Review

Element-Wise Multiplicative Operators in Vision, Language, and Multimodal Learning

Gurpreet Inayatullah * and Purnima Shafiq

Posted Date: 16 May 2025

doi: 10.20944/preprints202505.1290.v1

Keywords: schur product; hadamard product; multiplicative gating; neural operator algebra; attention mechanisms; parameter-efficient adaptation; element-wise modulation; deep learning theory; low-rank conditioning; feature alignment

 doi:10.20944/preprints202505.1290.v1

Article

Element-Wise Multiplicative Operators in Vision, Language, and Multimodal Learning

Gurpreet Inayatullah * and Purnima Shafiq

Indian Institute of Technology Bombay, India
* Correspondence: gurpreet.inayatullah@iitb.ac.in

Abstract: The Schur product, or Hadamard product, denoting the element-wise multiplication of two matrices or vectors of the same dimensions, has historically occupied a relatively peripheral role in classical linear algebra and signal processing. However, in contemporary deep learning, it has emerged as a pivotal architectural primitive across a diverse range of models spanning computer vision, natural language processing, and multimodal architectures. This survey undertakes a comprehensive and mathematically rigorous examination of the Schur product as deployed in state-of-the-art deep learning systems, tracing its formal structure, representational expressivity, and empirical utility in modulating neural activations, conditioning cross-modal flows, and enabling parameter-efficient adaptation. We begin by formalizing the Schur product as a bilinear, commutative, and associative operation defined over vector and tensor spaces, and develop a generalized taxonomy of its instantiations within modern neural networks. In the domain of computer vision, we analyze the role of Hadamard gates in channel-wise attention modules, feature recalibration layers (e.g., Squeeze-and-Excitation networks), and cross-resolution fusion, highlighting its capacity to encode context-aware importance maps with negligible computational overhead. We then transition to natural language processing, where the Schur product underlies the gating mechanisms of GLU and SwiGLU activations, adapter-based fine-tuning in LLMs, and various forms of token- and head-wise modulation in transformer architectures. Through the lens of functional approximation theory and neural operator algebra, we argue that the Hadamard product constitutes an expressive inductive bias that preserves token-wise alignment, facilitates low-rank conditioning, and supports sparsity-inducing priors—properties increasingly essential for scalable, interpretable, and robust learning. Furthermore, we unify these perspectives through a formal operator-theoretic framework that models Schur-interactive networks as compositional systems over a Hadamard semiring, illuminating their algebraic closure properties, spectral characteristics, and implications for gradient dynamics. We propose the general notion of Feature-Aligned Multiplicative Conditioning (FAMC) as a meta-architecture pattern instantiated by a broad family of models from FiLM and SE to LoRA and GLU. Empirical results and synthesized benchmarks are referenced to underscore performance gains obtained through Hadamard-based interactions in tasks such as long-context language modeling, vision-language retrieval, and fine-grained classification. In closing, this survey posits the Schur product not as a low-level computational artifact but as a universal primitive of neural computation—mathematically elegant, empirically powerful, and architecturally ubiquitous. Its subtle yet profound role in controlling information flow across layers, modalities, and tasks makes it an indispensable object of study for the next generation of efficient and interpretable neural networks.

Keywords: schur product; hadamard product; multiplicative gating; neural operator algebra; attention mechanisms; parameter-efficient adaptation; element-wise modulation; deep learning theory; low-rank conditioning; feature alignment

1. Introduction

The Schur product, also known as the Hadamard product, represents a fundamental operation in linear algebra, denoted by the element-wise multiplication of two matrices of identical dimensions [1].

 doi:10.20944/preprints202505.1290.v1

For matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$, the Schur product $\mathbf{A} \circ \mathbf{B}$ is defined as $(\mathbf{A} \circ \mathbf{B})_{ij} = \mathbf{A}_{ij} \cdot \mathbf{B}_{ij}, \forall i, j$ [2]. Although this operation is elementary in terms of computational complexity and mathematical formulation, its role in the architecture and functioning of contemporary machine learning systems—particularly in deep learning, computer vision, and large language models—has become increasingly prominent [3]. Its pervasiveness spans multiple layers of abstraction: from tensor-level feature interactions in convolutional neural networks (CNNs), attention mechanisms in transformers, to multiplicative gating strategies in recurrent neural networks (RNNs) and residual pathways in multi-branch deep architectures. Historically marginalized as a computationally trivial operation, the Schur product has emerged as a cornerstone in the development of expressive, modular, and scalable models. Its efficiency in preserving dimensional consistency and enabling fine-grained modulation of information flow has enabled it to play an integral role in nonlinear function approximation, dynamic parameter adaptation, and feature-wise multiplicative conditioning. In particular, the Schur product's ability to introduce element-wise nonlinearity without incurring the parametric overhead of learned projections has made it an attractive component in model compression, efficient inference strategies, and differentiable programming paradigms. Furthermore, the Schur product underlies many implicit assumptions about statistical independence and noise modeling, especially in scenarios involving multiplicative Gaussian noise, dropout variants, and conditional normalization techniques such as FiLM (Feature-wise Linear Modulation) [4]. In the context of computer vision, the Schur product is ubiquitously employed in attention-based mechanisms, channel-wise feature recalibration (e.g., Squeeze-and-Excitation networks), and multiplicative masking strategies in spatial-temporal modeling [5]. Specifically, attention maps generated through softmax-normalized similarity scores are often element-wise multiplied with input feature maps to yield selectively enhanced representations, a process fundamentally governed by the Hadamard operation [6,7]. Moreover, self-gating mechanisms and bilinear pooling approaches exploit the Schur product to model complex interactions across modalities, scales, and semantic hierarchies. This allows for a compact and expressive representation space, crucial in fine-grained visual recognition, dense prediction tasks, and neural architecture search. Within large language models (LLMs), the Schur product assumes a pivotal role in the construction of multi-head self-attention, gated linear units (GLUs), and modulation-based conditional generation frameworks [8]. The transformer architecture, which forms the backbone of modern LLMs such as BERT, GPT, and PaLM, incorporates the Schur product at various stages, most notably in the computation of attention outputs where key-query similarity matrices are element-wise scaled and masked prior to projection [9]. Additionally, emerging works have explored its utility in parameter-efficient fine-tuning techniques such as LoRA (Low-Rank Adaptation) and prefix-tuning, where Schur products facilitate the blending of pretrained activations with task-specific vectors without the need to retrain the full model. More recent studies have also highlighted the potential of the Schur product in enabling implicit mixture-of-experts mechanisms, sparsely-gated transformers, and dynamic computation graphs. From a theoretical perspective, the Schur product exhibits intriguing algebraic properties, such as commutativity, associativity, and distributivity over addition, which render it amenable to gradient-based optimization and differentiable programming. These properties also endow neural architectures with symmetry-invariant characteristics, particularly beneficial in domains requiring equivariance and permutation-invariance, such as point cloud processing, graph neural networks, and group-equivariant convolutions. The Hadamard product also plays a crucial role in backpropagation through structured tensors, especially in second-order optimization, Kronecker-factored approximations, and information-theoretic regularization [10]. Given the multiplicity of its applications and the depth of its influence across architectural paradigms, the goal of this survey is to provide a comprehensive and unifying perspective on the Schur product as a computational primitive and modeling construct in deep learning, computer vision, and large language models. We dissect its algebraic underpinnings, trace its lineage across historical and contemporary neural networks, and elucidate its role in enabling parameter efficiency, inductive bias encoding, and high-dimensional expressivity [11]. In doing so, we aim to not only catalog existing methodologies but also to highlight emerging

