2026.04.09 MOAI Lab Seminar

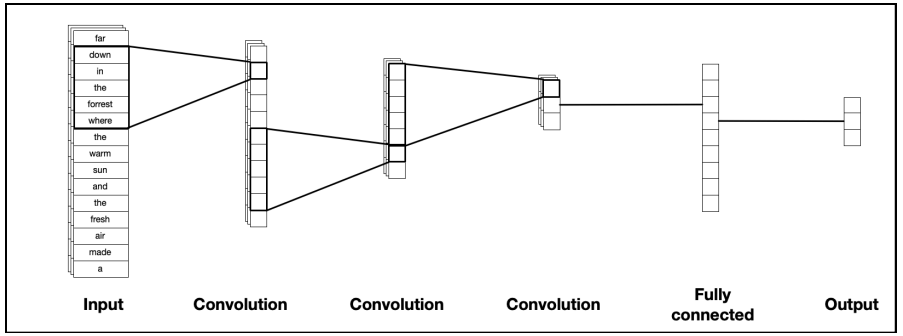
Improving Deep Learning Expressivity and Efficiency via Geometric(Clifford) Algebra

Sanghun Lee

#. Background(The “Why?”)

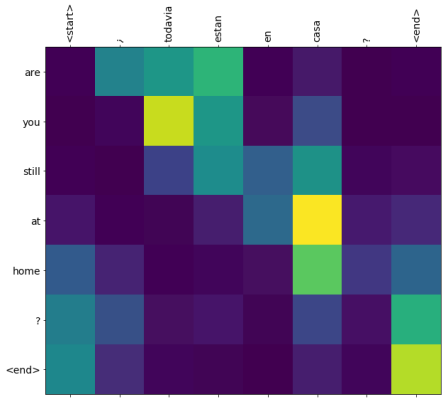
Dot Product

- Core operations underlying modern deep learning
  - e.g., **Attention, CNNs and MLPs**
- Capture feature relationships
  - **Similarity**
  - **Coherence**



Is Dot Product alone enough?

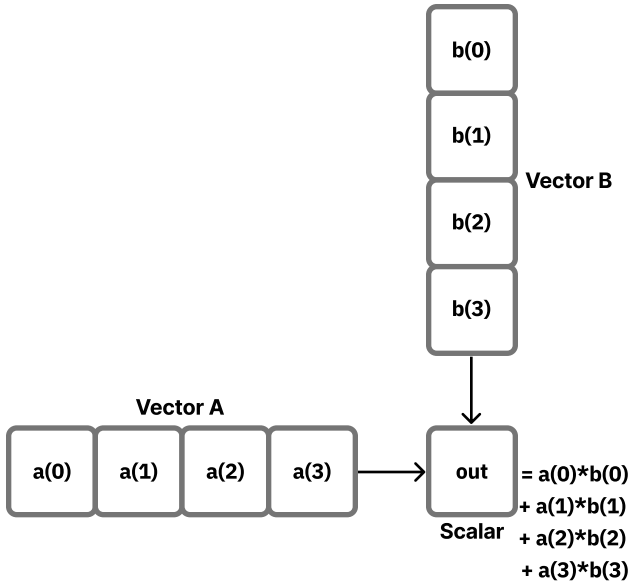
- Limitations of Dot Product
  - Loss of Geometric Structure
    : Vector (Direction O) → Scalar (Direction X)
  - Blind to Orthogonality
    : Failing to capture structural differences or orientation.

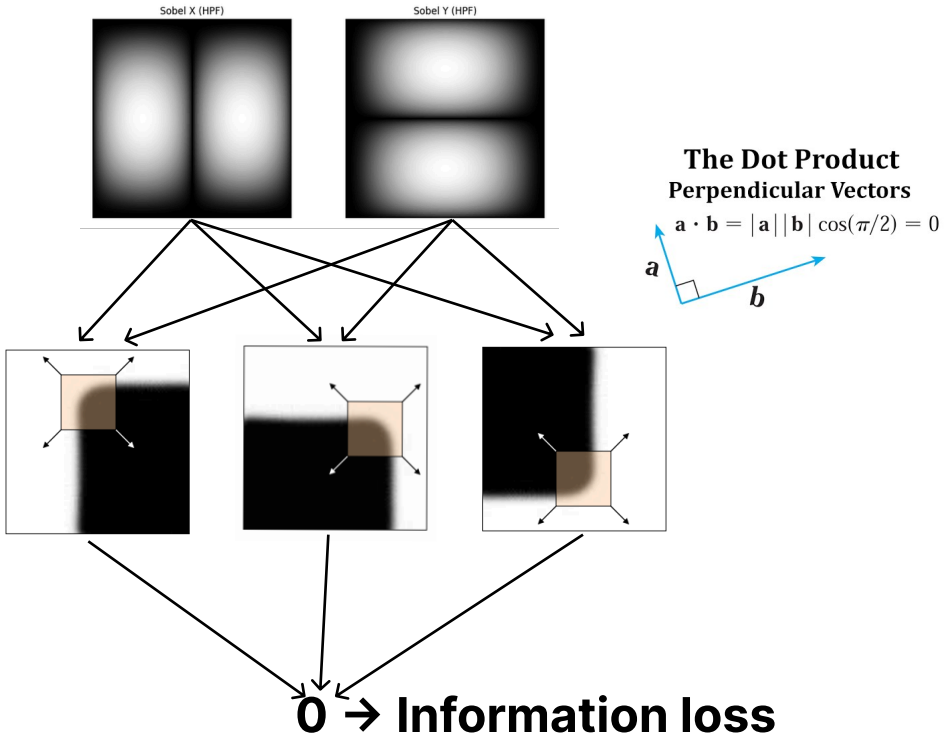


#. Background(The “Why?”)

