General Transform: A Unified Framework for Adaptive Transform to Enhance Representations

Gekko Budiutama[a,b,*], Shunsuke Daimon[c], Hirofumi Nishi[a,b], Yu-ichiro Matsushita[a,b,c]

[a] *Quemix Inc., Tokyo, 1030027, Japan*
[b] *The University of Tokyo, Department of Physics Tokyo, 1130033, Japan*
[c] *National Institutes for Quantum Science and Technology, Quantum Materials and Applications Research Center Tokyo, 1528550, Japan*

Abstract

Discrete transforms, such as the discrete Fourier transform, are widely used in machine learning to improve model performance by extracting meaningful features. However, with numerous transforms available, selecting an appropriate one often depends on understanding the dataset's properties, making the approach less effective when such knowledge is unavailable. In this work, we propose General Transform (GT), an adaptive transform-based representation designed for machine learning applications. Unlike conventional transforms, GT learns data-driven mapping tailored to the dataset and task of interest. Here, we demonstrate that models incorporating GT outperform conventional transform-based approaches across computer vision and natural language processing tasks, highlighting its effectiveness in diverse learning scenarios.

Keywords: machine learning, deep learning, feature extraction

1. Introduction

Deep neural networks have consistently pushed the boundaries of performance on tasks in computer vision, natural language processing, and beyond. A significant trend in improving the efficiency of these systems involves the integration of discrete mathematical transforms, such as the discrete Fourier

*Corresponding author
Email address: bgekko@quemix.com (Gekko Budiutama)

transform, discrete cosine transform, and others (Jassim and Harte, 2022; Yi et al., 2025).

The use of discrete transforms provides multiple advantages in deep learning, particularly in handling high-dimensional data. These transforms enable efficient signal compression and decorrelation, which are crucial for a wide range of data types. The discrete cosine transform has been widely employed in image compression schemes, such as JPEG, due to its ability to concentrate energy into a few significant coefficients (Ahmed et al., 1974; Rao and Yip, 1990). Similarly, the fast Fourier transform is fundamental in spectral analysis and audio compression, as seen in MP3 encoding and speech processing (Cooley and Tukey, 1965). The discrete wavelet transform further extends these benefits by allowing multi-resolution analysis, making it highly effective in image compression formats like JPEG 2000 (Daubechies, 1992). Beyond compression, discrete transforms may also contribute to improved generalization in deep learning by filtering high-frequency noise and emphasizing dominant spectral components for computer vision tasks (Xu et al., 2020). Additionally, frequency-domain representations improve computational efficiency, as certain operations—such as convolutions—can be performed more efficiently in the spectral domain. For example, Fourier-based convolutional architectures leverage the fast Fourier transform to replace spatial-domain convolutions with element-wise multiplications, significantly reducing computational complexity (Chitsaz et al., 2020; Highlander and Rodriguez, 2016).

Despite their advantages, a fundamental limitation of these approaches is their reliance on predefined operations. The chosen transform applies the same mapping regardless of the input's specific characteristics, thus may fail to optimally capture task-specific or domain-specific features. Moreover, selecting an appropriate transform often requires prior knowledge of the dataset. This restricts generality and reduces the suitability of these methods for multimodal or highly heterogeneous data.

To address these challenges, we introduce General Transform (GT), a parameterized approach enabling flexible adaptation across various transforms. GT dynamically learns the most suitable transform or combination of transforms directly from task and dataset. The additional parameters introduced by GT is directly proportional to the number of included transforms, which remains relatively insignificant compared to the overall network size. Here, we demonstrate the application of GT in replacing conventional transforms for computer vision and natural language processing tasks. We found that the proposed GT enhances performance in large models while adding only three

additional parameters. Although our proof-of-concept focuses on image and text classification tasks, GT is a general approach that can be extended to other applications, making it a versatile tool across various domains. Thus, our main contributions include:

- General Transform (GT) is introduced as a parameterized approach that dynamically adapts to different transforms, learning the most suitable transforms directly from the task and dataset.
- The impact of GT on enhancing the performance of large deep learning models is analyzed for computer vision and natural language tasks, requiring only three additional parameters.
- GT is proven to be capable of capturing meaningful differences across different input channels in both image and language data processing.

2. Related Works

Discrete transforms have become increasingly prominent in machine learning, offering efficient methods to manipulate and analyze data across various domains. From computer vision to natural language processing (NLP) and time series analysis, these mathematical tools enable both improved performance and reduced computational overhead. Below, we provide an in-depth exploration of the most widely used discrete transforms in machine learning, accompanied by their key applications and recent research trends.

2.1. Discrete Fourier Transform (DFT)

DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} e^{-\frac{j2\pi kn}{N}} x[n], \tag{1}$$

where n and k are indices, $x[n]$ represents the original sequence of length N, $X[k]$ is the transformed representation of $x[n]$, underpins many frequency-domain methods in machine learning. In computer vision, the introduction of Fourier Neural Operators (Guibas et al., 2022) uses DFT to carry out global operations in the Fourier space, improving both model generalization and

efficiency. Similarly, DFT-based positional encodings (Tancik et al., 2020) help capture high-frequency details within vision transformer architectures.

In NLP, FNet (Lee-Thorp et al., 2022) exemplifies how replacing self-attention in the Transformer architecture (Vaswani et al., 2017) with Fourier-based token mixing can significantly reduce computational complexity while preserving strong performance. Autoregressive models also benefit from frequency-domain mixing (Lou et al., 2021), facilitating more efficient sequence modeling.

Time series analysis has leveraged DFT to model periodic behavior. TimeMixer (Wang et al., 2024), for instance, employs Fourier transforms to capture extended temporal dependencies, resulting in more robust forecasting. Additionally, model merging can be streamlined by transforming network parameters via DFT, as shown in (Zheng and Wang, 2024), while (Gao et al., 2024) demonstrated that the Discrete Fourier Transform (DFT) can be leveraged for parameter-efficient fine-tuning of large language models.

2.2. Discrete Cosine Transform Type-II (DCT)

DCT, given by:

$$X[k] = \sum_{n=0}^{N-1} \cos\left(\frac{\pi k\,(n+0.5)}{N}\right) x[n], \tag{2}$$

is widely recognized for decorrelation and compression, making it pivotal in image processing pipelines (e.g., JPEG) (Wallace, 1992). Recent work in computer vision (Xu et al., 2020; Ng and Beng Jin Teoh, 2015; Karaoglu and Eksioglu, 2023; Lee and Kim, 2024; Su et al., 2024) integrates DCT features or DCT-based attention, enhancing image classification and feature extraction. DCT-based decorrelated attention (Pan et al., 2024) has further demonstrated computational gains over conventional self-attention approaches.

In NLP, DCT-derived spectral token representations (Scribano et al., 2023) have shown promise for various tasks, including sentiment analysis and machine translation, by emphasizing important frequency components of textual data.

2.3. Discrete Wavelet Transform (DWT)

The Haar DWT in pure element–wise form is:

$$X[k] \;=\; \sum_{n=0}^{N-1} H(k,n)\, x[n]\,, \quad k = 0, \ldots, N-1, \tag{3}$$

where $N = 2^J$, and the scalar-kernel $H(k,n)$ is

$$H(k,n) = \begin{cases} 2^{-J/2}, & k = 0, \\ 2^{-j/2}, & k = 2^{\,J-j} + m, \;\; 2^j m \le n < 2^j m + 2^{\,j-1}, \\ -2^{-j/2}, & k = 2^{\,J-j} + m, \;\; 2^j m + 2^{\,j-1} \le n < 2^j (m+1), \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

for levels $j = 1, \ldots, J$ and shifts $m = 0, \ldots, 2^{\,J-j} - 1$.