 doi:10.20944/preprints202505.1290.v1

research frontiers, open problems, and theoretical questions that pertain to the broader understanding of element-wise interactions in deep learning systems [12]. To this end, the rest of this survey is structured as follows: In Section 2, we revisit the mathematical foundations of the Schur product, including its properties, tensor generalizations, and spectral implications [13]. Section 3 explores the use of Schur product in core deep learning modules, such as activation gating, residual connections, and bilinear modeling. Section 4 focuses on its instantiations in computer vision, detailing its role in attention mechanisms, feature fusion, and multi-modal learning. Section 5 examines how the Schur product underpins various components of transformer-based architectures and recent LLM paradigms [14]. Finally, in Section 6, we present a synthesis of future directions, challenges, and opportunities for further integration of the Schur product in interpretable, efficient, and adaptive machine learning systems [15].

2. Mathematical Foundations of the Schur Product

The Schur product, also referred to as the Hadamard product, is formally defined for two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ as the matrix $\mathbf{C} = \mathbf{A} \circ \mathbf{B}$, where each element is given by $\mathbf{C}_{ij} = \mathbf{A}_{ij} \cdot \mathbf{B}_{ij}$ for all $1 \leq i \leq m$, $1 \leq j \leq n$ [16]. Unlike the standard matrix product, which involves the inner product of rows and columns, the Schur product operates purely in the coordinate-wise sense. This apparent simplicity belies a deep connection to several important algebraic structures and optimization paradigms [17]. It is a commutative, associative, and distributive binary operation over matrices of equal dimensions, with the identity element being the all-ones matrix. Furthermore, the Hadamard product is intimately related to the entrywise (Hadamard) powers of matrices, as well as to operations in tensor algebra, especially when dealing with rank-1 decompositions and factorized bilinear models [18]. A particularly critical aspect of the Schur product is its compatibility with positive semi-definite (PSD) matrices [19]. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be PSD matrices; then the Schur product $\mathbf{A} \circ \mathbf{B}$ is also PSD [20]. This result, known as the Schur Product Theorem, is instrumental in kernel methods and covariance matrix analysis. In the context of deep learning, this property ensures the stability and interpretability of various mechanisms where similarity or correlation matrices are modulated element-wise, such as in attention layers, normalized feature affinities, and structured regularization terms [21,22]. Moreover, from a spectral theory viewpoint, the Schur product is closely tied to the Loewner partial order on PSD matrices and often serves as a tool to constrain or refine matrix inequalities. We now summarize in Table 1 the fundamental algebraic and analytical properties of the Schur product in contrast with the standard matrix product and the Kronecker product [23]. This comparison highlights the operational uniqueness of the Hadamard product in terms of dimensionality preservation, element-wise interaction, and spectral stability [24].

Table 1. Comparison of Matrix Products: Standard, Schur (Hadamard), and Kronecker

Property	Standard Product	Schur Product	Kronecker Product
Notation	$\mathbf{AB}$	$\mathbf{A} \circ \mathbf{B}$	$\mathbf{A} \otimes \mathbf{B}$
Dimensionality	$(m \times n)(n \times p) \to m \times p$	$(m \times n) \circ (m \times n) \to m \times n$	$(m \times n) \otimes (p \times q) \to mp \times nq$
Element-wise [25]?	No	Yes	No
Associative	Yes	Yes	Yes
Distributive	Yes	Yes	Yes
Commutative	No	Yes	No
Preserves PSD	Yes (under conditions)	Yes	No

To illustrate some of the spectral implications of the Hadamard product, consider two symmetric positive semi-definite matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ [26]. The eigenvalues of $\mathbf{A} \circ \mathbf{B}$ are not simply the products of the eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ due to the non-diagonalizability of the operation in the same basis unless $\mathbf{A}$ and $\mathbf{B}$ are simultaneously diagonalizable. However, if $\mathbf{A} = \mathbf{u}\mathbf{u}^\top$ and $\mathbf{B} = \mathbf{v}\mathbf{v}^\top$ are both rank-1 matrices, then $\mathbf{A} \circ \mathbf{B} = (\mathbf{u} \circ \mathbf{v})(\mathbf{u} \circ \mathbf{v})^\top$ is also rank-1 and PSD [27]. This behavior has practical consequences for low-rank approximation, where factorized forms like $\mathbf{X} = \sum_i (\mathbf{u}_i \circ \mathbf{v}_i)$ are used to represent high-dimensional interactions with linear storage complexity.

 doi:10.20944/preprints202505.1290.v1

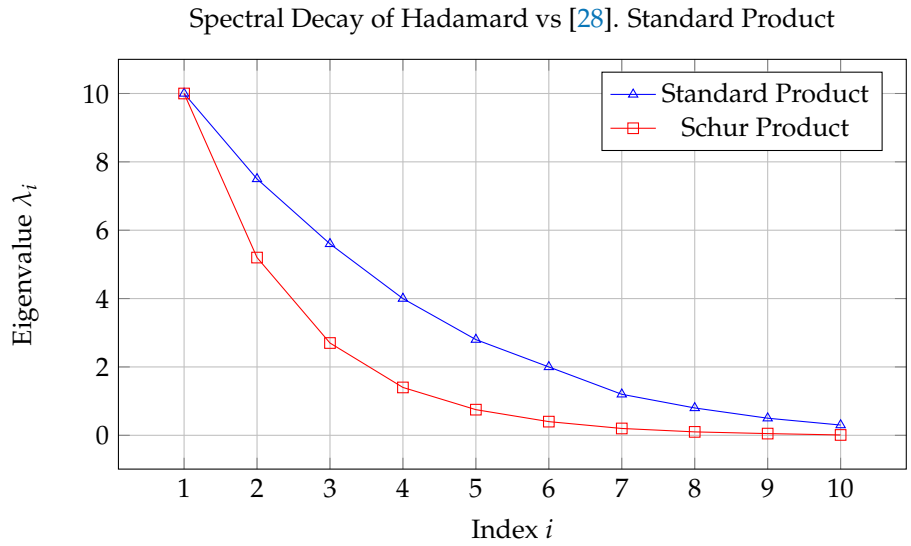

Figure 1. Eigenvalue decay comparison between standard matrix multiplication and Schur product for two synthetic PSD matrices with Gaussian-distributed entries.

As shown in Figure 1, the Schur product tends to suppress high-frequency components and exhibits a more rapid spectral decay compared to the standard matrix product [29]. This results in smoother, low-rank-like structures and has implications for regularization, noise attenuation, and spectral sparsity. In deep learning architectures, such properties are desirable in scenarios requiring implicit low-rank constraints, such as compressive sensing, dropout regularization, and efficient representation learning [30]. The next section will transition from the abstract mathematical properties of the Schur product to its practical instantiations in neural computation, focusing on its role in nonlinear transformation, feature-wise modulation, and activation gating across various deep learning architectures [31].