Dot Product

- Limitations of Dot Product
  - **Loss of Geometric Structure:**
    Vector (Direction O) → Scalar (Direction X)
  - **Blind to Orthogonality:**
    Failing to capture structural differences or orientation.





#. Background(Dot Product → Clifford Product)

Clifford Product

The core operation of Clifford Algebra that unifies the inner product (Dot product, $\cdot$) and exterior product (Wedge product, $\wedge$) into a single algebraic framework.

$$ab = a \cdot b + a \wedge b$$

1. Dot Product

The foundational operation driving modern deep learning mechanisms (e.g., Attention, CNNs, MLPs).

Captures scalar relationships:

- Coherence
- Similarity

2. Wedge Product (New!)

A complementary operation that captures geometric properties entirely missed by the dot product.

Captures structural relationships:

- **Orthogonality(How do features diverge?)**
- **Structure(Edges, Boundaries)**
- **Orientation (Directed area / Handedness)**

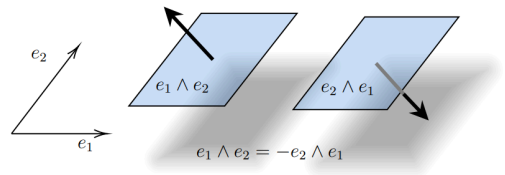
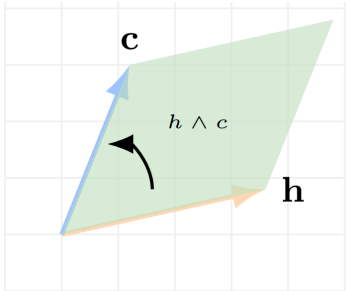


Figure 3: Antisymmetry of bivector exterior (wedge) product.



Geometric Meaning
Orthogonality & Area

#. Background(Clifford Algebra)

Clifford Algebra

Clifford (Geometric) Algebra, $\mathbf{Cl(p,q)}$, is a mathematical framework that **extends vector spaces** to represent and manipulate geometric objects such as **points(scalar), lines(vector), planes(bivector), and volumes(trivector)** within a unified algebraic space.

Multivector

A multivector is the fundamental element of Clifford algebra, formed as a linear combination of basis elements of different grades.

In $\mathbf{Cl(p,q)}$:

- $\mathbf{p}$: number of vector generators with $e_i^2 = +1$
- $\mathbf{q}$: number of vector generators with $e_i^2 = -1$
- Total number of vector generators: $n = p + q$
- Total number of basis elements: 2^n

For $\mathbf{Cl(2,0)}$:

$$e_1^2 = e_2^2 = +1$$

$$\{1, e_1, e_2, e_{12}\}$$

default — p — Combination of vector generators

- 1 Scalar
- 2 Vectors
- 1 Bivector

$$A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_{12} \quad (\because e_{12} = e_1 \wedge e_2 = e_1 e_2)$$

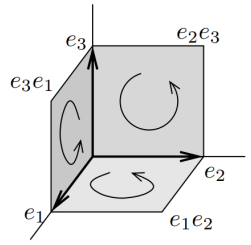
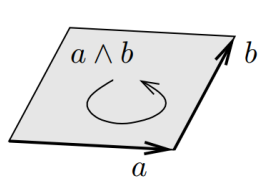


Figure 1.6: The standard unit blades of $\mathbb{R}^3$.



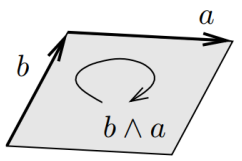


Figure 1.3: Geometric interpretation of $b \wedge a = -a \wedge b$.

#. Background(Clifford Algebra)

Example Interpretation of Cl(2,0) Multivector in Computer Vision

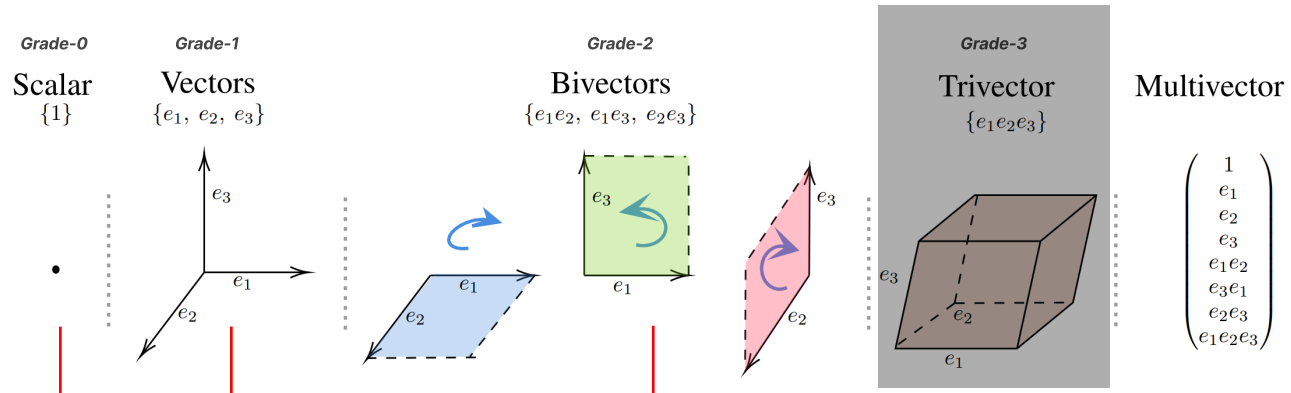
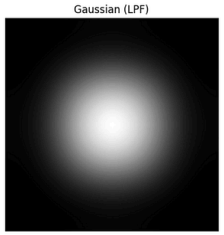


