CCMAMBA: RANK-AWARE STATE-SPACE MODELING FROM CAUSAL SCANS TO INCIDENCE-STRUCTURED MIXING ON COMBINATORIAL COMPLEXES

Anonymous authors
Paper under double-blind review

ABSTRACT

Selective state space models (SSMs) rely on ordered scans, whereas combinatorial complexes, a general framework encompassing graphs, hypergraphs, simplicial complexes, and cellular complexes, have no canonical ordering of their cells. Imposing a linearization, as in existing Graph Mamba approaches, can introduce order dependence because causal selective scans induce position-dependent, block lower-triangular mixing. We study this mismatch by separating the support of a mixing operator from its factorized content-dependent interactions, and introduce **Combinatorial Complex Mamba (CCMamba)**, a rank-aware framework that combines incidence-based cross-rank lifting with two complementary structured mixers. CCMamba-S preserves bidirectional selective SSM scans and efficient fused sequence processing, while remaining sensitive to the chosen within-rank ordering. CCMamba-K instead replaces causal sequence support with incidence-supported, index-free structural gating and factorized kernel mixing, providing permutation-equivariant mixing without requiring a cell linearization. We characterize the order sensitivity of causal scans, establish the permutation equivariance and structural factorization of CCMamba-K, and give a conditional characterization of its expressive power under explicit assumptions. Experiments across graph, hypergraph, simplicial-complex, and cellular-complex benchmarks show that CCMamba achieves competitive predictive performance, with memory and runtime advantages over the evaluated attention-based baselines in several higher-order settings. Code: https://anonymous.4open.science/r/CCMamba.

1 INTRODUCTION

Topological relational learning has emerged as an important paradigm for understanding complex systems. Real-world systems ranging from biomolecular interaction networks to traffic dynamics Truong & Chin (2024), image manifolds Love et al. (2023), and multi-scale geometric structures Du et al. (2026) exhibit relational patterns that extend far beyond pairwise interactions. While graphs provide a natural representation for pairwise interactions Pham et al. (2025), they do not explicitly capture the multi-rank relations found in hypergraphs Chen et al. (2025), simplicial complexes Wu et al. (2023a), and cellular complexes. This limitation motivates the adoption of combinatorial complexes (CCs). CCs bridge the gap between cell complexes and hypergraphs Hajij et al. (2023); Krahn et al. (2026) as illustrated in Figure 1(a), and CCs organize entities into ranked cells, such as nodes, edges, and faces, whose relations are encoded through incidence structure Hajij et al. (2022); Besta et al. (2025); Hajij et al. (2025). This makes CCs a natural domain for developing models that operate across different higher-order structures.

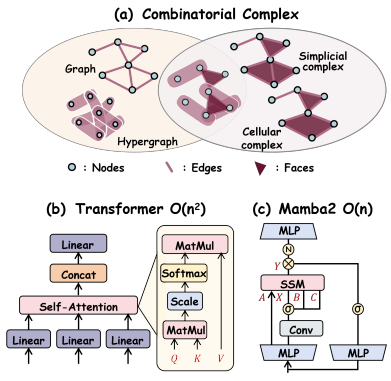


Figure 1: (a) Combinatorial complexes. (b) standard attention mechanisms, and (c) selective state-space models.

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.

A rich family of higher-order graph neural architectures has been proposed Akbari et al. (2026). Hypergraph neural networks typically focus on node-hyperedge interactions and do not explicitly represent the rank-stratified incidence relations Feng et al. (2019); Ding et al. (2020); Pellegrin et al. (2025), typically through two-stage aggregation schemes Ding et al. (2023), but fail to model higher-order and multi-rank dependencies (e.g., edge–face interactions). Beyond hypergraphs, parallel works have integrated topological priors via algebraic topology, introducing complex-specific neural networks such as UniGNN Huang & Yang (2021), CCNNs Hajij et al. (2022; 2023), MPSNs Bodnar et al. (2021a), HOMP Eitan et al. (2025), and simplicial or cellular neural networks Roddenberry et al. (2021); Lecha et al. (2025); Eijkelboom et al. (2023); Madhu & Chepuri (2023); Battiloro et al. (2024). These methods are often tied to specific structural types, remain iterative and local Naitzat et al. (2020); Wen et al. (2024); Hajij et al. (2025). Recent attempts to capture long-range, structured dependencies have extended attention and transformer-style architectures to graph domains Papamarkou et al. (2024a); Rampasek et al. (2022), and higher-order graph domains Deng et al. (2024); Lee et al. (2022); Gurugubelli & Chepuri (2024); Papillon et al. (2025). However, they suffer from the quadratic complexity of self-attention, limiting their scalability on large-scale graphs in Figure 1(b). This motivates a unified and efficient structured-mixing framework for higher-order relational data.

