IC^3: INPUT-CONDITIONED COMPLEX CONVOLUTION FOR NATURAL LANGUAGE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Input-Conditioned Complex Convolution (IC^3), an attention-free method for natural language inference. Building on a complex-valued representation that separates token and contextual information, IC^3 generates both a value sequence and a convolution kernel from each premise–hypothesis pair. The kernel determines how the values are combined across the sequence, and the real part of the complex convolution is used to update the token representations. Zero-padded Fourier transforms compute this convolution efficiently. Our analysis characterizes the resulting input-dependent operators, establishes limits on their approximation using fixed operators, and separates the effects of changes in the values and the kernel. With all models trained from scratch under a deterministic training protocol over ten random seeds, IC^3 achieves higher mean accuracy than baseline decoder architectures with comparable parameter counts on ANLI and FOLIO, with consistent gains on SNLI using reduced training data. Controlled operator comparisons and interventions further support the contributions of input conditioning and the imaginary component to predictive performance.

1 INTRODUCTION

Natural language inference (NLI) requires a model to determine the relation between a premise and a hypothesis, typically among entailment, contradiction, and neutral (Bowman et al., 2015). This task is not only a sentence-level classification problem, but also a structured sentence-pair reasoning problem, in that the model must identify local token-level cues, align related expressions across the two sentences, detect mismatches or contradictions, and compose these local semantics into an overall inference decision. Prior NLI models have therefore emphasized cross-sentence attention, local inference modeling, and inference composition (Parikh et al., 2016; Chen et al., 2017). Interaction-space models further show that dense premise–hypothesis interaction tensors contain rich semantic information for solving NLI (Gong et al., 2018). These observations motivate the joint modeling of token-level semantics and global contextual semantics to determine the relation between a premise and a hypothesis.

Most Transformer-based models represent each token with a real-valued hidden state and learn token–pair interactions implicitly through stacked self-attention layers (Vaswani et al., 2017). Although self-attention is highly effective, token-level semantic content, contextual information, and premise–hypothesis interaction signals are encoded in the same hidden representation and updated through repeated pairwise token routing. Self-attention provides a general-purpose interaction mechanism, but standard Transformer hidden states do not explicitly separate token-semantic information from complementary contextual information. Therefore, this raises a central question: **how can we design an input-conditioned update that explicitly couples token and contextual representations for premise–hypothesis relation modeling?**

Designing such an update is challenging because the relation between a premise and a hypothesis can depend on diverse token- and phrase-level semantic cues. The final label may depend on lexical alignment, phrase-level mismatch, negation, quantification, entity substitution, or condition changes. These phenomena motivate integrating token-level information with the context of the full sentence pair during representation updates. Existing interaction-based NLI models address this problem through explicit alignment or interaction spaces (Parikh et al., 2016; Gong et al., 2018), while sequential inference models compose local inference information into an overall decision (Chen et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

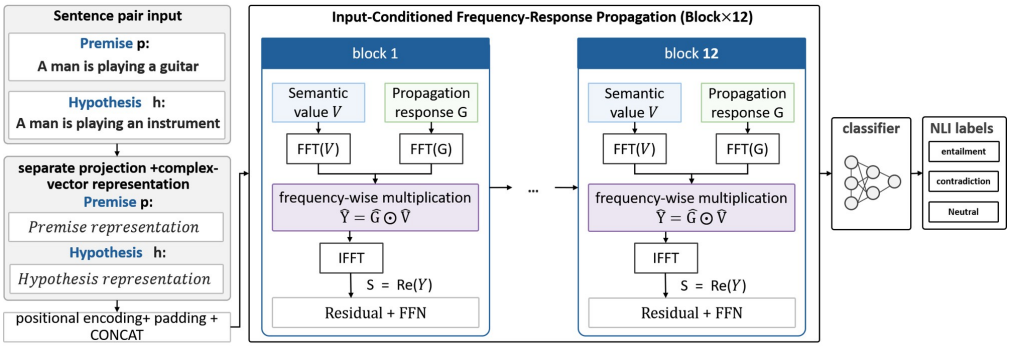


Figure 1: Input-conditioned complex spectral propagation for natural language inference.

2017). Separately, spectral methods provide efficient alternatives to self-attention, ranging from fixed Fourier mixing and learned global filters to input-dependent frequency filtering (Lee-Thorp et al., 2022; Rao et al., 2021; Huang et al., 2023). These approaches do not explicitly couple separately represented token-semantic and contextual components through the input-conditioned propagation operator. Our focus is therefore on such a structured coupling for premise–hypothesis relation modeling.

To address this problem, we propose Input-Conditioned Complex Convolution (**IC**^{3}) for premise–hypothesis relation modeling. Figure 1 provides an overview of the architecture. Given complex–vector token representations of a premise–hypothesis pair, IC^3 generates a complex semantic value sequence and an input-conditioned propagation response from the same input. Their zero-padded spectral convolution produces a token–context coupled signal whose real component updates the token-semantic representation. We adopt complex–vector token representations (Zhang & Sheng, 2026) because they explicitly separate token-semantic and complementary contextual information, providing a natural representation space for input-conditioned coupling.

This paper makes four contributions. **First**, we propose **IC**^{3}, an attention-free update layer that couples token-semantic and contextual components through input-conditioned complex convolution implemented with zero-padded Fourier transforms. **Second**, we characterize the conditional operator family, prove that the response parameterization preserves affine dimension, derive optimal low-dimensional affine approximation errors, and decompose output variation into contributions from changes in the value sequence and the operator. **Third**, we use empirical response-spectrum analysis, low-dimensional response projection, response swapping, and imaginary-response interventions to quantify variation in the learned operators and assess the contributions of input conditioning and the imaginary response to prediction. **Finally**, we compare **IC**^{3} with decoder-based baselines of comparable parameter counts, training all models from scratch under a common deterministic protocol over ten random seeds and obtaining higher mean accuracy across the five evaluated inference benchmarks.

2 Problem Formulation

Given a natural language inference instance, let $x = [p; h]$ denote the concatenated premise–hypothesis input, where p is the premise and h is the hypothesis. The goal is to predict the inference relation $y \in \{\text{entailment}, \text{neutral}, \text{contradiction}\}$. This decision may depend on token- and phrase-level semantic cues interpreted in the context of the full sentence pair.