Haar-based DWTs are used in deep learning pipelines to improve spatial-frequency modeling. For instance, wavelet CNNs (Liu et al., 2019) apply DWT to decompose inputs into multiple resolutions before convolution. In generative modeling, WaveletFlow (Yu et al., 2020) uses DWT for reversible transformations that facilitate likelihood computation in normalizing flows.

2.4. Discrete Legendre Transform (DLT)

DLT is defined as:

$$X[k] = \sum_{n=0}^{N-1} \hat{P}_k(t_n)\, x[n], \tag{5}$$

where

$$t_n = -1 + \frac{2n}{N-1}, \tag{6}$$

$$P_0(t_n) = 1, \quad P_1(t_n) = t_n, \tag{7}$$

$$P_{k+1}(t_n) = \frac{(2k+1)\, t_n\, P_k(t_n) - k\, P_{k-1}(t_n)}{k+1}, \quad k = 1, 2, \ldots, N-2, \tag{8}$$

$$\hat{P}_k(t_n) = \frac{P_k(t_n)}{\|P_k\|_2}, \quad \|P_k\|_2 = \sqrt{\sum_{n=0}^{N-1} [P_k(t_n)]^2}. \tag{9}$$

In machine learning, Discrete Legendre Transform (DLT) has been applied to solving nonlinear integral equations, particularly Volterra–Fredholm–Hammerstein integral equations (V-F-H-IEs), using Legendre polynomials as activation functions (Hajimohammadi et al., 2021). Additionally, Legendre polynomial-based neural networks have been employed for heat and mass transfer analysis in non-Newtonian fluids within porous channels (Khan et al., 2022) and for solving elliptic partial differential equations (PDEs) with enhanced accuracy and efficiency (Yang et al., 2019).

2.5. Other Discrete Transforms in Machine Learning

Beyond the above transforms, a range of other discrete transforms have been explored for machine learning applications:

- **Discrete Hartley transform**: Offers the advantage of using only real arithmetic, eliminating the need for complex-valued computations required in DFT. It has been applied in computer vision for various image processing tasks (Zhang and Ma, 2018; Wong et al., 2023; Mozafari et al., 2021) and in NLP for handling long-context representations (Giofré and Ghantasala, 2023).

- **Fractional Fourier transform**: Extends the standard Fourier transform to partial-frequency domains, benefitting signal denoising and feature extraction (Sahinuc and Koc, 2022; Kumar and Kansal, 2017; Kumari and Mustafi, 2022).

- **Walsh–Hadamard transform**: Offers binary orthogonal bases and fast computation, making it suitable for embedded systems and resource-constrained scenarios. WHT has been applied to vision tasks and integrated into neural networks (Pan et al., 2022a,b; Baldini et al., 2023).

3. Methods

3.1. General Transform

A major limitation of all the approaches above is their reliance on predefined, input-invariant operations, which may not optimally capture task-specific features and often require prior dataset knowledge, reducing their

generality for multimodal or heterogeneous data. To address these issues, we propose General Transform (GT).

Our definition of GT is as follows:

$$X[k] = \sum_{n=0}^{N-1} \left(\sum_{i=1}^{m} p_i f_i[n,k] + \left(1 - \sum_{i=1}^{m} p_i\right) f_{m+1}[n,k] \right) x[n] \tag{10}$$

Here, n and k are indices, $x[n]$ represents the original sequence of length N, $X[k]$ is the transformed representation of $x[n]$, $m+1$ is the number of transforms to be included in the GT, and f_i is the function for the i-th transform. The p_i is a trainable parameter for the i-th transform, optimized alongside the network parameters during training.

First, multiple transforms are selected to define the GT formula. During training, the weight of each transform (p_i) is optimized. Consequently, the GT mapping optimally converges to capture dataset features without requiring prior knowledge, dynamically adjusting the contribution of individual transforms, including their activation or deactivation, based on learned weights. The expressiveness of GT can be further enhanced by increasing the number of selected transforms. Moreover, the number of additional parameters introduced in this scheme corresponds directly to the number of included transformations.

While the above formulation uses addition to blend the different discrete transforms, the GT framework can accommodate any parametric combination (e.g., sums, products, or more complex mappings) of multiple discrete transforms, as long as the resulting operator (1) smoothly interpolates between two or more transforms as a function of one or more blending parameters, (2) recovers standard transforms at specific parameter values.

3.2. Experiments

3.2.1. Computer Vision Task

For the computer vision task, we implemented DFT (Eq. (1)), DCT (Eq. (2)), and DWT (Eq. (3)) as transformations, thereby defining GT for this task as:

$$Y[k] = \sum_{n=0}^{N-1} \left(p_1 \cos\!\left(\tfrac{\pi k\,(n+0.5)}{N}\right) + p_2\, e^{-\frac{j2\pi kn}{N}} + (1 - p_1 - p_2)\, H(k,n) \right) x[n]. \tag{11}$$