Recently, selective state-space models (SSMs), particularly Mamba (Figure 1(c)), have emerged as efficient alternatives for modeling long-range dependencies with linear-time complexity Dao & Gu (2024); Bick et al. (2024); Yuan et al. (2025); Lahoti et al. (2026). However, extending SSMs to topological domains is nontrivial because their mixing mechanism is inherently defined over ordered sequences Chennuru Vankadara et al. (2024); Montagna et al. (2025). Existing Graph Mamba methods Wang et al. (2024a); He et al. (2025; 2026) typically impose an ordering on graph elements before applying a selective scan. A causal selective scan induces a block lower-triangular mixing pattern whose interactions depend on positions in the resulting sequence. A combinatorial complex, in contrast, is defined by incidence relations among ranked cells and does not provide a canonical ordering of those cells. Consequently, different enumerations of the same complex can induce different causal mixing patterns, making selective-scan representations order-dependent and, in general, not permutation equivariant. This exposes the main question of this work: *Can the support structure of efficient sequence mixing be reformulated over incidence-defined higher-order relations without relying on an arbitrary cell ordering?*

Our main contributions are summarized as:

- We introduce Combinatorial Complex Mamba (CCMamba), a rank-aware framework that views structured mixing through the separation of interaction support and content-dependent mixing. It applies uniformly to graphs, hypergraphs, simplicial complexes, and cellular complexes.
- We analyze selective state-space mixing and the structural symmetries of combinatorial complexes. We show that causal selective scans are generally order-sensitive, and develop an incidence-aware alternative that replaces sequence-induced causal support with incidence-supported, index-free learned structural gating and factorized kernel mixing.
- We instantiate this framework with two complementary encoders. *CCMamba-S* retains bidirectional selective state-space scans for efficient sequence processing, while *CCMamba-K* realizes a structured kernel mixer with incidence-defined support that removes dependence on an imposed linearization. Experiments across graph and higher-order benchmarks show competitive performance with favorable memory and runtime efficiency.

2 PRELIMINARIES

Combinatorial Complexes A combinatorial complex $\mathcal{CC}$ generalizes graphs, hypergraphs, simplicial and cellular complexes into a unified algebraic-topological structure. A combinatorial complex is a triple $(\mathcal{S}, \mathcal{C}, \mathrm{rk})$ where $\mathcal{S}$ is a finite set of vertices, $\mathcal{C} \subseteq 2^{\mathcal{S}}$ is a collection of cells, $\mathrm{rk} : \mathcal{C} \to \mathbb{Z}_{\geq 0}$ is an order-preserving rank function such that if $\sigma \subseteq \tau$, then $\mathrm{rk}(\sigma) \leq \mathrm{rk}(\tau)$.

The elements of $\mathcal{C}$ are called cells. A cell $\sigma \in \mathcal{C}$ with rank $\mathrm{rk}(\sigma) = k$ is referred to as a k-cell. In particular, 0-cells, 1-cells, and 2-cells correspond to nodes, edges, and faces, respectively. The dimension of CCs is defined as $\dim(\mathcal{C}) := \max_{\sigma \in \mathcal{C}} \mathrm{rk}(\sigma)$. To construct higher-order domains from graphs, we adopt a *lifting operator* $f : \mathcal{G} \to \mathcal{CC}$, which maps an input graph to a higher-order

topological structure Akbari et al. (2026); Telyatnikov et al. (2024); Franco et al. (2026). Details of existing lifting methods are provided in Appendix D.1; Algorithm D.1 gives an example of cycle lifting from graphs to combinatorial complexes.

Definition 1 (Lifting Operation)**.** *Let $\mathcal{G}$ be the class of graphs, a lifting operation is a function $f : \mathcal{G} \to \mathcal{CC}$, which lifts a graph $G \in \mathcal{G}$ to a higher-order graph obtained from G by a prescribed construction, including hypergraph, simplicial complexes, and cellular complexes.*

Combinatorial Complex Neural Networks (CCNNs) CCNNs generalize classical message-passing frameworks to combinatorial complexes (CCs) Hajij et al. (2023; 2022); Papillon et al. (2025); Battiloro et al. (2025). A CCNN layer consists of neighbourhood-specific message construction, intra-neighbourhood aggregation, and inter-neighbourhood fusion. Let $\mathcal{N}_{\mathcal{C}}$ denote a set of neighbourhoods, each defining a relational operator on the combinatorial complex $\mathcal{CC}$. For each cell $\sigma \in \mathcal{C}$ with rank $\mathrm{rk}(\sigma)$ and feature $h_\sigma^l \in \mathbb{R}^{F^l}$ at layer l, the update is

$$h_\sigma^{l+1} = \phi\left(h_\sigma^l, \bigotimes_{\mathcal{N}\in\mathcal{N}_{\mathcal{C}}} \bigoplus_{\tau\in\mathcal{N}(\sigma)} \psi_{\mathcal{N},\,\mathrm{rk}(\sigma)}(h_\sigma^l, h_\tau^l)\right), \tag{1}$$

where $\bigoplus$ aggregates messages within each neighbourhood, $\bigotimes$ fuses information across neighbourhood types, $\psi_{\mathcal{N},k}$ is a neighbourhood-specific message function, and ϕ is the update function.