Following Zhang & Sheng (2026), we construct Complex–Vector Token Representations separately for the premise and hypothesis before padding and concatenation. Let L and d denote the resulting sequence length and feature dimension. For $0 \le t < L$, $X_t = X_t^r + iX_t^i \in \mathbb{C}^d$, where $X_t^r, X_t^i \in \mathbb{R}^d$ carry token-semantic and complementary contextual information. The construction is given in Appendix A.

Our objective is to learn an input-conditioned propagation operator that couples these components to update the token-semantic branch for premise–hypothesis relation modeling.

3 METHOD

Given the complex–vector input representation, $\mathbf{IC}^3$ uses two branches to generate a complex semantic value sequence and an input-conditioned complex impulse response. For a fixed input, the response defines a channel-wise convolution operator on the value sequence. After zero-padding both sequences to length $2L$, we apply Fourier transforms along the token dimension and multiply the resulting coefficients element-wise. We then apply the inverse Fourier transform, retain the first L positions, and use the real part of the result to update the token-semantic branch.

3.1 SEMANTIC AND PROPAGATION RESPONSE BRANCHES

Given the complex–vector input representation $X = X^r + iX^i$, where $X^r, X^i \in \mathbb{R}^{L\times d}$, we apply separate layer normalization to the real and imaginary components: $\bar{X}^r = \mathrm{LN}_r(X^r),\ \bar{X}^i = \mathrm{LN}_i(X^i)$. Layer normalization acts along the feature dimension at each token position. The normalized components are used to construct a semantic value sequence and an input-conditioned propagation response. All projection matrices belong to $\mathbb{R}^{d\times d}$, all bias vectors belong to $\mathbb{R}^d$, and their parameters are shared across token positions. In the token-wise equations below, individual token representations are written as column vectors.

The semantic value branch produces $V = V^r + iV^i \in \mathbb{C}^{L\times d}$. For $0 \le t < L$, its components are $V_t^r = W_V^r \bar{X}_t^r + b_V^r,\ V_t^i = W_V^i \bar{X}_t^i + b_V^i$. Thus, the real and imaginary value components are obtained from the token and contextual components, respectively.

The propagation response branch produces $G = G^r + iG^i \in \mathbb{C}^{L\times d}$, with $G_t^r = \sigma(W_G^r \bar{X}_t^r + b_G^r),\ G_t^i = \tanh(W_G^i \bar{X}_t^i + b_G^i),\ 0 \le t < L$. Here, σ denotes the logistic sigmoid function, and both activation functions are applied element-wise. The entries of G^r lie in $(0, 1)$, while those of G^i lie in $(-1, 1)$.

Although G_k is generated from input position k, it is used as the vector of channel-wise coefficients for relative lag k in the subsequent convolution. The same coefficients therefore apply to all source–target pairs with that lag. Thus, G defines an input-conditioned complex impulse-response sequence. It is not applied as a pointwise token gate, and its convolutional parameterization does not represent an arbitrary source–target attention matrix.

3.2 ZERO-PADDED SPECTRAL PROPAGATION

Let $\mathcal{F}_N$ denote the length-N discrete Fourier transform (DFT) along the token dimension, applied independently to each feature channel, and let $\mathcal{F}_N^{-1}$ denote its inverse. We use an unnormalized forward DFT and an inverse DFT with normalization factor $1/N$. The semantic value sequence V and the impulse-response sequence G are zero-padded to length $N = 2L$, with $(\widetilde{V}_t, \widetilde{G}_t) = (V_t, G_t)$ for $0 \le t < L$ and $(\widetilde{V}_t, \widetilde{G}_t) = (0, 0)$ for $L \le t < N$.

The Fourier representations are $\widehat{V} = \mathcal{F}_N(\widetilde{V}),\ \widehat{G} = \mathcal{F}_N(\widetilde{G})$. Here, G is the complex impulse-response sequence in the token domain, while $\widehat{G}$ is its frequency response. Propagation is computed by $\widehat{Y} = \widehat{G} \odot \widehat{V}$, where $\odot$ denotes element-wise complex multiplication at each frequency and feature channel.

We apply the inverse DFT and retain the first L token positions: $\widetilde{Y} = \mathcal{F}_N^{-1}(\widehat{Y}),\ Y = P_L \widetilde{Y}$, where P_L denotes truncation to those positions. Since $N = 2L \ge 2L - 1$, the zero-padding prevents circular wrap-around. Consequently, Y contains the first L terms of the linear convolution of G and V: $Y_t = \sum_{\tau=0}^{t} G_{t-\tau} \odot V_\tau,\ 0 \le t < L$. For a fixed input, this defines a channel-wise one-sided convolution of the value sequence.

The real-valued signal used for the token update is $S = \mathrm{Re}(Y)$. Expanding the complex convolution gives $S_t = \mathrm{Re}(Y_t) = \sum_{\tau=0}^{t} G_{t-\tau}^r \odot V_\tau^r - \sum_{\tau=0}^{t} G_{t-\tau}^i \odot V_\tau^i,\ 0 \le t < L$. The imaginary response and value components therefore contribute to the real-valued signal used to update the token-semantic branch.

3.3 Real-Valued Token Update

For each token position $0 \le t < L$, the real-valued propagation output S_t is projected and normalized: $\tilde{S}_t = \mathrm{LN}_o(W_s S_t + b_s)$, where $W_s \in \mathbb{R}^{d\times d}$ and $b_s \in \mathbb{R}^d$ are shared across token positions. The real component is then updated through a residual connection: $R = X^r + \mathrm{Dropout}(\tilde{S})$. A position-wise feed-forward network provides a second residual update: $X^r_{\mathrm{out}} = R + \mathrm{FFN}(R)$.

The imaginary component is preserved: $X^i_{\mathrm{out}} = X^i$. The resulting complex hidden state is $X_{\mathrm{out}} = X^r_{\mathrm{out}} + iX^i_{\mathrm{out}}$. Contextual information contributes to the real-valued update through S.

3.4 Stacked Propagation Layers

The complex–vector representation is constructed once before the first propagation layer. Each layer receives the preceding complex hidden state and uses its own projections to generate $V^{(\ell)}$ and $G^{(\ell)}$. The original contextual representation b_t, phase α_t, and directional gate γ_t are not recomputed between layers. Although the imaginary component is passed unchanged through the stack, each layer applies its own projections to this component when constructing the value and response sequences.