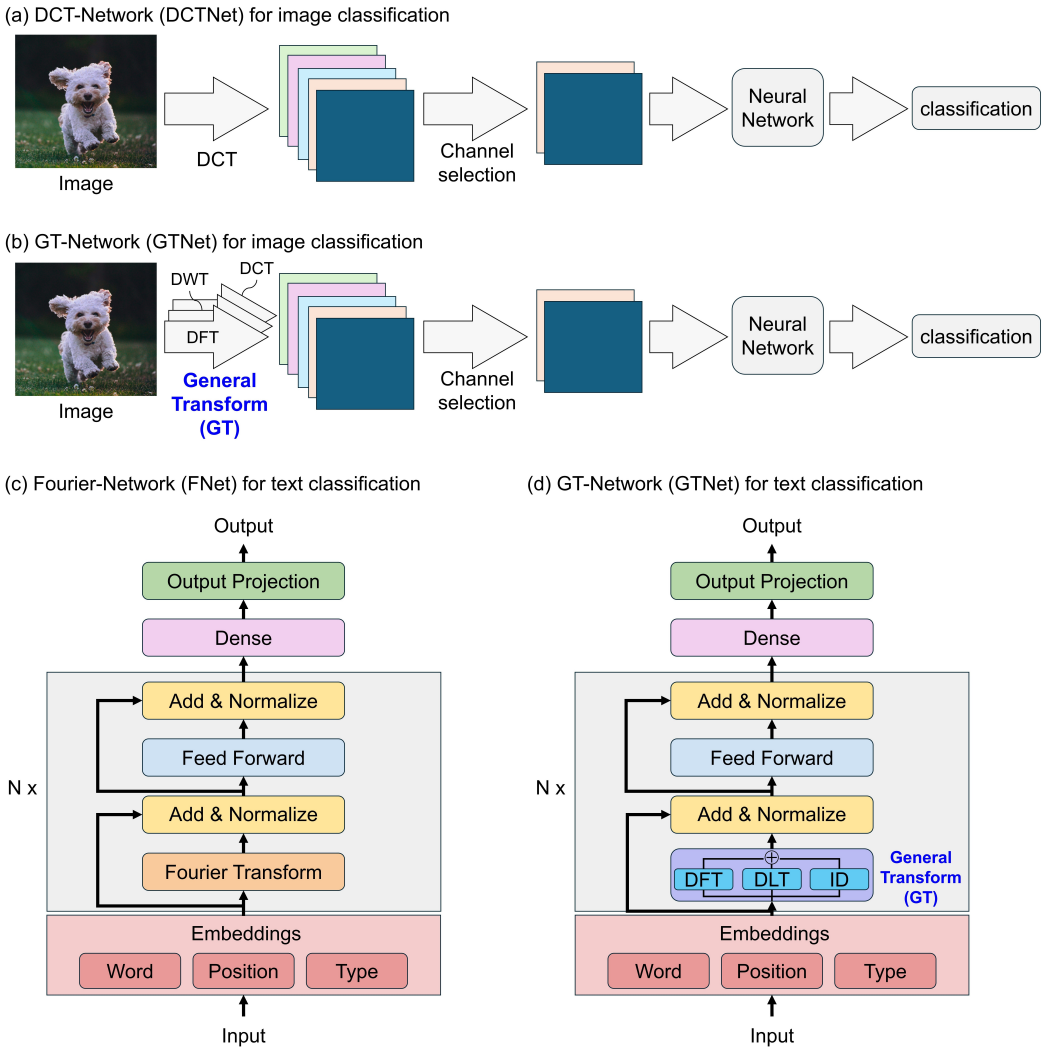

Figure 1: Image classification using RGB images as input, with DCT (Xu et al., 2020) (a) and GT (b) for feature extraction. Text classification with DFT (Lee-Thorp et al., 2022) (c) and GT (d) for token mixing.

The final output is a parameterized summation of the real and imaginary components of $Y[k]$.

$$X[k] = p_3\, Y[k]_{\text{real}} + (1 - p_3)\, Y[k]_{\text{imaginary}} \tag{12}$$

We replaced the DCT-based feature extraction used in (Xu et al., 2020) with the GT (Fig. 1(a) and (b)). Following this work, we implemented three versions of each model, each employing a different number of frequency channels (24, 48, and 64) as input to the ResNet-50 (He et al., 2016) architecture. In this experiment, the input image was first converted to the YCbCr color space, then divided into non-overlapping 8×8 blocks. A 2D DCT (DCT-Net) or GT (GTNet) was applied to each block, resulting in 64 frequency coefficients. Depending on the model variant, only the first 24, 48, or all 64 coefficients were retained. These selected frequency channels were aggregated and normalized by channel-specific mean and standard deviation. The processed data was then fed into neural networks.

The models were trained from scratch using the ImageNet 2012 Large-Scale Visual Recognition Challenge (ILSVRC-2012) (Deng et al., 2009) dataset. The cross-entropy loss function used as the objective function is given by:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{\exp(z_{i,y_i})}{\sum_{j=1}^{M}\exp(z_{i,j})}\right), \tag{13}$$

where N denotes the number of samples in the batch, M represents the total number of classes, and $z_{i,j}$ is the unnormalized logit score corresponding to the j-th class of the i-th sample. The ground-truth class label for the i-th sample is denoted by $y_i \in \{1, \ldots, M\}$. In this context, the softmax probability assigned to the correct class y_i is given as $q_{i,y_i} = \frac{\exp(z_{i,y_i})}{\sum_{j=1}^{M}\exp(z_{i,j})}$. During training, we adopted stochastic gradient descent (SGD) (Bottou, 2010) with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 10^{-4}. Training proceeded for 80 epochs with a batch size of 150, and the learning rate was decayed by a factor of 0.1 every 31 epochs. The objective function $\mathcal{L}$ and the top-1 accuracy are used to evaluate model performance. The top-1 accuracy is defined as:

$$A_{\text{Top-1}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i) \times 100\%, \tag{14}$$

where N denotes the number of samples in the batch, the indicator function $\mathbb{1}(\cdot)$ returns 1 if its argument is true and 0 otherwise. Here, y_i is the ground-truth label for the i-th sample, and $\hat{y}_i$ is the predicted label. The predicted label is computed as:

$$\hat{y}_i = \underset{j \in \{1,\ldots,M\}}{\arg\max} \frac{\exp(z_{i,j})}{\sum_{k=1}^{M} \exp(z_{i,k})}. \tag{15}$$

Because the softmax operation preserves the ordering of the logits—that is, the class with the highest logit will also have the highest softmax probability—we can simply choose the predicted label as:

$$\hat{y}_i = \underset{j \in \{1,\ldots,M\}}{\arg\max}\, z_{i,j}. \tag{16}$$

Thus, as the model outputs a probability distribution over the classes, the predicted label $\hat{y}_i$ is chosen as the class with the highest predicted probability.

For each model, the epoch with the highest validation $A_{\text{Top-1}}$ within the 80 training epochs is selected for comparison.

3.2.2. Natural Language Processing Task

For the NLP task, we implemented three transformation basis functions: the DFT (Eq. (1)), the DLT (Eq. (5)), and an identity transform. Thus the GT used for the task is defined as:

$$Y[k] \;=\; \sum_{n=0}^{N-1} \Big(p_1\, e^{-j\frac{2\pi kn}{N}} \;+\; p_2\, \hat{P}_k(t_n) \;+\; (1 - p_1 - p_2)\, \delta_{k,n} \Big)\, x[n], \tag{17}$$

where the $\delta_{k,n}$ is

$$\delta_{k,n} \;=\; \begin{cases} 1, & \text{if } k = n, \\ 0, & \text{otherwise.} \end{cases}$$

The final output is a parameterized summation of the real and imaginary components of $Y[k]$.

$$X[k] = p_3 Y[k]_{\text{real}} + (1 - p_3) Y[k]_{\text{imaginary}} \tag{18}$$

Then, we replaced the DFT-based token mixing proposed in (Lee-Thorp et al., 2022) with GT (Fig. 1(c) and (d)). Here, each text sample is first tokenized using a SentencePiece model with a 32k-vocabulary (Kudo and Richardson, 2018). Then, the tokens are mapped

to vectors through an embedding layer that includes token, positional, and segment embeddings. Within each encoder block, the self-attention sublayer of the Transformer is replaced by a two-dimensional DFT or GT applied across both the sequence and hidden dimensions. The processed data is then fed into feed-forward neural networks.

We evaluated the effectiveness of GT by fine-tuning pre-trained models. The model parameters were initialized using the pre-trained weights and subsequently fine-tuned on the SST-2 (Socher et al., 2013) and CoLA (Warstadt et al., 2019) datasets, using either DFT (FNet) or GT (GTNet) for token mixing. GT's parameters were initialized with a pure real-part DFT configuration ($p_1 = 1,\ p_2 = 0,\ p_3 = 1$), following the FNet setup.

During fine-tuning, the learning rate was set to increase to 10^{-5} by epoch 2 and gradually decayed to 0 by epoch 20. However, the primary analysis was focused on the first five epochs, as severe over-fitting was observed beyond this point. We used cross-entropy loss (Eq. (13)) as the training objective. The models were fine-tuned using the AdamW optimizer (Loshchilov and Hutter, 2019) with a weight decay coefficient of 0.01. Specifically, we employed $\beta_1 = 0.9$, $\beta_2 = 0.999$ and eps $= 1 \times 10^{-6}$. A maximum sequence length of 512 tokens was used, with a batch size of 64. The fine-tuning process was repeated 10 times, and the L and $A_{\text{Top-1}}$ were averaged for evaluation. For each model, the epoch that achieves the highest validation $A_{\text{Top-1}}$ within the 5 training epochs is chosen for comparison.

