Composing Language with Lie Groups: Bounded Operators for Compositional Generalization

Anonymous ACL submission

Abstract

We investigate whether algebraic structure in the composition mechanism can improve length generalization for sequential reasoning. We represent each sentence as an $\mathrm{SO}(n)$ rotation matrix via the Lie algebra: a sentence encoder produces an embedding, projected to skew-symmetric parameters and exponentiated to yield a rotation. Unlike fixed orthogonal recurrent networks, each sentence maps to a distinct input-dependent rotation, making composition non-commutative by construction. The $\mathrm{SO}(n)$ constraint guarantees norm preservation and unit condition number ($\kappa = 1.0$) regardless of chain length. On kinship chain reasoning, combining $\mathrm{SO}(n)$ composition with a lightweight Transformer over the state trajectory achieves 57.5% accuracy—best among all models—and eliminates the long-chain collapse observed with endpoint-only readout ($+22$pp at $4\times$ training chain length). Mechanistic analysis confirms that $\mathrm{SO}(n)$ maintains $\kappa = 1.0$ while unconstrained matrices degrade to $\kappa = 308$, directly explaining the advantage. On synthetic benchmarks, $\mathrm{SO}(n)$ operators outperform all baselines on order sensitivity ($+8.4$pp paired accuracy, $p = 2.8 \times 10^{-5}$) and length generalization ($+14.2$pp at $2\times$ training length).

1 Introduction

Understanding language requires composing meaning sequentially: "Alice opened the box" followed by "Bob closed the box" leaves the box closed, but reversing the sentence order leaves it open. This order sensitivity is fundamental to discourse understanding, yet current neural approaches—LSTMs (Hochreiter and Schmidhuber, 1997) and Transformers (Vaswani et al., 2017)—learn it implicitly through architecture rather than enforcing it through representation. As a result, these models often fail to generalize compositional reasoning to longer sequences than seen during training (Sinha et al., 2019; Dziri et al., 2023).

Why do sequential models struggle with length generalization? One reason is numerical: neither LSTM hidden states nor unconstrained matrix compositions have norm bounds, so information can be amplified or suppressed unpredictably over many steps. We observe condition numbers reaching $\kappa = 308$ after just 20 compositions (Section 4), meaning some directions are amplified $308\times$ relative to others and information in suppressed directions is effectively lost. A second reason is architectural: mean pooling discards order entirely, while attention-based models can overfit to the positional patterns seen during training.

These observations suggest a design principle: *use structured composition to build representations, and attention to read them out*. The composition mechanism should guarantee bounded conditioning regardless of chain length, while a flexible readout should extract information without being tied to a specific sequence length.

We instantiate this principle using $\mathrm{SO}(n)$, the rotation group. Each sentence maps to a rotation matrix via the Lie algebra $\mathfrak{so}(n)$: a sentence encoder produces an embedding, which is projected to skew-symmetric parameters and exponentiated to yield a valid rotation. Composition is matrix multiplication, producing a discourse state on the unit hypersphere S^{n-1}. The $\mathrm{SO}(n)$ constraint ensures non-commutativity by construction ($R_a R_b \neq R_b R_a$ for $n \geq 3$, so the model cannot ignore sentence order), bounded conditioning ($\kappa = 1$ regardless of chain length, preventing the numerical degradation described above), and exact invertibility ($R^{-1} = R^\top$, providing a natural mechanism for semantic negation). These properties are well-known in the orthogonal RNN literature (Mhammedi et al., 2017; Kerg et al., 2019), where constraining recurrent transitions to be orthogonal prevents vanishing and exploding gradients. Our approach differs in a key respect: orthogonal RNNs apply a *single learned* orthogonal matrix at every timestep, while

we learn a *distinct rotation for each input sentence* via the Lie algebra, making the composition input-dependent and semantically meaningful.

While $\mathrm{SO}(n)$ composition provides the right algebraic structure, we find that the composed endpoint state s_T alone is insufficient for long-chain generalization—it drifts to out-of-distribution regions of the sphere when the chain length exceeds training. Adding a lightweight Transformer over the full state trajectory $[s_0, s_1, \ldots, s_T]$ provides a length-invariant readout: each state s_t lies on S^{n-1} regardless of T, so attention patterns transfer from short training chains to long test chains. Neither component alone suffices; their combination achieves the best results across all experiments.

Our contributions:

- An empirical investigation of $\mathrm{SO}(n)$ rotation operators for sentence composition, showing that bounded conditioning and non-commutativity translate to measurable length generalization advantages: $+8.4$pp over the strongest baseline on order sensitivity ($p = 2.8 \times 10^{-5}$) and $+14.2$pp at $2\times$ training length.

- State-of-the-art on kinship chain reasoning (Sinha et al., 2019): $57.5\% \pm 1.1\%$ (5 seeds), outperforming Transformers (52.2%, $10\times$ more parameters) and all other baselines at every OOD chain length, with the combination of $\mathrm{SO}(n)$ composition and trajectory attention eliminating the long-chain collapse ($+22$pp at k=8).

- Validation on book-level retrieval (55K books, 30+ chapters) showing 10–20$\times$ stronger order sensitivity than Transformer, LSTM, and mean-pool baselines (Appendix F).

Related work. Orthogonal and unitary RNNs (Mhammedi et al., 2017; Kerg et al., 2019) constrain recurrent transitions to preserve norms and ensure $\kappa = 1$, preventing vanishing/exploding gradients. Our approach shares this motivation but learns a distinct $\mathrm{SO}(n)$ operator per input sentence via the Lie algebra, making composition input-dependent and non-commutative. Kerg et al. (2019) showed that strict orthogonality can limit expressivity; our trajectory Transformer readout addresses this (Section 2). Yang and Rush (2016) used Lie group actions to move a memory head for algorithmic tasks; we use them for semantic composition. LieRE (Ostmeier et al., 2024) learns skew-symmetric matrices mapped to rotations via the matrix exponential—the same construction we employ—but for positional encoding, not sequential composition. RoPE (Su et al., 2021) and RotatE (Sun et al., 2019) also use rotations for position encoding and knowledge graph relations, respectively. Socher et al. (2012) composed words as matrix-vector products without algebraic constraints. State-space models (S4/Mamba; Gu et al. 2022; Gu and Dao 2024) process sequences efficiently but lack explicit algebraic structure. Hahn (2020) and Dziri et al. (2023) show theoretical and empirical limitations of transformers on compositional reasoning, motivating structured alternatives.

2 Method

We describe our approach: map each sentence to a rotation matrix in $\mathrm{SO}(n)$ via the Lie algebra $\mathfrak{so}(n)$, then compose via matrix multiplication. The construction generalizes to other Lie groups G, but we focus on $\mathrm{SO}(n)$ for its simplicity and strong guarantees. Figure 1 illustrates the pipeline.

2.1 Sentence-to-Operator Mapping

Given a sentence x_t, an encoder produces embedding $h_t \in \mathbb{R}^d$. We map this to a rotation matrix $R_t \in \mathrm{SO}(n)$ through the Lie algebra $\mathfrak{so}(n)$:

$$h_t \xrightarrow{W} \theta_t \in \mathbb{R}^{\frac{n(n-1)}{2}} \xrightarrow{\text{skew}} A_t \in \mathfrak{so}(n) \xrightarrow{\exp} R_t \in \mathrm{SO}(n) \quad (1)$$

where $W \in \mathbb{R}^{d \times n(n-1)/2}$ is a learned projection, antisymmetrization gives $A_t = U - U^\top$, and $\exp : \mathfrak{so}(n) \to \mathrm{SO}(n)$ is the matrix exponential, which sends any skew-symmetric matrix to a valid rotation without projection or normalization. The construction is encoder-agnostic (frozen or trainable). Crucially, each sentence maps to a *distinct* rotation—unlike orthogonal RNNs (Mhammedi et al., 2017) which apply a single learned orthogonal matrix at every timestep.

Given T sentences, we maintain a state $s_t \in \mathbb{R}^n$ updated by each rotation:

$$s_0 \in S^{n-1}, \qquad s_t = R_t \cdot s_{t-1}, \qquad t = 1, \ldots, T \quad (2)$$

where s_0 is a learnable unit vector. This yields the endpoint $s_T = R_T \cdots R_1 \cdot s_0$ and the trajectory $\mathcal{S} = (s_0, \ldots, s_T)$. Since each $R_t \in \mathrm{SO}(n)$, the state remains on S^{n-1} at every step.