4 Theoretical Analysis

This section analyzes the propagation operator induced by the response branch. The complex-vector construction is treated as an input representation inherited from prior work. The results below concern the structured operator family selected by the current layer input, rather than the origin of that representation. We first characterize the exact conditional operator class, then quantify the variation available to the input-conditioned family, and finally separate changes in the propagated values from changes in the propagation operator.

4.1 Conditional Propagation Operator

Let $\mathcal{X}$ denote the set of admissible layer inputs. For $X \in \mathcal{X}$, let $G(X), V(X) \in \mathbb{C}^{L\times d}$ be the propagation response and semantic value matrices defined in the preceding section. For channel $c \in \{1,\dots,d\}$, write $g_c(X) = G_{:,c}(X)$ and $v_c(X) = V_{:,c}(X)$.

Let $\mathsf{J}_L \in \mathbb{R}^{L\times L}$ be the truncated shift matrix defined by $(\mathsf{J}_L u)_t = \begin{cases} 0, & t = 0, \\ u_{t-1}, & 1 \le t < L. \end{cases}$ For a response vector $g = (g_0,\dots,g_{L-1})^\top \in \mathbb{C}^L$, define $\Phi_L(g) = \sum_{k=0}^{L-1} g_k \mathsf{J}_L^k$. The channel-wise operator induced by the current input is $A_c(X) = \Phi_L(g_c(X))$. To represent all channels jointly, let $\mathbf{v}(X) = \mathrm{vec}(V(X))$, $\mathbf{y}(X) = \mathrm{vec}(Y(X))$, $\mathbf{\Phi}_{L,d}(G) = \mathrm{diag}\big(\Phi_L(G_{:,1}),\dots,\Phi_L(G_{:,d})\big)$, and $\mathbf{A}(X) = \mathbf{\Phi}_{L,d}(G(X))$. Here, vec stacks the columns of a matrix in channel order, and diag forms a block-diagonal matrix from the listed matrix blocks. The propagation stage can then be written as $\mathbf{y}(X) = \mathbf{A}(X)\mathbf{v}(X)$.

4.2 Shift-Commuting Characterization

Define the commutant of the truncated shift by $\mathcal{C}_L = \left\{B \in \mathbb{C}^{L\times L} : B\mathsf{J}_L = \mathsf{J}_L B\right\}$, and define its channel-wise extension by $\mathcal{C}_{L,d} = \{\mathrm{diag}(B_1,\dots,B_d) : B_c \in \mathcal{C}_L\}$.

Theorem 1 (Shift-commuting characterization)**.** *The map $\Phi_L : \mathbb{C}^L \to \mathcal{C}_L$ is a complex-linear isomorphism. Equivalently, a matrix $B \in \mathbb{C}^{L\times L}$ commutes with J_L if and only if there is a unique vector $g \in \mathbb{C}^L$ such that $B = \sum_{k=0}^{L-1} g_k \mathsf{J}_L^k$. Such a matrix is lower triangular and Toeplitz, and its action on $u \in \mathbb{C}^L$ is $(Bu)_t = \sum_{\tau=0}^{t} g_{t-\tau} u_\tau$, for $0 \le t < L$. Consequently, $\mathbf{\Phi}_{L,d} : \mathbb{C}^{L\times d} \to \mathcal{C}_{L,d}$ is also a complex-linear isomorphism, and $\dim_{\mathbb{C}}(\mathcal{C}_{L,d}) = Ld$. In particular, every target operator map $\mathbf{A}_{\mathrm{tar}} : \mathcal{X} \to \mathcal{C}_{L,d}$ admits a unique response map $G_{\mathrm{tar}} : \mathcal{X} \to \mathbb{C}^{L\times d}$ satisfying $\mathbf{A}_{\mathrm{tar}}(X) = \mathbf{\Phi}_{L,d}(G_{\mathrm{tar}}(X))$ for every $X \in \mathcal{X}$.*

Theorem 1 establishes a bijection between unrestricted complex response sequences and channel-wise operators that commute with the truncated shift. The implemented response generator selects a subset of this class through its parameterization and bounded activations; the theorem does not

establish that the generator can realize every operator or every input-dependent operator map. Since $\Phi_L(g)u$ gives the first L entries of the linear convolution of two length-L sequences, zero-padding to any transform length at least $2L-1$ yields an exact Fourier implementation. The choice $N = 2L$ used by $\mathbf{IC}^3$ therefore computes the propagation stage in $O(dL \log L)$ time and $O(dL)$ working storage, excluding the feature projections and feed-forward layers.

Lower triangularity describes the conditional routing map from $V(X)$ to $Y(X)$ after $G(X)$ has been generated. It does not make the complete layer causal, because both $G(X)$ and $V(X)$ may depend on the entire premise–hypothesis input.

4.3 Dimension of the Conditional Operator Family

We next quantify how much the selected operator can vary with the input. We measure affine dimension over $\mathbb{R}$ by jointly representing the real and imaginary components of each complex response as a real vector. This convention matches the real-valued coefficients used in the affine approximations below.