Table 1: ImageNet classification results using DCTNet and GTNet with varying numbers of channel inputs to ResNet-50.

		Training		**Validation**	
Model	**Channel**	L	$A_{\text{Top-1}}$	L	$A_{\text{Top-1}}$
DCTNet	24	0.9688	76.70	0.9388	76.53
GTNet	24	0.9599	76.90	0.9377	76.62
DCTNet	48	0.9637	76.87	0.9322	76.63
GTNet	48	0.9358	77.43	0.9203	76.90
DCTNet	64	0.9453	77.27	0.9246	76.67
GTNet	64	0.9503	77.16	0.9268	76.72

4. Results

4.1. Computer Vision Task

For computer vision task, replacing DCT with the GT on the same network improved validation $A_{\text{Top-1}}$ and reduces L on the ImageNet 2012 dataset classification. Table 1 compares the performance of DCTNet and the proposed GTNet for image classification using ImageNet dataset. GTNet consistently outperforms DCTNet in validation $A_{\text{Top-1}}$, with an increase from 76.53% to 76.62% for the 24-channel configuration, 76.63% to 76.90% for the 48-channel configuration, and 76.67% to 76.72% for the 64-channel configuration. Additionally, the validation L for GTNet with the 24-channel and 48-channel configurations is lower than their DCTNet counterparts: 0.9377 compared to 0.9388 for the 24-channel configuration, and 0.9203 compared to 0.9322 for the 48-channel configuration. Note that this improvement was achieved with the addition of only three parameters. Compared to the 25 million parameters of the ResNet-50 architecture, this additional computational cost is negligible.

Figure 2 presents the training and validation L curves (left) and $A_{\text{Top-1}}$ curves (right) for DCTNet and GTNet across different frequency channel configurations: 24 (a, b), 48 (c, d), and 64 (e, f). In the L curves (Fig. 2(a, c, e)), GTNet consistently exhibits lower validation loss than DCTNet, particularly at 24 channels (Fig. 2(a), GTNet-24: red solid line; DCTNet-24: blue solid line) and 48 channels (Fig. 2(c), GTNet-48: purple solid line; DCTNet-48: green solid line), indicating improved generalization. However, as the number of channels increases to 64, the performance gap narrows (Fig. 2(e)), likely due to the larger network capacity diminishing the relative benefits of GTNet. In the $A_{\text{Top-1}}$ curves (Fig. 2(b, d, f)), GTNet consistently outperforms DCTNet across all channel configurations, maintaining a similar accuracy improvement at 24 (Fig. 2(b), GTNet-24: red solid line; DCTNet-24: blue solid line), 48 (Fig. 2(d), GTNet-48: purple solid line; DCTNet-48: green solid line), and 64 (Fig. 2(f), GTNet-64: brown solid line; DCTNet-48: yellow solid line) channels.

GT as defined in Eq. (11) recovers standard transforms at specific parameter values. However, our analysis of the optimized GT parameters reveals that the optimal mapping is not a standalone component transform and differs between the input channels and the model size (Table 2). For instance, the optimal values of p_1 and p_2 for the Y-channel of GTNet-24 are 0.84 and 0.15, respectively, indicating that a mixture dominated by DCT and DFT,

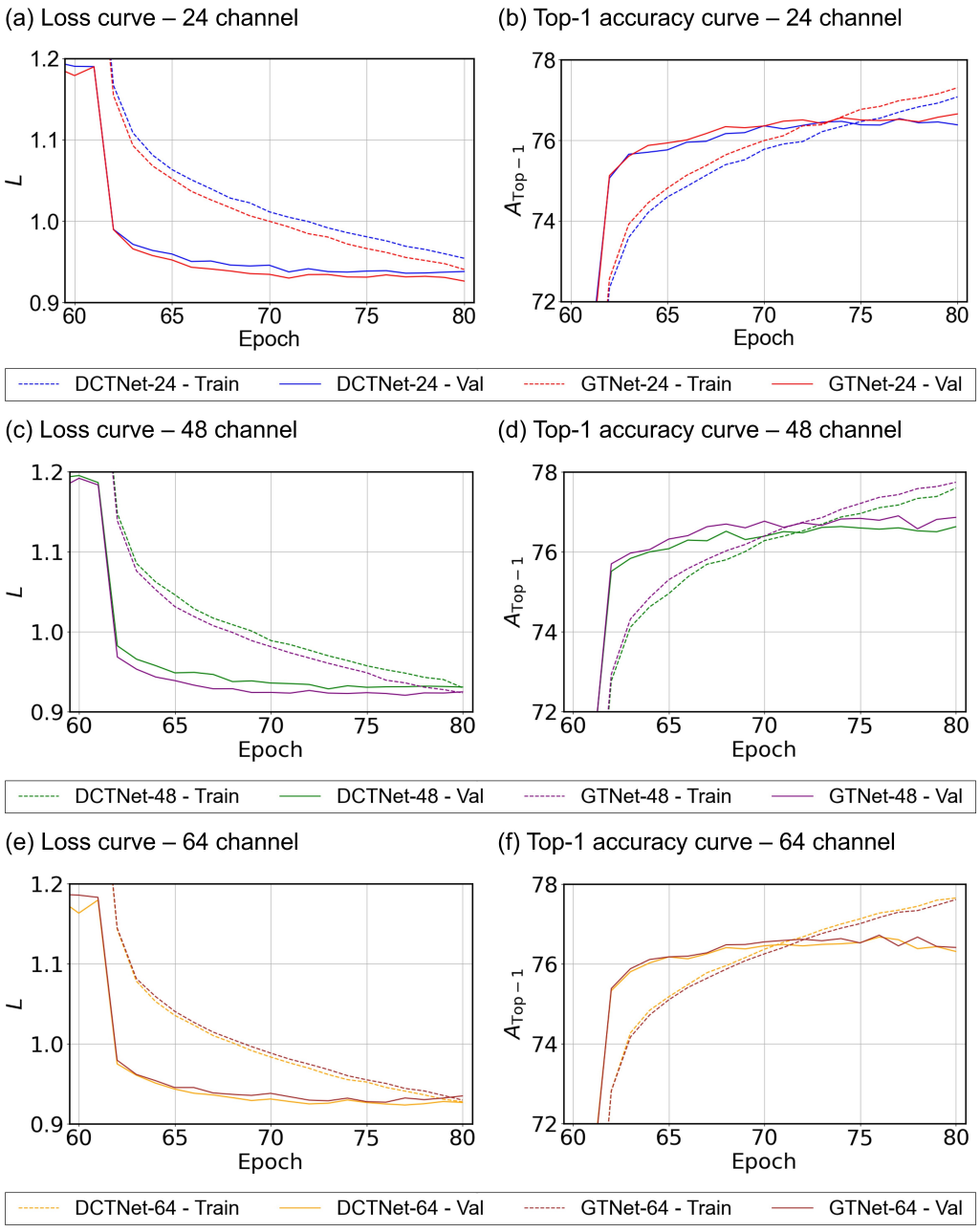

Figure 2: Loss and accuracy curves for image classification on the ImageNet 2012 dataset using DCTNet and GTNet, with 24 (a, b), 48 (c, d), and 64 (e, f) input channels.

Table 2: Optimized parameters of General Transform.

