Deep Tensor Network

Yifan Zhang
IIIS
Tsinghua University
yif-zhang@outlook.com

September 16, 2023

Abstract

In this paper, we delve into the foundational principles of tensor categories, harnessing the universal properties of tensor products to pioneer novel methodologies in deep network architectures. Our primary contribution is the introduction of the Tensor Attention and Tensor Interaction Mechanism, a groundbreaking approach that leverages the tensor category to enhance the computational efficiency and the expressiveness of deep networks, and can even generalized into the quantum realm.

1 Introduction

In the realm of deep learning, the quest for efficiency and efficacy in neural network architectures is perpetual. At the heart of this pursuit lies the exploration of vector spaces and their interactions, which are pivotal in understanding and improving neural network functions. This paper commences with a fundamental examination of two finite-dimensional vector spaces, $\mathcal{X}$ and $\mathcal{Y}$, and their tensor product space, $\mathcal{X} \otimes \mathcal{Y}$. The intricacies of these spaces are articulated through the homomorphism relations that provide a deeper insight into the tensor operations.

Moving beyond traditional dot-product attention mechanisms, which exhibit quadratic computational complexity, we propose a novel framework: Linear Tensor Attention. This framework significantly reduces computational demands to linear in training, thus addressing the scalability issues prevalent in existing models.

Furthermore, we introduce two pivotal concepts: Tensor Attention and Tensor Interaction. These concepts are not mere incremental improvements but represent a paradigm shift in the approach to neural network design. They are deeply intertwined with advanced mathematical theories such as Linear Logic, Dependent Type Theory, and even concepts akin to Feynman Diagrams in physics. By embedding these sophisticated mathematical structures into neural network architectures, we aspire to not only enhance the performance of AI systems but also ensure their operations are underpinned by robust, theoretically sound principles.

Our work, therefore, stands at the intersection of deep learning and advanced mathematics, signaling a leap towards the next generation of AI systems with Tensor Categorical Guarantees. This fusion of disciplines promises not only computational efficiency but also a richer, more profound understanding of the underlying mechanics of neural networks.

Consider two finite-dimensional vector spaces $\mathcal{X}$ and $\mathcal{Y}$ to form a tensor product space: $\mathcal{X} \otimes \mathcal{Y}$.

$$\mathrm{Hom}(\mathcal{X}, \mathcal{Y}) \cong \mathcal{Y} \otimes \mathcal{X}^* \cong \mathcal{X} \otimes \mathcal{Y}^* \cong \mathcal{X} \otimes \mathcal{Y}.$$

Preprint. Under review.

Dot-product attention is not Axiomatic (($O(n^2)$)

$$\mathrm{Att}_{\leftrightarrow}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathbf{D}^{-1}\mathbf{A}\mathbf{V}, \quad \mathbf{A} = \exp\left(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}\right), \quad \mathbf{D} = \mathrm{diag}\left(\mathbf{A}\mathbf{1}_L\right).$$

Linear Tensor Attention ($O(n)$)

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T}\mathbf{V},$$

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\mathbf{Q}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})$$

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\mathbf{K}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})(\mathbf{K}^\top\mathbf{V})$$

Tensor Attention

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T}\mathbf{V},$$

$$\mathbf{T}_{\mathbf{Q}} = (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top = \mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top$$

$$\mathbf{T}_{\mathbf{K}} = (\mathbf{Q}\mathbf{K}^\top)^\top(\mathbf{Q}\mathbf{K}^\top) = \mathbf{K}\mathbf{Q}^\top\mathbf{Q}\mathbf{K}^\top$$

$$\mathbf{T}_{\mathbf{Q}}^{\odot} = (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top = (\mathbf{Q}\mathbf{K}^\top) \odot (\mathbf{K}\mathbf{Q}^\top)$$

$$\mathbf{T}_{\mathbf{K}}^{\odot} = (\mathbf{Q}\mathbf{K}^\top)^\top(\mathbf{Q}\mathbf{K}^\top) = (\mathbf{K}\mathbf{Q}^\top) \odot (\mathbf{Q}\mathbf{K}^\top)$$

Tensor Interaction

$$\mathrm{TensorInteraction}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbb{T}))^{-1}\mathbb{T}\mathbf{V}^\top,$$

$$\mathbb{T}_{\mathbf{Q}} = (\mathbf{Q}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{K})^\top = (\mathbf{Q}^\top\mathbf{K})(\mathbf{K}^\top\mathbf{Q})$$

$$\mathbb{T}_{\mathbf{K}} = (\mathbf{Q}^\top\mathbf{K})^\top(\mathbf{Q}^\top\mathbf{K}) = (\mathbf{K}^\top\mathbf{Q})(\mathbf{Q}^\top\mathbf{K})$$

$$\mathbb{T}_{\mathbf{Q}}^{\odot} = (\mathbf{Q}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{K})^\top = (\mathbf{Q}^\top\mathbf{K}) \odot (\mathbf{K}^\top\mathbf{Q})$$

$$\mathbb{T}_{\mathbf{K}}^{\odot} = (\mathbf{Q}^\top\mathbf{K})^\top \odot (\mathbf{Q}^\top\mathbf{K}) = (\mathbf{K}^\top\mathbf{Q}) \odot (\mathbf{Q}^\top\mathbf{K})$$

2 Preliminaries

2.1 Scaled Dot-Product Attention

In the seminal paper "Attention is All You Need" (Vaswani et al., 2017), the authors introduced a specific attention mechanism called "Scaled Dot-Product Attention," which has since become the standard attention mechanism.

The input consists of queries of dimension d_q, keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed, divided by $\sqrt{d_k}$, and then renormalized using softmax function. The original version utilized the parallelism using matrix form, i.e.:

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

We know that $\mathbf{Q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \cdots, \mathbf{q}_n^\top]^\top \in \mathbb{R}^{n \times d_{\text{model}}}$, suppose token size equals n, detailed rollout for Attention i:

For simplicity, we denote $\operatorname{softmax}(\mathbf{q}_i^T \mathbf{k}_j) \triangleq \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_{j=1}^N \exp(\mathbf{q}_i^\top \mathbf{k}_j)}$.

$$\begin{aligned}
\operatorname{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_i &= \sum_{j=1}^N \frac{\exp\left(\mathbf{q}_i^\top \mathbf{k}_j\right)}{\sum_{j=1}^N \exp\left(\mathbf{q}_i^\top \mathbf{k}_j\right)} \mathbf{v}_j \\
&= \sum_{j=1}^N \operatorname{softmax}(\mathbf{q}_i^\top \mathbf{k}_j) \mathbf{v}_j
\end{aligned}$$

2.2 Original ViT model

$$\begin{aligned}
\mathbf{z}_0 &= \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \cdots; \mathbf{x}_p^N \mathbf{E}\right] + \mathbf{E}_{\text{pos}} \\
\mathbf{z}'_\ell &= \operatorname{MultiHead}\left(\operatorname{LayerNorm}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1} \\
\mathbf{z}_\ell &= \operatorname{MLP}\left(\operatorname{LayerNorm}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell \\
\mathbf{y} &= \operatorname{LayerNorm}\left(\mathbf{z}_L^0\right)
\end{aligned} \tag{1}$$

2.3 Multi-Head Attention

Multi-head attention mechanism (Vaswani et al., 2017) gives the model the ability to learn disentangled representation in several different subspaces with various positions to be focused on.

$$\begin{aligned}
\operatorname{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \operatorname{Concat}\left(\text{head}_1, \ldots, \text{head}_{\text{h}}\right) \mathbf{W}^O \\
\text{where head}_{\text{i}} &= \operatorname{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right)
\end{aligned}$$

The projections are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d \times d_q}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times d_v}$ and $\mathbf{W}^O \in \mathbb{R}^{d \times d}$. Typically, $d = 512$, $h = 8$, $d_q = d_k = d_v = d/h = 64$.

