATION

TuBA treats a tensor as a single computational unit. When a task requires causal or, more generally, autoregressive generation, the ordering is imposed over a sequence of tensors rather than being inherent to any individual tensor mode. Depending on the task, the choice of tensor determines the granularity at which causality is imposed. For example, a video clip may be processed non-causally as a single three-dimensional tensor, whereas causal generation can treat individual frames or contiguous blocks of frames as an ordered sequence of tensors. Let $\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_T \in \mathbb{R}^{I_1\times\cdots\times I_D\times d}$ be a sequence of tensor observations. TuBA independently reads each $\boldsymbol{\mathcal{X}}_t$ into a core $\mathbf{C}_t \in \mathbb{R}^{R\times d}$ and applies masked attention to their concatenation

$$\mathbf{C}_{1:T} = \begin{bmatrix}\mathbf{C}_1^\top & \cdots & \mathbf{C}_T^\top\end{bmatrix}^\top \in \mathbb{R}^{TR\times d}.$$

For core slots $a, b \in \{1, \ldots, R\}$ at times t and s, respectively, the causal mask is

$$\mathbf{\Lambda}_{(t,a),(s,b)} = \begin{cases} 0, & s \le t, \\ -\infty, & s > t. \end{cases} \tag{10}$$

The mask is added to the pre-softmax attention scores in Eq. (1). The usual triangular mask becomes block lower-triangular, with $R \times R$ blocks: all core slots within a tensor may interact, while each tensor attends only to itself and earlier tensors. Block-causal masking of this form has been used over raw token grids (Arriola et al., 2025; Yin et al., 2025; Meo et al., 2025); TuBA instead imposes it on low-rank Tucker cores to autoregressively generate a sequence of tensors. This construction allows the model to jointly predict the elements of the next tensor while conditioning on preceding tensors. The tensor size thus determines the granularity of each generation step, while the Tucker ranks determine the size of the representation used for attention.

3.4 COMPUTATIONAL, MEMORY, AND PARAMETER COST

Tucker core attention costs $O(R^2d+Rd^2)$ FLOPs, while the mode-wise read and write contractions are bounded by $O(N\bar{R}d)$, where $\bar{R} = \sum_{n=1}^{D} r_n$. Hence, TuBA's computational complexity is $O\big(N\bar{R}d + R^2d + Rd^2\big)$. Since $\bar{R} \le DR$, for fixed D, this complexity is asymptotically lower than the $O(N^2d+Nd^2)$ cost of standard self-attention whenever $R = o(N)$, i.e., $R/N \to 0$ as $N \to \infty$. TuBA requires $O(Nd + HR^2)$ activation memory, compared with $O(Nd + HN^2)$ for standard self-attention when attention matrices are explicitly stored; with FlashAttention (Dao et al., 2022), both require $O(Nd)$, while TuBA retains its computational advantage (see Appendix D). In terms of parameters, the read and write factors add $2\sum_{n=1}^{D} I_n r_n$ per layer, which grows with the sum of the mode sizes rather than with the number of tokens $N = \prod_n I_n$; hence, TuBA's parameter overhead remains negligible as the multidimensional input size grows.

3.5 TUBA TRANSFORMER BLOCK

TuBA replaces the multi-head self-attention sublayer of a standard pre-norm transformer block; all remaining components are unchanged. Given an input hidden tensor $\boldsymbol{\mathcal{H}}$, the TuBA sublayer is

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{H}} + \mathrm{TuBA}(\mathrm{LN}(\boldsymbol{\mathcal{H}}))\,, \tag{11}$$

where LN applies layer normalization along the hidden dimension. The resulting tensor then passes through a standard feed-forward sublayer:

$$\boldsymbol{\mathcal{H}}_{\mathrm{out}} = \boldsymbol{\mathcal{Z}} + \mathrm{FFN}(\mathrm{LN}(\boldsymbol{\mathcal{Z}}))\,. \tag{12}$$

TuBA changes only global token mixing, while normalization, residual connections, and channel-wise feed-forward computation remain those of a standard transformer block. The Tucker bottleneck constrains the attention update, while the residual path carries the incoming representation forward at its original resolution. Consequently, the rank bounds on the TuBA update do not impose the same bounds on the hidden representations passing through the network.

3.6 TRAINING AND INITIALIZATION

All TuBA parameters, including the read and write factors $\{\mathbf{A}_n, \mathbf{B}_n\}_{n=1}^{D}$, are optimized with standard backpropagation. The mode-wise contractions are differentiable, so TuBA requires neither alternating tensor-factor optimization nor a specialized training objective. The Tucker ranks $(r_1, \ldots, r_D)$ are considered to be hyperparameters and determined empirically. We initialize each $\mathbf{A}_n$ with orthonormal columns and set $\mathbf{B}_n \leftarrow \mathbf{A}_n$. The two factor sets are subsequently optimized independently. Thus, training begins from a shared Tucker basis while allowing the read and write subspaces to specialize to the task.

3.7 COMPARISON WITH PRIOR WORK

Structured token interaction methods restrict attention to sparse, windowed, or axis-wise patterns (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Liu et al., 2021; Ho et al., 2019; Bertasius et al., 2021), while reduction methods aggregate, select, or prune representations (Wang et al., 2022; Rao et al., 2021; Madinei et al., 2026). Their effectiveness depends on the chosen connectivity (Liu et al., 2021; Gao et al., 2022b) and composition of attention operations (Dong et al., 2022; Tarasiou et al., 2023; Tang et al., 2026). TuBA instead exploits low-rank tensor structure, learning mode-wise factors to perform global token mixing in a compact Tucker subspace. Prior low-rank (Wang et al., 2020) and latent-token methods (Lee et al., 2019; Ma et al., 2021; Jaegle et al., 2021) compress one side of attention while retaining full-length queries, whereas other methods approximate attention scores while retaining full-length representations (Xiong et al., 2021; Chen et al., 2021b;a). Tensorized attention (Omranpour et al., 2025; Feng et al., 2024) similarly exploits Kronecker structure in the attention matrix without compressing hidden representations. TuBA instead compresses the hidden tensor itself, performing attention and all query, key, value, and output projections within its Tucker core. Unlike Tucker-based model compression (Klein et al., 2026; Gu et al., 2025), which targets model weights, TuBA compresses hidden representations along token dimensions to reduce attention and projection costs.