Fix a reference input $X_\circ \in \mathcal{X}$. Let $\mathcal{G} = \{G(X) : X \in \mathcal{X}\}$ denote the family of input-conditioned propagation responses, and let $\mathcal{A} = \{\mathbf{A}(X) : X \in \mathcal{X}\}$ denote the corresponding family of conditional propagation operators. Define $r_G = \dim_{\mathbb{R}} \operatorname{span}_{\mathbb{R}}\{G(X) - G(X_\circ) : X \in \mathcal{X}\}$, which is the real affine dimension of the response family $\mathcal{G}$. Equivalently, r_G is the minimum number of fixed response directions needed to express all differences $G(X) - G(X_\circ)$ as linear combinations with real coefficients. For the quantitative statement, define $\mathsf{Q}_L = \operatorname{diag}(\sqrt{L}, \sqrt{L-1}, \ldots, 1)$ and the weighted realification map $\omega(H) = \begin{bmatrix} \operatorname{vec}(\operatorname{Re}(\mathsf{Q}_L H)) \\ \operatorname{vec}(\operatorname{Im}(\mathsf{Q}_L H)) \end{bmatrix} \in \mathbb{R}^{2Ld}$. Let $\boldsymbol{\xi}(X) = \omega(G(X))$. When X is random, assume $\mathbb{E}\|\boldsymbol{\xi}(X)\|_2^2 < \infty$, let $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{\xi}(X)]$, and define $\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{\xi}(X) - \boldsymbol{\mu})(\boldsymbol{\xi}(X) - \boldsymbol{\mu})^\top\right]$. Let $q = 2Ld$ and let $\lambda_1 \geq \cdots \geq \lambda_q \geq 0$ be the eigenvalues of $\boldsymbol{\Sigma}$. For an operator $C \in \mathcal{C}_{L,d}$ and a nonempty real affine subspace $\mathcal{U} \subseteq \mathcal{C}_{L,d}$, define $\operatorname{dist}_{\mathrm{F}}(C, \mathcal{U}) = \inf_{D \in \mathcal{U}} \|C - D\|_{\mathrm{F}}$, where D ranges over the operators in $\mathcal{U}$ and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Theorem 2 (Dimension preservation and finite-dictionary approximation)**.** *The response family and the induced operator family have the same real affine dimension:* $\operatorname{affdim}_{\mathbb{R}}(\mathcal{A}) = \operatorname{affdim}_{\mathbb{R}}(\mathcal{G}) = r_G$. *Moreover, the weighted coordinate map is isometric with respect to the operator Frobenius norm:* $\|\boldsymbol{\Phi}_{L,d}(H_1) - \boldsymbol{\Phi}_{L,d}(H_2)\|_{\mathrm{F}}^2 = \|\omega(H_1) - \omega(H_2)\|_2^2$ *for all* $H_1, H_2 \in \mathbb{C}^{L \times d}$.

For every integer K *satisfying* $0 \leq K < q$, *the best* K*-dimensional affine approximation of the random operator family satisfies* $\inf_{\substack{\mathcal{U} \subseteq \mathcal{C}_{L,d},\ \mathcal{U} \text{ affine} \\ \operatorname{affdim}_{\mathbb{R}}(\mathcal{U}) \leq K}} \mathbb{E}\left[\operatorname{dist}_{\mathrm{F}}(\mathbf{A}(X), \mathcal{U})^2\right] = \sum_{j=K+1}^{q} \lambda_j$. *Consequently, any dictionary-conditioned operator of the form* $\widetilde{\mathbf{A}}(X) = B_0 + \sum_{j=1}^{K} \beta_j(X) B_j$, *where* $B_0, B_1, \ldots, B_K \in \mathcal{C}_{L,d}$ *are fixed and* $\beta_j(X) \in \mathbb{R}$, *obeys* $\mathbb{E}\left[\|\mathbf{A}(X) - \widetilde{\mathbf{A}}(X)\|_{\mathrm{F}}^2\right] \geq \sum_{j=K+1}^{q} \lambda_j$.

Thus, an approximation using K *fixed operator directions cannot represent* $\mathbf{A}(X)$ *exactly for all* $X \in \mathcal{X}$ *when* $r_G > K$. *Similarly, an affine mixture of* $K \geq 1$ *fixed expert operators has the form* $\mathbf{A}_{\mathrm{mix}}(X) = \sum_{j=1}^{K} \alpha_j(X) B_j$, *where* $\alpha_j(X) \in \mathbb{R}$ *and* $\sum_{j=1}^{K} \alpha_j(X) = 1$. *Its range has real affine dimension at most* $K - 1$, *so it cannot represent the entire conditional operator family when* $r_G > K - 1$.

Corollary 3 (Empirical response spectrum)**.** *Let* $X_1, \ldots, X_{n_{\mathrm{s}}}$ *be observed inputs, let* $\overline{\boldsymbol{\xi}} = \frac{1}{n_{\mathrm{s}}} \sum_{i=1}^{n_{\mathrm{s}}} \boldsymbol{\xi}(X_i)$, *and form the centered matrix* $\mathbf{M}_\xi \in \mathbb{R}^{n_{\mathrm{s}} \times q}$ *whose* i*-th row is* $(\boldsymbol{\xi}(X_i) - \overline{\boldsymbol{\xi}})^\top$. *If* $s_1 \geq \cdots \geq s_{\min\{n_{\mathrm{s}}, q\}} \geq 0$ *are the singular values of* $\mathbf{M}_\xi$, *then, for every integer* $0 \leq K < q$, *the minimum average squared Frobenius error over real affine operator subspaces of dimension at most* K *is* $\frac{1}{n_{\mathrm{s}}} \sum_{j=K+1}^{\min\{n_{\mathrm{s}}, q\}} s_j^2$, *where the sum is interpreted as zero when* $K \geq \min\{n_{\mathrm{s}}, q\}$. *In particular,* $\operatorname{rank}(\mathbf{M}_\xi)$ *is the empirical affine dimension of the sampled conditional operator family.*

4.4 Value and Operator Variation

For a response matrix $G \in \mathbb{C}^{L \times d}$, define its maximum channel-wise ℓ_1 norm by $\|G\|_1 = \max_{1 \leq c \leq d} \sum_{k=0}^{L-1} |G_{k,c}|$.

Table 1: Performance comparison of $\mathbf{IC}^3$ and same-scale baseline models over 10 seeds.

Model	Metric	ANLI-R1	ANLI-R2	ANLI-R3	FOLIO	SNLI
IC^3	Acc %	**39.01**	**46.08**	**44.87**	**50.34**	**63.86**
	Δ CI	—	—	—	—	—
Llama 3.2	Acc %	36.96	43.61	42.48	45.22	59.15
	Δ CI	2.05 [0.70,3.40]	2.47 [1.63,3.31]	2.38 [1.04,3.72]	5.12 [2.16,8.09]	4.71 [4.30,5.11]
Phi 1.5	Acc %	37.66	43.57	41.74	44.19	61.80
	Δ CI	1.35 [0.12,2.58]	2.51 [1.50,3.52]	3.12 [2.36,3.88]	6.16 [3.27,9.04]	2.06 [1.76,2.37]
Pythia	Acc %	36.09	39.07	37.22	37.59	59.83
	Δ CI	2.92 [2.22,3.62]	7.01 [5.06,8.96]	7.65 [4.77,10.53]	12.76 [9.81,15.70]	4.03 [3.56,4.50]
Gemma 3	Acc %	36.77	41.51	41.73	43.69	60.99
	Δ CI	2.24 [1.46,3.02]	4.57 [3.84,5.30]	3.14 [2.09,4.19]	6.65 [4.22,9.08]	2.87 [2.19,3.55]

Acc denotes accuracy. Δ denotes the mean paired accuracy difference between $\mathbf{IC}^3$ and each baseline over ten matched random seeds. Brackets report two-sided 95% paired Student's t confidence intervals. All values are percentages.