2.4 Kernelized Scaled Dot-Product Attention

$$\operatorname{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \operatorname{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \mathbf{V}$$

We know that $\mathbf{Q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \cdots, \mathbf{q}_n^\top]^\top \in \mathbb{R}^{n \times d_{\text{model}}}$, suppose token size equals n, detailed roll-out for Attention i:

$$\begin{aligned}
\operatorname{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_i &= \sum_{j=1}^N \frac{\exp\left(\mathbf{q}_i^\top \mathbf{k}_j\right)}{\sum_{j=1}^N \exp\left(\mathbf{q}_i^\top \mathbf{k}_j\right)} \mathbf{v}_j \\
&= \sum_{j=1}^N \operatorname{softmax}(\mathbf{q}_i^\top \mathbf{k}_j) \mathbf{v}_j
\end{aligned}$$

Kernel Method:

$$\operatorname{sim}(\mathbf{q}_i, \mathbf{k}_j) = \phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)$$

e.g. (since $e^x \approx 1 + x$),

$$\phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j) = 1 + \left(\frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_2}\right)^\top \frac{\mathbf{k}_j}{\|\mathbf{k}_j\|_2}$$

Then

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})_i = \sum_{j=1}^{N} \frac{\phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)}{\sum_{j=1}^{N} \phi(\mathbf{q}_i)^\top \phi(\mathbf{k}_j)} \mathbf{v}_j$$

2.5 Linear Attention

In the context of Transformers, the computational bottleneck is often the attention mechanism, which has quadratic complexity due to the softmax function. Linear attention aims to reduce this to linear complexity. One popular approach to achieve this is by using ”kernelized” attention.

2.5.1 Linear Attention using Kernel Approximation

In the linear attention model, such as Performers, the softmax function is approximated using positive definite kernels ϕ. The idea is to rewrite the attention mechanism as:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^\top \mathbf{V}$$

The function $\phi(\cdot)$ maps the input vectors to a feature space where the dot product approximates the softmax operation.

3 Tensor Category

3.1 Tensor Product

Definition 3.1. Let V and W be two vector spaces. The tensor product of V and W denoted by $V \otimes W$ is a vector space with a bilinear map

$$\otimes : V \times W \to V \otimes W$$

which has the universal property.

In other words, if $\tau : V \times W \to Z$, then there exists a unique linear map, up to isomorphism, $\tilde{\tau} : V \otimes W \Rightarrow Z$ such that $\otimes \circ \tilde{\tau} = \tau$. The diagram for universal property can be seen in Fig. 1 below. Another way to say this is that a map $\tau \in \mathcal{L}^2(V \times W, Z)$ induces a map $\tilde{\tau} \in \mathcal{L}(V \otimes W, Z)$.

$$\begin{tikzcd} V \times W \arrow[r, "\otimes"] \arrow[dr, "\tau"'] & V \otimes W \arrow[d, dashed, "\tilde{\tau}"] \\ & Z \end{tikzcd}$$

Figure 1: Commutative diagram for the universal property of the tensor product.

3.2 Universal Property and Uniqueness of Tensor Product

The essence of tensor products lies not just in their construction through a bilinear map but also in their universal property. This universal property encapsulates the ”all-encompassing” nature of tensor products, making them a key element in understanding bilinear maps between vector spaces.

Definition 3.2. Given two vector spaces V and W, a tensor product $V \otimes W$ equipped with a bilinear map $\otimes : V \times W \to V \otimes W$ is said to satisfy the universal property if for every vector space Z and every bilinear map $\tau : V \times W \to Z$, there exists a unique linear map $\tilde{\tau} : V \otimes W \to Z$ such that $\tau = \tilde{\tau} \circ \otimes$.

Theorem 3.3 (Uniqueness Theorem of Tensor Products)**.** *If U and T are both tensor products of V and W (as defined above), then U and T are canonically isomorphic.*

Proof. Since U and T are both tensor products of V and W, they satisfy the universal property. According to the universal property, there exists unique linear maps $\alpha : U \to T$ and $\beta : T \to U$ such that the corresponding diagrams commute.

Utilizing the uniqueness part of the universal property again, we can establish that $\alpha \circ \beta = \mathrm{id}_T$ and $\beta \circ \alpha = \mathrm{id}_U$. Therefore, α and β are inverses of each other, making U and T isomorphic. This isomorphism is canonical, as it relies solely on the universal property. $\square$

Corollary 3.4. *$V \otimes W$ is canonically isomorphic to $W \otimes V$.*

Proof. To prove this, we merely need to show that $W \otimes V$ also satisfies the universal property for V and W. Then, by the uniqueness theorem, $V \otimes W$ and $W \otimes V$ must be canonically isomorphic.

First, note that there exists a bijection $\sigma : V \times W \to W \times V$ defined by $\sigma(v, w) = (w, v)$. Since $V \otimes W$ satisfies the universal property, we can find a unique map γ such that $\tau = \gamma \circ \sigma$.

We claim that γ satisfies the universal property for $W \otimes V$. For any bilinear map $\rho : W \times V \to Z$, the mapping $\rho \circ \sigma$ is bilinear. By the universal property of $V \otimes W$, there exists a unique $\eta : V \otimes W \to Z$ such that $\rho \circ \sigma = \eta \circ \otimes$.

Thus, $W \otimes V$ satisfies the universal property, confirming its canonical isomorphism to $V \otimes W$. $\square$

Lemma 3.5 (Bradley (2020))**.** *Given finite-dimensional vector spaces V and W there is an isomorphism*

$$\mathrm{End}(V \otimes W) \cong \mathrm{End}\, V \otimes \mathrm{End}\, W.$$

Proof. The proof quickly follows from the general fact that $\mathrm{Hom}(A, B) \cong B \otimes A^*$ for finite-dimensional spaces A and B. In particular, we have canonical isomorphisms

$$\begin{aligned}
\mathrm{End}(V \otimes W) &\cong (V \otimes W) \otimes (V \otimes W)^* \\
&\cong V \otimes W \otimes V^* \otimes W^* \\
&\cong V \otimes V^* \otimes W \otimes W^* \\
&\cong \mathrm{End}(V) \otimes \mathrm{End}(W).
\end{aligned}$$

$\square$

3.3 Partial Trace

Definition 3.6 (Partial trace (Bradley, 2020))**.** Suppose two vector space $V \in \mathbb{R}^m$ and $W \in \mathbb{R}^n$, let $\{e_i\}$ be a orthonormal basis for V and $\{f_i\}$ be a orthonormal basis for W. Let us denote $L(A)$ the space of linear operators on A, then the partial trace $\mathrm{Tr}_W : \mathrm{L}(V \otimes W) \to \mathrm{L}(V)$ is defined as follows: For any operator $T \in \mathrm{L}(V \otimes W)$, it can be represented by $\{e_i | i = 1, \cdots m\}$ and $\{f_i | i = 1, \cdots, n\}$:

$$T = \sum_{kl,ij} T_{kl,ij} \operatorname{vec}(e_k \otimes f_l) \operatorname{vec}(e_i \otimes f_j)^{\mathrm{H}}$$

$$(1 \le k, i \le m, \quad 1 \le l, j \le n)$$

Now we define $\mathrm{Tr}_W(T) \in \mathrm{L}(V)$ as:

$$\mathrm{Tr}_W(T) = \sum_{k,i} \sum_{j} T_{kj,ij} e_k e_i^{\mathrm{H}},$$

Also $\mathrm{Tr}_V(T) \in \mathrm{L}(W)$ as:

$$\mathrm{Tr}_V(T) = \sum_{l,j} \sum_{i} T_{il,ij} f_l f_j^{\mathrm{H}},$$

3.4 Tensor Simplification for Computational Feasibility

3.4.1 Preliminaries

Let $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{n \times d}$. We aim to explore the computational implications of working with tensor products and consider methods for "tensor simplification" to reduce computational complexity.