Table 1: Comparison of predictive accuracy and computational efficiency on Human3.6M.

Method	Accuracy					Efficiency			
	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓	Params	FLOPs	FPS	FPS*
TuBA	**113.21**	**1499.3**	0.9664	**33.79**	<u>0.0339</u>	11.1M	**63.5G**	<u>580</u>	<u>739</u>
SimVP	<u>116.91</u>	<u>1525.9</u>	**0.9824**	<u>33.72</u>	0.0340	11.1M	74.0G	**589**	576
ViT	141.64	1642.6	<u>0.9815</u>	33.50	**0.0335**	11.1M	172.8G	151	694
Linear	149.79	1639.4	0.9615	32.54	0.0397	11.1M	<u>70.6G</u>	555	**741**
Linformer	277.30	2107.8	0.9501	30.02	0.0406	23.7M	76.2G	552	701
Nyström	196.01	1809.8	0.9575	31.45	0.0439	11.1M	75.2G	473	537

Table 2: Comparison of predictive accuracy and computational efficiency on KittiCaltech.

Method	Accuracy					Efficiency			
	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓	Params	FLOPs	FPS	FPS*
TuBA	**156.07**	**1641.7**	<u>0.8963</u>	**27.25**	<u>0.0772</u>	7.0M	29.2G	142	382
SimVP	<u>156.94</u>	<u>1683.9</u>	**0.9019**	<u>26.87</u>	**0.0676**	8.6M	60.6G	218	197
ViT	207.33	1970.4	0.8621	25.85	0.1159	7.0M	87.5G	62	<u>392</u>
Linear	174.18	1732.0	0.8782	26.70	0.1026	7.0M	**25.2G**	**293**	**417**
Linformer	285.74	2303.4	0.8337	24.29	0.1346	16.8M	29.6G	<u>287</u>	381
Nyström	206.80	1972.4	0.8672	25.87	0.1047	7.0M	<u>29.0G</u>	216	185

4 EXPERIMENTS

We evaluate TuBA's accuracy and efficiency on video prediction and weather forecasting. On both tasks, we compare TuBA under a controlled setup with standard self-attention, efficient global attention methods—Linformer (Wang et al., 2020), Nyström attention (Xiong et al., 2021), and linear attention (Katharopoulos et al., 2020)—and the convolutional mixer SimVP (Gao et al., 2022a). For video prediction, we further compare with recurrent, convolutional, local-attention, MLP, and pooling-based predictors. For weather forecasting, we compare with Earthformer (Gao et al., 2022b) and autoregressive FourCastNet (Pathak et al., 2022), and evaluate causal TuBA for autoregressive forecasting. We also study the effects of Tucker core size and causal granularity on accuracy and efficiency. Bold and underlined values denote the best and second-best results, respectively. All experiments use a single NVIDIA H100 GPU, with implementation details in Appendix E.

4.1 VIDEO PREDICTION

For video prediction, we use two complementary benchmarks. Human3.6M captures structured human motion, where models predict four future frames from four observed frames at 256×256 resolution. KittiCaltech evaluates cross-dataset generalization to real-world driving scenes with camera motion, occlusions, and clutter: models trained on KITTI predict one future frame from ten observed frames on Caltech Pedestrian at 128×160 resolution. For controlled comparisons among attention mechanisms, all variants use the same shallow convolutional encoder–decoder and MLP, differing only in their attention blocks. SimVP (Gao et al., 2022a) uses the same encoder–decoder but replaces the transformer with its original Inception-UNet (IncepU) mixer. This setup isolates the effect of the token-mixing mechanism while keeping the remaining architecture fixed.

Prediction accuracy and computational efficiency. Tables 1 and 2 show that TuBA achieves the lowest MSE and MAE on both benchmarks, where FPS and FPS* denote `fp32` and mixed-precision inference, respectively. It also achieves the highest PSNR and second-best LPIPS. Compared with ViT using standard self-attention, TuBA reduces MSE by up to 24.7% and FLOPs by up to 66.6%, with 2.3–3.8$\times$ faster `fp32` inference and 3.3–4.3$\times$ faster training (Appendix E.4), while achieving comparable mixed-precision throughput. Under mixed precision, the gap narrows as ViT uses an optimized attention kernel, while TuBA's additional read and write operations lack specialized kernels. The MSE reduction is larger on KittiCaltech (24.7% versus 20.1%), demonstrating effectiveness in real-world scenes with camera motion and clutter. TuBA uses no more parameters than ViT or Linformer and reduces MSE over efficient-attention baselines by 10–59% at comparable or lower

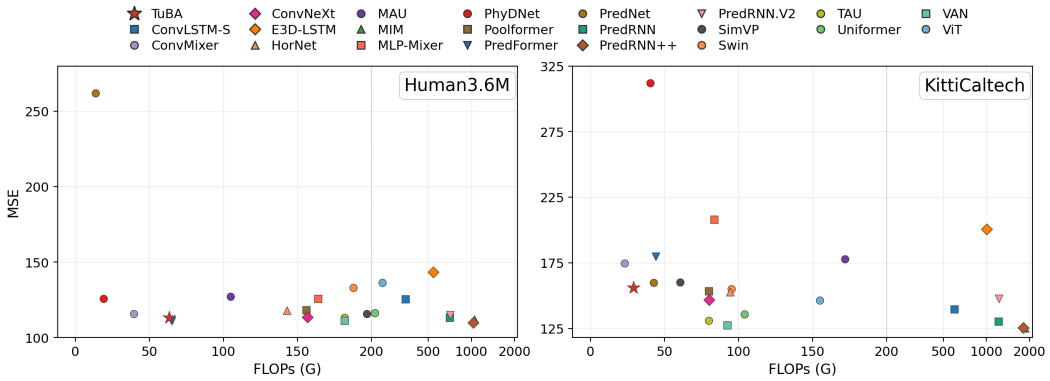

Figure 2: Comparison of prediction error (MSE) and computational cost (FLOPs) on Human3.6M and KittiCaltech; the FLOPs axis is compressed above 200G. TuBA lies on the FLOPs–MSE Pareto frontier on both benchmarks.

FLOPs. It also slightly outperforms SimVP in MSE with fewer FLOPs and 1.3–1.9× higher mixed-precision throughput. Despite attending over fewer core slots than ViT has tokens, TuBA achieves higher accuracy, suggesting that its low-rank tensor bottleneck provides a beneficial inductive bias rather than merely a lossy approximation.