Trajectory-aware readout. The endpoint s_T can drift out-of-distribution when T exceeds the

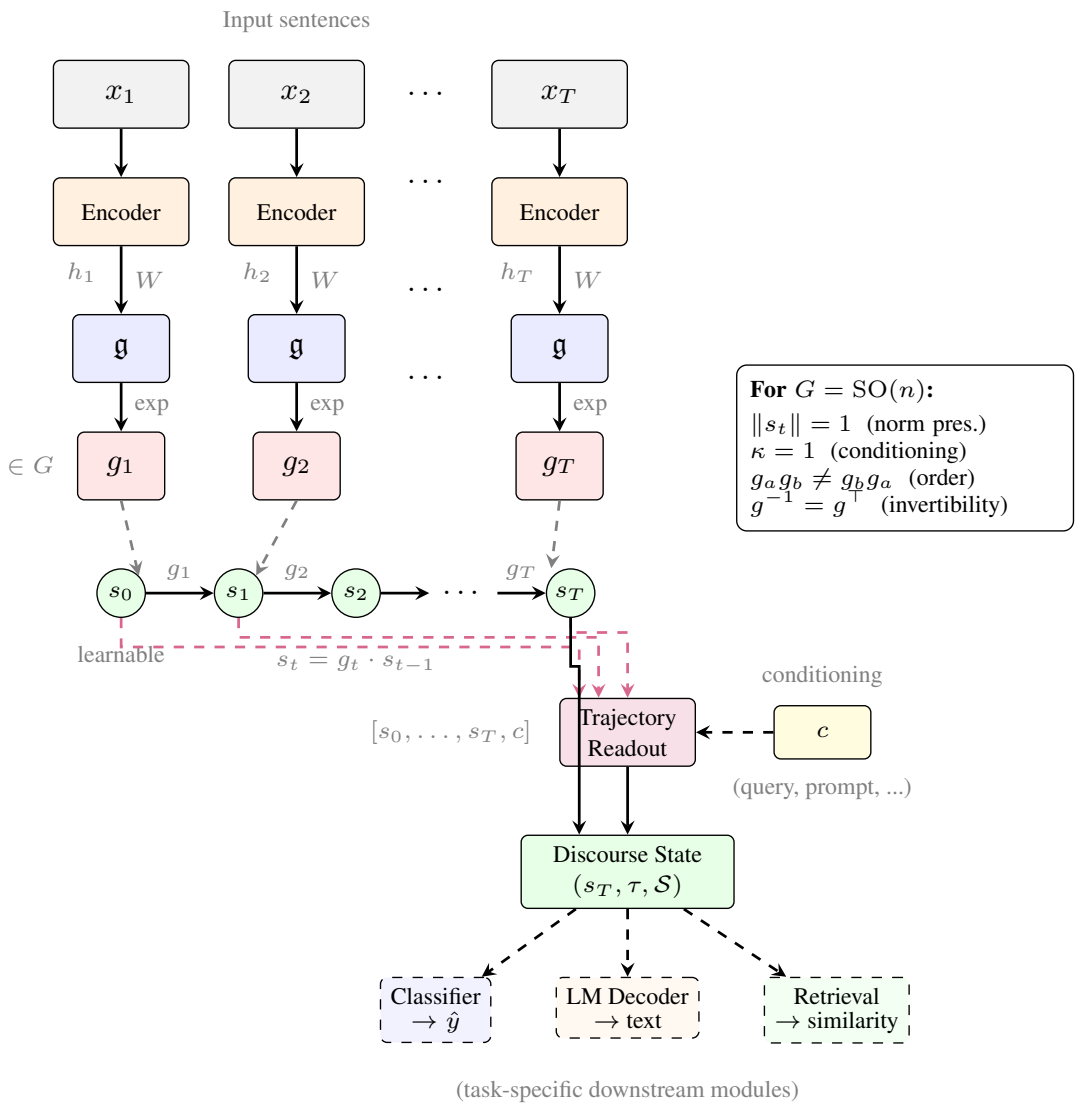

Figure 1: Framework overview. Sentences are encoded and projected through the Lie algebra $\mathfrak{g}$ to group elements $g_t \in G$. Operators compose sequentially on a state vector ($s_t = g_t \cdot s_{t-1}$), producing a state trajectory. A trajectory readout module processes the full trajectory with an optional conditioning input c, yielding a discourse state $(s_T, \tau, \mathcal{S})$ that can be consumed by any downstream module. We instantiate the framework with $G = \mathrm{SO}(n)$, which guarantees norm preservation, bounded conditioning ($\kappa = 1$), non-commutativity, and exact invertibility.

training range, but individual states s_t remain on S^{n-1} regardless of chain length. We process the trajectory with a 2-layer Transformer encoder, appending a conditioning token c (e.g., a query embedding):

$$\tau = \mathrm{TransformerEnc}\big([s_0, s_1, \ldots, s_T,\ W_c\, c]\big)_{T+1} \tag{3}$$

This readout is length-invariant: attention patterns transfer across lengths because each state occupies the same manifold. The model outputs (s_T, τ) for downstream classification.

2.2 Properties of $\mathrm{SO}(n)$ Composition

$\mathrm{SO}(n) = \{R \in \mathbb{R}^{n\times n} : R^\top R = I, \det(R) = 1\}$ provides four properties relevant to sequential composition (proofs in Appendix B):

Proposition 1 (Norm Preservation)**.** *For any $R \in \mathrm{SO}(n)$ and $s \in \mathbb{R}^n$: $\|Rs\| = \|s\|$. The state trajectory remains on S^{n-1}.*

Proposition 2 (Bounded Conditioning)**.** *For any product $M = R_T \cdots R_1$ with $R_t \in \mathrm{SO}(n)$: $\kappa(M) = 1$, independent of T. For unconstrained matrices, $\kappa(\prod M_t)$ can grow exponentially.*

Proposition 3 (Exact Invertibility)**.** *For any $R \in \mathrm{SO}(n)$: $R^{-1} = R^\top$, computable as a transpose*

without numerical inversion.

Proposition 4 (Non-Commutativity)**.** *For $n \geq 3$, $\mathrm{SO}(n)$ is non-abelian: $\exists\, R_a, R_b$ with $R_a R_b \neq R_b R_a$.*

These contrast with LSTMs (no norm bound, no conditioning guarantee, no invertibility) and unconstrained matrices (non-commutative but κ grows exponentially—verified in Section 4). The same properties hold for other compact Lie groups (e.g., $\mathrm{SU}(n)$); we choose $\mathrm{SO}(n)$ for simplicity and leave alternatives to future work. We contrast $\mathrm{SO}(n)$ composition with LayerNorm-based aggregation in Appendix G. For high-dimensional state spaces where full $\mathrm{SO}(n)$ is impractical, we explore a projected variant using $\mathrm{SO}(m)$ with $m \ll n$ and learned projections (Appendix F).

Why these properties matter for length generalization. Proposition 2 is the most directly relevant: it guarantees that composing T rotations does not amplify or suppress any direction, regardless of T. For unconstrained matrices, even small deviations from orthogonality compound exponentially—after 20 compositions, we observe $\kappa = 308$ empirically (Section 4), meaning some directions are amplified $308\times$ relative to others. Proposition 1 ensures that the state trajectory $\{s_t\}$ remains on S^{n-1}, which is critical for the trajectory Transformer: each state occupies the same manifold regardless of chain length, enabling attention patterns to transfer from short training chains to long test chains.

Relation to orthogonal RNNs. Orthogonal RNNs (Mhammedi et al., 2017; Kerg et al., 2019) constrain the recurrent transition matrix W to be orthogonal, ensuring $\kappa(W^T) = 1$ after T steps. Our approach shares this stability property but differs in two ways: (1) each sentence produces a *different* rotation R_t via the Lie algebra, making the composition input-dependent rather than a fixed recurrence; (2) we operate at the sentence level (composing sentence embeddings) rather than the token level. Kerg et al. (2019) showed that strict orthogonality can limit expressivity; our trajectory Transformer readout compensates for this by providing a flexible, length-invariant mechanism to extract information from the constrained state trajectory.

2.3 Training

We train end-to-end with a task-specific loss plus an orthogonality regularizer $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \cdot \frac{1}{T}\sum_{t=1}^{T} \|R_t^\top R_t - I\|_F^2$ ($\lambda = 0.01$), guarding against numerical drift under bfloat16 mixed precision.

3 Experiments

We validate the $\mathrm{SO}(n)$ composition approach on classification tasks that test specific algebraic properties, then on a kinship chain reasoning task following the protocol of CLUTRR (Sinha et al., 2019) for compositional chain-length generalization. We additionally validate on a book-level retrieval task using 55K real books, testing composition over 30+ chapters with a projected $\mathrm{SO}(m)$ operator (Appendix F).

3.1 Experimental Setup

Synthetic tasks. We design three controlled tasks using template-generated English sentences about entity state changes (e.g., "Alice opens the box" then "Bob closes the box" $\rightarrow$ box is closed; reversed $\rightarrow$ box is open):

- **Task A (Order Sensitivity):** Given two sentences x_1, x_2 and a query, classify the final state. The *paired accuracy* metric measures whether the model distinguishes (x_1, x_2) from (x_2, x_1).
- **Task B (Length Generalization):** Train on sequences of length 5–10, test on lengths 5, 10, 20, 50, 100. Measures how gracefully accuracy degrades beyond the training distribution.
- **Task C (Algebraic Structure):** Sequences include negation pairs ("open"/"close"). We measure *inversion error* $\|R_{\neg x} R_x - I\|_F$ and *commutativity gap* $\|R_a R_b - R_b R_a\|_F$ to assess whether the model learns algebraic properties without explicit supervision.

Kinship chain reasoning. Following the evaluation protocol of CLUTRR (Sinha et al., 2019), we construct a kinship reasoning task that tests compositional generalization over chain length. Given a story describing family relationships (e.g., "Alice is Bob's mother. Bob is Charlie's father."), the model must infer the relationship between two specified characters. We train on chains of length $k = 2, 3$ (5000 examples) and test on $k = 2, \ldots, 8$ (2100 examples, 300 per k), with 13 kinship relation classes. This directly tests whether models can compose more relationship hops than seen during

training. We additionally evaluate on the original CLUTRR benchmark with AMT-paraphrased natural language stories in Appendix E.