3.4.2 Natural Transformation

We first consider a operator $\mathbf{T} \in (\mathbf{Q} \otimes \mathbf{K}) \to (\mathbf{Q} \otimes \mathbf{K})$:

$$\mathbf{T}_{(n\times d)\times(n\times d)\times(n\times d)\times(n\times d)} = (\mathbf{Q} \otimes \mathbf{K}) \otimes (\mathbf{Q} \otimes \mathbf{K}).$$

Here, the symbol $\simeq$ denotes a form of isomorphism:

$$(\mathbf{Q} \otimes \mathbf{K}) \otimes (\mathbf{Q} \otimes \mathbf{K}) \simeq (\mathbf{Q} \to \mathbf{K}) \to (\mathbf{Q} \to \mathbf{K}).$$

3.4.3 Tensor Simplification

We have the following equivalence:

$$\mathbf{Q} \otimes \mathbf{K} \simeq \mathrm{vec}(\mathbf{Q}) \otimes \mathrm{vec}(\mathbf{K}) \simeq \mathrm{vec}(\mathbf{Q})\,\mathrm{vec}(\mathbf{K})^\top,$$

where vec is the vectorization operator. For computational efficiency, we aim to "simplify" $\mathbf{Q} \otimes \mathbf{K}$. We define a mapping:

$$\mathbf{Q} \otimes \mathbf{K} \to \mathbf{Q}\mathbf{K}^\top.$$

This leads to a new tensor $\mathbf{T}_{(n\times d)\times(n\times d)}$ given by:

$$\mathbf{T}_{(n\times d)\times(n\times d)} = (\mathbf{Q}\mathbf{K}^\top) \otimes (\mathbf{Q}\mathbf{K}^\top).$$

3.4.4 Further Simplification

We can further simplify the tensor $\mathbf{T}_{(n\times d)\times(n\times d)}$.

$$(\mathbf{Q}\mathbf{K}^\top) \otimes (\mathbf{Q}\mathbf{K}^\top) \simeq \mathrm{vec}(\mathbf{Q}\mathbf{K}^\top)\,\mathrm{vec}(\mathbf{Q}\mathbf{K}^\top)^\top,$$

We define a new mapping:

$$(\mathbf{Q}\mathbf{K}^\top) \otimes (\mathbf{Q}\mathbf{K}^\top) \to (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top.$$

Finally, we introduce a simplified tensor $\mathbf{T}_{n\times n}$ and $\mathbf{T}^{\odot}_{n\times n}$ as:

$$\mathbf{T}_{n\times n} = \mathbf{Q}\mathbf{K}^\top(\mathbf{Q}\mathbf{K}^\top)^\top,$$

$$\mathbf{T}^{\odot}_{n\times n} = \mathbf{Q}\mathbf{K}^\top \odot (\mathbf{Q}\mathbf{K}^\top)^\top.$$

3.5 Tensor Simplification Using Tensor Network Notation

Original Tensor $\mathbf{T}_{(n\times d)\times(n\times d)\times(n\times d)\times(n\times d)}$;

1. Tensor Nodes: $\mathbf{Q}$ and $\mathbf{K}$ are represented as nodes with two legs, one for each index n and d. 2. Tensor Product: $\mathbf{Q} \otimes \mathbf{K}$ is represented by tensor nodes $\mathbf{Q}$ and $\mathbf{K}$ not sharing any edge (indicating the tensor product). 3. Forming $\mathbf{T}$: This would involve a four-way tensor contraction, effectively creating a new node with four legs, one for each dimension $(n \times d)$.

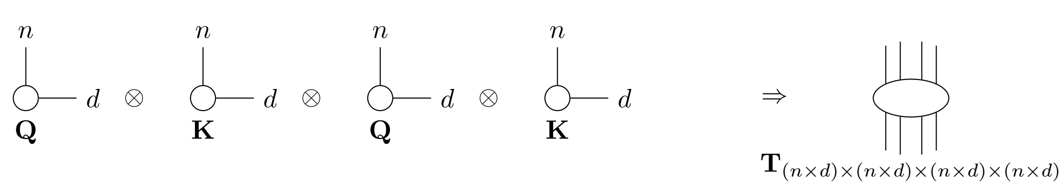
First Simplification $\mathbf{Q}\mathbf{K}^\top$;

1. Contraction: Represented by merging the d-dimensional legs of tensor nodes $\mathbf{Q}$ and $\mathbf{K}$. 2. Result: The resulting tensor node $\mathbf{Q}\mathbf{K}^\top$ will have two legs corresponding to dimensions n and n.

Further Simplification $\mathbf{T}_{n\times n}$;

1. Tensor Nodes: $\mathbf{Q}\mathbf{K}^\top$ is already available as a node with two legs (from the first simplification). 2. Contraction: Take two copies of $\mathbf{Q}\mathbf{K}^\top$ and perform a contraction along both n-dimensional legs to form $\mathbf{T}_{n\times n}$. 3. Result: The resulting tensor node $\mathbf{T}_{n\times n}$ will have two legs, one for each n-dimension.

3.5.1 Original Tensor T



3.5.2 First Simplification $\mathbf{QK}^\top$

3.5.3 Further Simplification $\mathbf{T}_{n\times n}$

4 Tensor Attention

Token Mixer, Superposition.

4.1 Tensor Attention

The input consists of queries of dimension d_q, keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed, divided by $\sqrt{d_k}$, and then renormalized using joint probability distribution.

We know that $\mathbf{Q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \cdots, \mathbf{q}_n^\top]^\top \in \mathbb{R}^{n\times d_{\text{model}}}$, suppose token size equals n.

Utilizing the parallelism using matrix form, i.e.:

$$\text{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\text{tr}(\mathbf{T}))^{-1}\mathbf{T}\mathbf{V},$$

$$\begin{aligned}\mathbf{T}_{\mathbf{Q}} &= (\mathbf{QK}^\top)(\mathbf{QK}^\top)^\top \\ &= \mathbf{QK}^\top\mathbf{KQ}^\top,\end{aligned}$$

$$\begin{aligned}\mathbf{T}_{\mathbf{K}} &= (\mathbf{QK}^\top)^\top(\mathbf{QK}^\top) \\ &= \mathbf{KQ}^\top\mathbf{QK}^\top,\end{aligned}$$

$\mathbf{T}_{\mathbf{Q}}$ and $\mathbf{T}_{\mathbf{K}}$ are symmetric positive semi-definite matrices (operators).

$$\begin{aligned}\mathbf{T}_{\mathbf{Q}}^{\odot} &= (\mathbf{QK}^\top)(\mathbf{QK}^\top)^\top \\ &= (\mathbf{QK}^\top)\odot(\mathbf{KQ}^\top),\end{aligned}$$

$$\begin{aligned}\mathbf{T}_{\mathbf{K}}^{\odot} &= (\mathbf{QK}^\top)^\top(\mathbf{QK}^\top) \\ &= (\mathbf{KQ}^\top)\odot(\mathbf{QK}^\top),\end{aligned}$$

Theorem 4.1. *Let $\mathbf{Q},\mathbf{K}\in\mathbb{R}^{n\times d}$. Then, the diagonal entries of $(\mathbf{QK}^\top)(\mathbf{QK}^\top)^\top$ are non-negative.*

Proof. Let $\mathbf{A} = \mathbf{QK}^\top$.

We consider $\mathbf{AA}^\top = (\mathbf{QK}^\top)(\mathbf{QK}^\top)^\top$.

For each diagonal entry $(\mathbf{AA}^\top)_{ii}$ of $\mathbf{AA}^\top$, it can be calculated as follows:

$$(\mathbf{AA}^\top)_{ii} = \sum_{j=1}^{n}\mathbf{A}_{ij}\mathbf{A}_{ij} = \sum_{j=1}^{n}(\mathbf{A}_{ij})^2.$$

Since $\mathbf{A}_{ij}$ is a real number, $(\mathbf{A}_{ij})^2$ is non-negative. Therefore, summing up these non-negative values will yield a non-negative number.