Theorem 4 (Value–operator decomposition and stability)**.** *For any $X, X' \in \mathcal{X}$, $\mathbf{y}(X') - \mathbf{y}(X) = \mathbf{A}(X')\big(\mathbf{v}(X') - \mathbf{v}(X)\big) + \big(\mathbf{A}(X') - \mathbf{A}(X)\big)\mathbf{v}(X)$. The first term is the value-variation term under the operator selected by X', whereas the second term is the operator-variation term acting on the value sequence from X. The decomposition implies $\|\mathbf{y}(X') - \mathbf{y}(X)\|_2 \le \|G(X')\|_1\|\mathbf{v}(X') - \mathbf{v}(X)\|_2 + \|G(X') - G(X)\|_1\|\mathbf{v}(X)\|_2$. If the response is input independent, so that $G(X) = G_0$ for all X, then $\mathbf{A}(X) = \mathbf{A}_0$ and the operator-variation term vanishes.*

Corollary 5 (Response-swap intervention)**.** *Define the swapped-response output by $\mathbf{y}(X \mid X') = \mathbf{A}(X')\mathbf{v}(X)$. Then $\mathbf{y}(X \mid X') - \mathbf{y}(X \mid X) = \big(\mathbf{A}(X') - \mathbf{A}(X)\big)\mathbf{v}(X)$. Thus, swapping the response while holding the value sequence fixed isolates the operator-variation term in Theorem 4 exactly at the propagation output.*

4.5 Relation to Existing Sequence Operators

The preceding results clarify the relation between IC^3 and several existing operator constructions. A static convolution or fixed spectral mixer uses an input-independent propagation operator. Its operator family therefore has affine dimension zero, and the operator-variation term in Theorem 4 vanishes. Conditional combinations of fixed kernels, as in dynamic convolution (Chen et al., 2020), restrict operator variation to the affine space determined by those kernels. Theorem 2 quantifies the approximation error when the IC^3 operator family is represented using finitely many fixed directions in $\mathcal{C}_{L,d}$. Direct filter generation, as in Dynamic Filter Networks (Jia et al., 2016), also produces input-dependent operators, so input conditioning alone does not distinguish IC^3.

IC^3 combines input conditioning with a channel-wise, one-sided convolution structure. For each fixed input, the propagation coefficients within each channel are shared across token pairs with the same lag, yielding a lower-triangular Toeplitz operator that commutes with J_L. Position-dependent dynamic convolution (Wu et al., 2019) and dense attention (Vaswani et al., 2017) need not satisfy this constraint. Consequently, the IC^3 propagation operator is specified by Ld complex lag coefficients, whereas an unrestricted channel-wise complex source–target operator requires L^2d coefficients. These counts describe the conditional propagation operators, not the numbers of trainable model parameters.

5 Results

Our experiments evaluate predictive performance and the role of input-conditioned propagation through benchmark comparisons, architectural ablations, response-spectrum analysis, response projection, response swapping, imaginary-response interventions, and input-order controls. We use FP32 training with deterministic PyTorch and CUDA settings, with TF32 and cuDNN benchmarking disabled. For each architecture and dataset, learning rate and dropout are selected by mean validation performance over three seeds, followed by evaluation on seven additional seeds. We report ten-seed test means and paired 95% Student's t CIs for model comparisons, and across-seed

Table 2: Additional ANLI-R1 comparison with encoder architectures over ten matched seeds.

Model	Params	Acc. (%)	Diff [95% CI]
IC^3	0.57B	**39.01**	—
ModernBERT-large	0.40B	35.25	3.76 [1.74, 5.78]
DeBERTa-v2-xlarge	0.89B	33.30	5.71 [5.02, 6.40]
RoBERTa-large	0.36B	33.30	5.71 [5.02, 6.40]

Table 3: Operator comparison over ten matched seeds. Each baseline entry reports accuracy followed by the paired Δ and its 95% confidence interval relative to IC^3.

Propagation operator	ANLI-R1	ANLI-R2	ANLI-R3
IC^3 propagation	**39.01**	**46.08**	**44.87**
FNet-style mixing	37.95; 1.06 $[0.04, 2.08]$	42.83; 3.25 $[2.45, 4.05]$	43.53; 1.34 $[0.72, 1.95]$
Orchid-based mixing	38.31; 0.70 $[-0.55, 1.95]$	42.98; 3.10 $[1.82, 4.38]$	41.53; 3.33 $[2.27, 4.40]$
AFF	36.51; 2.50 $[1.58, 3.42]$	43.00; 3.08 $[2.42, 3.74]$	41.13; 3.73 $[2.66, 4.80]$

means and SDs ($\pm$) for intervention analyses unless otherwise noted. All models are trained on a GB200/GH200 platform with 128GB memory and 8 CPU cores.

Since **IC^3** is trained end-to-end from scratch, we compare it against same-scale decoder-based baseline architectures. Our model uses 12 layers with a 1536-dimensional embedding size, resulting in 0.57B parameters. To ensure a comparable parameter budget, we reduce the number of layers in the 1B baseline architectures and compare all models at approximately the same parameter scale. The baselines are adapted from Llama 3.2-1B (Grattafiori et al., 2024), Phi-1.5 (Li et al., 2023), Pythia 1B (Biderman et al., 2023), and Gemma 3-1B (Gemma Team, 2025). All compared models are trained using matched random seeds, deterministic training, and the same validation-based hyperparameter selection protocol.

We evaluate on ANLI-R1, ANLI-R2, and ANLI-R3 (Nie et al., 2020) as the main adversarial NLI benchmarks, FOLIO (Han et al., 2024) for logically structured natural-language reasoning, and SNLI (Bowman et al., 2015) as supplementary directional validation. Due to computational constraints, SNLI uses 5% of the training set and 10% of the validation set, with evaluation on the full test set; it is therefore not reported as a full-data SNLI benchmark.