	Y-channel			**Cb-channel**			**Cr-channel**		
	p_1	p_2	p_3	p_1	p_2	p_3	p_1	p_2	p_3
GTNet-24	0.84	0.15	0.65	-1.85	-0.21	0.18	1.94	0.18	0.16
GTNet-48	-3.63	-0.06	0.15	2.10	0.74	0.30	2.11	0.74	0.31
GTNet-64	-0.90	-1.68	2.20	2.93	1.63	0.66	3.12	1.71	0.61

with a minor contribution from DWT, yields optimal performance. In contrast, for the larger GTNet-48, the optimal p_1 and p_2 for the Y-channel are -3.63 and -0.06, respectively, suggesting that a combination favoring DCT and DWT, with a negligible influence from DFT, is preferable. We also observed that, within the same model, the luminance (Y) channel has distinct p_1 and p_2 values compared to the chromatic (Cb, Cr) channels. This difference likely arises because the Y-channel captures more structural and high-frequency details compared to the chrominance channels (Cb and Cr), which primarily encode color information. Notably, GT effectively captures and adapts to these differences, learning distinct mappings for luminance and chrominance.

4.2. Natural Language Processing Task

GT also improved the model performance compared to DFT in natural language processing tasks. Table 3 summarizes the fine-tuning results of the pre-trained network, which employs DFT (FNet) and GT (GTNet) for token mixing, for both the base and large models with 83 million and 238 million parameters, respectively.

Table 3: Fine-tuning results of FNet and GTNet on CoLA and SST-2 tasks.

	CoLA				**SST-2**			
	Training		**Validation**		**Training**		**Validation**	
	L	$A_{\text{Top-1}}$	L	$A_{\text{Top-1}}$	L	$A_{\text{Top-1}}$	L	$A_{\text{Top-1}}$
FNet-base	0.4225	81.40	0.5875	72.48	0.1607	94.03	0.3876	87.58
GTNet-base	0.4192	81.58	0.5624	74.04	0.1256	95.47	0.3835	88.50
FNet-large	0.2828	89.49	0.5336	77.75	0.0825	97.29	0.3560	90.04
GTNet-large	0.2821	89.54	0.5238	78.74	0.0834	97.26	0.3342	90.70

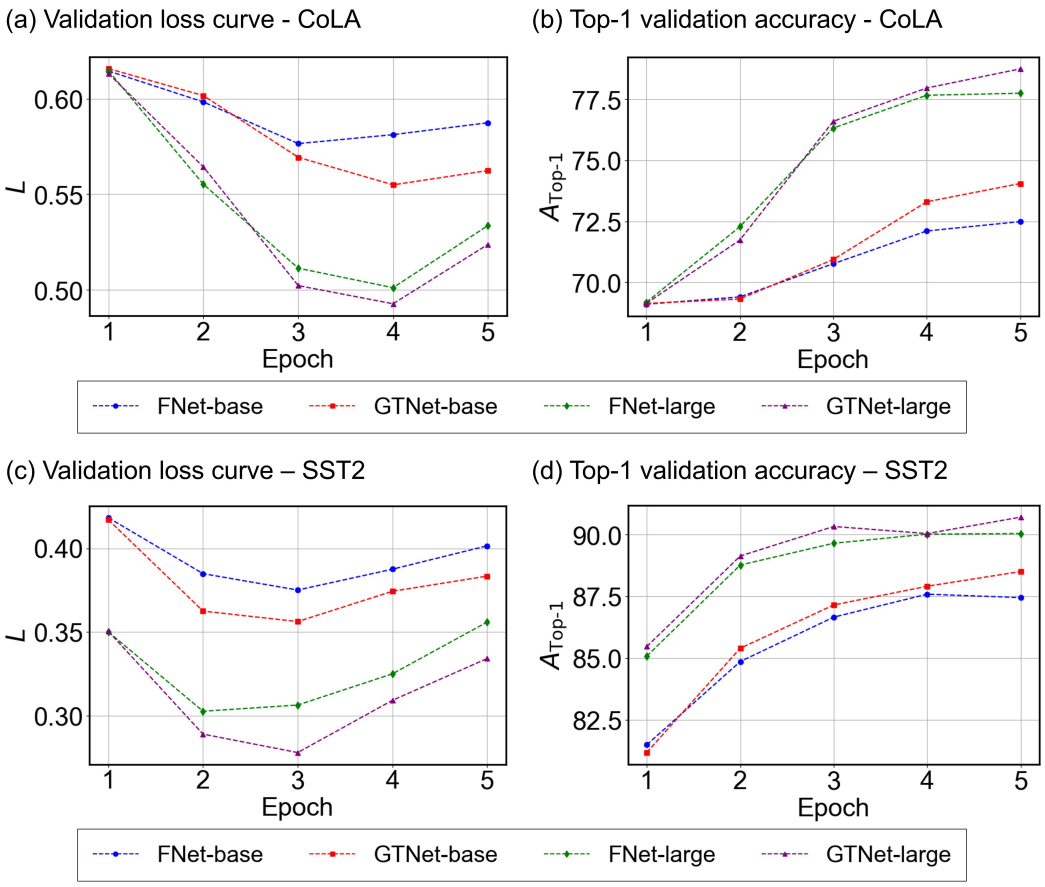

Figure 3: Validation loss (L) curve and top-1 accuracy ($A_{\text{Top-1}}$) curve for CoLA (a and b) and SST-2 (c and d) datasets.

For the CoLA task, GTNet consistently outperforms FNet in both validation $A_{\text{Top-1}}$ and L across both model sizes. GTNet-base achieves a validation $A_{\text{Top-1}}$ of 74.04%, compared to 72.48% for FNet-base. For large models, GTNet-large achieves 78.74%, surpassing the 77.75% of FNet-large. GTNet also demonstrates lower validation L compared to FNet across both base and large models. GTNet-base achieves a validation loss of 0.5624, compared to 0.5875 for FNet-base. For large models, GTNet-large achieves 0.5238, while FNet-large achieves 0.5336.

For the SST-2 task, GTNet also demonstrates superior performance compared to FNet. The validation $A_{\text{Top-1}}$ for GTNet-base is 88.50%, higher than the 87.58% of FNet-base. For the larger models, GTNet-large achieves 90.70%, outperforming the 90.04% of FNet-large. GTNet consistently achieves lower validation L as well, with GTNet-base achieving 0.3835, compared to 0.3876 for FNet-base. For the large models, GTNet-large achieves 0.3342, compared to 0.3560 for FNet-large.

Figure 3 presents the average validation L (a, c) and $A_{\text{Top-1}}$ (b, d) curves for FNet and GTNet on the CoLA and SST-2 datasets over the first five epochs of fine-tuning. In the CoLA validation L curve (Fig. 3(a)), GTNet-large (purple) achieves a noticeably lower loss than FNet-large (green). GTNet-base (red) also outperforms FNet-base (blue). A similar trend is observed in the SST-2 validation L curve (Fig. 3(c)), where GTNet-large (purple) achieves a lower L than FNet-large (green). GTNet-base (red) consistently outperforms FNet-base (blue). For validation $A_{\text{Top-1}}$ on CoLA (Fig. 3(b)), GTNet-large (purple) outperforms FNet-large (green), while GTNet-base (red) consistently surpasses FNet-base (blue). On SST-2 validation $A_{\text{Top-1}}$ (Fig. 3(d)), GTNet-large (purple) maintains a performance lead over FNet-large (green). GTNet-base (red) consistently outperforms FNet-base (blue) throughout training, demonstrating superior accuracy across all epochs.