Models. We compare six architectures. For the synthetic tasks and kinship chain reasoning, all models use a frozen sentence encoder (all-MiniLM-L6-v2, 384-dim) to isolate the contribution of the composition mechanism:

- **Rotation SO(n)** (n=32, 333K params): Our model with GRU trajectory (Section 2).
- **Rotation + StateTF** (347K params): Our model with Transformer trajectory readout.
- **LSTM** (h=256, 329K params): Processes sentence embeddings sequentially.
- **Transformer** (4 layers, 4 heads, 3.4M params): Full self-attention over all sentence embeddings, applied at the sentence level.
- **Bag of Sentences** (197K params): Mean-pools all sentence embeddings, discarding order (order-insensitive control).
- **Random Rotation** (142K params): Uses random fixed orthogonal matrices instead of learned operators (tests whether learned structure is necessary).

Additional baselines in Table 1: **Commutative Rotation** (148K) restricts operators to diagonal matrices, which commute by construction—isolating the contribution of non-commutativity. **Unconstrained Matrix** (291K, parameter-matched) uses general $n \times n$ matrices without the $\mathrm{SO}(n)$ constraint—testing whether the group structure itself matters beyond matrix composition. All models are trained with AdamW, cosine learning rate schedule, and 5 random seeds. See Appendix A for full hyperparameters.

3.2 Synthetic Task Results

Table 1 summarizes the synthetic results. Three findings stand out:

Order sensitivity (Task A). The rotation model achieves 85.3% paired accuracy versus 76.8% for LSTM (+8.4pp, $p = 2.8 \times 10^{-5}$, Cohen's $d = 9.6$ over 5 seeds). The bag-of-sentences baseline achieves exactly 50% paired accuracy (chance), confirming the task requires order sensitivity. The commutative rotation (75.8%) falls between, showing that non-commutativity specifically contributes +9.5pp.

Length generalization (Task B). Figure 2a shows accuracy versus sequence length. At length 20 (2× training maximum), rotation achieves 65.4% versus LSTM's 51.2% (+14.2pp). All models collapse to chance (∼50%) at length 50+. The parameter-matched unconstrained matrix (65.8%) performs comparably, suggesting that matrix composition—not specifically $\mathrm{SO}(n)$—drives the length generalization advantage over LSTMs.

Algebraic structure (Task C). All learned models achieve similar classification accuracy (∼74%), but the algebraic metrics differentiate them sharply. The rotation model's inversion error (0.182) is 40× lower than the random rotation baseline (7.999), indicating that the model learns to map negation pairs to approximate inverses without explicit supervision. The commutative model achieves the lowest inversion error (0.065) because diagonal matrices invert trivially.

3.3 Kinship Chain Reasoning: Compositional Generalization

Table 2 and Figure 2b present the kinship chain reasoning results. *Note on class distribution:* the number of reachable relation types decreases with chain length—k=2 has 10 classes (chance 10%), while $k \geq 6$ collapses to 2 classes (chance 50%). Raw accuracy at long chains is thus inflated relative to short chains; the models' advantage over chance is actually larger at short chains. We report raw accuracy for comparability across models. We highlight three findings:

Original rotation collapses at long chains. The base rotation model (with GRU trajectory) peaks at k=5 (65.0%) but drops to 37.3% at k=8—*below* the 50% chance level for the 2-class problem at that chain length. The endpoint state s_T after 8 rotations lands in a region of S^{n-1} where the 13-class classifier (trained on k=2, 3) confidently predicts labels that don't even exist at k=8, producing worse-than-random performance.

Trajectory attention fixes the cliff. Replacing the GRU trajectory encoder with a 2-layer Transformer over the state sequence (Section 2.1) eliminates the collapse: accuracy at k=8 improves from 37.3% to 59.3% (+22.0pp, 5-seed mean). The Rotation + StateTF model achieves the highest accuracy at every OOD chain length k=4, . . . , 8 and the best overall accuracy of 57.5%±1.1% (5 seeds). All pairwise comparisons are statistically significant at 95% confidence (Appendix C).

Neither component alone is sufficient. The

Table 1: Synthetic task results (test set). Paired accuracy measures order sensitivity; inv. error and comm. gap measure learned algebraic structure. Best in **bold**, second underlined. †Unfair comparison: 10× more parameters.

		Task A		Task B		Task C	
Model	Params	Overall	Paired	Overall	len=20	Acc	Inv. Err
Rotation SO(n)	333K	**82.7**	**85.3**	**76.9**	65.4	74.1	0.182
LSTM	329K	80.8	76.8	71.5	51.2	74.2	—
Transformer†	3.4M	72.2	57.5	73.0	58.0	—	—
Bag of Sentences	197K	70.6	50.0	—	—	—	—
Commutative Rot.	148K	80.7	75.8	—	—	73.8	**0.065**
Unconstrained	291K	**82.7**	84.8	75.7	**65.8**	**74.8**	0.071
Random Rotation	142K	70.8	50.0	50.5	49.8	71.6	7.999

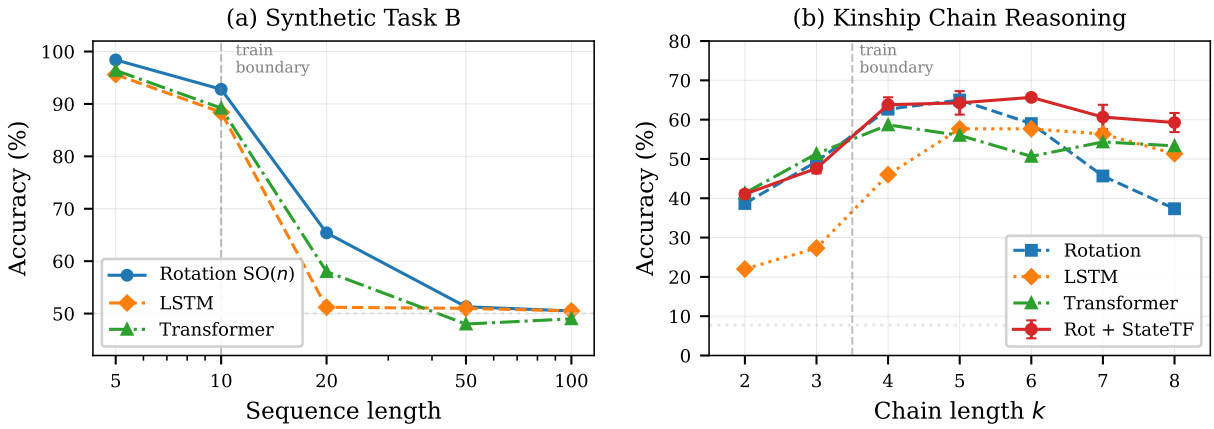

Figure 2: Length generalization on synthetic Task B (a) and kinship chain reasoning (b). Dashed vertical lines mark the training boundary. On kinship reasoning, error bars show ±1 std over 5 seeds for Rotation + StateTF. The rotation model degrades more slowly than LSTM on both tasks; adding trajectory attention (StateTF) eliminates the long-chain collapse at $k \geq 8$.

standalone Transformer baseline (52.2%)—which has full attention over all sentence embeddings and 10× more parameters (3.4M vs 347K)—lacks SO(n) structure and shows relatively flat performance across chain lengths. The base rotation model (51.1%) has the algebraic structure but collapses at $k \geq 7$. Only their combination achieves robust generalization across all chain lengths, suggesting that SO(n) composition and trajectory attention provide complementary inductive biases that neither full-context attention nor algebraic composition achieves alone.

3.4 Ablations

Figure 3 summarizes key ablations:

Operator dimension (panel a): Task A paired accuracy peaks at n=64 (85.8%) with diminishing returns. Task B length generalization improves monotonically—n=128 achieves 72.4% at length 20 (+7.0pp over n=32) and is the only configuration exceeding chance at length 50 (54.7%).

Trajectory variants (panel b): We test four trajectory encoders on kinship chain reasoning. Mean-pooled hop features provide modest improvement (+5pp at k=8). Query-attended hop features help substantially (+16.5pp at k=8). The State Transformer is best overall. A gated multiscale approach (blending endpoint and hop features) performs *worse* than the original, as the gate learns to over-rely on the endpoint.

Random rotation control (panel c): Replacing learned operators with random orthogonal matrices yields 50.0% paired accuracy on Task A (exact chance) and 49.8% at length 20 on Task B, confirming that the learned rotation structure—not the trajectory encoder—drives performance.

Parameter fairness. The LSTM at 4× the rotation model's parameters (1.38M vs 333K) still achieves only 75.5% paired accuracy versus rotation's 85.3%. The parameter-matched unconstrained matrix (291K) matches rotation on Task A (84.8% vs 85.3%), suggesting that the value of the SO(n) constraint lies in its theoretical guarantees (bounded conditioning, invertibility) rather than raw accuracy on short sequences. On synthetic tasks, removing the trajectory encoder entirely has

Table 2: Kinship chain reasoning test accuracy (%) by chain length k. Models train on k=2, 3 and are tested on k=2, . . . , 8 (300 examples per k). Rotation + StateTF results show mean ± std over 5 seeds. Best per column in **bold**.