Main results: Table 1 shows that **IC^3 consistently outperforms all same-scale decoder-based baselines across all five evaluated inference datasets**. The positive confidence intervals of Δ indicate that the gains are stable under the ten-seed evaluation protocol. The largest advantages appear on FOLIO and the later ANLI rounds, indicating that the gains are most pronounced on the more challenging inference settings in our evaluation. The SNLI result further suggests that the trend extends to a larger NLI dataset under a reduced-data training setting.

Additional encoder comparison. Table 2 extends the ANLI-R1 evaluation to randomly initialized encoder architectures. IC^3 achieves a mean accuracy of 39.01%, compared with 35.25% for ModernBERT-large (Warner et al., 2025) and 33.30% for both DeBERTa-v2-xlarge (He et al., 2021) and RoBERTa-large (Liu et al., 2019). The paired 95% confidence intervals exclude zero for all three comparisons. Because these encoder architectures differ in parameter count from IC^3, we report them as additional architecture-level comparisons rather than parameter-matched baselines.

Controlled operator comparison: Table 3 compares IC^3 with FNet-style (Lee-Thorp et al., 2022), Orchid-based (Karami & Ghodsi, 2024), and AFF (Huang et al., 2023) mixing under the same ten-seed protocol. IC^3 achieves higher mean accuracy than all three alternatives on ANLI-R1, ANLI-R2, and ANLI-R3. All paired 95% confidence intervals exclude zero except for Orchid-based mixing on ANLI-R1.

Ablations: Table 4 shows that replacing the proposed propagation mechanism, removing the complex token–context representation, or restricting the response substantially degrades performance. In particular, the static-response and imaginary-response ablations support the functional importance

Table 4: Component ablations on ANLI-R2

Variant	Acc %	Δ CI
Full **IC**^{3}	46.08	—
w/ token–context attention	42.82	3.26 [2.27, 4.25]
w/ real-valued input	39.29	6.79 [4.65, 8.93]
w/ static response	40.99	5.09 [4.29, 5.89]
w/o imaginary response	39.94	6.14 [4.88, 7.40]

Table 5: Layer-0 interventions on ANLI-R2

Intervention	Acc. drop (pp)
Mean response ($K = 0$)	9.58 ± 1.50
Random response swap	10.86 ± 1.96
Same-label response swap	8.21 ± 1.77
Remove imaginary response	12.27 ± 1.98
Reverse imaginary response	20.59 ± 1.34

of input-conditioned propagation and complex cross-branch coupling. Taken together, these results support the combined IC^3 design under the evaluated setting.

5.1 Mechanistic Validation of the Conditional Operator Analysis

The preceding ablations establish that input conditioning and the imaginary response are useful at the architectural level. We next examine whether the specific operator degrees of freedom characterized in Section 4 are actually used by the trained model. We focus on three questions: whether the learned conditional operator family remains genuinely multidimensional, whether this variation is functionally relevant to prediction, and whether the imaginary-response term identified by the complex decomposition contributes to the resulting token update.

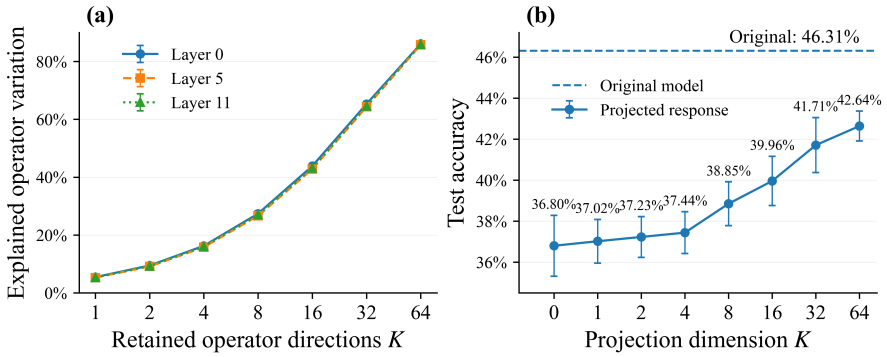


Figure 2: Geometry and task relevance of the learned conditional operator family.

Response variation across principal components. Figure 2(a) reports the empirical response spectrum characterized by Corollary 3. At Layer 0, explaining 90% of the observed variation requires 81.3 ± 2.5 principal components across seeds, with similar spectra at Layers 5 and 11. The observed response variation is therefore distributed over many principal components at the examined layers.

Predictive performance under response projection. Figure 2(b) shows accuracy rising from $(36.80 \pm 1.48)\%$ at $K = 0$ to $(42.64 \pm 0.73)\%$ at $K = 64$, versus $(46.31 \pm 0.75)\%$ for the seven-seed fixed-length reference. In the separate layer-wise $K = 0$ screening analysis, the Layer-0 intervention reduces accuracy by 9.58 ± 1.50 percentage points. The absolute mean accuracy changes for Layers 1–11 do not exceed 0.06 percentage points. Under these single-layer interventions, accuracy is most sensitive to input-dependent responses at Layer 0.

Response swapping tests input–operator matching. Corollary 5 shows that response swapping isolates the operator-variation term at the propagation output while holding the value sequence fixed at the intervened layer. As summarized in Table 5, replacing the Layer-0 response with that generated by another input reduces accuracy by 10.86 ± 1.96 percentage points, and a substantial degradation remains when the donor input has the same label. Response swapping in later layers has little effect on accuracy. The same-label result suggests that predictive performance depends on matching the Layer-0 response to the input, even when the donor has the same class label.

The imaginary response contributes to the real-valued update. The complex decomposition shows that the imaginary response and value components contribute jointly to the real-valued propagation output. Removing the Layer-0 imaginary response reduces accuracy by 12.27 ± 1.98 percentage points, while reversing its sign produces an even larger degradation, as shown in Table 5. These interventions provide evidence that the trained model uses the imaginary contribution for prediction. Together, the spectrum, projection, response-swap, and imaginary-response results provide empirical support for the conditional operator analysis in Section 4.