Hence, all diagonal entries of $\mathbf{A}\mathbf{A}^\top$ are non-negative. $\square$

Theorem 4.2. *Let $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{n\times d}$. Then, the diagonal entries of $(\mathbf{Q}\mathbf{K}^\top)^\top(\mathbf{Q}\mathbf{K}^\top)$ are non-negative.*

Proof. The same as the last proof. $\square$

4.1.1 Extremely fast to compute, Linear $O(n)$ complexity

The original expression $(\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top\mathbf{V}$ can be expanded as follows:

$$\begin{aligned}(\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top\mathbf{V} &= (\mathbf{Q}\mathbf{K}^T)(\mathbf{K}\mathbf{Q}^\top)\mathbf{V} \\ &= \mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{V} \\ &= \mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})\end{aligned}$$

Using these associative operations, the matrix multiplication complexity becomes $M(n \times d) \times [M(d \times n) \times M(n \times d)] \times [M(d \times n) \times M(n \times d)]$

Additionally, note that the trace of a matrix is invariant under cyclic permutations, i.e., $\text{tr}(\mathbf{ABC}) = \text{tr}(\mathbf{CAB}) = \text{tr}(\mathbf{BCA})$. Using this property, we can rewrite the expression as:

$$\text{tr}(\mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top) = \text{tr}(\mathbf{K}\mathbf{Q}^\top\mathbf{Q}\mathbf{K}^\top) = \text{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q})$$

Lemma 4.3. *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be square matrices such that $\mathbf{A} = \mathbf{B}$. Then the following equality holds:*

$$\operatorname{tr}(\mathbf{AB}) = \operatorname{sum}(\mathbf{A} \odot \mathbf{B}) \tag{2}$$

Proof. 1. The trace of $\mathbf{AB}$ is defined as the sum of the diagonal elements of $\mathbf{AB}$:

$$\operatorname{tr}(\mathbf{AB}) = \sum_{i=1}^{n} (\mathbf{AB})_{ii}$$

2. Expanding $(\mathbf{AB})_{ii}$ using the definition of matrix multiplication, we get:

$$(\mathbf{AB})_{ii} = \sum_{j=1}^{n} \mathbf{A}_{ij} \times \mathbf{B}_{ji}$$

3. Substituting this into the expression for $\operatorname{tr}(\mathbf{AB})$:

$$\operatorname{tr}(\mathbf{AB}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij} \times \mathbf{B}_{ji}$$

4. Since $\mathbf{A} = \mathbf{B}$, $\mathbf{A}_{ij} = \mathbf{B}_{ij}$ for all i, j. Therefore, we can rewrite $\operatorname{tr}(\mathbf{AB})$ as:

$$\operatorname{tr}(\mathbf{AB}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij} \times \mathbf{A}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij} \odot \mathbf{A}_{ij}$$

5. This can be compactly written as $\operatorname{sum}(\mathbf{A} \odot \mathbf{A})$ or $\operatorname{sum}(\mathbf{A} \odot \mathbf{B})$ since $\mathbf{A} = \mathbf{B}$.

Thus, we have proved that $\operatorname{tr}(\mathbf{AB}) = \operatorname{sum}(\mathbf{A} \odot \mathbf{B})$ when $\mathbf{A} = \mathbf{B}$. $\square$

The expression $\operatorname{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q})$ involves $\mathbf{K}^\top\mathbf{K}$ and $\mathbf{Q}^\top\mathbf{Q}$, both of which are square matrices of size $d \times d$. The trace of the product $\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q}$ can be computed as the sum of element-wise products of $\mathbf{K}^\top\mathbf{K}$ and $\mathbf{Q}^\top\mathbf{Q}$.

Lemma 4.4. *Let $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{n \times d}$. Then the following equality holds:*

$$\mathrm{tr}((\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top) = \mathrm{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q}) = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q})) \tag{3}$$

Proof. 1. First, let $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$. Then $\mathbf{A}\mathbf{A}^\top = (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top$.
Expanding $\mathbf{A}\mathbf{A}^\top$ gives:

$$\mathbf{A}\mathbf{A}^\top = \mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top$$

Taking the trace on both sides, we get:

$$\mathrm{tr}(\mathbf{A}\mathbf{A}^\top) = \mathrm{tr}(\mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top) = \mathrm{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q})$$

2. Based on Lemma 4.3, we have:

$$\mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q})) = \mathrm{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q})$$

Therefore, we have:

$$\mathrm{tr}((\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top) = \mathrm{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q}) = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))$$

□

In summary:

$$\mathrm{tr}((\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top) = \mathrm{tr}(\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top\mathbf{Q}) = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))$$

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\mathbf{Q}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})$$

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\mathbf{K}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})(\mathbf{K}^\top\mathbf{V})$$

Here, $\odot$ denotes element-wise multiplication.

4.1.2 Ensuring non-negative: $\mathrm{ReLU}(\mathbf{T})$

Notice that

$$\mathrm{tr}(\mathrm{ReLU}(\mathbf{T})) = \mathrm{tr}(\mathbf{T}),$$

Hence we have:

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\textbf{ReLU},\mathbf{Q}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\,\mathrm{ReLU}[\mathbf{Q}(\mathbf{K}^\top\mathbf{K})\mathbf{Q}^\top]\mathbf{V}$$

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\textbf{ReLU},\mathbf{K}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K}) \odot (\mathbf{Q}^\top\mathbf{Q}))^{-1}\,\mathrm{ReLU}[\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})\mathbf{K}^\top]\mathbf{V}$$

May have different places to introduce ReLU, but some of them may have $O(n^2)$ complexity.

4.1.3 Introducing element-wise exponential kernel

$$\mathrm{TensorAttention}_{\text{ElementExponential}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{element-wise-exp}((\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T})\mathbf{V},$$

4.1.4 Introducing matrix exponential kernel (seems equivalent to having more transformer layers)

$$\mathrm{TensorAttention}_{\text{MatrixExponential}}(\mathbf{Q},\mathbf{K},\mathbf{V}) = e^{(\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T}}\mathbf{V} = (\mathbf{I}+\hat{\mathbf{T}}+\frac{\hat{\mathbf{T}}^2}{2!}+\frac{\hat{\mathbf{T}}^3}{3!}+\cdots)\mathbf{V},$$

Can use Taylor Expansion or Pade Approximation.

The Pade Approximation is a technique used to approximate a function by a ratio of two polynomial functions. In the context of matrix exponentials, it is often used to approximate the matrix exponential function $e^{\mathbf{A}}$ as a ratio of two polynomial matrix functions:

$$e^{\mathbf{A}} \approx \mathbf{P}(\mathbf{A})/\mathbf{Q}(\mathbf{A})$$

Here, $\mathbf{P}(\mathbf{A})$ and $\mathbf{Q}(\mathbf{A})$ are polynomials of the matrix $\mathbf{A}$. These polynomials are usually obtained by truncating the Taylor series expansion of $e^{\mathbf{A}}$ and then using the coefficients to find the best ratio of polynomials that approximate the function within a certain range.

The advantage of using the Pade Approximation is that it often provides a better approximation over a wider range than a truncated Taylor series. This is particularly true near singularities or other points where the Taylor series converges slowly.

Steps to Compute Pade Approximation for $e^{\mathbf{A}}$

1. Choose the Degree: Decide the degree of the polynomials in the numerator and the denominator. Let's say m for the numerator and n for the denominator.

2. Compute Series Coefficients: Compute the first $m+n$ coefficients of the Taylor series expansion of $e^{\mathbf{A}}$.

3. Form Polynomials: Use these coefficients to form the polynomials $\mathbf{P}(\mathbf{A})$ and $\mathbf{Q}(\mathbf{A})$.

4. Compute the Approximation: The Pade approximation is then $\mathbf{P}(\mathbf{A})/\mathbf{Q}(\mathbf{A})$.

5. Matrix Division: Typically, one doesn't actually perform matrix division but solves a system of linear equations to find $\mathbf{Q}(\mathbf{A})^{-1}\mathbf{P}(\mathbf{A})$.

Example

For a simple example, the $[2/2]$ Pade Approximation of $e^{\mathbf{A}}$ is:

$$e^{\mathbf{A}} \approx \frac{(1+\frac{\mathbf{A}}{2}+\frac{\mathbf{A}^2}{6})}{(1-\frac{\mathbf{A}}{2}+\frac{\mathbf{A}^2}{6})}.$$

4.1.5 Diagonal Renormalization

Lemma 4.5. *Given matrices $\mathbf{Q} \in \mathbb{R}^{n\times d}$ and $\mathbf{K} \in \mathbb{R}^{n\times d}$, the diagonal elements of $\mathbf{M}\mathbf{M}^\top$, where $\mathbf{M} = \mathbf{Q}\mathbf{K}^\top$, can be efficiently computed as follows:*

$$\mathrm{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{k=1}^{d}\sum_{l=1}^{d}\mathbf{Q}[i,k]\mathbf{Q}[i,l]\mathbf{N}[k,l],$$

where $\mathbf{N} = \mathbf{K}^\top\mathbf{K}$.