		In-dist		Out-of-distribution					
Model	Params	k=2	k=3	k=4	k=5	k=6	k=7	k=8	Overall
Rot + StateTF	347K	41.1	47.6	**63.8**	**64.3**	**65.7**	**60.7**	**59.3**	**57.5**±1.1
Rotation	336K	38.7	**49.3**	62.7	65.0	59.0	45.7	37.3	51.1
Transformer	3.5M	**41.3**	**51.3**	58.7	56.0	50.7	54.3	53.3	52.2
LSTM	332K	22.0	27.3	46.0	57.7	57.7	56.3	51.3	45.5
Bag of Sent.	200K	22.0	22.7	51.7	53.3	50.7	49.7	51.0	43.0

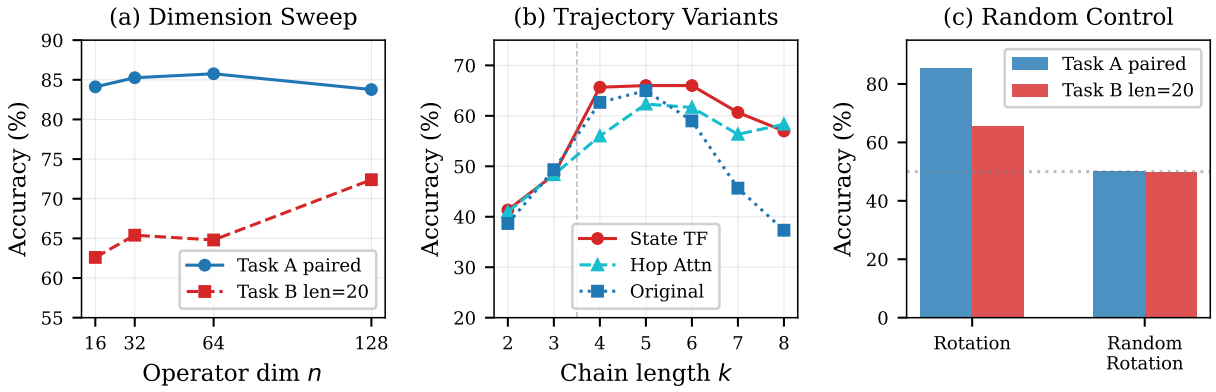

Figure 3: Ablation studies. (a) Operator dimension sweep on synthetic tasks: n=32 is near-optimal for Task A; larger n improves Task B length generalization. (b) Kinship chain reasoning trajectory encoder variants: State Transformer and Hop Attention both fix the $k \geq 8$ collapse; mean pooling helps modestly. (c) Random rotation control: replacing learned operators with random orthogonal matrices reduces paired accuracy to chance (50%), confirming the learned structure is necessary.

negligible impact ($<$ 1pp), confirming the operators do the compositional work; the trajectory encoder matters only at OOD chain lengths.

4 Analysis

We investigate the mechanistic basis for the rotation model's advantages through analyses of trained model internals.

Condition number explains length generalization. Figure 4a shows the condition number κ of the accumulated operator as a function of composition steps on Task B sequences. The rotation model maintains $\kappa = 1.0$ (up to floating-point precision) at every step, as guaranteed by Proposition 2. The unconstrained matrix model's condition number grows exponentially, reaching 308 at step 20. This means that after 20 compositions, the unconstrained model amplifies some directions by 308× relative to others—information in the suppressed directions is effectively lost. This directly explains why both models perform similarly at short chains (where κ is still moderate) but diverge at longer chains.

State norm stability. Figure 4b shows the state vector norm over composition steps. The rotation model's states remain at $\|s_t\| = 1.0$ to machine precision across all steps (Proposition 1). LSTM hidden states start near zero, grow to approximately 4–5 over the first 15 steps, then oscillate—the gates prevent explosion but provide no stability guarantee. PCA analysis of the state trajectories reveals that rotation states spread across the full n-dimensional manifold (PC1 explains only 11.2% of variance), while LSTM states collapse to 1–2 dominant directions (PC1 explains 72.0%).

Learned algebraic structure. We analyze the eigenvalue spectra of learned rotation operators on Task A (192 unique sentences). Eigenvalues of $\mathrm{SO}(n)$ matrices lie on the unit circle in $\mathbb{C}$. We find that semantically similar sentences (e.g., all "open" actions) cluster tightly in eigenvalue space, with a between-type to within-type distance ratio of 5.9. Negation pairs ("open"/"close", "turn on"/"turn off") exhibit similar eigenvalue angle magnitudes (difference < 0.08), consistent with the inverse relationship $R_{\text{close}} \approx R_{\text{open}}^{\top}$. This structure emerges

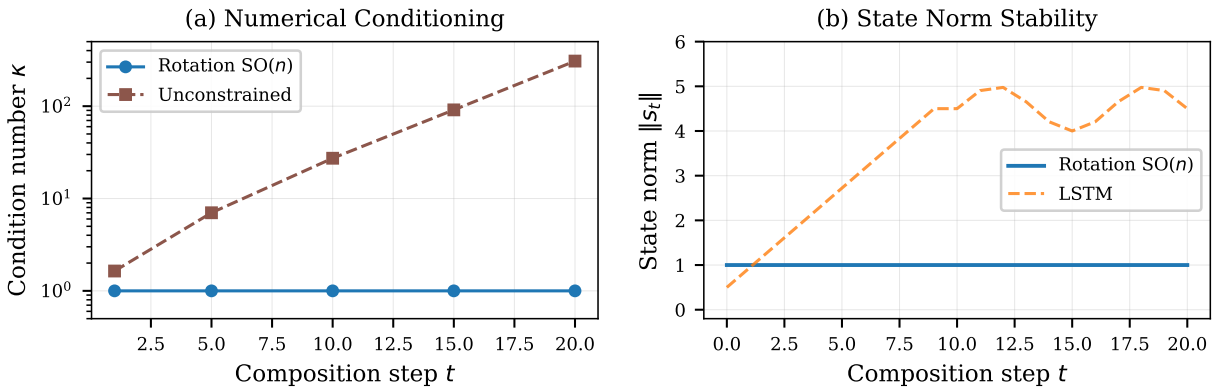

Figure 4: Mechanistic analysis. (a) Condition number of the accumulated operator $R_T \cdots R_1$ versus composition step. Rotation matrices maintain $\kappa = 1.0$ exactly (by construction), while unconstrained matrices degrade exponentially, reaching $\kappa = 308$ at step 20. (b) State vector norm $\|s_t\|$ over composition steps. Rotation states remain at $\|s_t\| = 1.0$ (machine precision), while LSTM hidden states grow to 4–5 and oscillate.

purely from the classification objective—no explicit algebraic supervision is applied.

Kinship chain reasoning per-class analysis. On kinship chain reasoning, the State Transformer variant's improvements over the base rotation model concentrate on relation types that require longer reasoning chains: great-grandparent (+20.8pp), brother/sister-in-law (+19.4pp), and spouse (+13.4pp). Short-chain relations (parent, child) show smaller gains. This confirms that the trajectory Transformer specifically helps with the long-chain readout problem rather than providing a general accuracy boost.

Order sensitivity at scale. On the book-level retrieval task (55K books, 30+ chapters; Appendix F), all models achieve ≥97.9% R@1, but order sensitivity sharply differentiates them. When chapter order is randomly shuffled, Transformer and LSTM baselines show <1% retrieval degradation—they barely use sequential structure. Mean pooling shows exactly 0% (order-invariant by construction). In contrast, projected $\mathrm{SO}(m)$ operators show 3–14% degradation, with the trajectory-attended variant reaching 10–14%. This 10–20× gap confirms that the algebraic non-commutativity of $\mathrm{SO}(n)$ captures sequential structure that attention and recurrence do not, even at realistic scale with 30+ composition steps.

5 Discussion

We investigated $\mathrm{SO}(n)$ rotation operators as a composition mechanism for sequential sentence understanding. The algebraic structure—bounded conditioning, norm preservation, non-commutativity—provides measurable advantages for length generalization, with $\mathrm{SO}(n)$ operators outperforming all baselines on order sensitivity and length generalization across all experiments.

The central finding is that $\mathrm{SO}(n)$ composition and trajectory attention are complementary. $\mathrm{SO}(n)$ composition ensures each step is well-conditioned ($\kappa = 1$), but the endpoint s_T drifts out-of-distribution at long chains. Trajectory attention provides length-invariant readout since each s_t lies on S^{n-1} regardless of T. Neither alone suffices: the standalone Transformer (no algebraic structure) and the base rotation model (no trajectory attention) both degrade. This supports the design principle motivating our approach: *structured composition for building representations, attention for reading them out.*

The discourse state s_T is a fixed-size (n-dim) summary computed in $O(T \cdot n^3)$ time (one matrix exponential and one matrix-vector product per step), compared to $O(T^2 d)$ for self-attention. For our experiments ($n = 32$, T up to 10), the per-step cost is modest; on the book retrieval task ($T = 30+$), the projected $\mathrm{SO}(m)$ variant with $m = 64$–128 keeps this manageable. The trajectory Transformer adds $O(T)$ cost at readout, applied once rather than at every layer.