Rank-Aware neighbourhood on Combinatorial Complexes. Let $\mathcal{CC} = (\mathcal{S}, \mathcal{C}, \mathrm{rk})$ be a 2-dimensional combinatorial complex. We consider rank-coupled neighbourhoods induced by incidence matrices $I_1 \in \mathbb{R}^{|V_0|\times|E_1|}$ and $I_2 \in \mathbb{R}^{|E_1|\times|F_2|}$, encoding node–edge and edge–face incidences, respectively. They induce four message-passing neighbourhoods: node-to-edge $\mathcal{N}_{0\to1}(e)$, edge-to-node $\mathcal{N}_{1\to0}(v)$, face-to-edge $\mathcal{N}_{2\to1}(e)$, and edge-to-face $\mathcal{N}_{1\to2}(f)$. Let $I_k \in \mathbb{R}^{n_{k-1}\times n_k}$ denote the incidence matrix between $(k-1)$- and k-cells, and $H_k \in \mathbb{R}^{n_k\times d}$ the rank-k feature matrix. Messages between adjacent ranks are aggregated using I_k or $I_k^\top$; for example, node-to-edge and edge-to-node aggregation are given by $I_1^\top H_0$ and $I_1 H_1$, respectively.

Figure 2 shows the incidence operators aggregate information from incident cells across ranks. Decomposing higher-order interactions into rank-specific channels produces rank-aware structural representations while preserving incidence-based aggregation over nodes, edges, and faces.



Figure 2: Rank-aware neighbourhood.

Rank-Aware CCNNs. Given a target cell σ with rank $k = \mathrm{rk}(\sigma)$, let $h_\sigma^{(l)}$ denote its feature at layer l, and let $\tilde{h}_\tau^{(l)} = W_r h_\tau^{(l)}$ be the rank-specific projection of a source cell τ of rank r. Based on the rank-aware neighbourhoods, we rewrite the l-th CCNN layer in equation 1 as

$$h_\sigma^{(l+1)} = \phi_1\left(h_\sigma^{(l)}, \phi_2\left(\phi_3\left(h_\sigma^{(l)}, \{\tilde{h}_\tau^{(l)}\}_{\tau\in\mathcal{N}_{r\to k}(\sigma)}\right)\right)_{r\in\mathcal{R}(\sigma)}\right), \tag{2}$$

where $\mathcal{R}(\sigma) \subseteq \{0, 1, 2\}$ denotes the set of incident source ranks and $\mathcal{N}_{r\to k}(\sigma)$ the rank-r neighbourhood of the target rank-k cell σ. Here, ϕ_3 performs neighbourhood-specific encoding, ϕ_2 fuses information across incident ranks, and ϕ_1 updates the cell representation. The resulting rank-aware aggregations provide the incidence-lifted representations for subsequent intra-rank mixing.

State Space Models State-space models (SSMs) can be viewed as linear dynamical systems Dao & Gu (2024); Lahoti et al. (2026) that map an input sequence $x(t) \in R^L$ to the embeddings $y(t) \in R^L$. SSM employs a latent state $h(t) \in R^N$, parameter matrices $A \in R^{N\times N}$ and $B \in R^{N\times 1}, C \in R^{1\times N}$, In practice, continuous-time state-space models are discretized for sequence modeling. defined as $h_t = \bar{A}\cdot h_{t-1} + \bar{B}x_t, y(t) = Ch(t)$, where $\bar{A} = \exp(\Delta A)$, and $\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I)\cdot \Delta B$. The selective mechanism of SSMs propagates information by updating the latent state as

$$y_j = \sum_{i=0}^{j} \mathbf{M}_{ji} x_i, \qquad \mathbf{M}_{ji} = \mathbf{C}_j^\top \left(\prod_{t=i+1}^{j} \bar{\mathbf{A}}_t\right) \bar{\mathbf{B}}_i. \tag{3}$$

where the empty product for $i = j$ is defined as the identity matrix. The coefficients $\mathbf{M}_{ji}$ are a path product along the scan order, where they sit relative to each other in the enumeration. On sequences this is desired, but on a combinatorial complex, where no canonical order exists, it becomes a liability.