Comparison to other predictors. Figure 2 compares TuBA with other video prediction methods in terms of prediction accuracy and computational demand. Since TuBA is trained with the standard MSE loss, we use MSE as the primary accuracy metric and FLOPs as a representative measure of computational cost. On both Human3.6M and KittiCaltech, TuBA lies on the FLOPs–MSE Pareto frontier: none of the compared methods achieves both lower MSE and lower FLOPs. TuBA therefore provides a favorable accuracy–efficiency trade-off, attaining low prediction error without the substantially larger computational cost of many recurrent and transformer-based predictors. These results provide broader context rather than a strictly controlled comparison, as methods differ in their training losses, architectures, skip connections, and input-processing pipelines.

Full-rank attention is not optimal. Figure 3 shows the effect of Tucker rank on model accuracy by varying the Tucker core sizes from $R = 720$ to the full-rank setting ($R = 3200$), which matches the number of tokens used by the standard self-attention (Appendix E.2). We observe that accuracy does not improve monotonically as the core size increases, despite the additional computation. The best results occur at the intermediate core size $R = 1287$, which attains the lowest MSE and the highest SSIM. Increasing the core beyond this point degrades prediction accuracy, and the full-rank core performs substantially worse. We also observe the same pattern comparing the standard ViT with TuBA in Tables 1 and 2. The fact that performance is maximized at an intermediate low rank rather than in the full-rank or standard self-attention supports TuBA's premise that global multidimensional interactions can be modeled effectively in a compact low-rank tensor representation.

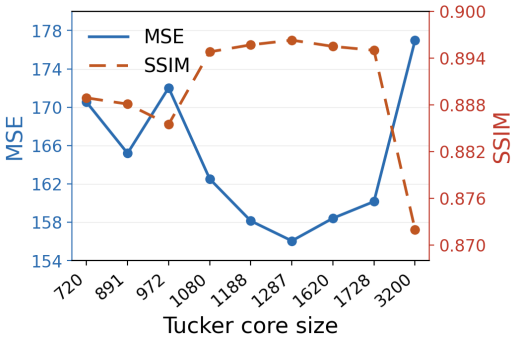

Figure 3: Effect of Tucker core size on TuBA's prediction accuracy on KittiCaltech, measured by MSE and SSIM.

Table 3: Comparison of predictive accuracy and computational efficiency on ERA5. Models are trained on 2011–2015 and evaluated on 2017–2018. (AR) denotes autoregressive generation.

Method	NRMSE↓	z_{500}↓	t_{850}↓	t_{2m}↓	u_{10}↓	v_{10}↓	Params	FLOPs	clips/s
TuBA	0.399	0.141	0.162	0.126	0.536	0.627	8.72M	**40.8G**	**600**
SimVP	0.373	0.127	0.158	0.122	0.484	0.572	8.60M	67.8G	192
Earthformer	0.415	0.148	0.182	0.194	0.526	0.613	8.66M	86.5G	191
ViT	0.492	0.212	0.204	0.147	0.671	0.793	8.71M	198G	362
Linformer	0.516	0.216	0.212	0.147	0.720	0.866	33.87M	56.7G	434
Linear	0.550	0.259	0.242	0.169	0.731	0.879	8.71M	45.4G	514
Nyström	0.552	0.259	0.243	0.172	0.731	0.883	8.71M	48.0G	401
TuBA (AR)	**0.349**	**0.113**	**0.136**	**0.107**	**0.458**	**0.534**	7.67M	67.2G	265
FourCastNet (AR)	0.442	0.153	0.167	0.123	0.559	0.659	8.80M	54.4G	64
ViT (AR)	0.555	0.203	0.203	0.184	0.652	0.767	7.26M	451.2G	62

4.2 Weather Prediction

We further evaluate TuBA on global weather forecasting to test its ability to model multivariate atmospheric dynamics. We use ERA5 (Hersbach et al., 2020) at $1.40625°$ resolution (128×256) with 25 atmospheric and surface variables, including geopotential, temperature, and wind at multiple pressure levels, sampled at six-hour intervals. Models observe 12 frames (three days) and predict the next 12 frames, either in one shot or autoregressively; we further evaluate autoregressive models over horizons up to 10 days (40 frames). For TuBA and the attention baselines, we use direct spatial patch embedding and a linear output head, isolating the ability to learn atmospheric dynamics without convolutional feature extraction. Autoregressive TuBA uses the block-causal formulation of Section 3.3. We also compare with SimVP and two Earth-system forecasting models: Earthformer (Gao et al., 2022b), a one-shot predictor combining a hierarchical convolutional encoder–decoder with cuboid attention, and FourCastNet (Pathak et al., 2022), an autoregressive model that uses adaptive Fourier neural operators (AFNO) for spatial mixing.

Prediction accuracy and computational efficiency. Table 3 reports the mean normalized RMSE (NRMSE) across all 25 channels and individual NRMSE values for five key weather fields. One-shot TuBA reduces the mean NRMSE of the standard ViT by 18.9% and that of the efficient-attention baselines by 22.7–27.7%, with improvements across all five individually reported fields. As in video prediction, the improvement over the standard self-attention supports the benefit of TuBA's structured low-rank inductive bias for learning spatiotemporal dynamics. These accuracy gains accompany the lowest FLOPs and highest forecast throughput among all methods in Table 3. TuBA requires 79.4% fewer FLOPs than ViT while delivering $1.66\times$ higher throughput. It also uses fewer FLOPs and achieves higher throughput than all three efficient attention baselines. Compared with Earthformer, TuBA achieves lower mean NRMSE, although Earthformer is slightly more accurate on the two surface wind components. TuBA nevertheless delivers $3.14\times$ higher throughput with 52.8% fewer FLOPs. One-shot TuBA has a 7.0% higher mean NRMSE than SimVP, but requires 39.8% fewer FLOPs and delivers $3.13\times$ higher throughput; the autoregressive formulation below closes this gap and surpasses SimVP. By contrast, the efficient-attention baselines, whose throughput is closest to TuBA's, have 38.3–48.0% higher mean NRMSE than SimVP. Even in its one-shot setting, TuBA outperforms autoregressive FourCastNet, achieving 9.7% lower mean NRMSE and $9.38\times$ higher throughput.