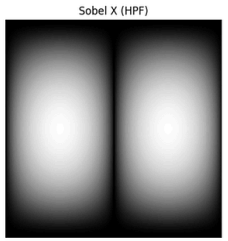
Figure 2: Multivector components of Clifford algebras.

$$A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_{12}$$

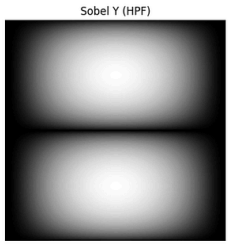
- Pixel energy
- **ex. RGB, brightness**



- Horizontal gradient
- **ex. Edge response along the w-axis**



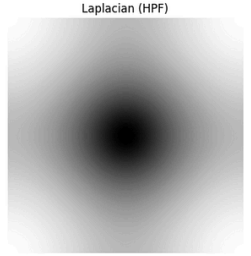
- Vertical gradient
- **ex. Edge response along the h-axis**



- Joint H–W interaction / orientation information
- Complex geometric structure
- **Anti-commutativity**

$$e_2 e_1 = -e_1 e_2$$

- Imaginary-like component($e_{12}^2 = -1$)
- **ex. Corner, rotation, texture pattern**



#. Related Work(Short version - Limitation)

Geometric Algebra Transformer(GATr) (NeurIPS, 2023)

- Limitation:
  b. 3D physics/robotics domain
  → Hard to generalize to Vision/NLP

Clifford Neural Layers for PDE Modeling (ICLR, 2023)

- Limitation:
  a. PDE domain
  → Data comes pre-structured as pressure, velocity, vortivity
  → Hard to generalize to Vision/NLP

Versor: A Geometric Sequence Architecture (2025)

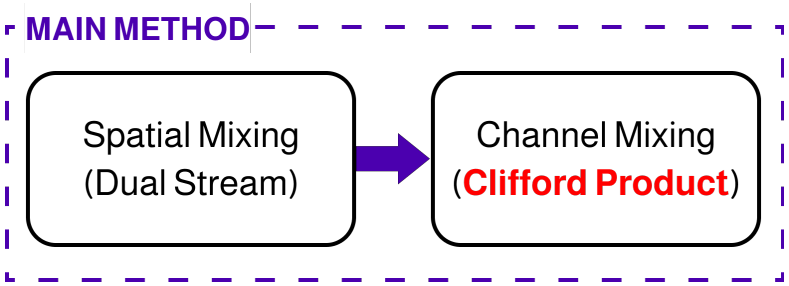
- Limitations:
  a. Claimed multimodal versatility, but **validated only at toy scale**
    - CIFAR-10: 49.63% (3 epochs, raw pixels) / WikiText-103: character-level only
  b. Cl(4,1)'s rich 32-dim structure is **too complex to recover via Linear Projection**
    - Shape: (B, L, D) → (Linear Projection) → (B, L, D//32, 32)

Grade	Name	Number
0	Scalar	1
1	Vectors	5
2	Bivectors	10
3	Trivectors	10
4	Quadvectors	5
5	Pseudoscalar	1
		32

#. Baseline(CliffordNet)

CliffordNet: All You Need is Geometric Algebra (arXiv Preprint, 2026)

- New vision backbone grounded purely in Geometric Algebra
- Dual Stream → **Clifford Product (Dot + Wedge Integration)**



- **No-FFN** Paradigm
- Reduces **Clifford Product complexity from $O(D^2)$ to $O(D)$** via Roll & Shift mechanism
- **Outperforms CNNs and ViT** despite small model scale

Table 2: **Comparison with Efficient Backbones on CIFAR-100. CliffordNet** outperforms strong baselines across all parameter scales.

Model	Params	MLP Ratio	Top-1 Acc
1.5M Scale (Nano)			
ShuffleNetV2 1.0×	1.4M	-	74.60%
CliffordNet-Nano	**1.4M**	**0.0**	**77.82%**
2.5M Scale (Tiny)			
MobileNetV2	2.3M	-	70.90%
ShuffleNetV2 1.5×	2.6M	-	75.95%
ViT-Tiny	2.7M	1.0	65.87%
CliffordNet-Lite	**2.6M**	**0.0**	**79.05%**
>3.0M Scale (Small)			
ResNet-18	11.2M	-	76.75%
ResNet-50	23.7M	-	79.14%
DenseNet-121	7.0M	-	80.20%
CliffordNet-32(3 shifts,cli_mode='full')	**4.8M**	**0.0**	**81.42%**
CliffordNet-64(5 shifts,cli_mode='inner')	**8.6M**	**0.0**	**82.46%**

New SOTA at each parameter scale

#. Baseline(CliffordNet)

MAIN METHOD

Spatial Mixing (Dual Stream) → Channel Mixing (Clifford Product)

1. Spatial Mixing

Input (B, H, W, C)



sunflower

DWConv 3×3 ×2

C: Global Context (B, D, H, W)
- local context aggregation

Conv1d

H: State (B, D, H, W)
- high-frequency details

$$\text{Clifford Product} = \underbrace{\text{Dot Product}}_{\text{Coherence}} + \underbrace{\text{Wedge Product}}_{\text{Structure}}$$

2. Channel Mixing (Clifford Product)

Conventional Channel Mixing (e.g., FFN, MLP):
- **Single input**

CliffordNet Channel Mixing:
- **Dual inputs (H: State, C: Context)**

#. Baseline(CliffordNet)