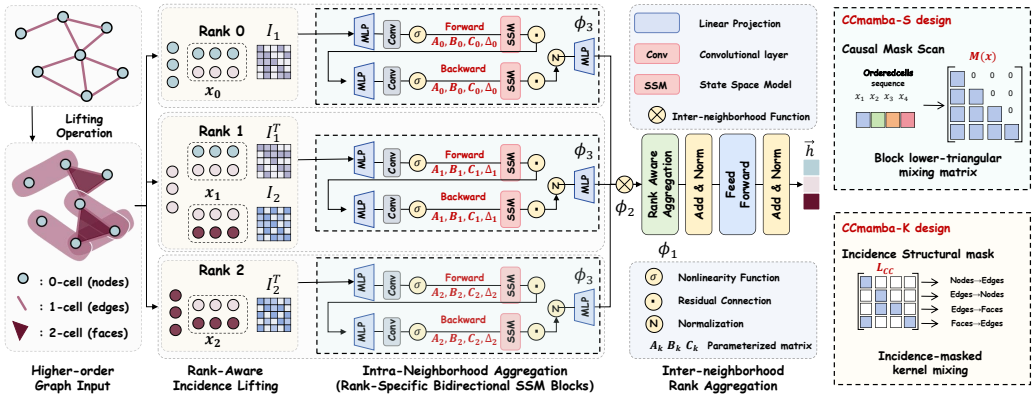


Figure 3: Overview of CCMamba. Rank-aware incidence lifting constructs multi-rank representations, which are processed by complementary structured mixers and followed by inter-rank incidence fusion. CCMamba-S applies rank-wise BiMamba scans, CCMamba-K performs incidence-supported structural gating and factorized kernel mixing without imposing a cell linearization.

3 Selective State Space Models on Combinatorial Complexes

Selective state space models operate on ordered sequences He et al. (2026); Wang et al. (2024a), whereas combinatorial complexes consist of cells of multiple ranks connected by incidence relations. As illustrated in Figure 3, CCMamba combines incidence lifting with structured mixing and inter-rank fusion. In terms of equation 2, we keep ϕ_2 fixed, redesign ϕ_1 (Sec. 3.3), and instantiate ϕ_3 using two complementary mixing mechanisms. CCMamba-S performs rank-wise selective scans over the lifted features, whereas CCMamba-K defines mixing directly over incidence-supported structural neighbourhoods. Incidence lifting determines the cross-rank information available to each cell before mixing, while ϕ_3 determines how the lifted representations interact.

3.1 Incidence-Aware Rank Lifting

Each CCMamba layer maintains rank-specific representations $H_0^{(l)}$, $H_1^{(l)}$, and $H_2^{(l)}$ for nodes, edges, and faces. Before any mixing occurs, we form the incidence-lifted input at rank k by concatenating a cell's own features with the messages gathered from its adjacent ranks,

$$X_k^{(l)} \;=\; \big[\, H_k^{(l)} \,\|\, I_k^\top H_{k-1}^{(l)} \,\|\, I_{k+1} H_{k+1}^{(l)} \,\big] W_k^{(l)}, \tag{4}$$

where $W_k^{(l)}$ is a rank-specific linear projection, $\|$ denotes channel-wise concatenation, and terms involving out-of-range ranks are omitted. Equation 4 implements three coupled message flows: edge-to-node propagation through $I_1 H_1^{(l)}$, node- and face-to-edge propagation through $I_1^\top H_0^{(l)}$ and $I_2 H_2^{(l)}$, and edge-to-face propagation through $I_2^\top H_1^{(l)}$.

3.2 Rank-Aware Structured Mixing

Bidirectional Selective Scan (CCMamba-S). For each rank k, CCMamba-S organizes the incidence-lifted cell representations according to a rank-aware ordering and applies a bidirectional selective state-space encoder,

$$H_{\text{intra},k}^{(l+1)} = \phi_3^{\mathrm{S}}\big(X_k^{(l)}\big) = \mathrm{MLP}_k^{(l)}\left(\mathrm{BiMamba}_k^{(l)}\big(X_k^{(l)}\big)\right). \tag{5}$$

Here, $\mathrm{BiMamba}_k^{(l)}$ combines selective scans over the forward and reversed rank-k sequences, allowing information to propagate along both scan directions. The resulting representation is then projected by a rank-specific MLP. This mitigates the directional bias of a single causal scan while retaining the computational structure of selective sequence processing. Since the input features of combinatorial complexes admit no canonical ordering, CCMamba-K defines the mixing coefficients directly from structural incidence relations, its properties are analyzed in Section 4.