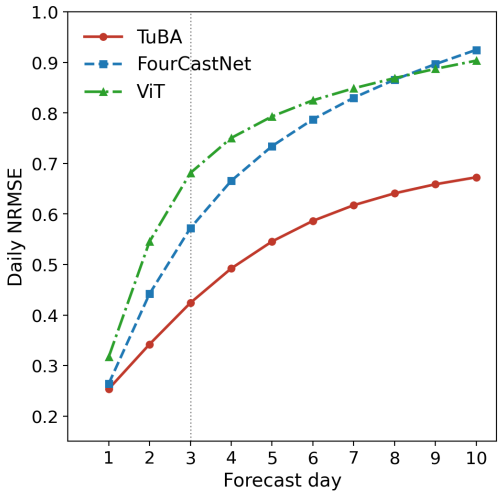

Figure 4: Daily NRMSE of autoregressive forecasts over ten days. The dotted line marks the three-day forecast horizon used during training.

Autoregressive TuBA. We additionally evaluate the causal formulation of TuBA from Section 3.3 for autoregressive forecasting. Each tensor contains four consecutive frames; attention is bidirectional within its Tucker core and causal across tensor cores. Autoregressive TuBA achieves the lowest mean NRMSE and the lowest error on every individual field reported in Table 3. It surpasses both SimVP and Earthformer in accuracy while requiring fewer FLOPs and achieving higher throughput, and reduces mean NRMSE by 12.5% relative to one-shot TuBA. Compared with autoregressive ViT, it achieves 37.1% lower mean NRMSE with 85.1% fewer FLOPs and $4.27\times$ higher throughput. Compared with FourCastNet, autoregressive TuBA reduces mean NRMSE by 21.0% and achieves $4.14\times$ higher throughput despite requiring more FLOPs. This throughput advantage reflects both TuBA's fewer autoregressive steps and differences in hardware efficiency between attention and AFNO's Fourier-domain operations. We further extend autoregressive forecasting to ten days. Figure 4 shows that TuBA maintains lower daily NRMSE than both FourCastNet and ViT throughout the forecast period. Its advantage persists beyond the three-day training horizon, with a substantial separation from both baselines through day ten. These results demonstrate autoregressive TuBA's accuracy and efficiency across short- and medium-range weather forecast horizons.

Frame-by-frame generation is not optimal. We study how the granularity of causality affects autoregressive TuBA by varying the number of frames in each tensor. As shown in Table 4, generating multiple frames jointly improves accuracy over frame-by-frame generation. The lowest NRMSE occurs with four-frame tensors; increasing the tensor size to six frames further improves computational efficiency, but lowers accuracy. Across the tested granularities, the parameter count increases modestly with tensor size. These results show that the choice of tensor size controls the trade-off between forecasting accuracy and computational efficiency, with the best accuracy achieved at an intermediate granularity.

Table 4: Effect of causal granularity (frames per tensor) on autoregressive TuBA's accuracy and computational efficiency on ERA5.

Granularity	NRMSE↓	Params	FLOPs	clips/s
1	0.426	**7.27M**	89.3G	100
2	<u>0.358</u>	<u>7.41M</u>	85.2G	163
4	**0.349**	7.67M	<u>67.2G</u>	<u>265</u>
6	0.368	7.93M	**62.9G**	**317**

The ablation also offers a possible explanation for part of the accuracy gap between TuBA and FourCastNet. Under frame-by-frame generation, TuBA's NRMSE rises to 0.426, approaching FourCastNet's 0.442, compared with 0.349 when generating four frames jointly. This suggests that TuBA's ability to combine bidirectional interactions within each tensor with causal attention across tensors contributes to its accuracy advantage, while other architectural differences should also be considered. Yet even at single-frame granularity, autoregressive TuBA substantially outperforms autoregressive ViT with standard self-attention, suggesting that Tucker core attention also contributes to its higher accuracy.

5 Conclusion and Future Work

We introduced Tucker bottleneck attention (TuBA), which exploits the low-rank tensor structure of hidden representations to perform global token mixing within a learned Tucker subspace. By computing standard self-attention and its linear projections entirely within compact cores, TuBA reduces computational cost. Our analysis establishes its Tucker interpretation, recovers standard self-attention as a special case, and shows the increased expressivity of independently learned read and write factors. Its autoregressive extension combines bidirectional interactions within each core with causal attention across cores. Experiments on video prediction and global weather forecasting demonstrate improved prediction accuracy and reduced computation relative to standard self-attention, with favorable accuracy–efficiency trade-offs against efficient-attention alternatives and task-specific models. The results further show that low-rank Tucker cores and multi-frame generation can outperform full-rank attention and frame-by-frame generation, respectively. These findings support low-rank tensor structure as a useful inductive bias for sequence modeling and motivate further exploration of attention within learned representation subspaces. Currently, Tucker ranks are selected empirically, and our evaluation focuses on data with native multidimensional structure. Future work could investigate adaptive rank selection and extend TuBA to long one-dimensional sequences, where the choice of tensorization introduces an additional modeling decision.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021.

Tom B. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. MAU: A motion-aware unit for video prediction and beyond. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a.

Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nyström method. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Stéphane d'Ascoli, Hugo Touvron, Matthew L. Leavitt, Ari S. Morcos, Giulio Biroli, and Levent Sagun. ConViT: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning (ICML)*, 2021.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. doi: 10.1137/S0895479896305696.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. CSWin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12124–12134, 2022. doi: 10.1109/CVPR52688.2022.01181.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Aosong Feng, Rex Ying, and Leandros Tassiulas. Long sequence modeling with attention tensorization: From sequence to tensor learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14642–14655, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.858.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3170–3180, 2022a. doi: 10.1109/CVPR52688.2022.00317.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 25390–25403, 2022b.

Yuxuan Gu, Wuyang Zhou, Giorgos Iacovides, and Danilo Mandic. TensorLLM: Tensorising multi-head attention for enhanced reasoning and compression in LLMs. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2025. doi: 10.1109/IJCNN64981.2025.11228585.

A. Hannachi, I. T. Jolliffe, and D. B. Stephenson. Empirical orthogonal functions and related techniques in atmospheric science: A review. *International Journal of Climatology*, 2007.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Philip Holmes, John L. Lumley, Gahl Berkooz, and Clarence W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 2 edition, 2012.

Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Timon Klein, Jonas Kusch, Sebastian Sager, Stefan Schnake, and Steffen Schotthöfer. Tucker attention: A generalization of approximate attention mechanisms. *arXiv preprint arXiv:2603.30033*, 2026.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 2009.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 2023.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021. doi: 10.1109/ICCV48922.2021.00986.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Parsa Madinei, Ryan Solgi, Ziqi Wen, Jonathan Skaza, Miguel Eckstein, and Ramtin Pedarsani. Interlace: Interleaved layer pruning and efficient adaptation in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 2947–2956, June 2026.

Cristian Meo, Varun Sarathchandran, Avijit Majhi, Shao-Hsuan Hung, Carlo Saccardi, Ruben Imhoff, Roberto Deidda, Remko Uijlenhoet, and Justin Dauwels. Blockgpt: Spatio-temporal modelling of rainfall via frame-level autoregression. *arXiv preprint arXiv:2510.06293*, 2025. URL https://arxiv.org/abs/2510.06293.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025.

Ken M. Nakanishi. Scalable-softmax is superior for attention. *arXiv preprint arXiv:2501.19399*, 2025.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. In *International Conference on Machine Learning (ICML)*, 2023.

Soroush Omranpour, Reihaneh Rabbany, and Guillaume Rabusseau. Higher-order transformers with kronecker-structured attention. *Transactions on Machine Learning Research (TMLR)*, 2025. arXiv:2412.02919.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations (ICLR)*, 2022.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Ryan Solgi, Seyedali Mousavinezhad, and Hugo A. Loaiciga. Do tensorized large-scale spatiotemporal dynamic atmospheric data exhibit low-rank properties? In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2025.

Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z. Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z. Li. OpenSTL: A comprehensive benchmark of spatio-temporal predictive learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pp. 69819–69831, 2023b.

Yujin Tang, Lu Qi, Xiangtai Li, Chao Ma, and Ming-Hsuan Yang. Video prediction transformers without recurrence or convolution. *Transactions on Machine Learning Research*, 2026.

Michail Tarasiou, Erik Chavez, and Stefanos Zafeiriou. ViTs for SITS: Vision transformers for satellite image time series. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10418–10428, 2023. doi: 10.1109/CVPR52729.2023.01004.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 2022.

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. Softmax is not enough (for sharp size generalisation). In *International Conference on Machine Learning (ICML)*, 2025.

John M. Wallace and David S. Gutzler. Teleconnections in the geopotential height field during the northern hemisphere winter. *Monthly Weather Review*, 1981.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. doi: 10.1007/s41095-022-0274-8.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning (ICML)*, 2018b.

Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S. Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14138–14148, 2021. doi: 10.1609/aaai.v35i16.17664.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.07772.

Hao Yu and Jianxin Wu. Compressing transformers: Features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11007–11015, 2023. doi: 10.1609/aaai.v37i9.26304. URL https://ojs.aaai.org/index.php/AAAI/article/view/26304.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 17283–17297, 2020.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

A SHARED TUCKER SUBSPACES OF HIDDEN REPRESENTATIONS

We examine whether the hidden representations of a standard transformer admit a low-rank Tucker structure whose factor matrices are shared across samples. We extract hidden tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D \times d}$ across the layers of the trained ViT baselines on Human3.6M and KittiCaltech and compare three Tucker representations. In all three, each sample has its own core, computed by projection onto the factors; they differ only in how the mode-wise factors are obtained. With shared factors, a single set of factors is fitted once to 128 training samples and applied unchanged to 128 held-out test samples. With per-sample factors, both the factors and the core are fitted to each test sample individually by HOSVD (De Lathauwer et al., 2000), which serves as a reference for how well each sample can be represented at the given ranks. Random orthonormal factors of the same ranks serve as a control. We measure the relative reconstruction error $\|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_F / \|\boldsymbol{\mathcal{X}}\|_F$, averaged over the test samples, as a function of the core size R/N.

Figure 5 shows the results for three representative layers (0, 5, and 8), with dotted lines marking the core sizes used by TuBA in the main experiments; random factors are included as a control. Across layers, datasets, and core sizes, shared factors with per-sample cores closely approach the reconstruction accuracy of per-sample factors, with a slightly larger gap on KittiCaltech, whose driving scenes are more diverse. Thus, the underlying Tucker subspace is largely shared across samples: fixed mode-wise factors learned from training data capture most of the structure of unseen samples, while the sample-specific information is carried by a compact core. This observation motivates TuBA's design, which learns read and write factors shared across inputs and applies attention to the input-dependent core.

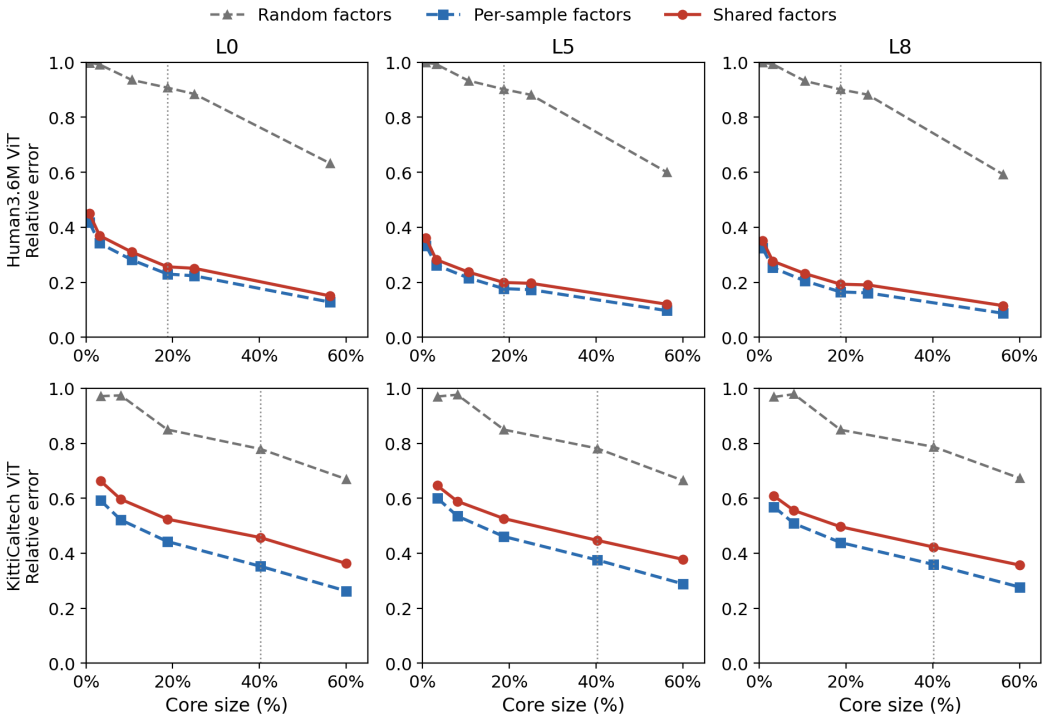

Figure 5: Tucker reconstruction error of held-out ViT hidden representations on Human3.6M (top) and KittiCaltech (bottom) at three representative layers. Shared factors with per-sample cores closely approach per-sample factors; dotted lines mark TuBA's core sizes. Random factors are included as a control.