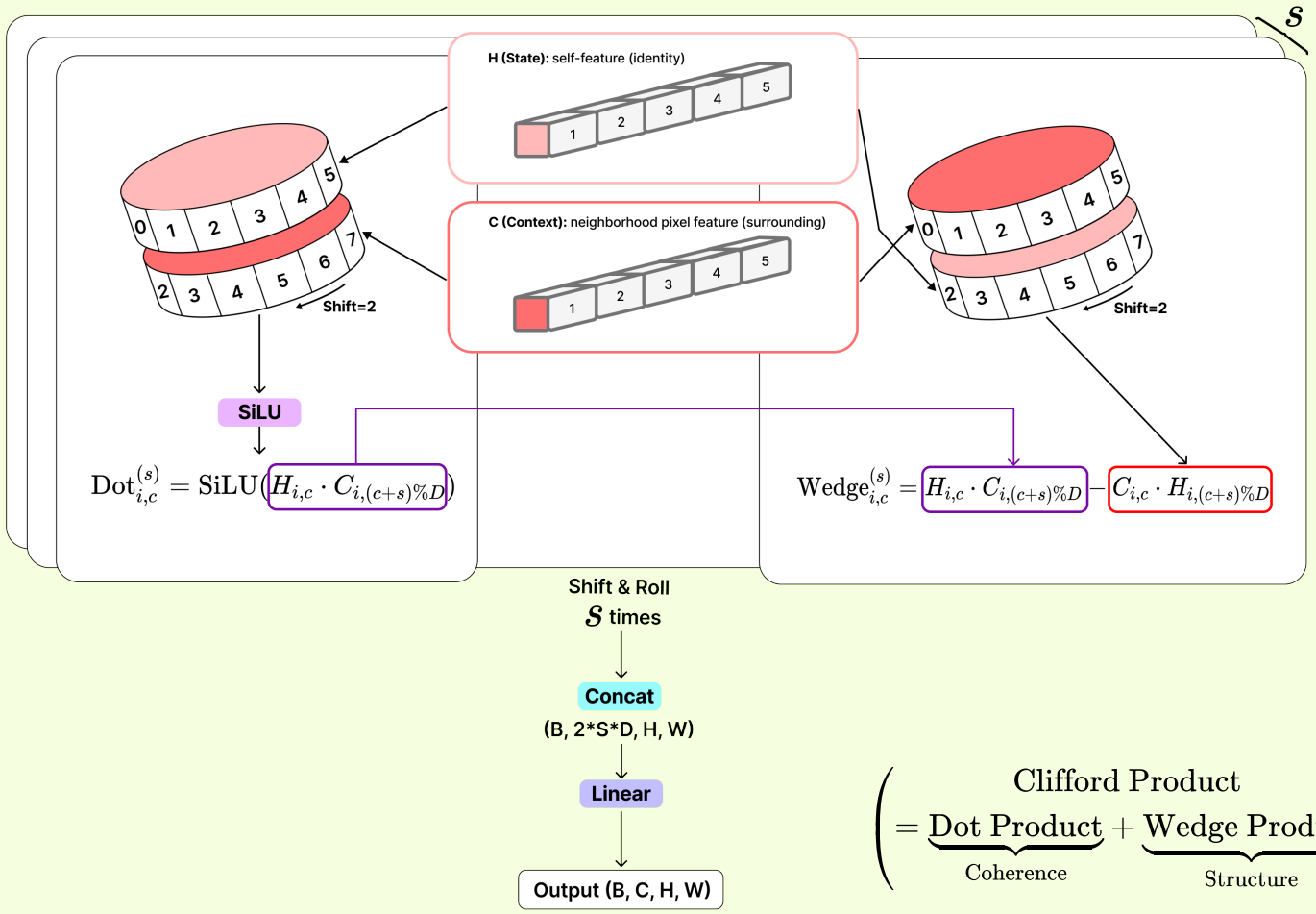
MAIN METHOD

Spatial Mixing (Dual Stream) → Channel Mixing (**Clifford Product**)

- **No-FFN** Paradigm
- Reduces **Clifford Product complexity from $O(D^2)$ to $O(D)$** via Roll & Shift mechanism

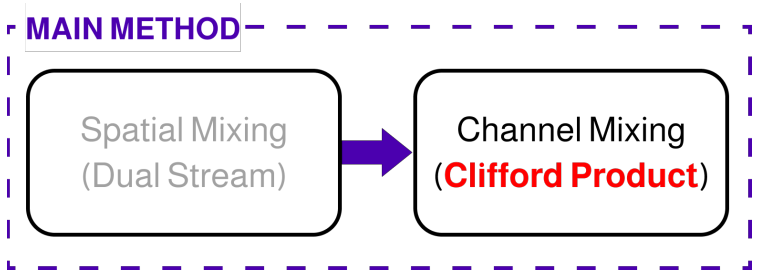
2. Channel Mixing (Clifford Product)

For Example. Visualized for a single pixel
(In practice, all pixels undergo Clifford Product simultaneously at the same spatial position)



$$\left(\begin{array}{c}\text{Clifford Product} \\ = \underbrace{\text{Dot Product}}_{\text{Coherence}} + \underbrace{\text{Wedge Product}}_{\text{Structure}}\end{array}\right)$$

#. Baseline(CliffordNet)



Dense Clifford Product ($D \times D = 4 \times 4$)

H\C	c=0	c=1	c=2	c=3
h=0	✓	✓	✓	✓
h=1	✓	✓	✓	✓
h=2	✓	✓	✓	✓
h=3	✓	✓	✓	✓

$\therefore D \times D = 16$

Roll & Shift ($D = 4, S = 2$)

Shift=1:

H\C	c=0	c=1	c=2	c=3
h=0		✓		
h=1			✓	
h=2				✓
h=3	✓			

Shift=2:

H\C	c=0	c=1	c=2	c=3
h=0			✓	
h=1				✓
h=2	✓			
h=3		✓		

$\therefore S \times D = 8$

#. Baseline(CliffordNet-Limitations)

CliffordNet

Output	Interpretation
Dot (scalar)	Coherence (how similar?)
Wedge (scalar)	Rotation / Structure

Limitations

- Treats channels as 1D vectors → **No Multivector**
  - Approximation of Clifford Product, not exact
- Dot & Wedge computed as **separate branches**
  - 2× intermediate tensors per shift $(\mathrm{B}, 2{*}\mathrm{S}{*}\mathrm{D}, \mathrm{H}, \mathrm{W})$
- **Heavy hyperparameter dependency** on Roll & Shift
  - D=128 → **64 possible** shift options
  - **shift=0 → Wedge=0**, completely unused

Multivector-based Clifford Product (Original)

$$\mathbf{H}_i = h_0 + h_1 e_1 + h_2 e_2 + h_{12} e_{12}$$
$$\mathbf{C}_i = c_0 + c_1 e_1 + c_2 e_2 + c_{12} e_{12}$$

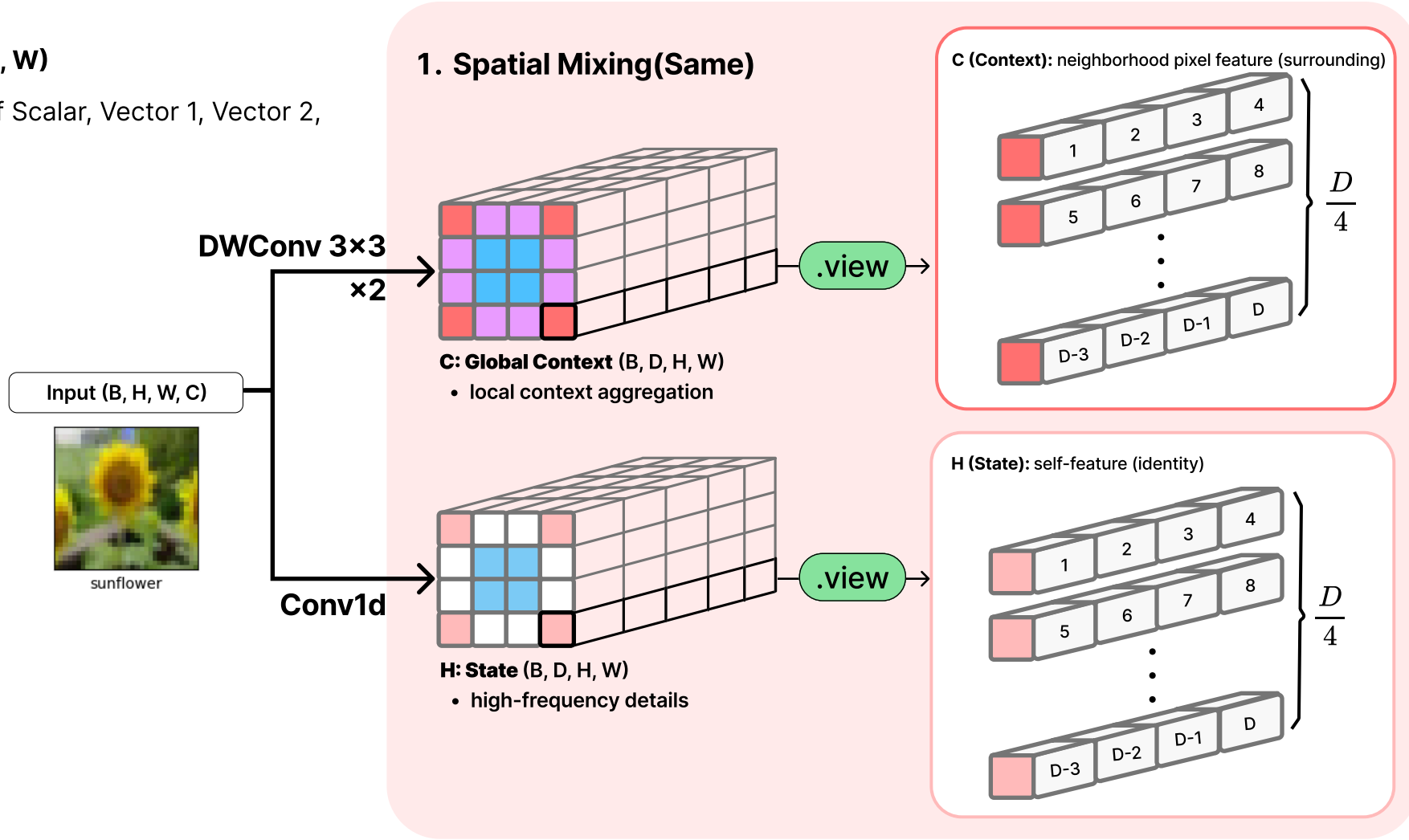
Output	Formula	Interpretation
Scalar	$h_0 c_0 + h_1 c_1 + h_2 c_2 - h_{12} c_{12}$	Coherence (how similar?)
Vector 1	$(h_0 c_1 + h_1 c_0 - h_2 c_{12} + h_{12} c_2) e_1$	x-directional variation
Vector 2	$(h_0 c_2 + h_1 c_{12} + h_2 c_0 - h_{12} c_1) e_2$	y-directional variation
Bivector	$(h_0 c_{12} + h_1 c_2 - h_2 c_1 + h_{12} c_0) e_{12}$	Rotation / Structure