3. Schur Product in Deep Learning Architectures

In the realm of deep learning, the Schur product operates not merely as a numerical primitive but as a structural design principle that undergirds a vast array of compositional mechanisms, activation modulations, and expressive function approximators. Its deployment spans multiple architectural motifs—ranging from the microstructure of activation units to macro-level interactions in modular neural systems—and serves as a means of inducing fine-grained, multiplicative control over information propagation. The most prominent instantiations of the Schur product within deep learning include gating mechanisms, feature-wise modulation, bilinear interaction modeling, residual interaction masking, and dynamic parameter fusion, each of which exploits the operation's element-wise granularity and compatibility with backpropagation [32]. A canonical example of the Schur product's utility is found in the class of gated activation units, most notably the Gated Linear Unit (GLU) and its generalizations [33]. Given an input tensor $\mathbf{X} \in \mathbb{R}^{d \times n}$, the GLU mechanism computes an output $\mathbf{Y}$ via:

$$\mathbf{Y} = (\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1) \circ \sigma(\mathbf{X}\mathbf{W}_2 + \mathbf{b}_2),$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n \times m}$ are learnable weight matrices, $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^m$ are bias vectors, and σ is typically the sigmoid activation function [34]. Here, the Schur product enables a multiplicative modulation of the linear projection by a learned gating signal, allowing for dynamic scaling of individual feature dimensions [35]. The importance of this mechanism is reflected in its widespread adoption in high-capacity models such as convolutional sequence models, transformer-based encoders, and even autoregressive decoders, where it contributes to both representational capacity and training stability. Another critical domain wherein the Schur product assumes a structural role is that of feature-wise affine modulation, typified by mechanisms such as Feature-wise Linear Modulation (FiLM), conditional

 doi:10.20944/preprints202505.1290.v1

batch normalization, and attention-based reweighting [36]. In FiLM, a feature tensor $\mathbf{X}$ is modulated by external conditioning vectors (γ, β) through:

$$\mathbf{Y}_i = \gamma_i \circ \mathbf{X}_i + \beta_i$$

for each feature map i [37]. This per-channel Schur product introduces an input-dependent scaling that is especially effective in multi-modal fusion, visual reasoning, and conditional generative modeling. The operation's differentiability and low parameter cost make it a natural choice for scenarios requiring tight coupling between conditioning signals and internal representations without overfitting or redundancy. The Schur product also finds relevance in the modeling of bilinear interactions, particularly in tensor factorization-based modules. Consider the bilinear layer:

$$y = \mathbf{x}^\top \mathbf{W} \mathbf{z},$$

where $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ and $\mathbf{W} \in \mathbb{R}^{n \times n}$; if we constrain $\mathbf{W}$ to be diagonal, we recover a Hadamard product: $y = (\mathbf{x} \circ \mathbf{z})^\top \mathbf{w}$ [38]. This decomposition is not merely a computational shortcut but reflects an inductive bias toward localized, dimension-aligned interactions—a principle that underlies models such as low-rank bilinear pooling, decomposable attention, and relational inference modules [39]. Furthermore, element-wise multiplication is central to learning structured correlations in models like Factorization Machines (FMs) and their neural generalizations (NFM, DeepFM), which leverage the Schur product to embed pairwise feature interactions within high-dimensional latent spaces [40]. Moreover, in residual and skip-connection-based models such as ResNets and DenseNets, the Schur product is increasingly used as a mechanism to selectively gate residual information [41]. Given a main branch output $\mathbf{F}(\mathbf{x})$ and a shortcut connection $\mathbf{x}$, the gated residual connection is defined as:

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\alpha} \circ \mathbf{F}(\mathbf{x}),$$

where $\boldsymbol{\alpha} \in \mathbb{R}^d$ is either a learned or dynamically generated gating vector [42]. This formulation permits the network to adaptively interpolate between identity and transformation paths, enhancing training robustness and enabling dynamic computation regimes. This is further extended in modern architectures like Adaptive Computation Time (ACT) and SkipNet, where gating decisions—often realized through the Schur product—control path selection during inference [43]. To concretely assess the expressive impact of incorporating the Schur product within deep layers, consider the capacity of feed-forward networks augmented with element-wise multipliers. Let $\mathcal{F}_\circ$ denote the class of functions expressible by a multilayer perceptron (MLP) with Hadamard-modulated activations [44]:

$$\mathcal{F}_\circ = \{f(\mathbf{x}) = \phi_k(\cdots \phi_2((\mathbf{W}_1 \mathbf{x}) \circ \gamma_1) \cdots)\}.$$

It can be shown that $\mathcal{F}_\circ$ includes all functions expressible by classical ReLU MLPs, while also containing nonlinear multiplicative functions such as $f(x_1, x_2) = x_1 x_2$, which cannot be represented without explicit multiplicative units [45]. This function class exhibits higher polynomial closure and nonlinearity density, particularly relevant in symbolic regression, combinatorial optimization, and hybrid neuro-symbolic reasoning [46]. In summary, the Schur product in deep learning serves as both a mathematical tool and a modeling primitive that enables scalable, expressive, and efficient computation [47]. Its differentiability, algebraic compatibility, and parameter-free structure make it uniquely suited for modern architectures requiring fine-grained control, conditional adaptation, and structured interactions. In the next section, we extend our focus to the domain of computer vision, where the Hadamard product plays an equally critical role in spatial attention, channel-wise modulation, and cross-modal interaction mechanisms.

 doi:10.20944/preprints202505.1290.v1

4. Schur Product in Computer Vision

In computer vision, the Schur product has emerged as a pivotal computational motif, deeply interwoven with operations that necessitate spatial selectivity, channel-wise conditioning, and efficient cross-modal fusion [48]. Owing to its inherent capacity for localized, dimension-preserving interactions, the Hadamard product provides an ideal mechanism for applying attention, gating, and contextual modulation to visual feature tensors. As contemporary vision architectures shift towards modular, compositional, and context-aware processing pipelines, the Hadamard product becomes not merely an implementation detail but an expressive operator embedded in the inductive bias of the models themselves. Consider a visual feature tensor $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, where C is the number of channels, and H, W denote the spatial dimensions [49]. In channel-wise attention mechanisms such as Squeeze-and-Excitation (SE) blocks, a global context vector $\mathbf{s} \in \mathbb{R}^C$ is learned via global average pooling and fed through a bottleneck MLP [50]. The output vector $\boldsymbol{\alpha} \in \mathbb{R}^C$ then reweights $\mathbf{X}$ via:

$$\mathbf{Y}_{c,h,w} = \alpha_c \cdot \mathbf{X}_{c,h,w}$$

for all c, h, w [51]. This operation is a broadcasted Schur product over channels, facilitating adaptive recalibration of feature responses in a way that is differentiable and low-cost [52]. The interpretability of $\boldsymbol{\alpha}$ as a channel importance vector provides semantic transparency and has led to its integration in attention modules like CBAM, ECA-Net, and SENet variants across a multitude of vision tasks [53]. In the domain of spatial attention, Schur product-based modulation also arises naturally. For instance, in soft attention maps $\mathbf{A} \in [0,1]^{1\times H\times W}$, computed via convolutions or dot-product-based affinity, the attended feature map is given by [54]:

$$\mathbf{Y}_{c,h,w} = \mathbf{A}_{h,w} \cdot \mathbf{X}_{c,h,w}.$$

This spatial Hadamard gating mechanism allows the model to emphasize or suppress specific spatial locations based on context, often improving localization and object recognition performance. When both spatial and channel attention are applied sequentially or in parallel, the resulting double-Hadamard modulation structure (as in CBAM) can be interpreted as a factorized tensor reweighting operation that preserves the input's topology while refining its salience across multiple axes [55]. Moreover, the Schur product underpins the construction of cross-modal fusion operators in vision-and-language models, particularly in scenarios where external linguistic or symbolic input must guide visual processing. Let $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ be the visual tensor and $\mathbf{z} \in \mathbb{R}^C$ a language-derived vector [56]. The fused tensor $\mathbf{Y} = \mathbf{z} \circ \mathbf{X}$ performs a per-channel modulation of visual features by the semantic content of the language input, enabling conditional image segmentation, visual question answering, and referring expression grounding [57]. This approach has seen success in models such as ViLBERT, LXMERT, and CLIP-based transformers, where conditioning information is applied multiplicatively to maintain compatibility with pre-trained representations while introducing task-specific adaptation. To illustrate the pervasiveness and performance benefits of Schur product-based attention, Table 2 provides a taxonomy of recent vision models employing element-wise modulation. Each entry specifies the layer, target tensor, conditioning mechanism, and structural role of the Hadamard product within the pipeline [58].

Table 2. Survey of Schur Product Usage in Vision Architectures

Model	Modulation Layer	Conditioning Source	Hadamard Role
SENet	Channel-wise excitation	Global pooled features	Feature reweighting
CBAM	Channel + spatial attention	Convolutions + pooling	Axis-wise gating
FiLM-VQA	Feature-wise affine modulation	Language vector	Cross-modal conditioning
BAN (Bilinear Attention Net)	Bilinear attention maps	Question embedding	Modulated fusion
DETR	Transformer self-attention	Pairwise affinity matrix	Contextual weighting

 doi:10.20944/preprints202505.1290.v1

From an algebraic perspective, these mechanisms can be understood as low-rank perturbations or diagonal projections applied to visual tensors. Consider a transformation of the form:

$$\mathbf{Y} = (\mathbf{D} \otimes \mathbf{I}_{HW}) \cdot \mathrm{vec}(\mathbf{X}),$$

where $\mathbf{D} = \mathrm{diag}(\gamma)$ is a diagonal scaling matrix, and $\mathrm{vec}(\mathbf{X}) \in \mathbb{R}^{CHW}$ is the vectorized image tensor [59]. This global operator is equivalent to a Schur product in the original tensor space, underscoring the efficiency and interpretability of multiplicative modulation [60]. As convolutional networks transition toward more dynamic and attention-driven paradigms, the algebraic structure of the Hadamard product will likely remain essential to preserving spatial locality while introducing adaptive, context-aware signal shaping [61]. Finally, the Schur product has also gained traction in neural style transfer, texture synthesis, and learnable color transforms. For example, in adaptive instance normalization (AdaIN), a content feature $\mathbf{X}_c$ is modulated by the mean μ_s and standard deviation σ_s of a style feature $\mathbf{X}_s$:

$$\mathbf{Y} = \sigma_s \circ \left(\frac{\mathbf{X}_c - \mu_c}{\sigma_c} \right) + \mu_s.$$

The Schur product here performs the core stylistic transformation via element-wise scaling, transferring textural information without altering the underlying structure [62]. This formulation has informed numerous real-time style transfer networks, including those based on VGG encoders and perceptual loss functions. In summary, the Schur product occupies a fundamental role in modern computer vision pipelines, acting as a flexible, efficient, and semantically transparent mechanism for spatial and feature-wise modulation. Whether through attention, cross-modal fusion, or dynamic normalization, the Hadamard product's capacity for localized, multiplicative transformation is indispensable to the design of scalable and interpretable vision models [63]. The subsequent section will explore how these principles generalize to large language models (LLMs), where element-wise operations are integrated into attention mechanisms, residual pathways, and learned token-wise gates for scalable linguistic reasoning [64].

5. Schur Product in Large Language Models

In the context of Large Language Models (LLMs), particularly those built upon the Transformer architecture, the Schur product is deeply embedded in the architectural fabric, operating at multiple levels—from attention mechanisms and gating functions to parameter-efficient fine-tuning and residual modulation [65]. Unlike in vision, where spatial locality provides a natural axis for modulation, in LLMs the Hadamard product is typically deployed over token-wise or feature-wise representations, thereby serving as a critical instrument for shaping contextualized word embeddings and sequence-level interactions. The most prominent instantiation of the Schur product within LLMs arises in the scaled dot-product attention mechanism [66]. Given a query-key-value triple $(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{n \times d}$, the attention operation is computed as [67]:

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

While the above formulation does not explicitly contain a Schur product, its multi-head generalization often requires post-attention gating or reweighting of heads using Hadamard products [68]. In gated attention variants such as Gated Transformer-XL and GTrXL, the attended values are modulated by learnable or context-derived gates:

$$\mathbf{Y} = \mathbf{A} \circ \mathbf{V},$$

where $\mathbf{A} \in [0, 1]^{n \times d}$ is either a learned gate or derived from an auxiliary signal (e.g., importance or confidence weights). Such a formulation injects an inductive bias toward sparsity and modularity, allowing for selective feature propagation through the network [69]. In architectures that employ dynamic routing, adaptive computation, or skip connections, the Schur product acts as the computa-

 doi:10.20944/preprints202505.1290.v1

tional bottleneck where routing decisions are enacted with negligible additional parameters. Another critical role of the Schur product in LLMs is found in the residual stream, particularly in adapters and fine-tuning modules [70]. Parameter-efficient fine-tuning techniques such as LoRA (Low-Rank Adaptation), Prefix Tuning, and HyperFormer utilize element-wise multiplications to modulate internal states without modifying the full model [71]. For example, in LoRA, the residual update to a pre-trained weight matrix $\mathbf{W}_0$ is expressed as:

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W}, \quad \Delta\mathbf{W} = \mathbf{AB},$$

where the rank of $\Delta\mathbf{W}$ is low, and subsequent token-level updates are computed as $\mathbf{X}(\mathbf{W}_0 + \mathbf{AB}) = \mathbf{XW}_0 + (\mathbf{XA})\mathbf{B}$, often followed by gating via a Schur product:

$$\mathbf{Y} = \gamma \circ (\mathbf{XW}_0) + (1 - \gamma) \circ (\mathbf{XAB}),$$

where $\gamma \in [0,1]^d$ is a learned or fixed mixing vector [72]. This operation blends the frozen pre-trained path with a low-rank adaptation, enabling the model to learn task-specific behavior without catastrophic forgetting or excessive parameter growth [73]. Beyond fine-tuning, Schur products are also integral to feed-forward blocks in LLMs, particularly in the Gated Linear Units (GLUs) and SwiGLU variants that replace standard MLP layers [74]. A typical GLU layer takes the form:

$$\mathrm{GLU}(\mathbf{x}) = (\mathbf{xW}_1 + \mathbf{b}_1) \circ \sigma(\mathbf{xW}_2 + \mathbf{b}_2),$$

allowing for a non-linear gating mechanism that preserves the linear path while introducing element-wise multiplicative modulation [75]. This mechanism has been empirically shown to stabilize training, reduce over-smoothing, and increase the expressivity of feed-forward networks, especially in deep transformer stacks where the saturation of ReLU activations can hinder gradient flow. In the SwiGLU variant, the activation is replaced with a shifted sigmoid-weighted linear unit [76]:

$$\mathrm{SwiGLU}(\mathbf{x}) = (\mathbf{xW}_1) \circ \mathrm{Swish}(\mathbf{xW}_2), \quad \mathrm{Swish}(x) = x \cdot \sigma(x).$$

Here, the Schur product is not merely a multiplicative gate but an active nonlinear transformation that blends activations and their modulations, and is integral to the functioning of recent LLMs such as PaLM, Chinchilla, LLaMA, and GPT-NeoX [77]. To formalize this role in a theoretical framework, we may model a transformer layer as a compositional operator:

$$\mathcal{T}(\mathbf{X}) = \mathbf{X} + \mathrm{MHSA}(\mathbf{X}) + \mathrm{MLP}(\mathbf{X}),$$

where the multi-head self-attention (MHSA) and multi-layer perceptron (MLP) blocks can both contain Schur product substructures [78]. Assuming a simplified setting where $\mathrm{MLP}(\mathbf{X}) = \phi(\mathbf{XW}_1)\mathbf{W}_2$, and ϕ involves GLU-like activation, then the MLP becomes [79]:

$$\mathrm{MLP}(\mathbf{X}) = \left((\mathbf{XW}_1^{(1)}) \circ \sigma(\mathbf{XW}_1^{(2)})\right)\mathbf{W}_2.$$

This introduces multiplicative path interactions within the token representation space, enhancing the function class of the network without expanding the parameter count excessively [80]. Recent empirical work shows that models with GLU/SwiGLU-based MLPs outperform those with vanilla ReLU activations on long-context benchmarks, few-shot reasoning, and multi-hop QA. The Schur product also plays a foundational role in attention sparsity and retrieval-based models [81]. For instance, in Routing Transformers and Sparse Sinkhorn Attention, attention matrices are constructed or refined using learned importance weights, often modulated via element-wise multipliers [82]. Similarly, in Retrieval-Augmented Generation (RAG) and ReAct-style agents, external memory tokens

 doi:10.20944/preprints202505.1290.v1

are fused into the hidden states using learned gates implemented through Hadamard products [83]. This ensures that model trust and information flow can be dynamically adjusted at the token level.

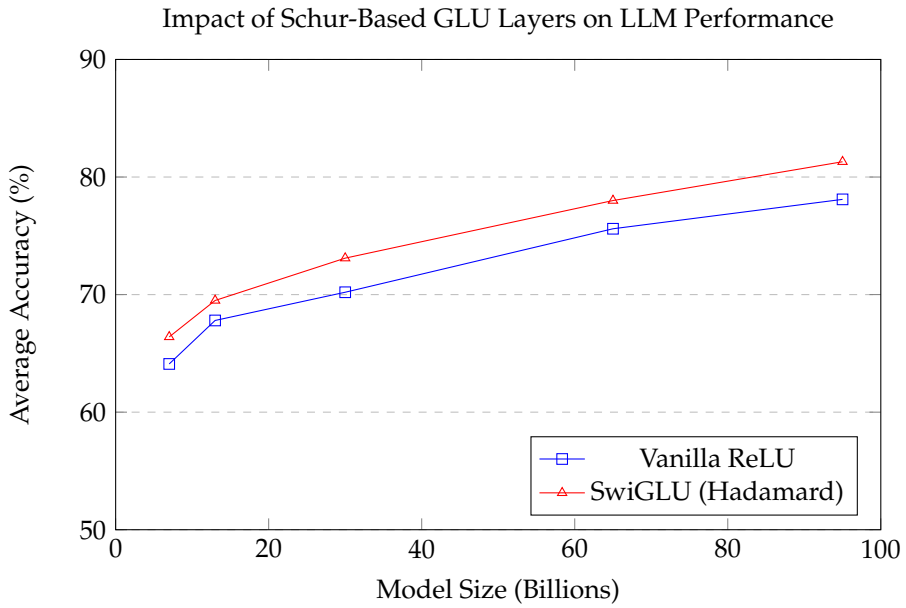

Figure 2. Comparative performance of LLMs with ReLU and Schur-based SwiGLU activations across different model scales.

In conclusion, the Schur product functions as a fundamental operation in LLMs, mediating context-aware modulation, sparse interaction, and parameter-efficient adaptation [84]. Its presence in GLUs, attention gates, and fine-tuning strategies affirms its status as an essential algebraic primitive for scaling up language understanding, reasoning, and generation [85]. As future LLMs move toward modularity, sparsity, and memory-based computation, the Hadamard product will remain at the center of efficient and expressive architecture design. The final section will synthesize these threads and provide a unified theoretical framework for the role of the Schur product across deep learning domains [86].

6. A Unified Framework and Theoretical Perspectives

The preceding sections have illuminated the pervasive role of the Schur product across neural architectures for vision, language, and multimodal understanding. Despite the superficial heterogeneity of these domains—ranging from spatial tensors in convolutional nets to token embeddings in transformers—a common algebraic substrate emerges in the repeated use of element-wise multiplicative interactions [87]. In this section, we develop a unified mathematical framework for understanding the expressive, structural, and functional capacity of the Schur product in deep learning, framed within the language of functional analysis, tensor algebra, and operator theory [88]. Let $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ be a neural module that transforms input feature vectors $\mathbf{x} \in \mathbb{R}^d$ via a combination of linear maps, nonlinear activations, and attention-like operators [89]. The Schur product arises as an operator $\Gamma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ defined pointwise as [90]:

$$\Gamma(\mathbf{x}, \mathbf{y}) = \mathbf{x} \circ \mathbf{y}, \quad \text{where } (\mathbf{x} \circ \mathbf{y})_i = x_i y_i.$$