B Proof of Proposition 1

Proof. For each token mode n, define $\mathbf{Q}_n = \mathbf{A}_n^\top \mathbf{A}_n$. Since $\mathbf{A}_n$ has full column rank, $\mathbf{Q}_n$ is positive definite and $\mathbf{U}_n = \mathbf{A}_n \mathbf{Q}_n^{-1}$ also has full column rank. Consequently,

$$\mathbf{U}_n^\top \mathbf{U}_n = \mathbf{Q}_n^{-1}, \qquad \mathbf{U}_n^\dagger = (\mathbf{U}_n^\top \mathbf{U}_n)^{-1} \mathbf{U}_n^\top = \mathbf{A}_n^\top.$$

To establish the least-squares statement, consider

$$\min_{\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times \cdots \times r_D \times d}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_D \mathbf{U}_D\|_F^2 .$$

Flatten the first D modes with the mode-1 index varying fastest, and let $\mathbf{X} \in \mathbb{R}^{N \times d}$ and $\mathbf{G} \in \mathbb{R}^{R \times d}$ be the resulting matrices, where $N = \prod_{n=1}^{D} I_n$ and $R = \prod_{n=1}^{D} r_n$. Define

$$\mathbf{U} = \mathbf{U}_D \otimes \cdots \otimes \mathbf{U}_1, \qquad \mathbf{A} = \mathbf{A}_D \otimes \cdots \otimes \mathbf{A}_1.$$

Reshaping preserves the Frobenius norm, so the objective becomes $\|\mathbf{X} - \mathbf{U}\mathbf{G}\|_F^2$. Its unique minimizer is

$$\mathbf{G}_* = \mathbf{U}^\dagger \mathbf{X} = \left(\mathbf{U}_D^\dagger \otimes \cdots \otimes \mathbf{U}_1^\dagger\right) \mathbf{X} = \mathbf{A}^\top \mathbf{X}.$$

Reshaping this matrix gives

$$\boldsymbol{\mathcal{G}}_* = \boldsymbol{\mathcal{X}} \times_1 \mathbf{A}_1^\top \times_2 \cdots \times_D \mathbf{A}_D^\top = \boldsymbol{\mathcal{C}}.$$

For the reconstruction, define

$$\mathbf{P}_n = \mathbf{U}_n \mathbf{A}_n^\top = \mathbf{A}_n (\mathbf{A}_n^\top \mathbf{A}_n)^{-1} \mathbf{A}_n^\top.$$

Each $\mathbf{P}_n$ is the orthogonal projector onto $\mathrm{range}(\mathbf{A}_n)$. Substituting the read core into the reconstruction yields

$$\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{C}} \times_1 \mathbf{U}_1 \times_2 \cdots \times_D \mathbf{U}_D = \boldsymbol{\mathcal{X}} \times_1 \mathbf{P}_1 \times_2 \cdots \times_D \mathbf{P}_D.$$

In flattened form,

$$\hat{\mathbf{X}} = (\mathbf{P}_D \otimes \cdots \otimes \mathbf{P}_1)\, \mathbf{X}.$$

The Kronecker product of these projectors is symmetric and idempotent, and its range is the tensor-product subspace determined by the read factors. Hence, this reconstruction is the orthogonal projection onto that subspace, with the channel mode unchanged.

Finally, for any transformed core $\boldsymbol{\mathcal{C}}'$, the output

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{C}}' \times_1 \mathbf{B}_1 \times_2 \cdots \times_D \mathbf{B}_D$$

is an exact Tucker representation, with the identity factor on the channel mode understood. For each token mode n, its mode-n unfolding has the form

$$\mathbf{Y}_{(n)} = \mathbf{B}_n \mathbf{Z}_n,$$

where $\mathbf{Z}_n$ collects the transformed core and the contractions along the remaining modes. Therefore,

$$\mathrm{rank}(\mathbf{Y}_{(n)}) \le \mathrm{rank}(\mathbf{B}_n) \le r_n, \qquad n = 1, \ldots, D.$$

This final claim does not require full-column-rank write factors. □

C Proof of Theorem 1

Proof. Every tied layer is an untied layer with $\mathbf{A}_n = \mathbf{B}_n$ for all n, so $\mathcal{F}_{\mathrm{tied}} \subseteq \mathcal{F}_{\mathrm{untied}}$. We prove strict inclusion by constructing an untied mapping that no tied layer can represent.

Consider a TuBA layer

$$F(\mathbf{X}) = \mathbf{B}\, \mathrm{MHSA}(\mathbf{A}^\top \mathbf{X}),$$

with MHSA as in Eq. (1). For a fixed input direction $\mathbf{Z}$ and scalar t, the query–key scores at input $t\mathbf{Z}$ are of order t^2. Hence, each core attention matrix satisfies

$$\widetilde{\mathbf{S}}_i(t\mathbf{A}^\top \mathbf{Z}) = \frac{1}{R} \mathbf{1}_R \mathbf{1}_R^\top + O(t^2).$$

Define

$$\mathbf{M} = \sum_{i=1}^{H} \mathbf{M}_i, \qquad \boldsymbol{a} = \mathbf{A}\mathbf{1}_R, \qquad \boldsymbol{b} = \mathbf{B}\mathbf{1}_R.$$

The layer therefore satisfies

$$F(t\mathbf{Z}) = \frac{t}{R}\boldsymbol{b}\boldsymbol{a}^\top \mathbf{Z}\mathbf{M} + O(t^3).$$

For a tied layer, let $\mathbf{T}$ denote its common Kronecker-product read and write matrix, and set $\boldsymbol{c} = \mathbf{T}\mathbf{1}_R$. Its derivative at the origin has the form

$$DF_{\text{tied}}(\mathbf{0})[\mathbf{Z}] = \frac{1}{R}\boldsymbol{c}\boldsymbol{c}^\top \mathbf{Z}\mathbf{M}_{\text{tied}}.$$

Now choose untied factors such that $\boldsymbol{a}$ and $\boldsymbol{b}$ are nonzero and noncollinear. Such a choice is possible because $N > 1$ implies $I_n > 1$ for at least one mode, and

$$\boldsymbol{a} = (\mathbf{A}_D\mathbf{1}_{r_D}) \otimes \cdots \otimes (\mathbf{A}_1\mathbf{1}_{r_1}), \qquad \boldsymbol{b} = (\mathbf{B}_D\mathbf{1}_{r_D}) \otimes \cdots \otimes (\mathbf{B}_1\mathbf{1}_{r_1}).$$

The column sums can be chosen nonzero and equal in all other modes, and noncollinear in a mode with $I_n > 1$.