Future work. Conditional generation (discourse state conditioning a decoder), richer Lie groups ($\mathrm{SU}(n)$, $\mathrm{SE}(n)$), and hybrid architectures combining Lie group composition with selective gating are natural extensions.

Limitations

Our experiments freeze the sentence encoder to isolate the composition mechanism; encoder–

composition interaction deserves further study. $\mathrm{SO}(n)$ composition accumulates all transformations without selective forgetting—on bAbI tasks requiring location tracking, LSTMs outperform (Appendix D), consistent with the expressivity limitations of strict orthogonality identified by Kerg et al. (2019). The experiments are small-scale (333K–347K composition parameters, frozen sentence encoder). Whether the advantages persist at larger scale remains open. Finally, the synthetic tasks and kinship chain reasoning benchmark use template-generated text; performance on naturalistic text with coreference and implicit relations is weaker (Appendix E).

References

Erik Arakelyan, Todor Mihaylov, Peter Clark, and Pasquale Minervini. 2025. FLARE: Faithful logic-aided reasoning and exploration. In *Proceedings of EMNLP*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Jiaxin Chen and 1 others. 2025. Do large language models excel in complex logical reasoning with formal language? *arXiv preprint arXiv:2505.16998*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Giancarlo Kerg, Kyle Goyette, Maximilian Puelma Touzel, Gauthier Gidel, Eugene Vorontsov, Yoshua Bengio, and Guillaume Lajoie. 2019. Non-normal recurrent neural network (nnRNN): Learning long time dependencies while improving expressivity with transient dynamics. In *Advances in Neural Information Processing Systems*.

Irtaza Khalid, Amir Masoud Nourollah, and Steven Schockaert. 2025. Large language and reasoning models are shallow disjunctive reasoners. *arXiv preprint arXiv:2503.23487*.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. 2017. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *International Conference on Machine Learning*, pages 2401–2409.

Sophie Ostmeier, Brian Axelrod, Michael E. Moseley, Akshay S. Chaudhari, and Curtis Langlotz. 2024. LieRE: Generalizing rotary position encodings. In *International Conference on Machine Learning*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. Clutrr: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of EMNLP-IJCNLP*, pages 4506–4515.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Yinya Wang and 1 others. 2025. Stepwise informativeness search for improving LLM reasoning. In *Proceedings of EMNLP*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Greg Yang and Alexander M. Rush. 2016. Lie access neural turing machine. *arXiv preprint arXiv:1611.02854*.

A Experimental Details

A.1 Training Configuration

All models are trained with the following shared configuration unless otherwise noted:

A.2 Model Configurations

Sentence encoder. For the synthetic tasks and kinship chain reasoning experiments, all models use `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019) (22M parameters, 384-dimensional output), frozen throughout training. This design choice isolates the contribution of the composition mechanism, ensuring that order sensitivity must emerge from the operators rather than from positional information in the embeddings. For the original CLUTRR benchmark (Appendix E), we use a trainable all-mpnet-base-v2 encoder (768-dim) with 2-phase training.

Rotation model specifics. The Lie algebra projection maps 384-dim embeddings to $n(n-1)/2$ upper-triangular parameters (496 for n=32). The skew-symmetric matrix is constructed by antisymmetrization, then exponentiated via `torch.linalg.matrix_exp` in float32 (to avoid NaN under bfloat16 autocast). The orthogonality regularizer weight is $\lambda_{\text{ortho}} = 0.01$.

State Transformer specifics. The trajectory Transformer uses 2 encoder layers with 4 attention heads, feedforward dimension $4n = 128$, GELU activation, and 0.1 dropout. The query embedding (384-dim) is projected to n-dim via a learned linear layer and appended as a special token. Total additional parameters over the base rotation model: $\sim$11K.

A.3 Synthetic Data Generation

Task A (Order Sensitivity). Template-generated English sentences about entity state changes involving 8 entity types (boxes, lights, doors, etc.) and 7 action types (open/close, turn on/off, put/take, give, move). Each example consists of 2–5 sentences, a query about the final state of one entity, and a binary label. The paired evaluation metric presents both (x_1, x_2) and (x_2, x_1) orderings and checks if the model correctly distinguishes them. Dataset: 10K train, 2K val, 2K test.

Example: Sentences: ["Alice opens the box.", "Bob closes the box."]. Query: "Is the box open?" Label: No. Swapping the sentence order changes the answer to Yes—the model must be sensitive to this ordering.

Task B (Length Generalization). Same sentence templates as Task A, but sequences are generated at specific lengths. Training: lengths 5 and 10 (5K examples each). Testing: lengths 5, 10, 20, 50, 100 (1K each). The key metric is accuracy at each test length, especially at lengths exceeding the training maximum (20, 50, 100).

Example (length 5): ["Alice opens the box.", "Bob turns on the light.", "Charlie moves to the kitchen.", "Alice closes the box.", "Bob turns off the light."]. Query: "Is the box open?" Label: No. At test time, the model must handle sequences of length 20–100 despite training only on lengths 5–10.

Task C (Algebraic Structure). Sequences include explicit negation pairs (e.g., "Alice opens the box" / "Alice closes the box"). In addition to classification accuracy, we compute: (1) *inversion error* $\|R_{\neg x}R_x - I\|_F$ measuring how close negation pairs map to inverse operators, and (2) *commutativity gap* $\|R_aR_b - R_bR_a\|_F$ measuring non-commutativity between operator pairs. Dataset: 10K train, 2K val, 2K test.

Example: ["Alice opens the box.", "Bob turns on the light.", "Alice closes the box."]. The open/close pair should map to approximate inverse operators: $R_{\text{close}} \approx R_{\text{open}}^{-1}$, so their composition $R_{\text{close}}R_{\text{open}} \approx I$.

A.4 Kinship Chain Reasoning Data Generation

Following the evaluation protocol of CLUTRR (Sinha et al., 2019), we generate synthetic kinship reasoning stories. Each story describes a chain of family relationships through natural language sentences, with 0–2 distractor sentences added and sentence order shuffled. The task is to infer the relationship between two specified characters.

Example (k=3): ["Alice is Bob's mother.", "Bob is Charlie's father.", "Charlie is Diana's brother."]. Query: "What is the relationship between Alice and Diana?" Label: grandmother. The model must compose three relationship hops: mother $\rightarrow$ father $\rightarrow$ brother $\Rightarrow$ grandmother.

- **Relations:** 13 kinship classes (parent, child, grandparent, grandchild, sibling, aunt/uncle, niece/nephew, cousin, spouse, father/mother-in-law, son/daughter-in-law, brother/sister-in-law, great-grandparent).

Table 3: Shared training hyperparameters.

Parameter	Value
Optimizer	AdamW
Learning rate	10^{-4}
Weight decay	10^{-5}
LR schedule	Cosine annealing with 5% linear warmup
Batch size	128
Gradient accumulation	1
Gradient clipping	Max norm 1.0
Max epochs	50
Early stopping	Patience 10 (validation loss)
Random seeds	42, 43, 44, 45, 46
Precision	Mixed (bfloat16 forward, float32 matrix_exp)

Table 4: Complete model parameter counts and configurations.

Model	Key Config	Trainable Params
Rotation SO(n)	n=32, traj=GRU(h=64)	336,285
Rotation + State Transformer	n=32, 2L/4H Transformer	347,005
Rotation + Hop Attention	n=32, attn_dim=64	354,525
Rotation + Hop Pool	n=32, mean pool	325,661
Rotation + Multiscale Gate	n=32, gate_dim=64	350,494
LSTM	h=256	332,365
Transformer	4 layers, 4 heads, d=384	3,454,733
Bag of Sentences	mean pool + MLP	200,205
Commutative Rotation	n=32, diagonal	148,000
Unconstrained Matrix	n=20 (param-matched)	291,000
Unconstrained Matrix	n=32 (original)	537,000
Random Rotation	n=32, frozen random	142,000

- **Composition rules:** 15+ rules mapping pairs of atomic relations to composed relations (e.g., parent + parent → grandparent).
- **Names:** 50 gender-balanced names, randomly assigned per story.
- **Templates:** Multiple natural language templates per relation type for linguistic diversity.
- **Splits:** Train: k=2 (2500) + k=3 (2500) = 5000. Val: k=2 (500) + k=3 (500) = 1000. Test: $k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$, ∼300 per length = 2100 total.

B Proofs

Proposition 5 (Norm Preservation)**.** *For any $R \in \mathrm{SO}(n)$ and $s \in \mathbb{R}^n$: $\|Rs\| = \|s\|$.*

Proof. $\|Rs\|^2 = (Rs)^\top(Rs) = s^\top R^\top Rs = s^\top Is = s^\top s = \|s\|^2$, where $R^\top R = I$ by definition of $\mathrm{SO}(n)$. □

Proposition 6 (Bounded Conditioning)**.** *For any product $M = R_T \cdots R_1$ with $R_t \in \mathrm{SO}(n)$: $\kappa(M) = 1$.*

Proof. Since $\mathrm{SO}(n)$ is closed under multiplication, $M \in \mathrm{SO}(n)$. For any $R \in \mathrm{SO}(n)$, $R^\top R = I$, so the singular values satisfy $\sigma_i^2 = 1$ for all i. Therefore $\kappa(M) = \sigma_{\max}/\sigma_{\min} = 1/1 = 1$, independent of T.