- **More Representation(Richer Interpretation), Efficiency(Algebraic Closure)**

#. Method(CliffordNet based on $\mathrm{Cl}(2,0)$)

Step 1. Cl(2,0) Multivectorization

- Projects D-dim embedding into 4 components (Scalar, Vector 1, Vector 2, Bivector)
- **(B, D, H, W) →(Convolution)→ (B, D//4, 4, H, W)**
- uses CNNs to learn D//4 group-wise roles of Scalar, Vector 1, Vector 2, Bivector suited for 2D images



#. Method(CliffordNet based on $\mathbf{Cl(2,0)}$)

Step 2. Complex Isomorphism - 1

In $\mathbf{Cl(2,0)}$, each element is:

$\mathbf{e_0}$ (Scalar, Grade-0), $\mathbf{e_1}$ (Vector, Grade-1),

$\mathbf{e_2}$ (Vector, Grade-1), $\mathbf{e_{12}}$ (Bivector, Grade-2)

This can be **mathematically decomposed into complex form**, where the Bivector e_{12} in Geometric Algebra is by definition the geometric product of e_1 and e_2: $e_{12}=e_1e_2$

There are **2 fundamental rules in Geometric Algebra:**

1. Squaring rule:

A basis vector squared equals 1

$$e_1e_1 = 1,\ e_2e_2 = 1$$

2. Anti-commutation rule:

Reverse the order of the different bases, the sign becomes negative.

$$e_2e_1 = -e_1e_2$$

- Therefore, **e_{12} acts as the imaginary unit(I) of complex numbers**

$$e_{12}^2 = (e_1e_2)(e_1e_2) = -e_1(e_2e_1)e_2 = -e_1e_1e_2e_2 = -(1)(1) = -1$$

#. Method(CliffordNet based on $\mathbf{Cl(2,0)}$)

Step 2. Complex Isomorphism - 2

Given two $\mathbf{Cl(2,0)}$ Multivectors

$$\mathbf{H}_i = h_0 + h_1 e_1 + h_2 e_2 + h_{12} e_{12}$$
$$\mathbf{C}_i = c_0 + c_1 e_1 + c_2 e_2 + c_{12} e_{12}$$

Complex Transform →

$\mathbf{Cl(2,0)}$ Multivector can be decomposed into **2 complex elements**

α : Even subalgebra part(Energy/Rotation)

$$\alpha_H = h_0 + h_{12} e_{12} = h_0 + h_{12} I$$

β : Odd grade vector part(Edge)

$$\begin{aligned} & h_1 e_1 + h_2 e_2 \\ & \rightarrow h_1 e_1 - h_2(-e_2) \\ & = h_1 e_1 - h_2 e_1 e_2 e_1 \\ & = (h_1 - h_2 e_{12}) e_1 \\ & = \beta_H e_1 \quad (\beta_H := h_1 - h_2 e_{12}) \end{aligned}$$

Clifford Product Output (Real-valued form):

$$\begin{aligned} \mathbf{H}_i \mathbf{C}_i = & h_0 c_0 + h_1 c_1 + h_2 c_2 - h_{12} c_{12} \\ & + (h_0 c_1 + h_1 c_0 - h_2 c_{12} + h_{12} c_2) e_1 \\ & + (h_0 c_2 + h_1 c_{12} + h_2 c_0 - h_{12} c_1) e_2 \\ & + (h_0 c_{12} + h_1 c_2 - h_2 c_1 + h_{12} c_0) e_{12} \end{aligned}$$

Complex form of Cl(2,0) Multivector:

$$\mathbf{H}_i = \alpha_H + \beta_H e_1$$
$$\mathbf{C}_i = \alpha_C + \beta_C e_1$$

Clifford Product Output (Complex Form):

$$\mathbf{H}_i \mathbf{C}_i = (\alpha_H + \beta_H e_1)(\alpha_C + \beta_C e_1)$$

#. Method(CliffordNet based on $\mathrm{Cl}(2,0)$)

Step 2. Complex Isomorphism - 3

Clifford Product(Complex Form):

$$\begin{aligned}\mathbf{H}_i\mathbf{C}_i &= (\alpha_H + \beta_H e_1)(\alpha_C + \beta_C e_1) \\ &= \alpha_H\alpha_C + \alpha_H\beta_C e_1 + \beta_H e_1\alpha_C + \beta_H e_1\beta_C e_1\end{aligned}$$

- $\beta_H(e_1\alpha_B) \to \beta_H\tilde{\alpha_C}e_1$ (where $\tilde{}$ denotes the conjugate)
- $\beta_H(e_1\beta_B)e_1 \to \beta_H\tilde{\beta_C}e_1e_1 \to \beta_H\tilde{\beta_C}$

$$= (\alpha_H\alpha_C + \beta_H\tilde{\beta_C}) + (\alpha_H\beta_C + \beta_H\tilde{\alpha_C})e_1$$

$$\begin{aligned}\alpha_{new} &= \alpha_H\alpha_C + \beta_H\tilde{\beta_C} \\ &= \underbrace{[(h_0c_0 - h_{12}c_{12}) + (h_1c_1 + h_2c_2)]}_{\text{New Scalar } (o_0)} + \underbrace{[(h_0c_{12} + h_{12}c_0) + (h_1c_2 - h_2c_1)]I}_{\text{New Bivector } (o_{12})\cdot e_{12}}\end{aligned}$$