Set the query and key projections to zero, making every attention matrix exactly $\mathbf{1}_R\mathbf{1}_R^\top/R$. Choose the value and output projections so that

$$\mathbf{M} = \mathbf{E} := \boldsymbol{e}_1\boldsymbol{e}_1^\top,$$

where $\boldsymbol{e}_1 \in \mathbb{R}^d$ is the first coordinate vector. This rank-one channel map is realizable by one head, with all other heads contributing zero. The resulting untied layer is the linear map

$$F_{\text{untied}}(\mathbf{X}) = \frac{1}{R}\boldsymbol{b}\boldsymbol{a}^\top \mathbf{X}\mathbf{E}.$$

Since this map is linear, its derivative is

$$DF_{\text{untied}}(\mathbf{0})[\mathbf{Z}] = \frac{1}{R}\boldsymbol{b}\boldsymbol{a}^\top \mathbf{Z}\mathbf{E}.$$

Suppose a tied layer represents this same mapping. Their derivatives at the origin must agree. Applying them to $\mathbf{Z} = \boldsymbol{x}\boldsymbol{e}_1^\top$ and taking the first output channel gives

$$\boldsymbol{b}\boldsymbol{a}^\top \boldsymbol{x} = \gamma \boldsymbol{c}\boldsymbol{c}^\top \boldsymbol{x} \quad \text{for every } \boldsymbol{x} \in \mathbb{R}^N, \qquad \gamma = \boldsymbol{e}_1^\top \mathbf{M}_{\text{tied}}\boldsymbol{e}_1.$$

Thus,

$$\boldsymbol{b}\boldsymbol{a}^\top = \gamma \boldsymbol{c}\boldsymbol{c}^\top.$$

$\boldsymbol{b}\boldsymbol{a}^\top = \gamma \boldsymbol{c}\boldsymbol{c}^\top$ is symmetric, contradicting the noncollinearity of $\boldsymbol{a}$ and $\boldsymbol{b}$. Hence, $\mathcal{F}_{\text{tied}} \subsetneq \mathcal{F}_{\text{untied}}$. □

D TuBA COMPUTATIONAL COST DETAILS

Consider a read contraction order $\pi = (\pi_1, \ldots, \pi_D)$. Before the kth contraction, modes $\pi_1, \ldots, \pi_{k-1}$ have been reduced to their ranks, while the remaining modes retain their ambient sizes. Multiplication by $\mathbf{A}_{\pi_k}^\top$ therefore costs

$$d\, I_{\pi_k} r_{\pi_k} \Big(\prod_{j<k} r_{\pi_j} \Big) \Big(\prod_{j>k} I_{\pi_j} \Big) = dN\, r_{\pi_k} \prod_{j<k} \frac{r_{\pi_j}}{I_{\pi_j}} \;\le\; dN\, r_{\pi_k},$$

since $r_n \le I_n$. Summing over modes, the read costs at most $N\bar{R}d$ with $\bar{R} = \sum_{n=1}^{D} r_n$, for any contraction order. The write applies the factors $\mathbf{B}_{\pi_k}$ in reverse order at the same cost, so the read and write together cost $O(N\bar{R}d)$. Contracting modes with smaller r_n/I_n first further reduces the exact cost.

Since $r_n \le R$, we have $\bar{R} \le DR$. Hence, for fixed D and $R = o(N)$, $N\bar{R}d = o(N^2 d)$, $R^2 d = o(N^2 d)$, and $Rd^2 = o(Nd^2)$, so the TuBA cost $O(N\bar{R}d + R^2 d + Rd^2)$ is asymptotically lower than the $O(N^2 d + Nd^2)$ cost of full attention.

For memory, every intermediate tensor in the read and write has at most Nd entries, since each contraction only shrinks the tensor and each expansion recovers at most the ambient size; TuBA therefore requires $O(Nd + HR^2)$ activation memory, and $O(Nd)$ with FlashAttention.

E Implementation Details

E.1 Video-prediction implementation details

Tasks and metrics. Human3.6M uses $4\to4$ prediction at $256\times256\times3$, and KittiCaltech uses $10\to1$ prediction at $128\times160\times3$ (Tan et al., 2023b); KittiCaltech models are trained on KITTI and evaluated on the $3{,}631$-clip Caltech Pedestrian test set. We report mean squared error (MSE), mean absolute error (MAE), structural similarity index measure (SSIM) (Wang et al., 2004), peak signal-to-noise ratio (PSNR), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018). MSE and MAE are summed over pixels and channels, giving scales of $196{,}608$ for Human3.6M and $61{,}440$ for KittiCaltech. Parameter counts and FLOPs are reported per input clip.

Tokenization. The models in Tables 1 and 2 share the SimVP encoder–decoder (Gao et al., 2022a), with four convolutional blocks in both the encoder and the decoder, 64 hidden channels, and 3×3 convolutions, and differ only in the token mixer between them. The encoder maps the input sequence to a latent tensor, and 2×2 latent patches define the tokens, yielding token tensors of size $4\times32\times32$ ($4{,}096$ tokens) on Human3.6M and $10\times16\times20$ ($3{,}200$ tokens) on KittiCaltech. All attention-based models add a learned absolute positional embedding, with one vector per token, before the first block.