3.3 INTER-RANK FUSION AND REFINEMENT

The structured mixer ϕ_3 produces cell representations from the incidence-lifted inputs. CCMamba-S performs within-rank bidirectional selective scans, whereas CCMamba-K mixes over incidence-supported structural neighbourhoods. A shared inter-rank fusion step then aggregates representations from each incident source rank to update the target cell. This stage is defined as follows:

$$h^{(l+1)}_{\mathrm{inter},\sigma} = \phi_1\Big(h^{(l)}_{\sigma},\ \mathrm{MLP}\big(\big\|_{r\in\mathcal{R}(\sigma)} \mathrm{Combine}_r(\sigma)\big)\Big), \qquad \mathrm{Combine}_r(\sigma) = \sum_{\tau\in\mathcal{N}_{r\to k}(\sigma)} h^{(l+1)}_{\mathrm{intra},\tau}, \quad (6)$$

where $k = \mathrm{rk}(\sigma)$, $\|$ concatenates over ranks, and ϕ_1 is a non-linear update, $\mathrm{Combine}_r$ is an unnormalized sum function (Proposition 5). Concatenating over r keeps per-rank contributions and stacking $h^{(l+1)}_{\mathrm{inter},\sigma}$ into rank-level matrices $H^{(l+1)}_{\mathrm{inter},k}$, a residual block and FFN stabilize optimization:

$$\begin{aligned} \widehat{H}^{(l+1)}_k &= \mathrm{LayerNorm}\big(H^{(l)}_k + \mathrm{Dropout}(H^{(l+1)}_{\mathrm{inter},k})\big), \\ H^{(l+1)}_k &= \mathrm{LayerNorm}\big(\widehat{H}^{(l+1)}_k + \mathrm{FFN}(\widehat{H}^{(l+1)}_k)\big), \end{aligned} \quad (7)$$

for $k \in \{0,1,2\}$. The resulting representations are refined by the post-processing block, and the final embeddings $\{H^{(L)}_k\}_{k=0}^{2}$ are pooled for downstream tasks.

4 ANALYSIS OF THE CCMAMBA LAYER

We analyze CCMamba in terms of permutation equivariance and expressive power. We first show that causal selective scans do not guarantee permutation equivariance under arbitrary reorderings. We then analyze CCMamba-K, which replaces sequence-induced causal support with index-free structural gating and factorized kernel interactions, yielding permutation-equivariant mixing.

Proposition 1 (Order sensitivity of selective state-space mixing)**.** *Let $f(\mathbf{x}) = \mathbf{M}(\mathbf{x})\mathbf{x}$ denote the sequence-to-sequence mapping induced by a causal selective state-space model, where $\mathbf{x} = [x_0, \ldots, x_{T-1}]^\top$ and $\mathbf{M}(\mathbf{x})$ is block lower-triangular with*

$$\mathbf{M}_{ji}(\mathbf{x}) = \mathbf{C}_j^\top \left(\prod_{t=i+1}^{j} \bar{\mathbf{A}}_t\right) \bar{\mathbf{B}}_i, \qquad i \le j, \quad (8)$$

and $\mathbf{M}_{ji} = 0$ for $i > j$. Permutation equivariance under a permutation matrix $\mathbf{P}$ requires $f(\mathbf{P}\mathbf{x}) = \mathbf{P}f(\mathbf{x})$. This property is not guaranteed by the causal selective-scan architecture: there exist valid model parameters, inputs, and permutations for $f(\mathbf{P}\mathbf{x}) \neq \mathbf{P}f(\mathbf{x})$. Hence, permutation equivariance under arbitrary token reorderings is not guaranteed by the causal selective-scan architecture.

Incidence-Masked Kernel Mixing (CCMamba-K). Motivated by Proposition 1, causal selective scans are not permutation-equivariant under arbitrary reorderings (Appendix D.4). We define mixing coefficients from structural invariants rather than enumeration positions as the order-free structural alternative. We next show that the resulting layer is permutation-equivariant and admits an incidence-supported factorization in which structural coefficients modulate factorized kernel interactions.