Effect of input order: Under the ten-seed evaluation protocol, the premise–hypothesis and hypothesis–premise input orders achieve mean accuracies of 46.08% and 45.82% on ANLI-R2, respectively. The mean paired difference is 0.26 percentage points, with a 95% confidence interval of $[-0.79, 1.31]$. The difference is not statistically significant, although this result does not establish order invariance.

6 Related Work

Natural language inference and structured sentence-pair reasoning. NLI benchmarks such as SNLI, MultiNLI, and ANLI evaluate entailment, contradiction, and neutral relations under increasingly challenging settings (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020), while LogicNLI and FOLIO emphasize logically structured reasoning (Tian et al., 2021; Han et al., 2024). Neural approaches include token alignment in Decomposable Attention (Parikh et al., 2016), local inference composition in ESIM (Chen et al., 2017), dense premise–hypothesis interaction modeling (Gong et al., 2018), and pretrained cross-encoders such as BERT and DeBERTa (Devlin et al., 2019; He et al., 2021).

Attention-free and spectral sequence operators. Structured alternatives to self-attention include S4 (Gu et al., 2022), Mamba (Gu & Dao, 2024), RetNet (Sun et al., 2024), Gated Linear Attention (Yang et al., 2024), Hyena (Poli et al., 2023), and lightweight or dynamic convolutions (Wu et al., 2019). Spectral token mixing includes FNet (Lee-Thorp et al., 2022), GFNet (Rao et al., 2021), AFNO (Guibas et al., 2022), Caracal (Gan et al., 2026), and Adaptive Frequency Filtering (Huang et al., 2023).

Complex-valued representations. Complex-valued neural models exploit magnitude and phase structure (Lee et al., 2022). In language modeling and matching, CNM uses complex semantic representations (Li et al., 2019), EulerFormer applies polar-form complex representations (Tian et al., 2024), and Complex–Vector Token Representations separate token-semantic and complementary contextual information (Zhang & Sheng, 2026).

Input-conditioned operators. Dynamic Filter Networks generate filters from the current input (Jia et al., 2016), HyperNetworks generate parameters for another network (Ha et al., 2016), and dynamic convolution combines kernels using input-dependent weights (Chen et al., 2020). These approaches establish input-conditioned parameterization as a general mechanism for adaptive computation.

IC^3 differs from fixed spectral mixers such as FNet and GFNet by generating its propagation response from the current premise–hypothesis input. Unlike general dynamic filtering, it restricts the conditional operator to channel-wise one-sided convolution with coefficients shared across equal relative lags. Relative to Complex–Vector Token Representations with Token–Context Attention, it retains the token–context decomposition but replaces attention-based routing with zero-padded spectral convolution.

7 Conclusion

We introduced **IC^3**, an attention-free architecture for input-conditioned complex propagation over premise–hypothesis representations. Our analysis characterizes its shift-commuting operator family and separates value from operator variation. Experiments show consistent gains over same-scale baselines and confirm that the learned operator family is multidimensional, input-dependent, and functionally reliant on the imaginary cross-branch contribution.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this paper, to support the retrieval and discovery of related work, and to assist in checking and revising portions of the mathematical arguments and proofs, and to support the verification of portions of the experimental code. All AI-assisted writing was independently verified by the authors against the stated assumptions, derivations, and reported results, and all references identified with AI assistance were checked against the original publications. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The supplementary material provides the source code and complete experimental configurations used to reproduce the experiments and numerical results reported in this paper.

REFERENCES

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075/.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668, 2017.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11030–11039, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Bingzheng Gan, Tianyi Zhang, Yusu Li, Jing Huang, Wei Shi, Yangkai Ding, and Tao Yu. Caracal: Causal architecture via spectral mixing. *arXiv preprint arXiv:2605.00292*, 2026.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Yichen Gong, Heng Luo, and Jian Zhang. Natural language inference over interaction space. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1dHXnH6-.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EXHG-A3jlM.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, et al. Folio: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, 2024.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Zheng-Jun Zha, Yan Lu, and Baining Guo. Adaptive frequency filters as efficient global token mixers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6026–6036. IEEE, 2023.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

Mahdi Karami and Ali Ghodsi. Orchid: Flexible and data-adaptive convolution for sequence modeling. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=44VDxQwWXN.

ChiYan Lee, Hideyuki Hasegawa, and Shangce Gao. Complex-valued neural networks: A comprehensive survey. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1406–1426, 2022.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. FNet: Mixing tokens with Fourier transforms. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4296–4313, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.319. URL https://aclanthology.org/2022.naacl-main.319/.

Qiuchi Li, Benyou Wang, and Massimo Melucci. Cnm: An interpretable complex-valued network for matching. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4139–4148, 2019.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL https://api.semanticscholar.org/CorpusID:198953378.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://aclanthology.org/2020.acl-main.441/.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://aclanthology.org/D16-1244/.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2024. URL https://openreview.net/forum?id=UU9Icwbhin.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. Diagnosing the first-order logical reasoning ability through logicnli. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3738–3747, 2021.

Zhen Tian, Wayne Xin Zhao, Changwang Zhang, Xin Zhao, Zhongrui Ma, and Ji-Rong Wen. Eulerformer: Sequential user behavior modeling with complex vector attention. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1619–1628, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.127. URL https://aclanthology.org/2025.acl-long.127/.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Xin Zhang and Victor S Sheng. Token-context attention for nli: An alternative to self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34836–34844, 2026.

A Complex–Vector Token Representation

Following Zhang & Sheng (2026), we apply the complex–vector construction separately to the projected premise and hypothesis sequences; the derivation below applies to either sequence. For the t-th token, let $a_t \in \mathbb{R}^d$ denote its real-valued token representation. The global semantic

vector is computed dimension-wise as $g = \sqrt{\sum_s a_s \odot a_s}$, and the contextual representation of token t is constructed from the remaining tokens as $b_t = \sqrt{\sum_{s \neq t} a_s \odot a_s}$, where $\odot$ denotes element-wise multiplication. The token–global alignment phase is defined dimension-wise by $\alpha_t = \operatorname{atan2}\left(\sqrt{1 - \left(\frac{a_t}{g}\right)^2}, \frac{a_t}{g}\right)$. This phase characterizes the angular relation between the token representation and the global semantic vector. The polar-form complex representation uses the adapted magnitude $g + a_t$ and the phase α_t: $Z_t = (g + a_t) \odot e^{i\alpha_t}$. After converting Z_t to Cartesian form, the real and imaginary components can be written as $X_t^r = \gamma_t \odot a_t, X_t^i = \gamma_t \odot b_t$, where $\gamma_t = \cos(\alpha_t) + 1$ is the directional gate. Thus, the complex-vector token representation used in this paper is $X_t = X_t^r + iX_t^i$.