5. Extension to Quantum Computing Architectures

5.1. Parameterized Quantum General Transform via Linear Combination of Unitaries

The above General Transform (GT) can be further extended to quantum computing by leveraging the principle of Linear Combination of Unitaries (LCU) (Childs et al., 2017; Chakraborty, 2024; Kosugi and Matsushita, 2020). This extension yields the Quantum General Transform (QGT), a

quantum-native, trainable operator that performs adaptive feature mapping through a coherent superposition of multiple unitaries. Rather than selecting a single fixed unitary transformation U, the QGT is defined as a variationally parameterized operator of the form:

$$U_{\mathrm{QGT}} := \sum_{i=0}^{m-1} p_i U_i \tag{19}$$

Here, $\{U_i\}_{i=0}^{m-1}$ are known n-qubit unitary matrices, such as Quantum Fourier Transform (QFT) (Shor, 1994; Nielsen and Chuang, 2000), Instantaneous Quantum Polynomial (IQP) (Shepherd and Bremner, 2009; Bremner et al., 2017), or Clifford+T operators (Bravyi and Gosset, 2016; Nielsen and Chuang, 2000), and $\{p_i\}$ are trainable, positive weights constrained by $\sum_i p_i = 1$.

A typical LCU implementation introduces ancilla qubits prepared in a superposition state:

$$|\chi\rangle = \sum_i \sqrt{p_i}|i\rangle \tag{20}$$

where $|i\rangle$ is the computational basis state of the ancilla that acts as a selector for the corresponding unitary U_i.

Following the standard LCU framework, we define the multiplexed unitary:

$$\mathrm{SELECT}(U) := \sum_i |i\rangle\langle i| \otimes U_i \tag{21}$$

Post-selecting the ancilla back onto the state $|\chi\rangle$ yields the effective operation:

$$\langle\chi|\mathrm{SELECT}(U)|\chi\rangle = U_{\mathrm{QGT}}, \tag{22}$$

recovering the non-unitary operator via coherent projection. In physical implementations, amplitude amplification may be used to boost the success probability of this postselection. Thus, QGT acts as a learnable quantum module with learnable inductive bias over known quantum primitives.

5.2. Quantum General Transform for Image Classification Tasks

5.2.1. Experiments

We investigate the application of the Quantum General Transform (QGT) in the context of image classification, using the ImageNet2012 dataset as a benchmark (Deng et al., 2009). In this setting, QGT acts as a trainable,

Table 4: Experimental Configurations for QGT-based ImageNet2012 Classification

Experiment ID	**#LCU Unitaries**	**Unitary Types Used**
S-1	2	IQP, QFT
S-2	2	IQP, Clifford+T
S-3	2	Clifford+T, QFT
S-4	4	Clifford+T, QFT, IQP, QNN (5-layers)

quantum-native transformation applied to image patches prior to classification. Following the experimental setup in (Xu et al., 2020), we substitute their DCT-based feature extraction pipeline with QGT of 3-qubit system, using 64 frequency channels as input to the ResNet-50 architecture. The classical input vectors are embedded into quantum Hilbert space through amplitude encoding, serving as the initial state for the QGT circuit.

To explore the expressive capacity and trainability of the QGT under various quantum gate compositions, we test configurations comprising either two or four unitaries (Table 4). The unitary operators considered include the Quantum Fourier Transform (QFT) (Shor, 1994; Nielsen and Chuang, 2000), Instantaneous Quantum Polynomial (IQP) circuits (Shepherd and Bremner, 2009; Bremner et al., 2017), Clifford+T (Bravyi and Gosset, 2016; Nielsen and Chuang, 2000), and parameterized Quantum Neural Networks (QNNs) (Schuld et al., 2020) (Fig. 4).

5.2.2. Results and Discussion

Figure 5 presents the training and validation performance for all four QGT-based configurations on the ImageNet2012 dataset. Figure 5(a) shows the evolution of the cross-entropy loss L over 80 training epochs, while Figure 5(b) illustrates the corresponding $A_{\text{Top-1}}$. Across all experimental settings (S1–S4), the QGT exhibits stable convergence, with monotonically decreasing training loss and improving accuracy, demonstrating consistent learnability.

Although the performance does not surpass that of classical General Transforms (GT), it is important to note that all hyperparameters—including optimizer settings and total training epochs—were inherited directly from the classical baseline, without any additional tuning for the quantum setting. Despite this, the QGT proves to be fully trainable on a large-scale, real-world classification task, underscoring its viability as a quantum-native

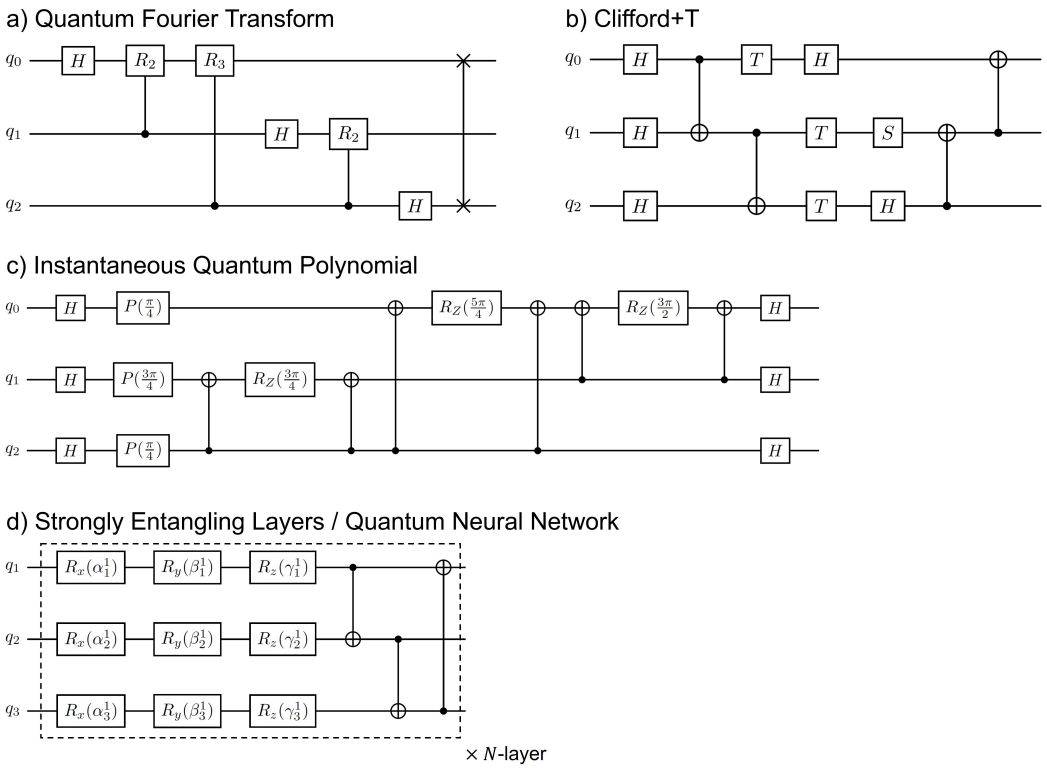

Figure 4: Different unitaries used for QGT: (a) Quantum Fourier Transform (QFT) where $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$, (b) Clifford+T, (c) Instantaneous Quantum Polynomial (IQP) and (d) Quantum Neural Network (QNN).