Definition 2. *Let $\mathcal{CC} = (\mathcal{S}, \mathcal{C}, \mathrm{rk})$ be a combinatorial complex, where $\tau \prec \sigma$ denotes that τ is an immediate face of σ. Define $\partial\sigma = \{\tau \in \mathcal{C} : \tau \prec \sigma\}$ and $\mathrm{co}\partial\sigma = \{\gamma \in \mathcal{C} : \sigma \prec \gamma\}$. For cells $\sigma, \tau \in \mathcal{C}$, let $\zeta(\sigma) = \big(\mathrm{rk}(\sigma), |\partial\sigma|, |\mathrm{co}\partial\sigma|\big)$, and pairwise invariants be $\iota(\sigma,\tau) = \big(\mathbf{1}[\sigma \prec \tau], \mathbf{1}[\tau \prec \sigma], |\sigma \cap \tau|, |\gamma : \sigma \prec \gamma \succ \tau|\big)$. These quantities depend only on local incidence structure and are invariant under incidence-preserving relabeling π, i.e., $\zeta(\pi\sigma) = \zeta(\sigma)$ and $\iota(\pi\sigma, \pi\tau) = \iota(\sigma,\tau)$.*

Definition 3. *Let $\mathcal{N}$ be a rank-aware neighbourhood family satisfying $\mathcal{N}(\pi\sigma) = \pi(\mathcal{N}(\sigma))$ under any rank- and incidence-preserving relabeling π. We call $\{a_{\sigma\tau}\}$* index-free structural *if $a_{\sigma\tau} = \alpha(\zeta(\sigma), \zeta(\tau), \iota(\sigma,\tau), h_\sigma, h_\tau)\,\mathbf{1}[\tau \in \mathcal{N}(\sigma)]$, where ζ and ι are defined in Definition 2. Then $a_{\pi\sigma,\pi\tau} = a_{\sigma\tau}$, so the coefficients depend on structural relations and features rather than cell indices.*

Definition 3 specifies order-independent structural coefficients for incidence-supported kernel mixing. In practice, $a_{\sigma\tau}$ is implemented by a shared structural-selective gate based on incidence-derived quantities and equivariantly transformed cell features; details are given in Remark 3. Unlike CCMamba-S, which applies sequence mixing within each rank, CCMamba-K operates directly on the incidence-supported neighbourhood $\mathcal{N}(\sigma)$ (for a rank-k cell σ, $\mathcal{N}(\sigma) = \mathcal{N}_{k-1\to k}(\sigma) \cup \mathcal{N}_{k+1\to k}(\sigma)$).

Definition 4 (Incidence-masked kernel mixing layer)**.** *Let $\psi : \mathbb{R}^d \to \mathcal{H}_\psi$ be a feature map into an inner-product space, and let $q_\sigma = \mathbf{W}_Q X_\sigma$, $k_\tau = \mathbf{W}_K X_\tau$, $v_\tau = \mathbf{W}_V X_\tau$ be computed from the incidence-lifted input $X^{(l)}$ of equation 4. Given index-free structural coefficients $\{a_{\sigma\tau}\}$, the* incidence-masked kernel mixing layer *computes the structural aggregates*

$$\mathbf{S}_\sigma = \sum_{\tau \in \mathcal{N}(\sigma)} a_{\sigma\tau}\, \psi(k_\tau) \otimes v_\tau, \qquad \mathbf{z}_\sigma = \sum_{\tau \in \mathcal{N}(\sigma)} a_{\sigma\tau}\, \psi(k_\tau), \tag{9}$$

$$h^{(l+1)}_{\mathrm{intra},\sigma} = \phi_3^{\mathrm{K}}\big(X^{(l)}\big)_\sigma = \Big[\, \psi(q_\sigma)^\top \mathbf{S}_\sigma \,\Big\|\, \mathbf{z}_\sigma \,\Big], \tag{10}$$

where $\|$ denotes channel-wise concatenation. The first block is the query-contracted, value-weighted response; the second carries the aggregate neighbourhood embedding uncontracted*.*

Lemma 1 (Permutation equivariance)**.** *Let $\pi : \mathcal{CC}_1 \to \mathcal{CC}_2$ be a rank- and incidence-preserving bijection with permutation matrix $\mathbf{P}_\pi$. If $\{a_{\sigma\tau}\}$ is index-free structural, then $\mathcal{F}(\mathbf{P}_\pi \mathbf{X}; \mathcal{CC}_2) = \mathbf{P}_\pi\, \mathcal{F}(\mathbf{X}; \mathcal{CC}_1)$. In particular, the layer is equivariant to automorphisms when $\mathcal{CC}_1 = \mathcal{CC}_2$.*

Lemma 2 (Exact factorization)**.** *On its first block of equation 10, CCMamba-K induces the structural mixing matrix $\mathbf{M}_{\sigma\tau} = a_{\sigma\tau}\, \psi(q_\sigma)^\top \psi(k_\tau)$. Let $\mathbf{L}_{\mathcal{CC}}(\cdot)$ denote the matrix with $[\mathbf{L}_{\mathcal{CC}}(\cdot)]_{\sigma\tau} = a_{\sigma\tau}$, and let $\mathbf{\Psi}_Q$ and $\mathbf{\Psi}_K$ stack the query and key feature maps. Then $\mathbf{M} = \mathbf{L}_{\mathcal{CC}} \circ (\mathbf{\Psi}_Q \mathbf{\Psi}_K^\top)$. The resulting Hadamard product $\mathbf{M}$ itself need not be low rank and $\mathbf{L}_{\mathcal{CC}}(\mathbf{X})$ has incidence-defined support.*

