Symmetry-Based Structured Matrices for Efficient Approximately Equivariant Networks

Ashwin Samudre¹, Mircea Petrache², Brian D. Nord³, Shubhendu Trivedi⁴

¹SFU ²PUC Chile & CENIA ³FNAL & UChicago ⁴Independent

AISTATS 2025

► Equivariance in Neural Networks: Output transforms predictably under input symmetry (e.g., rotation of images).

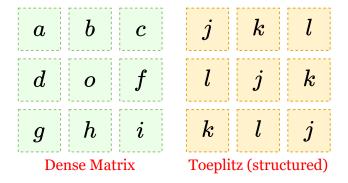
- ► Equivariance in Neural Networks: Output transforms predictably under input symmetry (e.g., rotation of images).
- ► **Challenge**: Strict equivariance can fail with imperfect real-world symmetries.

- ► Equivariance in Neural Networks: Output transforms predictably under input symmetry (e.g., rotation of images).
- Challenge: Strict equivariance can fail with imperfect real-world symmetries.
- Solution: Approximately equivariant networks balance flexibility and symmetry.

- Equivariance in Neural Networks: Output transforms predictably under input symmetry (e.g., rotation of images).
- Challenge: Strict equivariance can fail with imperfect real-world symmetries.
- Solution: Approximately equivariant networks balance flexibility and symmetry.
- Our Approach: Use Group Matrices (GMs) to generalize CNNs efficiently.

► **Structured Matrices**: Matrices with patterns (e.g., Toeplitz, Hankel, circulant) that reduce parameters.

Dense Vs. Structured Matrix



- ► **Structured Matrices**: Matrices with patterns (e.g., Toeplitz, Hankel, circulant) that reduce parameters.
- ▶ **Displacement Rank (DR)**: Measures structure via $\nabla_{A,B}(M) = AM MB$. DR = rank of $\nabla_{A,B}(M)$.

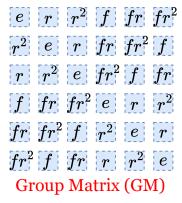
- ► **Structured Matrices**: Matrices with patterns (e.g., Toeplitz, Hankel, circulant) that reduce parameters.
- ▶ **Displacement Rank (DR)**: Measures structure via $\nabla_{A,B}(M) = AM MB$. DR = rank of $\nabla_{A,B}(M)$.
- Low DR preserves structure under operations.

- ► **Structured Matrices**: Matrices with patterns (e.g., Toeplitz, Hankel, circulant) that reduce parameters.
- ▶ **Displacement Rank (DR)**: Measures structure via $\nabla_{A,B}(M) = AM MB$. DR = rank of $\nabla_{A,B}(M)$.
- Low DR preserves structure under operations.
- Equivariance: Circulant matrices (used in CNNs) are GMs for cyclic groups, enabling shift equivariance.

Group Matrices

Definition: Matrices encoding group symmetries, $M_{gh} = M_{g'h'}$ if gh = g'h'.

Example: Dihedral group matrix (D_3)



Group Matrices

- **Definition**: Matrices encoding group symmetries, $M_{gh} = M_{g'h'}$ if gh = g'h'.
- ▶ **Group Diagonals**: Basis matrices B_g , sparse, with $(B_g)_{h,h'} = \delta(h = gh')$.

Group Matrices

- **Definition**: Matrices encoding group symmetries, $M_{gh} = M_{g'h'}$ if gh = g'h'.
- ▶ **Group Diagonals**: Basis matrices B_g , sparse, with $(B_g)_{h,h'} = \delta(h = gh')$.
- **Example**: For cyclic group C_3 , B_g shifts entries cyclically.

Example: Group Diagonal for C_3

For $C_3 = \{e, a, a^2\}$ with $a^3 = e$, the group diagonal B_a is:

$$B_{a} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

where rows and columns are labeled by e, a, a^2 .

Standard Convolution: $\phi \star \psi(x) = \sum_{g \in G} \phi(g) \psi(g^{-1}x)$.

- **Standard Convolution**: $\phi \star \psi(x) = \sum_{g \in G} \phi(g) \psi(g^{-1}x)$.
- ▶ **GM Formulation**: Conv_{ψ} $\vec{\phi} = \sum_{g \in G} \phi(g) B_g \vec{\psi}$.

- ► Standard Convolution: $\phi \star \psi(x) = \sum_{g \in G} \phi(g) \psi(g^{-1}x)$.
- **► GM Formulation**: Conv $_{\psi}\vec{\phi} = \sum_{g \in G} \phi(g) B_g \vec{\psi}$.
- ▶ **Efficient kernels:** Restrict ϕ to a radius k neighborhood in G's Cayley graph, similar to $k \times k$ CNN kernels.

- ► Standard Convolution: $\phi \star \psi(x) = \sum_{g \in G} \phi(g) \psi(g^{-1}x)$.
- ▶ **GM Formulation**: Conv $_{\psi}\vec{\phi} = \sum_{g \in G} \phi(g)B_g\vec{\psi}$.
- ▶ **Efficient kernels:** Restrict ϕ to a radius k neighborhood in G's Cayley graph, similar to $k \times k$ CNN kernels.
- ▶ **Complex groups**: Compute GMs for direct/semi-direct products via Kronecker products, e.g., $B_{(g,h)}^{G \times H} = B_g^G \otimes B_h^H$.