Proof. Step 1: Expression of $\mathbf{M}\mathbf{M}^\top$;

The i-th diagonal element of $\mathbf{M}\mathbf{M}^\top$ can be expressed as

$$\mathrm{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{j=1}^{n}\mathbf{M}[i,j]\mathbf{M}[i,j]$$

Substituting $\mathbf{M}[i,j] = \mathbf{Q}[i,:]\cdot\mathbf{K}[j,:]$, we get

$$\text{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{j=1}^{n} (\mathbf{Q}[i,:] \cdot \mathbf{K}[j,:])^2$$

Step 2: Expanding the Dot Product;

Expanding the dot product, we have:

$$\text{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{j=1}^{n} \left(\sum_{k=1}^{d} \mathbf{Q}[i,k]\mathbf{K}[j,k] \right)^2$$

Further expanding, we get:

$$\text{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{j=1}^{n} \left(\sum_{k=1}^{d} \sum_{l=1}^{d} \mathbf{Q}[i,k]\mathbf{K}[j,k]\mathbf{Q}[i,l]\mathbf{K}[j,l] \right)$$

Step 3: Factorizing the Terms;

Notice that $\mathbf{Q}[i,k]\mathbf{Q}[i,l]$ does not depend on j. We can rewrite as:

$$\text{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{k=1}^{d} \sum_{l=1}^{d} \mathbf{Q}[i,k]\mathbf{Q}[i,l] \left(\sum_{j=1}^{n} \mathbf{K}[j,k]\mathbf{K}[j,l] \right)$$

By definition, $\sum_{j=1}^{n} \mathbf{K}[j,k]\mathbf{K}[j,l]$ is the (k,l)-th element of $\mathbf{N} = \mathbf{K}^\top\mathbf{K}$.

Thus, we can write:

$$\text{diag}(\mathbf{M}\mathbf{M}^\top)[i] = \sum_{k=1}^{d} \sum_{l=1}^{d} \mathbf{Q}[i,k]\mathbf{Q}[i,l]\mathbf{N}[k,l]$$

□

Algorithm 1 Efficiently Compute Diagonal Elements

Require: Matrices $\mathbf{Q} \in \mathbb{R}^{n \times d}$, $\mathbf{K} \in \mathbb{R}^{n \times d}$
Ensure: Diagonal elements $\text{diag}(\mathbf{M}\mathbf{M}^\top)$

function COMPUTEDIAGONAL($\mathbf{Q}$, $\mathbf{K}$)
  $\mathbf{N} \leftarrow \mathbf{K}^\top\mathbf{K}$
  Initialize diag_elements as an empty array of size n
  for $i = 1$ to n **do**
    diag_elements$[i] \leftarrow \sum_{k=1}^{d} \sum_{l=1}^{d} \mathbf{Q}[i,k]\mathbf{Q}[i,l]\mathbf{N}[k,l]$
  end for
  return diag_elements
end function

To assess the computational complexity, let's consider the two different approaches separately: vanilla matrix multiplication and the efficient method proposed in the lemma.

Vanilla Matrix Operations

1. Computing $\mathbf{M} = \mathbf{Q}\mathbf{K}^\top$: The complexity is $O(n \times d \times n) = O(n^2 d)$. 2. Computing $\mathbf{M}\mathbf{M}^\top$: The resulting matrix $\mathbf{M}$ is of size $n \times n$. Multiplying it by its transpose will have a complexity of $O(n \times n \times n) = O(n^3)$. 3. Extracting the diagonal: This will take $O(n)$.

So, the overall complexity for vanilla operations is $O(n^2 d + n^3 + n) = O(n^3 + n^2 d)$.

Efficient Method

1. Computing $\mathbf{N} = \mathbf{K}^\top \mathbf{K}$: This takes $O(d \times n \times d) = O(d^2 n)$. 2. Computing each diagonal element: The sum over k and l takes $O(d \times d) = O(d^2)$ for each diagonal element. 3. Computing all diagonal elements: For n diagonal elements, this will take $O(n \times d^2) = O(nd^2)$.

So, the overall complexity for the efficient method is $O(d^2 n + nd^2) = O(nd^2)$.

Comparison

1. Vanilla: $O(n^3 + n^2 d)$

2. Efficient: $O(nd^2)$

The efficient method is advantageous especially when $d^2 \ll n^2$ or when $d \ll n$, which is often the case in many practical applications. This could be particularly useful in scenarios where the dimensionality d is much smaller than the number of samples n.

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\text{DiagRenormalization},\mathbf{Q}} = \mathrm{diag}(\mathbf{T}_\mathbf{Q})^{-1} \mathbf{Q}(\mathbf{K}^\top \mathbf{K})(\mathbf{Q}^\top \mathbf{V})$$

$$\mathrm{TensorAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\text{DiagRenormalization},\mathbf{K}} = \mathrm{diag}(\mathbf{T}_\mathbf{K})^{-1} \mathbf{K}(\mathbf{Q}^\top \mathbf{Q})(\mathbf{K}^\top \mathbf{V})$$

4.1.6 Row Renormalization

$$\begin{aligned}\mathrm{TensorAttention}&(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\text{RowRenormalization},\mathbf{Q}} \\ &= \mathrm{diag}(\mathbf{T}_\mathbf{Q} \mathbf{1_d})^{\mathbf{-1}} \mathbf{Q}(\mathbf{K}^\top \mathbf{K})(\mathbf{Q}^\top \mathbf{V})\end{aligned}$$

$$\begin{aligned}\mathrm{TensorAttention}&(\mathbf{Q}, \mathbf{K}, \mathbf{V})_{\text{RowRenormalization},\mathbf{K}} \\ &= \mathrm{diag}(\mathbf{T}_\mathbf{K} \mathbf{1_d})^{\mathbf{-1}} \mathbf{K}(\mathbf{Q}^\top \mathbf{Q})(\mathbf{K}^\top \mathbf{V})\end{aligned}$$

4.1.7 Introducing unidirectional attention mask

$$\mathrm{TensorAttention}_{\rightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\mathrm{tr}(\mathbf{T}))^{-1}\, \mathrm{tril}(\mathbf{T})\mathbf{V},$$

4.1.8 Introducing L_2 normalization

$$\mathrm{TensorAttention}_{\text{Residual}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\mathbf{T} + \lambda \cdot \mathrm{tr}(\mathbf{T}) \cdot \mathbf{I})\mathbf{V},$$

Notice that

$$\mathrm{tr}(\mathbf{T}) = \|\mathbf{Q}\mathbf{K}^\top\|_F^2,$$

Deeply connected to Compressed Sensing.

4.1.9 Generalize to Complex-valued Tensors

$$\mathbf{T}_\mathbf{Q} = (\mathbf{Q}\mathbf{K}^\mathrm{H})(\mathbf{Q}\mathbf{K}^\mathrm{H})^\mathrm{H}$$

$$\mathbf{T}_\mathbf{K} = (\mathbf{Q}\mathbf{K}^\mathrm{H})^\mathrm{H}(\mathbf{Q}\mathbf{K}^\mathrm{H})$$

4.2 Introducing Hardmard Product

$$\mathbf{T}_\mathbf{Q}^\odot = (\mathbf{Q}\mathbf{K}^\mathrm{H}) \odot (\mathbf{Q}\mathbf{K}^\mathrm{H})^\mathrm{H}$$

$$\mathbf{T}_\mathbf{K}^\odot = (\mathbf{Q}\mathbf{K}^\mathrm{H})^\mathrm{H} \odot (\mathbf{Q}\mathbf{K}^\mathrm{H})$$

5 Tensor Interaction

The input consists of queries of dimension d_q, keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed, divided by $\sqrt{d_k}$, and then renormalized using joint probability distribution.

We know that $\mathbf{Q} = [\mathbf{q}_1^\top, \mathbf{q}_2^\top, \cdots, \mathbf{q}_n^\top]^\top \in \mathbb{R}^{n \times d_{\text{model}}}$, suppose token size equals n.