CCMamba-K replaces sequence-induced causal support with incidence-supported, index-free structural gating while parameterizing content-dependent interactions through a factorized kernel. Under the conditions of Lemma 1, this removes dependence on an imposed cell linearization and yields permutation-equivariant mixing. Unlike the order-sensitive CCMamba-S, CCMamba-K aggregates over unordered incidence neighbourhoods and therefore admits a multiset-based analysis. The unnormalized kernel aggregate $\sum_{\tau \in \mathcal{N}(\sigma)} \psi(k_\tau)$ preserves multiplicities under an appropriate injective kernel. The upper-bound characterization applies to the implemented layer under the coefficient restrictions and related experiments are stated in Appendix B.4.

5 Experiments

We evaluate CCMamba on graph, hypergraph, simplicial, and cellular benchmarks for node- and graph-level tasks Liu et al. (2025); Kipf (2016). All models are implemented in PyTorch. Experiments were conducted on six NVIDIA RTX 6000 GPUs (48 GB each). Dataset statistics, evaluation protocols, baseline details, and implementation settings are provided in Appendix C. Our experiments address four research questions: **RQ1:** How does CCMamba perform across graph-lifted and native higher-order benchmarks? **RQ2:** How does each module within CCMamba affect its performance? **RQ3:** What impact do different hyperparameter settings have on CCMamba? **RQ4:** How efficiently does CCMamba scale compared with attention-based models?

5.1 Comparison to state-of-the-art methods (RQ1)

CCMamba on Graph Benchmarks. We evaluate CCMamba on a diverse collection of real-world graph benchmarks. For graph-origin benchmarks, we construct hypergraph, simplicial-complex, and cellular-complex representations using fixed lifting procedures. Within each topological domain, CCMamba and the corresponding baselines use the same lifting functions Telyatnikov et al. (2024) (Appendix D.2). For graph classification, we consider TUDatasets Morris et al. (2020a), including MUTAG, PROTEINS, IMDB-B, and IMDB-M. For node classification, we include Roman-Empire, Amazon-Ratings, and Minesweeper from the Heterophilic Graph Dataset Platonov et al. (2023), together with Cora, CiteSeer, and PubMed Yang et al. (2016). Table 1 shows that CCMamba-S and CCMamba-K achieve competitive performance across both graph- and node-level tasks. Under identical cellular-complex settings, CCMamba-K improves over CWN by **5.77%** and **5.80%** relatively on MUTAG and PROTEINS, respectively. On the graph domain, CCMamba-S achieves a **3.20%** relative improvement over MbaGCN on MUTAG. For node classification, CCMamba-K (cellular) achieves 89.96% on Cora and 77.79% on CiteSeer, while CCMamba-S (cellular) achieves 89.29% on PubMed. Notably, the cellular instantiation yields the highest mean accuracy among the four topological variants for CCMamba-S (**76.41%**) and CCMamba-K (**76.04%**). Moreover, CCMamba-S improves from 79.75% in the graph domain to 85.11% in the cellular domain on

Table 1: Test accuracy (%) of CCMamba and baseline methods on graph benchmarks. Best results (**bold**) and second-best scores (underlined). Label $*$ indicates with the lifting operation.