Architecture. On each benchmark, TuBA, ViT, linear attention, Linformer, and Nyström attention use the same transformer block, a pre-norm attention residual followed by a pre-norm SwiGLU residual, and differ only in the attention operation. The block has width 256, eight heads of size 32, and dropout 0.1, with SwiGLU hidden size 682 on Human3.6M and 256 on KittiCaltech. TuBA uses Tucker ranks $(3, 16, 16)$ on Human3.6M and $(9, 11, 13)$ on KittiCaltech; ViT applies full attention over all tokens; Linformer projects the keys and values to length 128; and Nyström attention uses 64 landmarks. SimVP instead uses its original IncepU translator with width 256.

Optimization. All models are trained in `fp32` using AdamW, weight decay 0.01, and a cosine schedule decaying to 10^{-6}, without data augmentation, gradient clipping, or drop path. Human3.6M models are trained for 50 epochs with batch size 4, and KittiCaltech models for 100 epochs with batch size 8.

Throughput. FPS is measured at inference with batch size 4 in `fp32`. FPS* uses mixed precision, with a fused FlashAttention kernel (Dao et al., 2022) when available and `bf16` arithmetic otherwise.

Table 5: Effect of Tucker ranks on TuBA's accuracy on KittiCaltech.

R	Ranks	MSE↓	MAE↓	SSIM↑	PSNR↑	LPIPS↓
720	$(9, 8, 10)$	170.54	1737.3	0.8889	26.74	0.0881
891	$(9, 9, 11)$	165.20	1665.5	0.8881	26.97	0.0898
972	$(9, 9, 12)$	172.03	1757.1	0.8855	26.66	0.0972
1080	$(9, 10, 12)$	162.53	1641.4	0.8948	27.15	0.0836
1188	$(9, 11, 12)$	158.15	1654.2	0.8957	27.14	0.0832
1287	$(9, 11, 13)$	**156.07**	1641.7	**0.8963**	**27.25**	**0.0772**
1620	$(9, 12, 15)$	158.43	**1608.7**	0.8955	27.17	0.0820
1728	$(9, 12, 16)$	160.17	1624.8	0.8950	27.15	0.0841
3200	$(10, 16, 20)$	176.98	1770.7	0.8720	26.56	0.1132

E.2 Rank ablation details

Table 5 reports the full results of the Tucker rank sweep on KittiCaltech. All models share the same backbone as in Appendix E.1, with the $10\times16\times20$ token tensor, and differ only in the Tucker ranks; the core size R is their product. The temporal rank is fixed at 9 for all reduced cores, so the sweep varies the two spatial ranks, while $R = 3200$ corresponds to the full-rank setting with no reduction. The core $R = 1287$ attains the best MSE, SSIM, PSNR, and LPIPS, while the next larger core, $R = 1620$, attains the lowest MAE.

E.3 Weather Forecasting Implementation Details

Data and forecasting task. We use ERA5 (Hersbach et al., 2020) on a 128×256 latitude–longitude grid at 1.40625° resolution, subsampled from the hourly archive at six-hour intervals. Each example contains 12 input frames and 12 target frames. Models are trained on 2011–2015 and evaluated on 2017–2018. The state contains 25 channels: geopotential (z), relative humidity (r), temperature (t), and horizontal wind components (u, v), each at 50, 500, 850, and 1000 hPa, together with five surface or single-level fields: 2 m temperature ($t_{2\mathrm{m}}$), 10 m wind components (u_{10}, v_{10}), total cloud cover (tcc), and top-of-atmosphere incident solar radiation (tisr).

Tokenization and architecture. TuBA and the attention baselines use non-overlapping 8×8 spatial patches, producing a 16×32 token grid per frame and a $12\times16\times32$ token tensor ($6{,}144$ tokens) per 12-frame context, and reconstruct predictions with a linear output head followed by unpatching, without a convolutional encoder–decoder. All attention-based models, including TuBA, add a learned absolute positional embedding, with one vector per token, before the first block. TuBA, ViT, linear attention, Linformer, Nyström attention, and the autoregressive TuBA and ViT use the same transformer block as in the video experiments, with width 256, four heads of size 64, SwiGLU hidden size 682, and no dropout, and differ only in the attention operation. One-shot TuBA uses Tucker ranks $(10, 8, 16)$ along the temporal, latitudinal, and longitudinal token axes; Linformer projects the keys and values to length 256, and Nyström attention uses 64 landmarks. SimVP retains its convolutional encoder–decoder and IncepU translator with spatial width 64 and translator width 176.

Autoregressive generation. Autoregressive TuBA uses block-causal attention with a granularity of four frames per tensor, so the 12-frame context forms three tensors, each with Tucker ranks $(3, 8, 16)$. Training uses teacher forcing over the concatenated context and target sequence in a single forward pass. At inference, the model generates three successive four-frame tensors, extending the sequence from 12 to 24 frames, with a per-layer key–value cache that processes the context once and is then updated with each generated tensor. Learned token positional embeddings span one four-frame tensor and are shared across tensors.

Evaluation metrics. For each channel c, we compute $\mathrm{NRMSE}_c = \mathrm{RMSE}_c/\sigma_c$, where σ_c is the standard deviation estimated from the training data. The aggregate score is the mean of the 25 channel-wise values. FLOPs are measured per complete 12-frame forecast, and throughput is the best measured number of complete forecasts per second at batch size eight.

E.4 Training memory and wall-clock time

Table 6 reports the peak GPU memory of one `fp32` training step and the average wall-clock time per training epoch on a single NVIDIA H100 GPU. Memory is the total held by the training process at that peak, read from `nvidia-smi`, and therefore includes live tensors, the reserved allocator cache, and the CUDA context. TuBA reduces training memory relative to standard self-attention by 3.5–5.1× and trains 1.7–4.3× faster per epoch across the three benchmarks, while remaining comparable to the efficient-attention baselines.

Table 6: Peak training GPU memory (GiB) and wall-clock time per epoch (minutes) on a single H100 GPU in `fp32`. Batch sizes are 4 for Human3.6M and 8 for KittiCaltech and ERA5.

	Human3.6M		KittiCaltech		ERA5	
Method	Memory	Time/epoch	Memory	Time/epoch	Memory	Time/epoch
TuBA	**10.22**	29.3	15.05	1.68	**11.99**	7.50
ViT	38.72	95.7	52.28	7.23	61.61	12.68
Linear	10.61	**23.7**	10.86	**1.10**	13.35	7.67
Linformer	10.47	25.2	**10.04**	1.30	13.42	7.37
Nyström	10.96	29.5	12.54	1.57	12.96	**7.31**