▶ **Strides**: Restrict output to a subgroup $H \subseteq G$.

- ▶ **Strides**: Restrict output to a subgroup $H \subseteq G$.
- ▶ **Pooling**: Pool_H[□](ψ)(h) = \square { ψ (g') : $g' \in P_h$ }, over cosets.

- ▶ **Strides**: Restrict output to a subgroup $H \subseteq G$.
- ▶ **Pooling**: Pool $_H^\square(\psi)(h) = \square\{\psi(g') : g' \in P_h\}$, over cosets.
- ▶ Closure: M^T , MM', $M \otimes N$ remain GMs.

- ▶ **Strides**: Restrict output to a subgroup $H \subseteq G$.
- ▶ **Pooling**: Pool_H(ψ)(h) = \square { ψ (g') : $g' \in P_h$ }, over cosets.
- ▶ Closure: M^T , MM', $M \otimes N$ remain GMs.
- ► Extensions: Applies to homogeneous spaces and infinite groups.

► **Goal**: Relax strict equivariance for flexibility.

- ► Goal: Relax strict equivariance for flexibility.
- ▶ **Method**: $M = \sum_{g \in G} \operatorname{diag}(F_g)B_g$, with learnable F_g .

- ► **Goal**: Relax strict equivariance for flexibility.
- ▶ **Method**: $M = \sum_{g \in G} \text{diag}(F_g)B_g$, with learnable F_g .
- ▶ **Displacement Operator**: D(M) = F(M) PF(M), where P is a permutation. D(M) = 0 for exact GMs.

- ► **Goal**: Relax strict equivariance for flexibility.
- ▶ **Method**: $M = \sum_{g \in G} \text{diag}(F_g)B_g$, with learnable F_g .
- ▶ **Displacement Operator**: D(M) = F(M) PF(M), where P is a permutation. D(M) = 0 for exact GMs.
- ▶ Low Displacement Rank: $rank(D(M)) \le r$ reduces parameters while controlling error.

Experimental Setup

► Tasks: Dynamics prediction (Plumes, JetFlow), image classification (MNIST, CIFAR-10).

Experimental Setup

- ► **Tasks**: Dynamics prediction (Plumes, JetFlow), image classification (MNIST, CIFAR-10).
- Baselines: CNN, E2-CNN, RPP, LIFT, structured matrix methods.

Experimental Setup

- ► Tasks: Dynamics prediction (Plumes, JetFlow), image classification (MNIST, CIFAR-10).
- Baselines: CNN, E2-CNN, RPP, LIFT, structured matrix methods.
- Metrics: RMSE, accuracy, parameter count.

Results: JetFlow

Task	GM-CNN RMSE \downarrow	Best Baseline RMSE \downarrow	Param Ratio
Translation (F) Rotation (F) Scaling (F)	$\begin{array}{c} 0.15\pm0.01 \\ 0.17\pm0.01 \\ 0.15\pm0.01 \end{array}$	$0.15 \pm 0.00 \; ext{(Lift)} \ 0.16 \pm 0.01 \; ext{(RSteer)} \ 0.14 \pm 0.01 \; ext{(RSteer)}$	1:75 1:32 1:255

Note : F = Future.

Results: Plumes

Task	GM-CNN RMSE↓	Best Baseline RMSE \downarrow	Param Ratio
Translation (F) Rotation (F) Scaling (F)	$egin{array}{l} 0.72 \pm 0.01 \ 0.92 \pm 0.01 \ 0.92 \pm 0.01 \end{array}$	$0.72 \pm 0.01 \; (RGroup) \ 0.80 \pm 0.01 \; (RSteer) \ 0.82 \pm 0.01 \; (RSteer)$	1:100 1:50 1:350

Note : F = Future.

Results: Image Classification

Dataset	GM-CNN Acc (%) \uparrow (Params) \downarrow	Best Baseline Acc (%) (Params)
M-bg-rot	55.29 (5.9k)	45.81 (14k) [LDR]
M-noise	90.20 (5.9k)	82.55 (25k) [GM(n=3,E)]
CIFAR-10	58.31 (14k)	45.33 (18k) [LDR]
SmallNORB	85.22 (25k)	83.23 (14k) [LDR]
Rot-MNIST	84.50 (15k)	83.78 (49k) [GM(n=3)]
Rect	99.89 (5.9k)	99.60 (12k) [GM(n=3)]

Note: n = neighborhood radius, E = error addition.

Conclusion

► **Contributions**: GM-CNNs generalize CNN operations using GMs, enabling efficient approximate equivariance.

Conclusion

- ► **Contributions**: GM-CNNs generalize CNN operations using GMs, enabling efficient approximate equivariance.
- ► Impact: Achieves high accuracy with 1-2 orders fewer parameters, ideal for resource-limited settings.

Conclusion

- Contributions: GM-CNNs generalize CNN operations using GMs, enabling efficient approximate equivariance.
- ► Impact: Achieves high accuracy with 1-2 orders fewer parameters, ideal for resource-limited settings.
- ► **Future Work**: Extend to continuous groups, incorporate steerability, and improve scalability.

Thank You

Questions?