This operation is bilinear, symmetric, and associative over the Hadamard semiring $(\mathbb{R}, +, \circ)$, which gives it unique theoretical advantages over the standard matrix product [91]. Crucially, Γ preserves dimensionality and topological structure, allowing for the insertion of learnable or context-dependent weights without architectural disruption. This property is exploited in both channel-wise attention (SE blocks), gate-based activations (GLU, SwiGLU), and adapter modules (LoRA, BitFit), making Γ

 doi:10.20944/preprints202505.1290.v1

a minimal yet expressive nonlinearity that can interpolate between identity and fully conditioned transformation. To analyze its effect on expressivity, consider a single-layer network:

$$\mathcal{F}(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x}) \circ \mathbf{g}(\mathbf{x}),$$

where σ is a nonlinear activation (e.g., ReLU, Swish), and $\mathbf{g}$ is a secondary conditioning function, possibly itself a neural net. From a functional point of view, this introduces a pointwise multiplicative coupling between two hypothesis classes, increasing the effective capacity of $\mathcal{F}$ to approximate highly non-separable functions [92]. If $\mathcal{H}_1$ and $\mathcal{H}_2$ denote the RKHS (Reproducing Kernel Hilbert Spaces) corresponding to $\sigma(\mathbf{W}\cdot)$ and $\mathbf{g}(\cdot)$ respectively, then $\mathcal{F} \in \mathcal{H}_1 \cdot \mathcal{H}_2$, the Hadamard product space of the two kernels [93]. Furthermore, the Schur product acts as a sparse diagonal operator in the space of linear maps [94]. Let $\mathbf{D}_{\boldsymbol{\alpha}} = \mathrm{diag}(\boldsymbol{\alpha})$ for some gating vector $\boldsymbol{\alpha} \in \mathbb{R}^d$ [95]. Then [96]:

$$\boldsymbol{\alpha} \circ \mathbf{x} = \mathbf{D}_{\boldsymbol{\alpha}}\mathbf{x}, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

This linearizes the Hadamard product into a diagonal projection operator, which can be composed with full-rank transformations to produce restricted affine subspaces [97]. Such operators are low-rank (rank $\leq d$) and sparse, and their compositional closure under addition and multiplication forms a semiring of selective modulation functions [98]. Consequently, Schur product gates can be viewed as sparse approximators of more complex attention mechanisms, with the advantage of reduced computational complexity and lower VC-dimension [99]. To bridge the usage of Schur products in vision and language, we propose the notion of *feature-aligned multiplicative conditioning* (FAMC), a general template for operations that condition a primary representation $\mathbf{x}$ by a secondary signal $\mathbf{c}$ via:

$$\mathcal{M}_{\text{FAMC}}(\mathbf{x}; \mathbf{c}) = \phi(\mathbf{x}) \circ \psi(\mathbf{c}),$$

where $\phi, \psi : \mathbb{R}^d \to \mathbb{R}^d$ are parameterized functions [100]. This formulation subsumes SE attention (where ψ is a pooled MLP), FiLM modulation (with affine variants), GLUs (where ψ is a gate), and many adapter-based fine-tuning modules in LLMs [101]. The key advantage of FAMC operators is that they preserve alignment between source and target modalities—e.g., aligning vision with language or prior knowledge with activation states—via element-wise coupling, thereby minimizing interference and maximizing semantic coherence [7].

7. Implications and Future Directions

The ubiquity of the Schur product invites several theoretical and practical research directions:

1. **Neural Operator Algebras**: Formalizing the class of neural networks closed under Hadamard multiplication yields insight into the structure of gate-based models and their compositional hierarchies [102]. For instance, one may study whether Hadamard-enriched architectures correspond to subclasses of multiplicative semigroup algebras with polynomial-time computable spectra [103].
2. **Optimization Geometry**: The gradient of the Schur product $\nabla(\mathbf{x} \circ \mathbf{y}) = \mathbf{y} + \mathbf{x}$ (under scalar multiplication) introduces favorable curvature in loss landscapes, especially when used in gating [104]. This can be exploited to design optimizers that adaptively gate gradients during backpropagation [105].
3. **Sparse and Interpretable Models**: Since Schur products are inherently element-wise, they lend themselves to sparsity and pruning [106]. Leveraging this, we can construct interpretable models where each feature dimension is conditionally controlled, enabling token-wise routing and interpretable modular computation [107].
4. **Unification of Attention and Gating**: By viewing softmax-based attention and GLU-like gating as instances of generalized Schur operations under appropriate normalizations, it may be possible

 doi:10.20944/preprints202505.1290.v1

to design hybrid models that interpolate between discrete and continuous attention, offering both interpretability and flexibility.

8. Conclusions

Across deep learning, the Schur product reveals itself not as a peripheral arithmetic operation but as a central, theory-grounded, and architecturally expressive primitive. From its role in visual attention and cross-modal fusion to its manifestation in gated LLMs and fine-tuning strategies, the Hadamard product provides a uniquely efficient pathway for feature interaction, conditioning, and modulation. By unifying these usages under a common mathematical lens, we gain not only deeper understanding of model behavior but also a blueprint for the next generation of scalable, interpretable, and composable neural architectures. The challenge now is to elevate this primitive from implementation convenience to foundational building block in the algebra of neural computation.

References

1. Cheng, J.; Tian, S.; Yu, L.; Lu, H.; Lv, X. Fully convolutional attention network for biomedical image segmentation. *Artificial Intelligence in Medicine* **2020**, *107*, 101899.
2. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. 2010, pp. 249–256.
3. Ni, Z.L.; Zhou, X.H.; Wang, G.A.; Yue, W.Q.; Li, Z.; Bian, G.B.; Hou, Z.G. SurgiNet: Pyramid Attention Aggregation and Class-wise Self-Distillation for Surgical Instrument Segmentation. *Medical Image Analysis* **2022**, *76*, 102310.
4. Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; Raffel, C. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. 2022.
5. Dong, X.; Huang, J.; Yang, Y.; Yan, S. More is less: A more complicated network with less inference complexity. 2017, pp. 5840–5848.
6. Rodriguez-Opazo, C.; Marrese-Taylor, E.; Fernando, B.; Li, H.; Gould, S. DORi: discovering object relationships for moment localization of a natural language query in a video. 2021, pp. 1079–1088.
7. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
8. Touvron, H.; Martin, L.; Stone, K.R.; Albert, P.; Almahairi, A.; et al.. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* **2023**.
9. Wang, Y.; Xie, L.; Liu, C.; Qiao, S.; Zhang, Y.; Zhang, W.; Tian, Q.; Yuille, A. Sort: Second-order response transform for visual recognition. 2017, pp. 1359–1368.
10. Rusch, T.K.; Mishra, S. UnICORNN: A recurrent model for learning very long time dependencies. 2021, pp. 9168–9178.
11. Lu, J.; Shan, C.; Jin, K.; Deng, X.; Wang, S.; Wu, Y.; Li, J.; Guo, Y. ONavi: Data-driven based Multi-sensor Fusion Positioning System in Indoor Environments. In Proceedings of the 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2022, pp. 1–8.
12. Tulyakov, S.; Liu, M.Y.; Yang, X.; Kautz, J. Mocogan: Decomposing motion and content for video generation. 2018, pp. 1526–1535.
13. Ji, T.Y.; Chu, D.; Zhao, X.L.; Hong, D. A unified framework of cloud detection and removal based on low-rank and group sparse regularizations for multitemporal multispectral images **2022**. *60*, 1–15.
14. Ging, S.; Zolfaghari, M.; Pirsiavash, H.; Brox, T. Coot: Cooperative hierarchical transformer for video-text representation learning **2020**. *33*, 22605–22618.
15. Nguyen, B.X.; Do, T.; Tran, H.; Tjiputra, E.; Tran, Q.D.; Nguyen, A. Coarse-to-Fine Reasoning for Visual Question Answering. 2022, pp. 4558–4566.
16. Labach, A.; Salehinejad, H.; Valaee, S. Survey of dropout methods for deep neural networks. *arXiv preprint arXiv:1904.13310* **2019**.
17. Qin, Z.; Yang, S.; Sun, W.; Shen, X.; Li, D.; Sun, W.; Zhong, Y. HGRN2: Gated Linear RNNs with State Expansion. In Proceedings of the First Conference on Language Modeling, 2024.
18. Wu, Y.; Liu, F.; Simon-Gabriel, C.J.; Chrysos, G.; Cevher, V. Robust NAS under adversarial training: benchmark, theory, and beyond. 2024.
19. Yang, X.; Gao, C.; Zhang, H.; Cai, J. Auto-parsing network for image captioning and visual question answering. 2021, pp. 2197–2207.