For comparison, consider unconstrained matrices $M_t \in \mathbb{R}^{n\times n}$ with $\|M_t\| = 1 + \epsilon$ for small $\epsilon > 0$. Then $\kappa(\prod_{t=1}^T M_t) \le \prod_{t=1}^T \kappa(M_t)$, which can grow as $(1+\epsilon)^T$—exponentially in T. □

Proposition 7 (Exact Invertibility)**.** *For any $R \in \mathrm{SO}(n)$: $R^{-1} = R^\top$.*

Proof. By definition, $R \in \mathrm{SO}(n)$ satisfies $R^\top R = RR^\top = I$ and $\det(R) = 1$. Therefore $R^{-1} = R^\top$, computable as a transpose without numerical matrix inversion. □

Proposition 8 (Non-Commutativity)**.** *For $n \ge 3$, $\mathrm{SO}(n)$ is non-abelian.*

Proof. Consider $n = 3$. Let $R_x(\theta)$ denote rotation by angle θ about the x-axis and $R_z(\phi)$ rotation by

ϕ about the z-axis. For $\theta = \phi = \pi/2$:

$$R_x(\tfrac{\pi}{2})R_z(\tfrac{\pi}{2}) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

$$R_z(\tfrac{\pi}{2})R_x(\tfrac{\pi}{2}) = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}$$

These are not equal, so $\mathrm{SO}(3)$ is non-abelian. Since $\mathrm{SO}(3)$ embeds in $\mathrm{SO}(n)$ for all $n \geq 3$ (as rotations in a 3-dimensional subspace), $\mathrm{SO}(n)$ is non-abelian for $n \geq 3$. □

C Full Results

C.1 Synthetic Task Results: 5-Seed Validation

Statistical test (Task A, rotation vs LSTM): Paired t-test over 5 seeds: $p = 2.8 \times 10^{-5}$, Cohen's $d = 9.6$. The effect size is extraordinarily large, though the confidence interval on d is wide with n=5.

C.2 Operator Dimension Sweep (Complete)

Key observations: Task A paired accuracy peaks at n=64 (85.8%) with diminishing returns. Task B length generalization improves monotonically with n—n=128 achieves 72.4% at length 20 and is the only configuration exceeding chance at length 50 (54.7%). Task C inversion error increases with n (0.113 → 0.528), reflecting the increased difficulty of learning precise inverses in higher dimensions. This reveals a capacity-precision trade-off: larger operators compose more stably but learn less precise algebraic structure.

C.3 Trajectory Encoder Ablation (Synthetic Tasks)

Key finding: Removing the trajectory encoder entirely has negligible impact on synthetic tasks ($<$ 1pp on Task A, identical on Task B). The rotation operators themselves perform the compositional work. This contrasts with kinship chain reasoning, where the trajectory encoder is critical for long-chain generalization (Section 3.3).

C.4 Kinship Chain Reasoning: Statistical Significance

All key pairwise comparisons are statistically significant at the 95% confidence level. Confidence intervals are computed using the Wilson score method for the difference in independent proportions, which is more accurate than the normal approximation for moderate sample sizes (N = 2100).

C.5 Kinship Chain Reasoning: 5-Seed State Transformer Results

Results are highly consistent across seeds (overall std = 1.1%).

C.6 Kinship Chain Reasoning: Trajectory Variant Ablation (Complete)

Analysis of trajectory variant results:

Why hop pooling helps modestly. Mean-pooled hop features $d_t = [s_t; s_{t-1}; s_t - s_{t-1}]$ capture per-hop information that is in-distribution regardless of chain length. However, mean pooling treats all hops equally, losing the ability to focus on query-relevant hops. The +5pp improvement at k=8 over the original confirms that hop features are useful, but the blunt aggregation limits the benefit.

Why query attention helps substantially. Cross-attention from the query to hop features allows the model to select which hops are most relevant for answering the specific query. This is particularly important for kinship chain reasoning, where the query asks about two specific characters and only a subset of hops involve those characters. The +16.5pp improvement at k=8 demonstrates the value of selective readout.

Why the State Transformer is best. The 2-layer Transformer over the full state trajectory can capture both local (per-hop) and global (trajectory-level) patterns through self-attention. It subsumes the hop attention mechanism while also having access to the raw state values and their ordering. The query token serves as a natural aggregation point.

Why multiscale gating fails. The gated multiscale approach learns a scalar gate $\alpha \in [0, 1]$ to blend the endpoint s_T with hop-pooled features. In principle, the gate should learn to trust hop features more at long chains. In practice, the gate converges to $\alpha \approx 1$ (trust endpoint) because the training data only contains k=2, 3 chains where s_T is in-distribution and highly informative. At test time on k=8+, the gate still trusts the now-OOD endpoint, causing worse performance than even the original model. This is a cautionary result: naive gating cannot solve the OOD readout problem because the gate itself is trained on in-distribution data.

Table 5: Synthetic task validation accuracy (mean $\pm$ std over 5 seeds).

Model	Task A	Task A Paired	Task B	Task C
Rotation n=32	0.840 ± 0.008	0.853 ± 0.010	0.964 ± 0.010	0.741 ± 0.003
LSTM h=256	0.809 ± 0.002	0.768 ± 0.003	0.909 ± 0.006	0.743 ± 0.004
Transformer	0.722 ± 0.011	0.575 ± 0.015	0.925 ± 0.003	—
Bag of Sentences	0.706 ± 0.003	0.500 ± 0.000	—	—
Commutative	0.795 ± 0.012	0.758 ± 0.015	—	0.733 ± 0.002
Unconstrained n=32	0.831 ± 0.005	0.866 ± 0.008	—	0.744 ± 0.002

Table 6: Rotation model performance across operator dimensions (test set).

n	Params	Task A		Task B			Task C	
		Overall	Paired	len=5	len=10	len=20	Acc	Inv.Err
16	182K	81.2	84.1	97.8	91.4	62.6	73.5	0.113
32	333K	82.7	85.3	98.4	92.8	65.4	74.1	0.182
64	933K	83.1	85.8	98.8	93.2	64.8	74.3	0.284
128	3.3M	82.0	83.8	100.0	95.0	72.4	73.9	0.528

C.7 Kinship Chain Reasoning: Per-Class Accuracy

The State Transformer's largest improvements over the base rotation model concentrate on relations that typically require longer reasoning chains: great-grandparent ($+20.8$pp), brother/sister-in-law ($+19.4$pp), spouse ($+13.4$pp), and cousin ($+12.0$pp). Short-chain relations (parent, child) show smaller or mixed changes. This confirms that the trajectory Transformer specifically addresses the long-chain readout problem.

C.8 LSTM Parameter Sweep

Scaling the LSTM from 395K to 1.38M parameters ($3.5\times$) does not close the Task A paired accuracy gap (75.5% vs 85.3%). The LSTM at h=384 does match rotation on Task B length-20 (65.2% vs 65.4%), but at $4\times$ the parameter cost. Task C accuracy is completely flat across LSTM sizes, confirming it is not capacity-limited.

D bAbI Natural Language Experiment

We evaluate on five bAbI tasks (Weston et al., 2015) that test different reasoning abilities. This experiment reveals a structural limitation of $\mathrm{SO}(n)$ composition.

D.1 Results

Table 13 shows test accuracy across all five bAbI tasks. The rotation model is competitive on cumulative state-tracking tasks (qa7, qa8) but underperforms LSTM on location-tracking tasks (qa1–3).

D.2 Scaling Study

To verify that the rotation model's weakness on qa1–3 is structural rather than capacity-limited, we scale both rotation (n=32, 64, 128) and LSTM (h=256, 512) models. Results are shown in Table 14.

D.3 Interpretation

The rotation model is competitive on cumulative state-tracking tasks (qa7: 74.8%, qa8: 73.4%, beating LSTM by $+2.7$pp on qa8) but underperforms on location-tracking tasks (qa1–3). The key difference: qa7–8 require tracking cumulative state changes (how many objects is someone holding?), which aligns with $\mathrm{SO}(n)$ composition that accumulates all transformations. qa1–3 ask "where is X?"—the answer depends on the *most recent* mention of a character's location, requiring selective overwriting of old information. LSTM's forget gate enables this; $\mathrm{SO}(n)$ composition cannot selectively discard past rotations.

The scaling study confirms this is structural, not a capacity limitation: increasing n from 32 to 128 ($10\times$ parameters) does not improve qa1–3 and actually slightly *worsens* qa2–3. Meanwhile, qa7–8 improve modestly with larger n. This demonstrates that the $\mathrm{SO}(n)$ inductive bias is genuinely helpful for cumulative composition but genuinely limiting for selective memory—a fundamental trade-off inherent to the algebraic structure.

Table 7: Trajectory encoder ablation on synthetic tasks (rotation n=32).

Trajectory Mode	Params	Task A Paired	Task B len=20	Task C Inv.Err
GRU (default)	333K	85.3%	65.4%	0.182
Mean pooling	306K	84.9%	63.4%	0.164
None (final state only)	298K	84.4%	65.6%	0.175

Table 8: Statistical significance of kinship chain reasoning overall accuracy comparisons (Wilson score intervals).