SiLU

$$\begin{aligned}\beta_{new} &= \alpha_H\beta_C + \beta_H\tilde{\alpha_C} \\ &= \underbrace{[(h_0c_1 + h_{12}c_2) + (h_1c_0 - h_2c_{12})]}_{\text{New Vector 1 (Real, } o_1)} + \underbrace{[(h_{12}c_1 - h_0c_2) - (h_1c_{12} + h_2c_0)]I}_{\text{New Vector 2 (Imag, } o_2)} \\ &= \underbrace{[(h_0c_1 + h_{12}c_2) + (h_1c_0 - h_2c_{12})]}_{\text{New Vector 1 (Real, } o_1)} - \underbrace{[(-h_{12}c_1 + h_0c_2) + (h_1c_{12} + h_2c_0)]I}_{\text{New Vector 2 (Imag, } o_2)}\end{aligned}$$

The final Clifford Product output is:

$$O_i = \alpha_{new} + \beta_{new}e_1$$

Rearranging into $[o_0, o_1, o_2, o_{12}]$ form:

Identical to the Clifford Product (Real-valued form)

- Scalar output is essentially a Dot Product across dimensions
  → apply SiLU for enhanced expressivity

- **Why compute via Complex form**

	1. Cl(2,0) Native	2. Cl(2,0) Complex
FLOPs	28 (16 mult + 12 add/sub)	6 (4 complex mult + 2 add)
Sign process ($e_{12}^2 = -1$)	Hard Coding	Handled automatically by Complex ALU
Backprop efficiency	16 intermediate tensors	No intermediate tensors
Custom kernel required	Yes	PyTorch Native

#. Method(CliffordNet based on $\mathrm{Cl}(2,0)$)

Step 2. Complex Isomorphism - 4

Input (B, H, W, C)



sunflower

DWConv 3×3 ×2

Conv1d

1. Spatial Mixing(Same)

C: Global Context (B, D, H, W)
- local context aggregation

H: State (B, D, H, W)
- high-frequency details

.view

.view

2. Channel Mixing

H (State): self-feature (identity)

C (Context): neighborhood pixel feature (surrounding)

$\mathbf{H}_i\mathbf{C}_i = (\alpha_H + \beta_H e_1)(\alpha_C + \beta_C e_1)$

$O_i = \alpha_{new} + \beta_{new} e_1$

$[\mathrm{SiLU}(o_0), o_1, o_2, o_{12}]$: (B, D//4, 4, H, W)

Complex to Float
: (B, D, H, W)

intermediate tensors X

Shift & Roll
S times

Concat
(B, S*D, H, W)

Linear

Output (B, C, H, W)

Forward pass memory reduced by 2×

#. Results(CliffordNet based on $\mathrm{Cl}(2,0)$)

▶ Limitations & Improvements of CliffordNet

1. No proper Multivector structure (treats channels as plain 1D vectors)
   - Does not follow Clifford Algebra
   - **Merely an approximation of the exact Clifford Product**

→

(Improvement)

- **Algebraically exact Clifford Product**

  with proper Multivector structure
- Geometric representation ↑

2. Dot & Wedge computed as separate branches
   - **2× intermediate tensors per shift**

→

(Improvement)

- Dot & Wedge unified into a **single Algebraic Operation**
- 1× intermediate tensor per shift

3. **Heavy hyperparameter dependency on Roll & Shift**
   - ex. D=128 → **64 possible shift options**
   - shift=0 → Wedge=0, completely unused

→

(Improvement)

- **Group-wise Roll & Shift** reduces hyperparameter dependency
- ex. D=128 → **17(+zero) possible shift options**
- shift=0 → full Clifford Product result still available

#. Results(CliffordNet based on $\mathrm{Cl}(2,0)$)

▶ Performance

Training from Scratch results under identical conditions(CliffordNet vs. Ours)

Model	Params	MLP Ratio	Top-1 Acc
1.5M Scale (Nano)			
ShuffleNetV2 1.0×	1.4M	-	74.60%
CliffordNet-Nano†	1.4M	0.0	77.82%
Ours-Nano(16-2 shifts)	**1.4M**	**0.0**	**78.37%**
2.5M Scale (Tiny)			
MobileNetV2	2.3M	-	70.90%
ShuffleNetV2 1.5×	2.6M	-	75.95%
ViT-Tiny	2.7M	1.0	65.87%
CliffordNet-Lite†	2.6M	0.0	79.05%
Ours-Lite(18-5 shifts)	**2.4M**	**0.0**	**79.92%**
Ours-Lite(25-3 shifts)	**2.6M**	**0.0**	**80.57%**
>3.0M Scale (Small)			
ResNet-18	11.2M	-	76.75%
ResNet-50	23.7M	-	79.14%
DenseNet-121	7.0M	-	80.20%
CliffordNet-32†	4.8M	0.0	81.42%
CliffordNet-64 (Dot Product only)†	8.6M	0.0	82.46%
Ours(32-5 shifts)	**4.3M**	**0.0**	**81.45%**
Ours(48-2 shifts)	**4.1M**	**0.0**	**81.79%**
Ours(52-3 shifts)	**5.3M**	**0.0**	**82.46%**

† Baseline (Previous SOTA at each scale).