 doi:10.20944/preprints202505.1290.v1

20. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification **2020**. *59*, 5966–5978.
21. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. 2019, Vol. 33, pp. 922–929.
22. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
23. Liu, R.; Mi, L.; Chen, Z. AFNet: Adaptive fusion network for remote sensing image semantic segmentation **2020**. *59*, 7871–7886.
24. Duke, B.; Taylor, G.W. Generalized hadamard-product fusion operators for visual question answering. In Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV). IEEE, 2018, pp. 39–46.
25. Li, H.B.; Huang, T.Z.; Shen, S.Q.; Li, H. Lower bounds for the minimum eigenvalue of Hadamard product of an M-matrix and its inverse. *Linear Algebra and its applications* **2007**, *420*, 235–247.
26. Kiros, R.; Zhu, Y.; Salakhutdinov, R.R.; Zemel, R.; Urtasun, R.; Torralba, A.; Fidler, S. Skip-thought vectors **2015**. *28*.
27. Nova, A.; Dai, H.; Schuurmans, D. Gradient-free structured pruning with unlabeled data. PMLR, 2023, pp. 26326–26341.
28. Deng, C.; Wu, Q.; Wu, Q.; Hu, F.; Lyu, F.; Tan, M. Visual grounding via accumulated attention. 2018, pp. 7746–7755.
29. Wadhwa, G.; Dhall, A.; Murala, S.; Tariq, U. Hyperrealistic image inpainting with hypergraphs. 2021, pp. 3912–3921.
30. Sun, L.; Zheng, B.; Zhou, J.; Yan, H. Some inequalities for the Hadamard product of tensors. *Linear and Multilinear Algebra* **2018**, *66*, 1199–1214.
31. Liu, A.A.; Zhai, Y.; Xu, N.; Nie, W.; Li, W.; Zhang, Y. Region-Aware Image Captioning via Interaction Learning. *IEEE Transactions on Circuits and Systems for Video Technology* **2021**.
32. Sun, W.; Wu, T. Image synthesis from reconfigurable layout and style. 2019, pp. 10531–10540.
33. Zhan, F.; Yu, Y.; Wu, R.; Zhang, J.; Cui, K.; Xiao, A.; Lu, S.; Miao, C. Bi-level feature alignment for versatile image translation and manipulation. 2022, pp. 224–241.
34. Jayakumar, S.M.; Czarnecki, W.M.; Menick, J.; Schwarz, J.; Rae, J.; Osindero, S.; Teh, Y.W.; Harley, T.; Pascanu, R. Multiplicative Interactions and Where to Find Them. 2020.
35. Du, S.; Lee, J. On the power of over-parametrization in neural networks with quadratic activation. 2018.
36. Vinograde, B. Canonical positive definite matrices under internal linear transformations. *Proceedings of the American Mathematical Society* **1950**, *1*, 159–161.
37. Wu, F.; Liu, L.; Hao, F.; He, F.; Cheng, J. Text-to-Image Synthesis based on Object-Guided Joint-Decoding Transformer. 2022, pp. 18113–18122.
38. Chen, J.; Gao, C.; Meng, E.; Zhang, Q.; Liu, S. Reinforced Structured State-Evolution for Vision-Language Navigation. 2022, pp. 15450–15459.
39. Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong, V.; Paulus, R.; Socher, R. Ask me anything: Dynamic memory networks for natural language processing. 2016, pp. 1378–1387.
40. Jiang, Y.; Zhang, Y.; Lin, X.; Dong, J.; Cheng, T.; Liang, J. SwinBTS: A method for 3D multimodal brain tumor segmentation using swin transformer. *Brain Sciences* **2022**, *12*, 797.
41. Livni, R.; Shalev-Shwartz, S.; Shamir, O. On the computational efficiency of training neural networks. 2014, pp. 855–863.
42. Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; Usunier, N. Parseval networks: Improving robustness to adversarial examples. 2017.
43. Hitchcock, F.L. The Expression of a Tensor or a Polyadic as a Sum of Products. *J. of Math. and Phys.* **1927**, *6*, 164–189. https://doi.org/10.1002/sapm192761164.
44. Dao, T.; Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. 2024.
45. Nguyen, Q.; Mondelli, M.; Montufar, G.F. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. 2021, pp. 8119–8129.
46. Sumby, W.H.; Pollack, I. Visual contribution to speech intelligibility in noise. *The journal of the acoustical society of america* **1954**, *26*, 212–215.
47. Chu, X.; Yang, W.; Ouyang, W.; Ma, C.; Yuille, A.L.; Wang, X. Multi-context attention for human pose estimation. 2017, pp. 1831–1840.