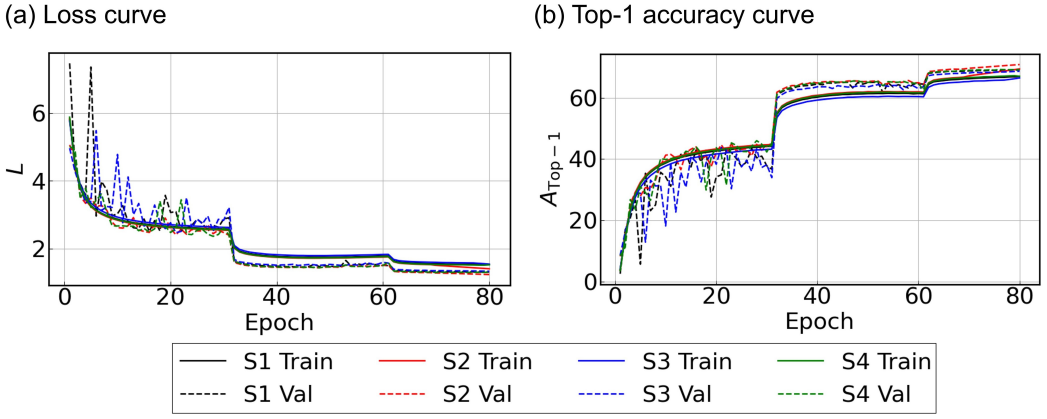

Figure 5: Training and validation loss (L) curves (a) and top-1 accuracy ($A_{\text{Top-1}}$) curves (b) for image classification on the ImageNet 2012 dataset using quantum general transform.

preprocessing strategy.

Among the tested variants, S2 and S4—which incorporate Clifford+T unitaries—tend to yield slightly better generalization. Interestingly, S2 (comprising only two unitaries) outperforms S4 (which combines four unitaries including a parameterized QNN) in validation $A_{\text{Top-1}}$, suggesting that increasing the number of unitaries does not straightforwardly lead to better performance. This highlights the importance of careful circuit composition.

This study constitutes a preliminary exploration of the circuit design space for QGT in vision tasks. These results should be interpreted as a proof of concept, rather than an optimized system. Future work will focus on circuit architecture search, adaptive training protocols, and improved parameter initialization schemes to better exploit the quantum model capacity.

6. Conclusion

In this study, we proposed General Transform (GT) for machine learning applications. GT generalizes the use of transforms in machine learning by allowing models to dynamically determine the optimal transform or combination of transforms based on the task and dataset. Through experiments on computer vision and natural language processing tasks, we demonstrated that GT consistently improves the performance of large-scale deep learning models. Notably, these gains are achieved with minimal overhead, requiring only a few additional parameters. The ability of GT to adapt to diverse datasets and tasks without requiring prior knowledge of their structure underscores its robustness and generalizability. These properties make GT a promising paradigm for more flexible and efficient transform-based learning in neural networks.

Acknowledgements

This work was supported by JSPS KAKENHI under Grant-in-Aid for Early-Career Scientists No. JP24K16985 and JSPS KAKENHI under Grant-in-Aid for Transformative Research Areas No. JP22H05114. This study was partially carried out using the facilities of the Supercomputer Center, the Institute for Solid State Physics, the University of Tokyo and the TSUBAME4.0 supercomputer at the Institute of Science Tokyo. This work was partially supported by the Center of Innovations for Sustainable Quantum AI (JST Grant Number JPMJPF2221). The author acknowledges the contributions and discussions provided by the members of Quemix Inc.

References

Ahmed, N., Natarajan, T., Rao, K., 1974. Discrete cosine transform. IEEE Transactions on Computers C-23, 90–93. doi:10.1109/T-C.1974.223784.

Baldini, G., Bonavitacola, F., Chareau, J.M., 2023. Fading channel classification with walsh-hadamard transform and convolutional neural network, in: 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), IEEE. p. 1–6. URL: http://dx.doi.org/10.1109/SmartNets58706.2023.10215941, doi:10.1109/smartnets58706.2023.10215941.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent, in: Lechevallier, Y., Saporta, G. (Eds.), Proceedings of COMPSTAT'2010, Physica-Verlag HD, Heidelberg. pp. 177–186.

Bravyi, S., Gosset, D., 2016. Improved classical simulation of quantum circuits dominated by clifford gates. Phys. Rev. Lett. 116, 250501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.250501, doi:10.1103/PhysRevLett.116.250501.

Bremner, M.J., Montanaro, A., Shepherd, D.J., 2017. Achieving quantum supremacy with sparse and noisy commuting quantum computations. Quantum 1, 8. URL: https://doi.org/10.22331/q-2017-04-25-8, doi:10.22331/q-2017-04-25-8.

Chakraborty, S., 2024. Implementing any Linear Combination of Unitaries on Intermediate-term Quantum Computers. Quantum 8, 1496. URL: https://doi.org/10.22331/q-2024-10-10-1496, doi:10.22331/q-2024-10-10-1496.

Childs, A.M., Kothari, R., Somma, R.D., 2017. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. SIAM J. Comput. 46, 1920–1950. URL: https://doi.org/10.1137/16M1087072, doi:10.1137/16M1087072.

Chitsaz, K., Hajabdollahi, M., Karimi, N., Samavi, S., Shirani, S., 2020. Acceleration of convolutional neural network using fft-based split convolutions. ArXiv abs/2003.12621. URL: https://api.semanticscholar.org/CorpusID:214713959.

Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine calculation of complex fourier series. Mathematics of Computation 19, 297–301. URL: http://www.jstor.org/stable/2003354.

Daubechies, I., 1992. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611970104, doi:10.1137/1.9781611970104, arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611970104.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

Gao, Z., Wang, Q., Chen, A., Liu, Z., Wu, B., Chen, L., Li, J., 2024. Parameter-efficient fine-tuning with discrete fourier transform, in: Proceedings of the 41st International Conference on Machine Learning, JMLR.org.

Giofré, D., Ghantasala, S., 2023. Legal-hnet: Mixing legal long-context tokens with hartley transform. URL: https://arxiv.org/abs/2311.05089, arXiv:2311.05089.

Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A., Catanzaro, B., 2022. Adaptive fourier neural operators: Efficient token mixers for transformers. URL: https://arxiv.org/abs/2111.13587, arXiv:2111.13587.

Hajimohammadi, Z., Parand, K., Ghodsi, A., 2021. Legendre deep neural network (ldnn) and its application for approximation of nonlinear volterra fredholm hammerstein integral equations. URL: https://arxiv.org/abs/2106.14320, arXiv:2106.14320.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. doi:10.1109/CVPR.2016.90.

Highlander, T., Rodriguez, A., 2016. Very efficient training of convolutional neural networks using fast fourier transform and overlap-and-add. ArXiv abs/1601.06815. URL: https://api.semanticscholar.org/CorpusID:2543043.

Jassim, W.A., Harte, N., 2022. Comparison of discrete transforms for deep-neural-networks-based speech enhancement. IET Signal Processing 16, 438–448. doi:https://doi.org/10.1049/sil2.12109.

Karaoglu, H.H., Eksioglu, E.M., 2023. Dctnet: deep shrinkage denoising via dct filterbanks. Signal, Image and Video Processing 17, 3665–3676. URL: http://dx.doi.org/10.1007/s11760-023-02593-0, doi:10.1007/s11760-023-02593-0.