9.87x less

▼ 38.4%

SOTA

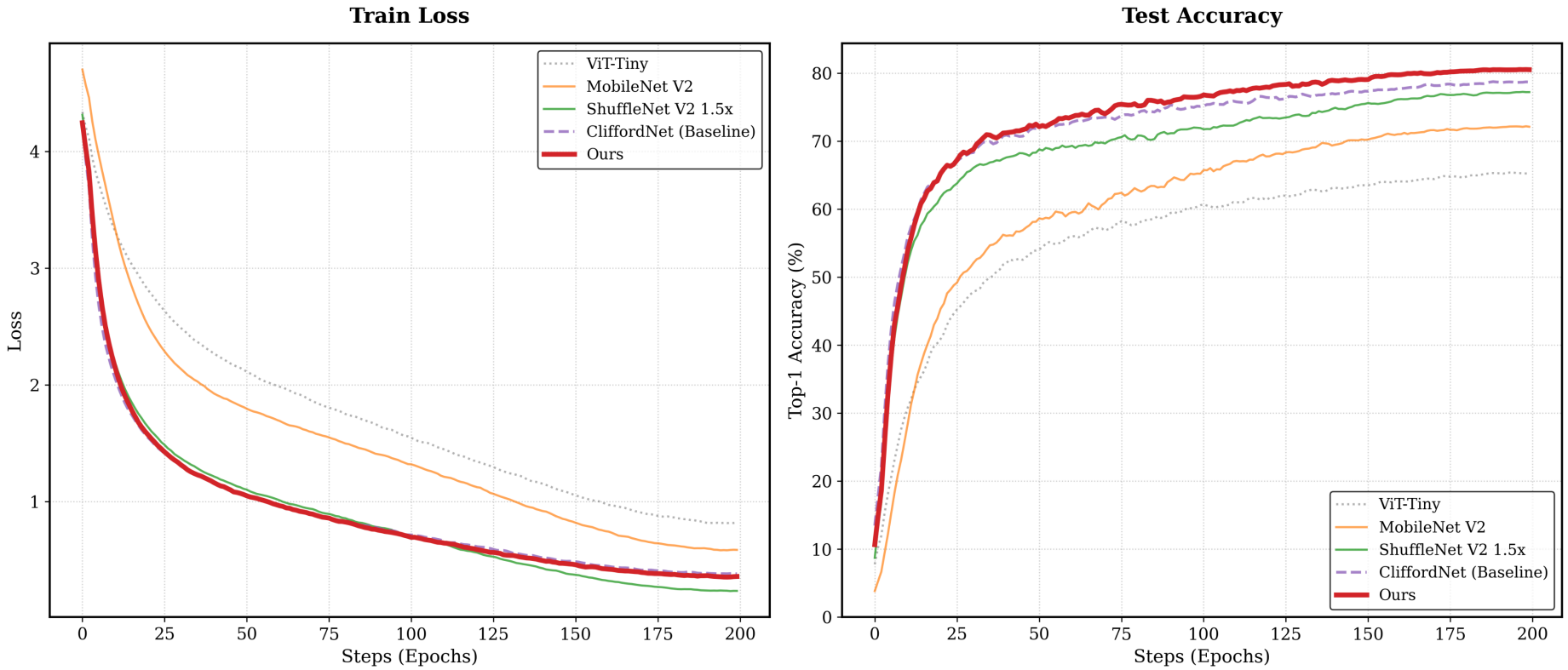
[Experimental Setup (Training from Scratch)]

- **Dataset:** CIFAR-100 (Batch Size: 128)
- **Optimization:** AdamW (LR: 1e-3, Weight Decay: 0.1), Cosine Annealing
- **Augmentation:** AutoAugment, Random Erasing (p=0.25), Crop & Flip
- **Environment:** Fixed Seed (Fully Reproducible)

- **[SOTA]** New SOTA across all parameter scales on CIFAR-100, outperforming baseline CliffordNet at every scale.
- **[Parameter Efficiency]** Via **Cl(2,0) Multivector-based Clifford Product**, achieved **38.4% parameter reduction over SOTA model (82.46%).** Outperforms ResNet-50 with up to **9.87× fewer parameters.**

#. Results(CliffordNet based on $\mathrm{Cl}(2,0)$)

Geometric Inductive Bias Comparison



ShuffleNet V2 1.5x (previous 2.6M scale SOTA) **shows a strong overfitting tendency** after epoch 100, **pushing Train Loss to its minimum.** In contrast, both CliffordNet and **Ours maintain relatively higher Train Loss, yet achieve significantly higher Test Accuracy than ShuffleNet.**

This strongly suggests that the **Geometric Inductive Bias** of the proposed model **structurally suppresses memorization** of local noise, instead learning the essential structure of the data → **leading to superior generalization performance.**

#. Limitations & Future Work

Limitations

1. Scale & Pipeline

- **[Limitation]** Hardware constraints limit validation to low-resolution datasets (CIFAR-100). **Experiments conducted in a controlled setting with modern training techniques (e.g., Warm-up)** fully excluded for pure architecture comparison.
- **[Future Work]** Generalization/scalability validation via **large-scale, high-resolution datasets (ImageNet)**, and potential performance gains through **modern Training Recipe.**

2. Architectural & Algebraic Extension

- **[Limitation]** Unidirectional Shift introduces geometric asymmetry, and **the current design is limited to 2D Vision (Cl(2,0)) Channel Mixing.**
- **[Future Work] NLP Extension:** The unidirectional operation can be leveraged as an Auto-Regressive property for language model Token Mixing.
  - Higher-dim Extension: Via Cl(p,q)-based algebra, extensible to **higher-dimensional geometric processing such as 3D/Video.**

3. System Bottleneck & Math Optimization

- **[Limitation]** Tensor casting overhead and **Memory-Bound (I/O) bottleneck inherent to torch.roll** remain unresolved.
- **[Future Work]** Engineering approach: extreme speed optimization via Triton/CUDA-based Custom Kernel.
  - Mathematical approach: Introducing **Clifford Fourier Transform (CFT)** closely tied to Cl(2,0). **Eliminates inefficient physical Roll & Shift,** enabling **Global Channel Mixing in the frequency domain** and novel Token/Spatial Mixing structures.