Comparison	Difference	95% CI	Significant
Rot+StateTF vs Rotation	+6.8pp	$(+3.8, +9.8)$	Yes
Rot+StateTF vs LSTM	+12.4pp	$(+9.4, +15.4)$	Yes
Rot+StateTF vs Transformer	+5.6pp	$(+2.6, +8.6)$	Yes
Rotation vs LSTM	+5.6pp	$(+2.6, +8.6)$	Yes
Rot+HopAttn vs Rotation	+3.8pp	$(+0.7, +6.8)$	Yes

E Evaluation on the Original CLUTRR Benchmark

To validate our approach on a standard benchmark, we evaluate on the original CLUTRR dataset (Sinha et al., 2019) with AMT-paraphrased natural language stories. Unlike the template-based kinship reasoning task in the main paper (where each sentence describes exactly one relationship hop), the original CLUTRR stories are written by Amazon Mechanical Turk workers as natural paragraphs with coreference, implicit relations, and linguistic variation.

E.1 Setup

We use the `data_db9b8f04` split (train k=2, 3, 4; test k=2, . . . , 10) with the `clean_story` field (logically ordered sentences). The encoder is all-mpnet-base-v2 (768-dim, 109M params) with 3-phase curriculum training: Phase 1 frozen encoder on k=2, Phase 2 frozen on k=2, 3, Phase 3 unfrozen on k=2, 3, 4. We add an auxiliary per-sentence relation classification loss (λ_{aux}=0.5, 14 relation types) that forces the encoder to learn relational structure. Models are larger than the main experiments: rotation n=128, LSTM h=1024. 18 answer classes (gendered relations). 3 seeds per model.

E.2 Results

All models achieve 87–100% on in-distribution chain lengths (k=2, 3, 4) but drop to 10–23% on OOD lengths ($k\geq5$). The Rotation + State Transformer achieves the best overall accuracy (34.4%) and the best k=8 performance (18.5%). The Transformer baseline is consistently worst at OOD lengths (30.6% overall), confirming that full attention without algebraic composition structure overfits to training chain lengths.

E.3 Comparison with Published Results

Direct comparison is difficult because published results use different data versions and evaluation protocols. The high accuracy of QwQ-32B (99.5%) and GPT-4o (78.8%) reported by Chen et al. (2025) is on the basic template version of CLUTRR in a zero-shot setting, where the relational structure is explicit in the text. Our evaluation uses the harder AMT-paraphrased version where relations are embedded in natural language narratives.

On the AMT version, our Rotation + State Transformer (34.4%, 112M params) is competitive with Llama-3.1-8B + FLARE (Arakelyan et al., 2025) (36.8%, 8B params) despite being $\sim$70$\times$ smaller. This suggests that the SO(n) composition structure provides an efficient inductive bias for relational chain reasoning, partially compensating for the massive parameter gap.

The key challenge on the original CLUTRR remains OOD generalization: all models (including LLMs) show significant accuracy drops beyond training chain lengths. Recent work confirms this is a fundamental limitation of current approaches (Khalid et al., 2025; Wang et al., 2025), with Wang et al. (2025) showing that LLM accuracy on CLUTRR degrades as the distance between referencing and referenced reasoning steps increases—precisely the problem our trajectory attention mechanism addresses.

F Book-Level Retrieval: Scaling to Real-World Composition

We validate the Lie group composition framework on a large-scale retrieval task: composing chapter-level embeddings into a whole-book embedding

Table 9: Per-seed kinship chain reasoning accuracy (%) for Rotation + State Transformer (corrected chain length k=2, . . . , 8).

k	Seed 42	Seed 43	Seed 44	Seed 45	Seed 46	Mean ± Std
2	41.3	42.3	39.7	41.7	40.3	41.1 ± 1.0
3	48.3	46.0	47.0	49.7	47.0	47.6 ± 1.3
4	65.7	62.0	63.3	66.3	61.7	63.8 ± 1.9
5	66.0	61.7	69.3	62.7	61.7	64.3 ± 3.0
6	66.0	65.3	66.0	65.0	66.0	65.7 ± 0.4
7	60.7	58.3	66.3	57.3	60.7	60.7 ± 3.1
8	57.0	57.3	63.7	58.3	60.0	59.3 ± 2.4
Overall	57.9	56.1	59.3	57.3	56.8	57.5 ± 1.1

Table 10: Kinship chain reasoning accuracy (%) by chain length for all trajectory variants (single seed, corrected k).

Variant	k=2	3	4	5	6	7	8	Overall
State Transformer	**41.3**	48.3	**65.7**	**66.0**	**66.0**	**60.7**	57.0	**57.9**
Hop Attention	41.0	48.3	56.0	62.3	61.7	56.3	**58.3**	54.9
Original (GRU)	38.7	**49.3**	62.7	65.0	59.0	45.7	37.3	51.1

for 55,487 real books. This experiment tests three aspects not covered by the main experiments: (1) composition over 30+ steps (vs. 2–10 in the main paper), (2) scaling to tens of thousands of training examples, and (3) the retrieval application (vs. classification in the main paper).

F.1 Task and Data

Source data. 55,487 unique books from BookCorpusOpen (17,868 books) and Project Gutenberg (61,340 books), deduplicated by book identifier. Each book is chunked into chapters via regex detection of chapter headings; books with $\leq$5 detected chapters are uniformly split into 32 segments. Median: $\sim$30 chunks per book, $\sim$2000 words per chunk.

Summary generation. Each chunk is summarized into $\leq$10 sentences using Llama 3.1 8B Instruct via vLLM on 150 parallel GPU instances. For each book, the concatenation of all chunk summaries serves as the "whole-book summary." This yields per-book: `chunk_summaries[]` (one per chapter) and `concat_summaries` (all chapters concatenated).

Embedding. All text is encoded with a frozen Qwen3-Embedding-0.6B encoder (1024-dim, 32K token context). Each chunk summary produces a 1024-dim embedding; the concatenated summary ($\sim$30 summaries $\times$ 300 words $\approx$ 9000 words) produces a single 1024-dim target embedding. Embeddings are pre-computed and stored as tensors.

Split. 44,391 train / 5,548 val / 5,548 test (80/10/10 random split, seed 42).

Task. Given a book's chapter embeddings, compose them into a single embedding and retrieve the correct whole-book target from a pool of 5,548 test books. The composition model and the target both use the same rotation operator (for rotation models) or the same embedding space (for baselines), ensuring the comparison is in the same space.

F.2 Models

All models use the same frozen Qwen3-Embedding-0.6B encoder (1024-dim). The composition mechanism varies:

- **Mean pool** (0 params): Average all chunk embeddings. Order-invariant baseline.
- **LSTM** (2.1M–17.8M params): 2-layer LSTM over chunk embeddings, final hidden state projected to 1024-dim.
- **Transformer** (25.5M–50.6M params): Self-attention over chunk embeddings with positional encoding, mean-pooled output.
- **Projected SO(m)** (3.3M–60.5M params): Our model. Each chunk embedding maps to a rotation in $\mathrm{SO}(m)$ via the Lie algebra, projected into the 1024-dim state space via learned linear maps. Sequential composition produces a state on the unit hypersphere. The same operator applied to the whole-book embedding produces the target state. Variants: with trajectory attention (StateTransformer

Table 11: Kinship chain reasoning per-class accuracy (%) for key models. Classes requiring longer chains show larger improvements from the State Transformer.

Relation	Rotation	Rot+StateTF	LSTM	Transformer
parent	41.7	48.3	5.0	56.7
child	37.0	31.5	13.0	35.2
grandparent	0.0	0.0	0.0	12.0
grandchild	32.3	24.2	33.9	19.4
sibling	13.6	22.7	0.0	18.2
aunt/uncle	48.6	45.7	34.3	48.6
niece/nephew	76.9	80.8	11.5	92.3
cousin	72.0	84.0	48.0	80.0
spouse	31.7	45.1	24.3	64.2
in-law (parent)	76.9	79.3	80.6	45.0
in-law (child)	61.0	54.2	37.3	54.2
in-law (sibling)	0.0	19.4	0.0	0.0
great-grandparent	33.3	54.2	8.3	58.3

Table 12: LSTM performance across parameter counts (synthetic tasks, test set).

Config	Params	Task A Paired	Task B len=20	Task C Acc
LSTM h=128	395K	73.2%	51.0%	74.0%
LSTM h=256	329K	76.8%	51.2%	74.2%
LSTM h=384	1.38M	75.5%	65.2%	74.2%
Rotation n=32	333K	85.3%	65.4%	74.1%

Table 13: bAbI test accuracy (%). Tasks qa1–3 require selective memory; qa7–8 require cumulative state tracking.

Model	qa1 (1-fact)	qa2 (2-fact)	qa3 (3-fact)	qa7 (count)	qa8 (lists)
Rotation n=32	48.4	30.8	24.3	74.8	73.4
LSTM h=256	**52.6**	**38.0**	**41.3**	75.7	70.7
Transformer	51.1	36.7	16.5	**78.5**	**74.5**
Bag of Sentences	43.6	31.0	27.8	68.5	51.3

Table 14: bAbI validation accuracy (%) across model scales. Rotation is flat on qa1–3 regardless of size.