B Proofs of Theoretical Results

This appendix provides the proofs of the theoretical results stated in Section 4, following the same order as in the main text.

B.1 Proof of Theorem 1

Proof. Every polynomial in J_L commutes with J_L, so $\Phi_L(g) \in \mathcal{C}_L$ for every $g \in \mathbb{C}^L$. Conversely, let $B \in \mathcal{C}_L$ and let $e_0, \ldots, e_{L-1}$ be the standard basis of $\mathbb{C}^L$. Since $e_t = \mathsf{J}_L^t e_0$, commutation gives $Be_t = B\mathsf{J}_L^t e_0 = \mathsf{J}_L^t B e_0$. Write $Be_0 = \sum_{k=0}^{L-1} g_k e_k$. Then $Be_t = \sum_{k=0}^{L-1-t} g_k e_{k+t}$, which shows that $B = \sum_{k=0}^{L-1} g_k \mathsf{J}_L^k$. The coefficients are unique because they form the first column of B. Since Φ_L is linear by definition, it is a complex-linear isomorphism. The convolution formula follows by direct multiplication. Applying the same argument independently to each channel proves the block-diagonal statement and the dimension formula. The final claim follows from the bijectivity of $\mathbf{\Phi}_{L,d}$. □

B.2 Proof of Theorem 2

Proof. The map $\mathbf{\Phi}_{L,d}$ is injective and real-linear, so it preserves affine independence and therefore preserves affine dimension. For a single channel, the coefficient at lag k appears on a subdiagonal containing $L - k$ entries. Hence $\|\Phi_L(h)\|_{\mathrm{F}}^2 = \sum_{k=0}^{L-1} (L-k)|h_k|^2$. Summing over channels gives the stated isometry.

The isometry reduces affine approximation of $\mathbf{A}(X)$ under the Frobenius norm to affine approximation of $\boldsymbol{\xi}(X)$ under the Euclidean norm. The optimal affine subspace contains the mean $\boldsymbol{\mu}$, and its optimal K directions are the eigenvectors associated with the largest K eigenvalues of $\mathbf{\Sigma}$. The residual expected squared distance is therefore $\sum_{j=K+1}^{q} \lambda_j$.

The range of $\widetilde{\mathbf{A}}$ is contained in the affine space $B_0 + \operatorname{span}_{\mathbb{R}}\{B_1, \ldots, B_K\}$, whose affine dimension is at most K. Applying the optimal affine-approximation identity proves the lower bound. The exact-representation statement follows from affine-dimension preservation. For an affine mixture of $K \geq 1$ fixed expert operators with real coefficients summing to one, the range lies in their real affine hull, whose dimension is at most $K - 1$. This proves the corresponding exact-representation statement for affine mixtures. □

B.3 Finite-Expert Mixture Extension

For $K \geq 1$, if a model instead mixes K fixed experts as $\widetilde{\mathbf{A}}_{\mathrm{mix}}(X) = \sum_{j=1}^{K} \beta_j(X) E_j, \quad E_1, \ldots, E_K \in \mathcal{C}_{L,d}, \quad \beta_j(X) \geq 0, \quad \sum_{j=1}^{K} \beta_j(X) = 1$, then its operator family has affine dimension at most $K - 1$, and $\mathbb{E}\left[\|\mathbf{A}(X) - \widetilde{\mathbf{A}}_{\mathrm{mix}}(X)\|_{\mathrm{F}}^2\right] \geq \sum_{j=K}^{q} \lambda_j$. Thus, a K-expert affine mixture cannot exactly represent the conditional operator family whenever $r_G > K - 1$.

Proof. The range of a K-expert affine mixture is contained in the affine hull of $E_1, \ldots, E_K$, whose affine dimension is at most $K - 1$. Applying the optimal affine-approximation identity in Theorem 2 proves the lower bound. The exact-representation statement follows from affine-dimension preservation. □

B.4 Proof of Corollary 3

Proof. Apply Theorem 2 to the empirical distribution supported uniformly on $X_1, \ldots, X_{n_s}$ and use the singular-value characterization of the best rank-K approximation. □

B.5 Proof of Theorem 4

Proof. Add and subtract $\mathbf{A}(X')\mathbf{v}(X)$ to obtain the exact decomposition.

For each channel, extend g and u by zero outside $\{0, \ldots, L-1\}$. Young's convolution inequality and the fact that truncation does not increase the ℓ_2 norm give $\|\Phi_L(g)u\|_2 \le \|g\|_1\|u\|_2$. Since $\mathbf{\Phi}_{L,d}(G)$ is block diagonal, it follows that $\|\mathbf{\Phi}_{L,d}(G)\|_{2\to 2} \le \max_{1\le c\le d}\|G_{:,c}\|_1 = \|G\|_1$. By linearity, $\mathbf{A}(X') - \mathbf{A}(X) = \mathbf{\Phi}_{L,d}\big(G(X') - G(X)\big)$. Applying the operator-norm bound to $G(X')$ and $G(X') - G(X)$, and then using the triangle inequality in the decomposition, proves the stated stability bound.

If $G(X) = G_0$ for all X, then $\mathbf{A}(X) = \mathbf{\Phi}_{L,d}(G_0)$ is constant. Hence $\mathbf{A}(X') - \mathbf{A}(X) = 0$, and the operator-variation term vanishes. □

B.6 Proof of Corollary 5

Proof. By definition, $\mathbf{y}(X \mid X) = \mathbf{A}(X)\mathbf{v}(X)$ and $\mathbf{y}(X \mid X') = \mathbf{A}(X')\mathbf{v}(X)$. Subtracting gives $\mathbf{y}(X \mid X') - \mathbf{y}(X \mid X) = \big(\mathbf{A}(X') - \mathbf{A}(X)\big)\mathbf{v}(X)$. The value sequence is identical in both outputs, so their difference is exactly the operator-variation term in Theorem 4. □