Utilizing the parallelism using matrix form, i.e.:

$$\text{TensorInteraction}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\text{tr}(\mathbb{T}))^{-1} \mathbb{T} \mathbf{V}^\top,$$

$$\begin{aligned} \mathbb{T}_{\mathbf{Q}} &= (\mathbf{Q}^\top \mathbf{K})(\mathbf{Q}^\top \mathbf{K})^\top \\ &= (\mathbf{Q}^\top \mathbf{K})(\mathbf{K}^\top \mathbf{Q}), \end{aligned}$$

$$\begin{aligned} \mathbb{T}_{\mathbf{K}} &= (\mathbf{Q}^\top \mathbf{K})^\top (\mathbf{Q}^\top \mathbf{K}) \\ &= (\mathbf{K}^\top \mathbf{Q})(\mathbf{Q}^\top \mathbf{K}), \end{aligned}$$

$\mathbb{T}_{\mathbf{Q}}$ and $\mathbb{T}_{\mathbf{K}}$ are $\mathbb{R}^{d \times d}$ symmetric positive semi-definite matrices (operators). It is inherently $O(n)$ Training and $O(1)$ Inference.

$$\begin{aligned} \mathbb{T}_{\mathbf{Q}}^{\odot} &= (\mathbf{Q}^\top \mathbf{K})(\mathbf{Q}^\top \mathbf{K})^\top \\ &= (\mathbf{Q}^\top \mathbf{K}) \odot (\mathbf{K}^\top \mathbf{Q}), \end{aligned}$$

$$\begin{aligned} \mathbb{T}_{\mathbf{K}}^{\odot} &= (\mathbf{Q}^\top \mathbf{K})^\top \odot (\mathbf{Q}^\top \mathbf{K}) \\ &= (\mathbf{K}^\top \mathbf{Q}) \odot (\mathbf{Q}^\top \mathbf{K}). \end{aligned}$$

6 Experiments

Consider two finite-dimensional vector spaces $\mathcal{X}$ and $\mathcal{Y}$ to form a tensor product space: $\mathcal{X} \otimes \mathcal{Y}$. $(\text{Hom}(\mathcal{X}, \mathcal{Y}) \cong \mathcal{Y} \otimes \mathcal{X}^* \cong \mathcal{X} \otimes \mathcal{Y}^* \cong \mathcal{X} \otimes \mathcal{Y})$

Dot-Product Attention is not Axiomatic

$$\text{Att}_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{D}^{-1} \mathbf{A} \mathbf{V}, \quad \mathbf{A} = \exp\left(\mathbf{Q}\mathbf{K}^\top / \sqrt{d}\right), \quad \mathbf{D} = \text{diag}\left(\mathbf{A} \mathbf{1}_L\right).$$

Tensor Attention is All You Need

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T}\mathbf{V},$$

$$\mathbf{T}_{\mathbf{Q}} = (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top = \mathbf{Q}\mathbf{K}^\top\mathbf{K}\mathbf{Q}^\top$$

$$\mathbf{T}_{\mathbf{K}} = (\mathbf{Q}\mathbf{K}^\top)^\top(\mathbf{Q}\mathbf{K}^\top) = \mathbf{K}\mathbf{Q}^\top\mathbf{Q}\mathbf{K}^\top$$

$$\mathbf{T}_{\mathbf{Q}}^{\odot} = (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top = (\mathbf{Q}\mathbf{K}^\top)\odot(\mathbf{K}\mathbf{Q}^\top)$$

$$\mathbf{T}_{\mathbf{K}}^{\odot} = (\mathbf{Q}\mathbf{K}^\top)^\top(\mathbf{Q}\mathbf{K}^\top) = (\mathbf{K}\mathbf{Q}^\top)\odot(\mathbf{Q}\mathbf{K}^\top)$$

May or may not introduce element-wise ReLU:

$$\mathbf{T} \leftarrow \mathrm{ReLU}(\mathbf{T})$$

May or may not introduce matrix exponential:

$$\mathrm{TensorAttention}_{\text{MatrixExponential}}(\mathbf{Q},\mathbf{K},\mathbf{V}) = e^{\mathrm{tr}(\mathbf{T})^{-1}\mathbf{T}}\mathbf{V} = (\mathbf{I}_n + \hat{\mathbf{T}} + \frac{\hat{\mathbf{T}}^2}{2!} + \frac{\hat{\mathbf{T}}^3}{3!} + \cdots)\mathbf{V},$$

Linear Tensor Attention is All You Need

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbf{T}))^{-1}\mathbf{T}\mathbf{V},$$

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\mathbf{Q}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K})\odot(\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})$$

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\mathbf{K}} = \mathrm{sum}((\mathbf{K}^\top\mathbf{K})\odot(\mathbf{Q}^\top\mathbf{Q}))^{-1}\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})(\mathbf{K}^\top\mathbf{V})$$

Different Normalization Methods

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\text{DiagNormalization},\mathbf{Q}} = \mathrm{diag}(\mathbf{T}_{\mathbf{Q}})^{-1}\mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})$$

$$\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\text{DiagNormalization},\mathbf{K}} = \mathrm{diag}(\mathbf{T}_{\mathbf{K}})^{-1}\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})(\mathbf{K}^\top\mathbf{V})$$

$$\begin{aligned}\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\text{RowNormalization},\mathbf{Q}} \\ = \mathrm{diag}(\mathbf{T}_{\mathbf{Q}}\mathbf{1}_{\mathbf{n}})^{\mathbf{-1}}\mathbf{Q}(\mathbf{K}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{V})\end{aligned}$$

$$\begin{aligned}\mathrm{TensorAttention}(\mathbf{Q},\mathbf{K},\mathbf{V})_{\text{RowNormalization},\mathbf{K}} \\ = \mathrm{diag}(\mathbf{T}_{\mathbf{K}}\mathbf{1}_{\mathbf{n}})^{\mathbf{-1}}\mathbf{K}(\mathbf{Q}^\top\mathbf{Q})(\mathbf{K}^\top\mathbf{V})\end{aligned}$$

Tensor Interaction is All You Need

$$\mathrm{TensorInteraction}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\mathrm{tr}(\mathbb{T}))^{-1}\mathbb{T}\mathbf{V}^\top,$$

$$\mathbb{T}_{\mathbf{Q}} = (\mathbf{Q}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{K})^\top = (\mathbf{Q}^\top\mathbf{K})(\mathbf{K}^\top\mathbf{Q})$$

$$\mathbb{T}_{\mathbf{K}} = (\mathbf{Q}^\top\mathbf{K})^\top(\mathbf{Q}^\top\mathbf{K}) = (\mathbf{K}^\top\mathbf{Q})(\mathbf{Q}^\top\mathbf{K})$$

$$\mathbb{T}_{\mathbf{Q}}^{\odot} = (\mathbf{Q}^\top\mathbf{K})(\mathbf{Q}^\top\mathbf{K})^\top = (\mathbf{Q}^\top\mathbf{K})\odot(\mathbf{K}^\top\mathbf{Q}),$$

$$\mathbb{T}_{\mathbf{K}}^{\odot} = (\mathbf{Q}^\top\mathbf{K})^\top\odot(\mathbf{Q}^\top\mathbf{K}) = (\mathbf{K}^\top\mathbf{Q})\odot(\mathbf{Q}^\top\mathbf{K}).$$

May or may not introduce element-wise ReLU:

$$\mathbb{T} \leftarrow \mathrm{ReLU}(\mathbb{T})$$

May or may not introduce matrix exponential:

$$\mathrm{TensorInteraction}_{\text{MatrixExponential}}(\mathbf{Q},\mathbf{K},\mathbf{V}) = e^{\mathrm{tr}(\mathbb{T})^{-1}\mathbb{T}}\mathbf{V} = (\mathbf{I}_d+\hat{\mathbb{T}}+\frac{\hat{\mathbb{T}}^2}{2!}+\frac{\hat{\mathbb{T}}^3}{3!}+\cdots)\mathbf{V},$$

If the **TensorAttention** and **TensorInteraction** work, we will embrace the next-generation AI, which has Tensor Categorical Guarantees. Deeply connected to Linear Logic, Dependent Type Theory, and even Feynman Diagrams.

7 Related Work

7.1 Large Language Models

Large Language Models (LLMs) represent a significant leap in the field of AI and machine learning, particularly in understanding and generating natural languages and programming languages.