Khan, N.A., Sulaiman, M., Kumam, P., Alarfaj, F.K., 2022. Application of legendre polynomials based neural networks for the analysis of heat and mass transfer of a non-newtonian fluid in a porous channel. Advances in Continuous and Discrete Models 2022. URL: http://dx.doi.org/10.1186/s13662-022-03676-x, doi:10.1186/s13662-022-03676-x.

Kosugi, T., Matsushita, Y.i., 2020. Linear-response functions of molecules on a quantum computer: Charge and spin responses and optical absorption. Phys. Rev. Res. 2, 033043. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.2.033043, doi:10.1103/PhysRevResearch.2.033043.

Kudo, T., Richardson, J., 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, in: Blanco, E., Lu, W. (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Brussels, Belgium. pp. 66–71. URL: https://aclanthology.org/D18-2012/, doi:10.18653/v1/D18-2012.

Kumar, P., Kansal, S., 2017. Noise removal in speech signal using fractional fourier transform, in: 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp. 1–4. doi:10.1109/ICOMICON.2017.8279117.

Kumari, R., Mustafi, A., 2022. Denoising of images using fractional fourier transform, in: 2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), pp. 1–6. doi:10.1109/ICEFEET51821.2022.9848244.

Lee, J., Kim, H., 2024. Dct-vit: High-frequency pruned vision transformer with discrete cosine transform. IEEE Access

12, 80386–80396. URL: http://dx.doi.org/10.1109/ACCESS.2024.3410231, doi:10.1109/access.2024.3410231.

Lee-Thorp, J., Ainslie, J., Eckstein, I., Ontanon, S., 2022. Fnet: Mixing tokens with fourier transforms. URL: https://arxiv.org/abs/2105.03824, arXiv:2105.03824.

Liu, P., Zhang, H., Lian, W., Zuo, W., 2019. Multi-level wavelet convolutional neural networks. IEEE Access 7, 74973–74985. doi:10.1109/ACCESS.2019.2921451.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Lou, T., Park, M., Ramezanali, M., Tang, V., 2021. Fnetar: Mixing tokens with autoregressive fourier transforms. URL: https://arxiv.org/abs/2107.10932, arXiv:2107.10932.

Mozafari, S.H., Clark, J.J., Gross, W.J., Meyer, B.H., 2021. Hartley stochastic computing for convolutional neural networks, in: 2021 IEEE Workshop on Signal Processing Systems (SiPS), pp. 1–6. doi:10.1109/SiPS52927.2021.00049.

Ng, C.J., Beng Jin Teoh, A., 2015. Dctnet: A simple learning-free approach for face recognition, in: 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 761–768. doi:10.1109/APSIPA.2015.7415375.

Nielsen, M.A., Chuang, I.L., 2000. Quantum Computation and Quantum Information. Cambridge University Press.

Pan, H., Badawi, D., Cetin, A.E., 2022a. Block walsh–hadamard transform-based binary layers in deep neural networks. ACM Trans. Embed. Comput. Syst. 21. URL: https://doi.org/10.1145/3510026, doi:10.1145/3510026.

Pan, H., Badawi, D., Chen, C., Watts, A., Koyuncu, E., Cetin, A.E., 2022b. Deep neural network with walsh-hadamard transform layer for ember detection during a wildfire, in: Proceedings of the IEEE/CVF Conference

on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 257–266.

Pan, H., Hamdan, E., Zhu, X., Biswas, K., Cetin, A.E., Bagci, U., 2024. Dct-based decorrelated attention for vision transformers. URL: https://arxiv.org/abs/2405.13901, arXiv:2405.13901.

Rao, K.R., Yip, P., 1990. Discrete cosine transform: algorithms, advantages, applications. Academic Press Professional, Inc., USA.

Sahinuc, F., Koc, A., 2022. Fractional fourier transform meets transformer encoder. IEEE Signal Processing Letters 29, 2258–2262. URL: http://dx.doi.org/10.1109/LSP.2022.3217975, doi:10.1109/lsp.2022.3217975.

Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N., 2020. Circuit-centric quantum classifiers. Phys. Rev. A 101, 032308. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.032308, doi:10.1103/PhysRevA.101.032308.

Scribano, C., Franchini, G., Prato, M., Bertogna, M., 2023. Dct-former: Efficient self-attention with discrete cosine transform. Journal of Scientific Computing 94. URL: http://dx.doi.org/10.1007/s10915-023-02125-5, doi:10.1007/s10915-023-02125-5.

Shepherd, D., Bremner, M.J., 2009. Temporally unstructured quantum computation. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 465, 1413–1439. URL: http://dx.doi.org/10.1098/rspa.2008.0443, doi:10.1098/rspa.2008.0443.

Shor, P., 1994. Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. doi:10.1109/SFCS.1994.365700.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C., 2013. Recursive deep models for semantic compositionality over a sentiment treebank, in: Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., Bethard, S. (Eds.), Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA. pp. 1631–1642. URL: https://aclanthology.org/D13-1170/.

Su, K., Cao, L., Zhao, B., Li, N., Wu, D., Han, X., Liu, Y., 2024. Dctvit: Discrete cosine transform meet vision transformers. Neural Networks 172, 106139. URL: https://www.sciencedirect.com/science/article/pii/S0893608024000558, doi:https://doi.org/10.1016/j.neunet.2024.106139.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in neural information processing systems 33, 7537–7547.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA. p. 6000–6010.

Wallace, G., 1992. The jpeg still picture compression standard. IEEE Transactions on Consumer Electronics 38, xviii–xxxiv. doi:10.1109/30.125072.

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J.Y., ZHOU, J., 2024. Timemixer: Decomposable multiscale mixing for time series forecasting, in: The Twelfth International Conference on Learning Representations. URL: https://openreview.net/forum?id=7oLshfEIC2.

Warstadt, A., Singh, A., Bowman, S.R., 2019. Neural network acceptability judgments. Trans. Assoc. Comput. Linguist. 7, 625–641.

Wong, K.C.L., Wang, H., Syeda-Mahmood, T., 2023. HartleyMHA: Self-attention in frequency domain for resolution-robust and parameter-efficient 3D image segmentation, in: Lecture Notes in Computer Science. Springer Nature Switzerland, Cham. Lecture notes in computer science, pp. 364–373.

Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.K., Ren, F., 2020. Learning in the frequency domain, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1737–1746. doi:10.1109/CVPR42600.2020.00181.

Yang, Y., Hou, M., Sun, H., Zhang, T., Weng, F., Luo, J., 2019. Neural network algorithm based on legendre improved extreme learning machine for solving elliptic partial differential equations. Soft Computing 24,

1083–1096. URL: http://dx.doi.org/10.1007/s00500-019-03944-1, doi:10.1007/s00500-019-03944-1.

Yi, K., Zhang, Q., Fan, W., Cao, L., Wang, S., Long, G., Hu, L., He, H., Wen, Q., Xiong, H., 2025. A survey on deep learning based time series analysis with frequency transformation. URL: https://arxiv.org/abs/2302.02173, arXiv:2302.02173.

Yu, J.J., Derpanis, K.G., Brubaker, M.A., 2020. Wavelet flow: Fast training of high resolution normalizing flows, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 6184–6196. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf.

Zhang, H., Ma, J., 2018. Hartley spectral pooling for deep learning. URL: https://arxiv.org/abs/1810.04028, doi:10.48550/ARXIV.1810.04028.

Zheng, S., Wang, H., 2024. Free-merging: Fourier transform for model merging with lightweight experts. URL: https://arxiv.org/abs/2411.16815, arXiv:2411.16815.