Model	Params	qa1	qa2	qa3	qa7	qa8
Rotation n=32	333K	49.3	33.1	24.4	76.4	74.6
Rotation n=64	933K	49.4	31.6	23.4	76.3	75.7
Rotation n=128	3.3M	49.4	31.0	21.3	77.7	76.8
LSTM h=256	920K	54.4	40.5	42.8	76.7	74.2
LSTM h=512	3.4M	53.7	41.0	43.8	76.7	74.3

over $[s_0, \ldots, s_T]$) and without (final state s_T only).

Projected rotation operator. When the state dimension n is large (1024), full $\mathrm{SO}(n)$ requires $n(n-1)/2 = 523{,}776$ Lie algebra parameters per rotation—impractical with ∼44K training books. We instead use $\mathrm{SO}(m)$ with $m \in \{64, 128, 256\}$ and learned projections $W_{\text{down}} \in \mathbb{R}^{m\times n}$, $W_{\text{up}} \in \mathbb{R}^{n\times m}$:

$$R_{\text{full}} = I_n + W_{\text{up}}(R_m - I_m)W_{\text{down}} \qquad (4)$$

where $R_m \in \mathrm{SO}(m)$ is constructed via the standard Lie algebra parameterization. This preserves non-commutativity (the projected operators do not commute) while reducing the Lie algebra dimension to $m(m-1)/2$ (e.g., 2,016 for m=64). Intermediate states are normalized to the unit hypersphere at each composition step.

Training. InfoNCE contrastive loss with in-batch negatives plus a cosine similarity auxiliary term ($\lambda_{\cos} = 0.1$). Both the composed state and the target state are compared on the unit hypersphere. AdamW optimizer, cosine LR schedule with warmup, 15 epochs. Hyperparameter sweep over rotation dimension $m \in \{64, 128, 256\}$ and learning rate $\in \{10^{-4}, 3 \times 10^{-4}\}$ for rotation mod-

Table 15: Original CLUTRR benchmark accuracy (%) by chain length k. Train on k=2, 3, 4; test on k=2, . . . , 10. Our models use a trainable mpnet encoder (109M + composition params).

		In-distribution			Out-of-distribution			
Model	Params	k=2	k=3	k=4	k=5	k=6	k=7	k=8
Rot+StateTF n=128	112M	**100**	94.4	94.8	17.8	11.4	21.3	**18.5**
Rotation n=128	112M	94.7	86.0	88.3	18.4	12.4	**23.2**	12.6
LSTM h=1024	113M	89.5	94.4	87.0	**20.0**	**16.2**	**23.2**	14.1
Transformer 4L/8H	113M	97.4	94.4	87.0	15.1	15.2	18.7	9.6
Bag of Sentences	109M	97.4	**95.3**	**97.4**	17.3	14.3	21.9	17.8

Table 16: Comparison with published CLUTRR results. Methods vary in setup (few-shot prompting vs. fine-tuning) and data version (basic templates vs. AMT paraphrases). †Zero-shot on basic templates. ‡8-shot prompting.

Method	Model	Params	Accuracy
LLMs with prompting (no fine-tuning):			
Zero-shot text†	GPT-4o	>100B	78.8%
Zero-shot text†	QwQ-32B	32B	99.5%
Zero-shot text†	Llama-3.1-8B	8B	17.7%
CoT prompting‡	Llama-3.1-8B	8B	45.7%
FLARE‡	Llama-3.1-8B	8B	36.8%
Fine-tuned models (ours, AMT paraphrases):			
Rot+StateTF n=128	mpnet + SO(n)	112M	34.4%
Rotation n=128	mpnet + SO(n)	112M	31.9%
LSTM h=1024	mpnet + LSTM	113M	32.9%
Transformer 4L/8H	mpnet + Attn	113M	30.6%

els; hidden dimension and learning rate for baselines.

F.3 Results

Table 17 presents the main results. Three findings stand out:

All models achieve near-perfect retrieval. With 44K training books and a 5,548-book test pool, all models (except full $\mathrm{SO}(1024)$) achieve ≥97.9% R@1. The task is well-solved by all composition mechanisms, confirming that the contrastive training objective works at this scale.

Order sensitivity differentiates the models. The key metric is shuffle degradation—the drop in R@1 when chapter order is randomly permuted. Mean pool shows exactly 0% degradation (order-invariant by construction). LSTM and Transformer show <1% degradation despite being sequential models—they barely use order information for this task. The projected $\mathrm{SO}(m)$ models without trajectory show 2.8–3.7% degradation, and with trajectory attention show **10–14% degradation**. This 10–20× gap between rotation models and baselines demonstrates that the algebraic non-commutativity of $\mathrm{SO}(n)$ captures sequential structure that attention and recurrence do not.

Full $\mathrm{SO}(1024)$ collapses. When the rotation dimension matches the encoder dimension (n=1024), the model fails completely (0.16% R@1). Analysis reveals that the Lie algebra basis matrix (256 × 523,776 = 134M parameters) creates a parameter-to-sample ratio of ∼3000:1, causing the state space to collapse—33% of book pairs have cosine similarity >0.5 in state space, compared to 1.6% in the original encoder space. The projected operator avoids this by operating in a lower-dimensional subgroup ($\mathrm{SO}(64)$–$\mathrm{SO}(256)$) with far fewer parameters, maintaining a healthy parameter-to-sample ratio.

F.4 Analysis: Why Projection Works

The projected rotation operator $R_{\text{full}} = I + W_{\text{up}}(R_m - I_m)W_{\text{down}}$ can be understood as operating in a learned subspace of the full state space. The rotation $R_m \in \mathrm{SO}(m)$ acts in an m-dimensional subspace defined by W_{down}, and the result is lifted back to n dimensions by W_{up}. This is analogous to low-rank adaptation (LoRA) in fine-tuning: rather

Table 17: Book-level retrieval results (test-only pool, 5,548 books). R@k: fraction of queries where the correct target is in the top-k retrieved. Shuffle degradation: drop in R@1 when chapter order is randomly permuted (5 trials). Best per column in **bold**.

Model	Params	R@1	R@5	Cosine	MRR	Shuffle Degrad.
Baselines (comparison in encoder space):						
Mean pool	0	99.82	**100.0**	0.840	0.999	0.00
LSTM (h=512, lr=10^{-3})	5.8M	99.98	**100.0**	0.840	**1.000**	0.63
Transformer (2L/8H, lr=3×10^{-4})	25.5M	**100.0**	**100.0**	0.876	**1.000**	0.57
Projected $\mathrm{SO}(m)$ *without trajectory (comparison in state space):*						
m=64, lr=10^{-4}	3.3M	99.96	**100.0**	0.959	**1.000**	3.09
m=128, lr=10^{-4}	9.6M	99.98	**100.0**	0.958	**1.000**	3.66
m=256, lr=10^{-4}	35.0M	99.96	**100.0**	0.953	**1.000**	3.00
Projected $\mathrm{SO}(m)$ *with trajectory attention:*						
m=64, lr=10^{-4}	28.7M	99.17	**100.0**	0.960	0.996	10.38
m=128, lr=10^{-4}	35.1M	98.63	**100.0**	**0.961**	0.992	11.98
m=256, lr=10^{-4}	60.5M	97.87	99.9	0.957	0.988	**13.60**
Full $\mathrm{SO}(1024)$ *(state space collapsed—see text):*						
Full $\mathrm{SO}(1024)$ + traj	162.2M	0.16	0.97	0.798	0.012	—

than learning a full $n \times n$ transformation, we learn a rank-m perturbation of the identity.

The key insight is that the Lie group guarantees (non-commutativity, bounded composition) hold within the m-dimensional subspace, while the projection allows the model to select which subspace is most informative for the task. With m=64 (only 2,016 Lie algebra parameters), the model achieves 99.96% R@1 and 3.1% shuffle degradation—demonstrating that a small rotation group suffices when combined with learned projections.

This result also supports the framework's generality: the choice of Lie group can be adapted to the problem scale. For small state spaces (n=32–128), full $\mathrm{SO}(n)$ works directly (as in the main paper experiments). For large state spaces (n=1024), projected $\mathrm{SO}(m)$ provides a practical instantiation that preserves the algebraic structure while scaling gracefully.

G Contrast with LayerNorm

While LayerNorm (Ba et al., 2016) also projects to the unit hypersphere, it is a pointwise normalization, not a composition operator. We highlight three key differences:

Commutativity. Common aggregation strategies paired with LayerNorm remain commutative—e.g., $\mathrm{LN}(\mathrm{mean}(h_1, h_2)) = \mathrm{LN}(\mathrm{mean}(h_2, h_1))$—so they cannot distinguish orderings without additional positional encoding. $\mathrm{SO}(n)$ composition is non-commutative by construction (Proposition 4).

Invertibility. LayerNorm is non-invertible: many different inputs map to the same normalized output. $\mathrm{SO}(n)$ rotations are exactly invertible ($R^{-1} = R^\top$, Proposition 3).

Conditioning. The Jacobian conditioning of LayerNorm depends on the input distribution, while $\mathrm{SO}(n)$ guarantees $\kappa = 1$ regardless of input (Proposition 2).

Our attention+LayerNorm baseline (the Transformer in Section 3) uses $10\times$ more parameters than the rotation model yet achieves lower paired accuracy on order sensitivity (57.5% vs 85.3%), confirming that normalization alone does not provide the compositional structure of group multiplication.