Large Language Models. Language models have evolved into extremely large-scale neural networks Devlin et al. (2018); Raffel et al. (2020); Radford et al. (2018, 2019); Brown et al. (2020); OpenAI (2023), which have shown impressive results across various tasks. GPT-3 Brown et al. (2020) and its successors, such as Gopher Rae et al. (2021), PaLM Chowdhery et al. (2022), GLaM Du et al. (2022), Chinchilla Hoffmann et al. (2022), Megatron–Turing NLG Smith et al. (2022), LaMDA Thoppilan et al. (2022), OPT Zhang et al. (2022b), LLaMA Touvron et al. (2023), PaLM 2 Anil et al. (2023) and GPT-4 OpenAI (2023), have demonstrated that large auto-regressive language models can achieve high-quality results without extensive task-specific data collection or parameter updates.

7.2 Vision Foundation Models

Convolutional Neural Networks and Vision Transformers. Convolutional Neural Networks (CNNs) have long been the standard for visual tasks, supported by significant research (LeCun et al., 1998; Krizhevsky et al., 2012; Szegedy et al., 2015; He et al., 2016; Xie et al., 2017; Howard et al., 2017; Tan & Le, 2019; Liu et al., 2022b). In parallel, Transformers have revolutionized language tasks (Devlin et al., 2018; Radford et al., 2018; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Chowdhery et al., 2022; Fedus et al., 2022; Ouyang et al., 2022). Extending this, Vision Transformers (ViT) proposed by Dosovitskiy et al. (2021) have shown comparable or superior performance to CNNs in image recognition, inspiring further model advancements (Liu et al., 2021; Wang et al., 2021, 2022b; Dong et al., 2022; Ali et al., 2021; Zhou et al., 2021b; Han et al., 2021; Liu et al., 2022a; Riquelme et al., 2021; Zhai et al., 2022; Dai et al., 2021).

Self-Supervised Learning: Contrastive, Non-Contrastive, and Masked Image Modeling. Self-Supervised Learning (SSL) methods, encompassing both contrastive and non-contrastive approaches, learn rich representations by utilizing diverse views or augmentations of inputs, independent of human-annotated labels (Chen et al., 2020; Hjelm et al., 2018; Wu et al., 2018; Tian et al., 2019; Chen & He, 2021; Gao et al., 2021; Bachman et al., 2019; Oord et al., 2018; Ye et al., 2019; Henaff, 2020; Misra & Maaten, 2020; Caron et al., 2020; HaoChen et al., 2021; Caron et al., 2021; Li et al., 2021; Zbontar et al., 2021; Tsai et al., 2021; Bardes et al., 2021; Tian et al., 2020; Robinson et al., 2021; Dubois et al., 2022). These methods have been demonstrated to surpass supervised learning in various tasks. Simultaneously, Masked Image Modeling (MIM), such as MAE and SimMIM, has emerged as an innovative approach in the visual domain, showing promising results in vision tasks (He et al., 2022b; Xie et al., 2022a; Huang et al., 2022; Ren et al., 2023; Bao et al., 2022; Cao et al., 2022). MIM, taking inspiration from Masked Language Modeling (MLM) in NLP and exemplified by BERT (Devlin et al., 2018), has been instrumental in advancing visual representation learning.

Advancements and Innovations in Masked Image Modeling. In the realm of MIM, techniques like iBOT (Zhou et al., 2021a), SimMIM (Xie et al., 2022b), and MAE (He et al., 2022a) have been particularly influential. MAE, diverging from BERT-like methodologies, employs a unique approach with a pixel-level reconstruction loss and a fully non-linear encoder-decoder architecture. This design choice, which includes the exclusion of masked tokens from the encoder input, enables MAE to intricately capture spatial information within images. Such advancements highlight the ongoing evolution and effectiveness of MIM methods in various applications, despite their theoretical underpinnings still being actively explored.

Theoretical Understanding of Self-Supervised Learning. The field of Self-Supervised Learning (SSL) has witnessed significant theoretical advancements, particularly in the context of contrastive learning and its mechanisms. A body of research (Arora et al., 2019; HaoChen et al., 2021, 2022; Tosh et al., 2020, 2021; Lee et al., 2020; Wang et al., 2022c; Nozawa & Sato, 2021; Huang et al., 2021; Tian, 2022; Hu et al., 2022; Tan et al., 2023) has been dedicated to exploring the workings of contrastive loss. Central to this is the study by (Wang & Isola, 2020), which provides an insightful analysis of the InfoNCE loss, breaking it down into alignment and uniformity terms and thereby deepening the understanding of SSL. Other studies like HaoChen et al. (2021); Wang et al. (2022c); Tan et al. (2023) examine SSL methods from a spectral graph perspective, while Saunshi et al. (2022); HaoChen & Ma (2022) highlight the role of inductive bias in shaping SSL performance. Additionally, Cabannes et al. (2023) presents a comprehensive framework that intertwines data augmentation techniques, network architectures, and training algorithms in SSL.

Feature decorrelation-based methods within SSL have also been thoroughly investigated. Studies such as Wen & Li (2022); Tian et al. (2021); Garrido et al. (2022); Balestriero & LeCun (2022); Tsai et al. (2021); Pokle et al. (2022); Tao et al. (2022); Lee et al. (2021) have contributed significantly to this area. Notably, Balestriero & LeCun (2022) draws parallels between SimCLR, Barlow Twins, and VICReg with classical unsupervised learning techniques. The resilience of methods like SimSiam against collapse, as analyzed by Tian et al. (2021), and the comparative study of loss landscapes in Pokle et al. (2022) reveal intricate details of these methods. Moreover, connections between Barlow Twins' criterion and the Hilbert-Schmidt Independence Criterion are established in Tsai et al. (2021), while (Huang et al., 2021; Wen & Li, 2021) explore the theoretical aspects of data augmentation in sample-contrastive learning.

Compared to these, the theoretical understanding of masked image modeling, a different strand of SSL, is relatively nascent. Cao et al. (2022) and Zhang et al. (2022a) have offered novel perspectives, using the integral kernel and masked graph concepts, respectively, to understand the Masked AutoEncoder (MAE). Recent work by Kong et al. (2023) demonstrates how MAE effectively identifies specific latent variables through a hierarchical model.

7.3 Multi-modal Foundation Models (Vision-Language Models)

CLIP-based Models. The intersection of vision and language models has been significantly advanced by the development of CLIP (Contrastive Language-Image Pre-training) (Radford et al., 2021). CLIP learns visual models from natural language supervision, marking a departure from the traditional approach of training computer vision systems on fixed object categories. This method leverages a vast dataset of image-text pairs to predict which caption matches which image, enabling zero-shot transfer to various computer vision tasks. Building upon CLIP's foundation, LaCLIP enhances the original model by incorporating language rewrites. This process introduces diverse textual descriptions for the same images, significantly improving the model's performance on multiple benchmarks including zero-shot accuracy on ImageNet (Fan et al., 2023). Such advancements demonstrate the robustness and versatility of CLIP-based models in bridging the gap between vision and language.

Advancements with Large Language Models. Recent trends in vision-language models are increasingly leveraging large language models (LLMs) for enhanced multi-modal understanding. For instance, BLIP-2 (Li et al., 2023) introduces a novel pre-training strategy that combines off-the-shelf frozen image encoders with large language models. This approach, which employs a Querying Transformer and a two-stage pre-training, leads to state-of-the-art performance in various vision-language tasks with substantially fewer trainable parameters. Similarly, BEIT-3 (Wang et al., 2022a) pushes the boundaries of multimodal foundation models. It uses Multiway Transformers and a unified pretraining approach on images, texts, and image-text pairs, achieving top-tier results across a wide range of tasks. Recently, MiniGPT-4 (Zhu et al., 2023) exemplifies the integration of advanced LLMs in vision-language models. By aligning a visual encoder with the Vicuna LLM, MiniGPT-4 demonstrates capabilities like detailed image description generation, creative writing inspired by images, and even instructional content based on visual input. This model emphasizes the importance of high-quality, aligned datasets and the efficiency of training primarily a projection layer for aligning image-text pairs. Together, these advancements underscore the significant role of large language models in enhancing vision-language understanding and generative capabilities.