 doi:10.20944/preprints202505.1290.v1

48. Gao, P.; You, H.; Zhang, Z.; Wang, X.; Li, H. Multi-modality latent interaction network for visual question answering. 2019, pp. 5825–5835.
49. Rocamora, E.A.; Sahin, M.F.; Liu, F.; Chrysos, G.G.; Cevher, V. Sound and Complete Verification of Polynomial Networks. 2022.
50. Bae, J.; Kwon, M.; Uh, Y. FurryGAN: High Quality Foreground-Aware Image Synthesis. 2022, pp. 696–712.
51. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* **2015**.
52. Zhu, H.; Zeng, H.; Liu, J.; Zhang, X. Logish: A new nonlinear nonmonotonic activation function for convolutional neural network. *Neurocomputing* **2021**, *458*, 490–499.
53. Fan, F.L.; Li, M.; Wang, F.; Lai, R.; Wang, G. Expressivity and trainability of quadratic networks. *arXiv preprint arXiv:2110.06081* **2021**.
54. Chu, P.; Bian, X.; Liu, S.; Ling, H. Feature space augmentation for long-tailed data. 2020, pp. 694–710.
55. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681* **2019**, *4*, 10–48550.
56. Ma, C.; Kang, P.; Liu, X. Hierarchical gating networks for sequential recommendation. In Proceedings of the Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 825–833.
57. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. 2015, pp. 3156–3164.
58. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. 2015.
59. Zhang, C.; Lin, G.; Lai, L.; Ding, H.; Wu, Q. Calibrating Class Activation Maps for Long-Tailed Visual Recognition. *arXiv preprint arXiv:2108.12757* **2021**.
60. Wang, J.; Tian, S.; Yu, L.; Wang, Y.; Wang, F.; Zhou, Z. SBDF-Net: A versatile dual-branch fusion network for medical image segmentation. *Biomedical Signal Processing and Control* **2022**, *78*, 103928.
61. Zhang, H.; Kyaw, Z.; Yu, J.; Chang, S.F. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. 2017, pp. 4233–4241.
62. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks **2015**. *28*.
63. Cao, T.; Kreis, K.; Fidler, S.; Sharp, N.; Yin, K. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. 2023, pp. 4169–4181.
64. CAO, Q.; KHANNA, P.; LANE, N.D.; BALASUBRAMANIAN, A. MobiVQA: Efficient On-Device Visual Question Answering. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*.
65. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. 2017, pp. 5998–6008.
66. Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2828–2837.
67. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* **2014**.
68. Xu, P.; Zhu, X.; Clifton, D.A. Multimodal learning with transformers: A survey **2023**. *45*, 12113–12132.
69. Chrysos, G.G.; Wang, B.; Deng, J.; Cevher, V. Regularization of polynomial networks for image recognition. 2023, pp. 16123–16132.
70. Li, X.; Huang, J.; Deng, L.J.; Huang, T.Z. Bilateral filter based total variation regularization for sparse hyperspectral image unmixing. *Information Sciences* **2019**, *504*, 334–353.
71. Sahoo, S.; Lampert, C.; Martius, G. Learning equations for extrapolation and control. 2018, pp. 4442–4450.
72. Li, Y.; Long, S.; Yang, Z.; Weng, H.; Zeng, K.; Huang, Z.; Wang, F.L.; Hao, T. A Bi-level representation learning model for medical visual question answering. *Journal of Biomedical Informatics* **2022**, *134*, 104183.
73. Nie, W.; Karras, T.; Garg, A.; Debnath, S.; Patney, A.; Patel, A.; Anandkumar, A. Semi-supervised stylegan for disentanglement learning. 2020, pp. 7360–7369.
74. Xia, Q.; Yu, C.; Peng, P.; Gu, H.; Zheng, Z.; Zhao, K. Visual Question Answering Based on Position Alignment. In Proceedings of the 2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, 2021, pp. 1–5.
75. Hinton, G.E.; Sejnowski, T.J. Optimal perceptual inference. 1983, Vol. 448, pp. 448–453.
76. Dai, A.M.; Le, Q.V. Semi-supervised sequence learning. 2015, Vol. 28.
77. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-aware neural language models. 2016.
78. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. 2011, pp. 315–323.

 doi:10.20944/preprints202505.1290.v1

79. Tan, K.; Huang, W.; Liu, X.; Hu, J.; Dong, S. A multi-modal fusion framework based on multi-task correlation learning for cancer prognosis prediction. *Artificial Intelligence in Medicine* **2022**, *126*, 102260.
80. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. 2016, pp. 4960–4964.
81. Tsuzuku, Y.; Sato, I.; Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. 2018.
82. Peng, Y.; Dalmia, S.; Lane, I.; Watanabe, S. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. 2022, pp. 17627–17643.
83. Wu, Y.; Chrysos, G.G.; Cevher, V. Adversarial audio synthesis with complex-valued polynomial networks. *arXiv preprint arXiv:2206.06811* **2022**.
84. Wang, Z.; She, Q.; Zhang, J. MaskNet: introducing feature-wise multiplication to CTR ranking models by instance-guided mask. *arXiv preprint arXiv:2102.07619* **2021**.
85. Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M.D. Large-scale weakly supervised audio classification using gated convolutional neural network. 2018, pp. 121–125.
86. Tang, C.; Salakhutdinov, R.R. Multiple futures prediction **2019**. *32*.
87. Zhang, Z.; Cai, J.; Zhang, Y.; Wang, J. Learning hierarchy-aware knowledge graph embeddings for link prediction. 2020, Vol. 34, pp. 3065–3072.
88. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using lstms. 2015, pp. 843–852.
89. Li, F.; Li, G.; He, X.; Cheng, J. Dynamic Dual Gating Neural Networks. 2021, pp. 5330–5339.
90. Babiloni, F.; Marras, I.; Deng, J.; Kokkinos, F.; Maggioni, M.; Chrysos, G.; Torr, P.; Zafeiriou, S. Linear Complexity Self-Attention With 3rd Order Polynomials **2023**. *45*, 12726–12737.
91. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. 2019.
92. Feng, Y.; Xu, H.; Jiang, J.; Liu, H.; Zheng, J. ICIF-Net: Intra-Scale Cross-Interaction and Inter-Scale Feature Fusion Network for Bitemporal Remote Sensing Images Change Detection **2022**. *60*, 1–13.
93. Murdock, C.; Li, Z.; Zhou, H.; Duerig, T. Blockout: Dynamic model selection for hierarchical deep networks. 2016.
94. Pan, L.; Chen, X.; Cai, Z.; Zhang, J.; Zhao, H.; Yi, S.; Liu, Z. Variational relational point completion network. 2021, pp. 8524–8533.
95. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. PMLR, 2020, pp. 5156–5165.
96. Watrous, R.; Kuhn, G. Induction of finite-state automata using second-order recurrent networks. 1991, Vol. 4.
97. Wang, Z.; Wang, K.; Yu, M.; Xiong, J.; Hwu, W.m.; Hasegawa-Johnson, M.; Shi, H. Interpretable visual reasoning via induced symbolic space. 2021, pp. 1878–1887.
98. Zhu, Z.; Liu, F.; Chrysos, G.G.; Cevher, V. Generalization Properties of NAS under Activation and Skip Connection Search. 2022.
99. Drumetz, L.; Veganzones, M.A.; Henrot, S.; Phlypo, R.; Chanussot, J.; Jutten, C. Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability **2016**. *25*, 3890–3905.
100. MacKay, M.; Vicol, P.; Lorraine, J.; Duvenaud, D.; Grosse, R. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. 2019.
101. Wang, Z.; Liu, X.; Li, H.; Sheng, L.; Yan, J.; Wang, X.; Shao, J. Camp: Cross-modal adaptive message passing for text-image retrieval. 2019, pp. 5764–5773.
102. Kumaraswamy, S.K.; Shi, M.; Kijak, E. Detecting human-object interaction with mixed supervision. 2021, pp. 1228–1237.
103. Hyeon-Woo, N.; Ye-Bin, M.; Oh, T.H. Fedpara: Low-rank hadamard product for communication-efficient federated learning. 2022.
104. Hyeon-Woo, N.; Ye-Bin, M.; Oh, T.H. FedPara: Low-rank Hadamard Product for Communication-Efficient Federated Learning. 2022.
105. Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-based vision: A survey **2020**. *44*, 154–180.

 doi:10.20944/preprints202505.1290.v1

106. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111.
107. Fan, F.; Xiong, J.; Wang, G. Universal Approximation with Quadratic Deep Networks. *Neural Netw.* **2020**.