8 Conclusion

In this work, we introduce the Tensor Attention Mechanism and Tensor Interaction Mechanism, which are deeply rooted in Tensor Category Theory, from the first principles of universal properties of the tensor product. In conclusion, our work not only marks a significant advancement in the field of deep learning but also bridges the gap between advanced mathematical theories and practical AI applications. It is a step towards realizing the full potential of AI, guided by the principles of mathematical rigor, and can even generalized into the quantum realm.

References

Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., and Jégou, H. Xcit: Cross-covariance image transformers. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 20014–20027, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/a655fbe4b8d7439994aa37ddad80de56-Abstract.html`. 15

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 15

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 2019. 16

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 16

Balestriero, R. and LeCun, Y. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*, 2022. 16

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: BERT pre-training of image transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=p-BhZSz59o4`. 16

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 16

Bradley, T.-D. *At the interface of algebra and statistics*. PhD thesis, City University of New York, 2020. 5

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 15

Cabannes, V., Kiani, B. T., Balestriero, R., LeCun, Y., and Bietti, A. The ssl interplay: Augmentations, inductive bias, and generalization. *arXiv preprint arXiv:2302.02774*, 2023. 16

Cao, S., Xu, P., and Clifton, D. A. How to understand masked autoencoders. *arXiv preprint arXiv:2202.03670*, 2022. 16

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 16

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 16

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020. 16

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 16

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 15

Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 3965–3977, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/20568692db622456cc42a2e853ca21f8-Abstract.html`. 15

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 15, 16

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., and Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 12114–12124. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01181. URL `https://doi.org/10.1109/CVPR52688.2022.01181`. 15

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`. 15

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569. PMLR, 2022. 15

Dubois, Y., Hashimoto, T., Ermon, S., and Liang, P. Improving self-supervised learning by characterizing idealized representations. *arXiv preprint arXiv:2209.06235*, 2022. 16

Fan, L., Krishnan, D., Isola, P., Katabi, D., and Tian, Y. Improving clip training with language rewrites. *arXiv preprint arXiv:2305.20088*, 2023. 17

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022. 15

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021. 16

Garrido, Q., Chen, Y., Bardes, A., Najman, L., and Lecun, Y. On the duality between contrastive and non-contrastive self-supervised learning. *arXiv preprint arXiv:2206.02574*, 2022. 16

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., and Wang, Y. Transformer in transformer. *CoRR*, abs/2103.00112, 2021. URL `https://arxiv.org/abs/2103.00112`. 15

HaoChen, J. Z. and Ma, T. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022. 16

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34: 5000–5011, 2021. 16

HaoChen, J. Z., Wei, C., Kumar, A., and Ma, T. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. *Advances in Neural Information Processing Systems*, 2022. 16

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL `https://doi.org/10.1109/CVPR.2016.90`. 15

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022a. 16

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15979–15988. IEEE, 2022b. doi: 10.1109/CVPR52688.2022.01553. URL `https://doi.org/10.1109/CVPR52688.2022.01553`. 16

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020. 16

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 16

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 15

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL `http://arxiv.org/abs/1704.04861`. 15

Hu, T., Liu, Z., Zhou, F., Wang, W., and Huang, W. Your contrastive learning is secretly doing stochastic neighbor embedding. *arXiv preprint arXiv:2205.14814*, 2022. 16

Huang, L., You, S., Zheng, M., Wang, F., Qian, C., and Yamasaki, T. Green hierarchical vision transformer for masked image modeling. *CoRR*, abs/2205.13515, 2022. doi: 10.48550/arXiv.2205.13515. URL `https://doi.org/10.48550/arXiv.2205.13515`. 16

Huang, W., Yi, M., and Zhao, X. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021. 16

Kong, L., Ma, M. Q., Chen, G., Xing, E. P., Chi, Y., Morency, L.-P., and Zhang, K. Understanding masked autoencoders via hierarchical latent variable models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7918–7928, 2023. 16

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 1106–1114, 2012. 15

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL `https://doi.org/10.1109/5.726791`. 15

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting what you already know helps: Provable self-supervised learning. *arXiv preprint arXiv:2008.01064*, 2020. 16

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021. 16

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 17

Li, Y., Pogodin, R., Sutherland, D. J., and Gretton, A. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34:15543–15556, 2021. 16

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 15

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9992–10002. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00986. URL `https://doi.org/10.1109/ICCV48922.2021.00986`. 15

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. Swin transformer V2: scaling up capacity and resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11999–12009. IEEE, 2022a. doi: 10.1109/CVPR52688.2022.01170. URL `https://doi.org/10.1109/CVPR52688.2022.01170`. 15

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11966–11976. IEEE, 2022b. doi: 10.1109/CVPR52688.2022.01167. URL `https://doi.org/10.1109/CVPR52688.2022.01167`. 15

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020. 16

Nozawa, K. and Sato, I. Understanding negative samples in instance discriminative self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:5784–5797, 2021. 16

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 16

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 15

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 15

Pokle, A., Tian, J., Li, Y., and Risteski, A. Contrasting the landscape of contrastive and non-contrastive learning. *arXiv preprint arXiv:2203.15702*, 2022. 16

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. *openai.com*, 2018. 15

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 15

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. 16

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021. 15

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 15

Ren, S., Wei, F., Albanie, S., Zhang, Z., and Hu, H. Deepmim: Deep supervision for masked image modeling. *arXiv preprint arXiv:2303.08817*, 2023. 16

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8583–8595, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/48237d9f2dea8c74c2a72126cf63d933-Abstract.html`. 15

Robinson, J. D., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. In *ICLR*, 2021. 16

Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora, S., Kakade, S., and Krishnamurthy, A. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pp. 19250–19286. PMLR, 2022. 16

Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022. 15

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 1–9. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298594. URL `https://doi.org/10.1109/CVPR.2015.7298594`. 15

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019. URL `http://proceedings.mlr.press/v97/tan19a.html`. 15

Tan, Z., Zhang, Y., Yang, J., and Yuan, Y. Contrastive learning is spectral clustering on similarity graph. *arXiv preprint arXiv:2303.15103*, 2023. 16

Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., and Dai, J. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14431–14440, 2022. 16

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022. 15

Tian, Y. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022. 16

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 16

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 16

Tian, Y., Chen, X., and Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pp. 10268–10278. PMLR, 2021. 16

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive estimation reveals topic posterior information to linear models. *arXiv:2003.02234*, 2020. 16

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR, 2021. 16

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 15

Tsai, Y.-H. H., Bai, S., Morency, L.-P., and Salakhutdinov, R. A note on connecting barlow twins with negative-sample-free contrastive learning. *arXiv preprint arXiv:2104.13712*, 2021. 16

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`. 2, 3

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020. 16

Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 548–558. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00061. URL `https://doi.org/10.1109/ICCV48922.2021.00061`. 15

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022a. 17

Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. PVT v2: Improved baselines with pyramid vision transformer. *Comput. Vis. Media*, 8(3):415–424, 2022b. doi: 10.1007/s41095-022-0274-8. URL `https://doi.org/10.1007/s41095-022-0274-8`. 15

Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022c. 16

Wen, Z. and Li, Y. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pp. 11112–11122. PMLR, 2021. 16

Wen, Z. and Li, Y. The mechanism of prediction head in non-contrastive self-supervised learning. *arXiv preprint arXiv:2205.06226*, 2022. 16

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. 16

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5987–5995. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.634. URL `https://doi.org/10.1109/CVPR.2017.634`. 15

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: a simple framework for masked image modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 9643–9653. IEEE, 2022a. doi: 10.1109/CVPR52688.2022.00943. URL `https://doi.org/10.1109/CVPR52688.2022.00943`. 16

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9653–9663, 2022b. 16

Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019. 16

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021. 16

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 1204–1213. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01179. URL `https://doi.org/10.1109/CVPR52688.2022.01179`. 15

Zhang, Q., Wang, Y., and Wang, Y. How mask matters: Towards theoretical understandings of masked autoencoders. *Advances in Neural Information Processing Systems*, 35:27127–27139, 2022a. 16

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022b. 15

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021a. 16

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A. L., and Kong, T. ibot: Image BERT pre-training with online tokenizer. *CoRR*, abs/2111.07832, 2021b. URL `https://arxiv.org/abs/2111.07832`. 15

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 17