	Model	Graph Level Tasks				Node Level Tasks						Mean
		MUTAG	PROTEINS	IMDB-B	IMDB-M	ROMAN-E	AMAZON-R	Minesweeper	Cora	Citeseer	PubMed	
Graph	GCN Kipf (2016)	70.21 ±2.43	72.76 ±1.78	70.80 ±2.04	49.33 ±4.52	78.10 ±1.15	49.84 ±3.71	83.48 ±0.69	87.00 ±2.11	75.51 ±3.47	89.48 ±1.29	72.65
	GAT Veličković et al. (2018)	72.34 ±1.69	74.91 ±2.34	68.40 ±3.73	46.93 ±4.25	83.25 ±1.82	49.29 ±3.94	83.84 ±1.01	86.71 ±2.25	73.59 ±3.77	89.21 ±1.25	72.85
	GIN Xu et al. (2018)	78.72 ±2.61	76.70 ±1.87	71.20 ±1.86	46.40 ±4.07	79.92 ±2.20	49.22 ±3.58	83.60 ±0.37	87.59 ±1.92	73.11 ±4.08	**89.51** ±0.94	73.60
	Graph-Mamba Wang et al. (2024a)	72.34±3.49	**77.78** ±4.46	70.13 ±2.08	46.27 ±3.11	78.84±0.41	48.46±0.93	**86.44** ±0.40	**88.18** ±1.86	77.79 ±2.28	89.45±0.38	73.57
	GraphGPS Rampasek et al. (2022)	78.85±3.31	76.70±1.88	72.35 ±2.60	48.83 ±2.71	79.17±0.37	46.99±0.76	85.80 ±0.44	86.41±0.97	**78.27** ±1.92	88.92±0.33	74.23
	MbaGCN He et al. (2025)	77.28 ±5.13	73.98 ±2.59	70.57 ±2.34	47.44 ±3.49	82.36 ±1.68	50.03 ±3.15	84.49 ±0.58	87.79 ±0.60	76.68 ±0.96	89.32 ±0.24	73.99
	CCMamba*-S(graph)	79.75 ±2.40	77.42 ±4.12	**73.20** ±3.56	**52.81** ±3.78	**83.46** ±2.40	**52.92** ±2.85	85.52 ±1.10	88.12 ±0.93	76.11 ±2.60	**89.51** ±1.87	**75.88**
	CCMamba*-K(graph)	**80.85** ±3.36	74.62 ±1.80	68.32 ±1.68	49.97 ±3.50	82.38 ±0.87	45.37 ±0.60	85.14 ±1.08	81.51 ±1.37	71.43 ±1.66	89.05 ±0.40	72.86
Hypergraph	HyperGCN Yadati et al. (2019)	75.53 ±2.05	73.82 ±2.17	67.60 ±2.38	45.87 ±4.18	76.40 ±1.92	48.91 ±3.66	80.24 ±1.27	86.35 ±4.26	72.41 ±2.04	88.35 ±0.97	71.55
	HyperGAT Ding et al. (2020)	68.51 ±12.04	70.25 ±2.08	70.08 ±2.38	34.67 ±2.08	71.56 ±0.39	48.11 ±0.16	79.82 ±0.49	85.05 ±1.03	72.45 ±1.83	82.87 ±0.39	68.34
	UniGNNII Huang & Yang (2021)	**82.98** ±1.46	73.48 ±2.68	71.20 ±2.03	47.20 ±3.72	73.00 ±2.98	48.75 ±3.50	80.72 ±1.55	**87.59** ±1.73	74.79 ±3.08	88.59 ±0.77	72.83
	UniGIN Huang & Yang (2021)	79.23 ±4.81	71.26 ±2.09	68.48 ±2.17	32.69 ±1.06	73.29 ±0.28	48.75 ±0.43	80.36 ±0.79	84.64 ±0.78	72.53 ±1.09	83.08 ±0.41	69.43
	UniSAGE Huang & Yang (2021)	78.72 ±1.43	74.55 ±3.76	71.60 ±4.72	34.67 ±2.74	74.53 ±3.81	49.70 ±0.48	80.85 ±0.62	85.55 ±0.76	73.78 ±1.39	83.08 ±0.41	70.70
	AllSet Chien et al. (2021)	74.47 ±2.55	**76.70** ±1.97	64.40 ±2.92	47.47 ±3.81	77.52 ±2.38	**50.02** ±3.27	81.88 ±1.49	83.60 ±2.83	74.31 ±3.29	89.05 ±1.75	71.94
	EdGNN Jo et al. (2021)	80.85 ±1.63	74.91 ±2.26	70.40 ±2.48	**49.53** ±4.15	**79.99** ±1.78	48.38 ±3.85	83.12 ±0.99	84.44 ±2.04	**75.03** ±3.19	89.39 ±1.15	73.60
	CCMamba*-S(hypergraph)	78.72 ±2.10	76.67 ±2.31	**72.40** ±2.66	46.40 ±3.92	78.52 ±1.83	49.53 ±2.23	**83.48** ±1.05	**87.59** ±3.16	**75.03** ±4.16	**89.53** ±2.63	**73.79**
	CCMamba*-K(hypergraph)	81.70 ±4.90	73.26 ±3.09	68.40 ±2.42	48.32 ±2.23	64.26 ±0.29	40.84 ±0.73	79.59 ±0.90	82.38 ±1.08	72.10 ±2.23	86.37 ±0.67	69.72
Simplex	SCCN Yang et al. (2022)	76.60 ±2.06	75.63 ±1.88	69.60 ±2.84	48.20 ±3.43	83.06 ±1.07	50.15 ±3.49	83.92 ±0.75	81.24 ±3.57	69.03 ±4.55	87.85 ±1.92	72.53
	SAN Lee et al. (2022)	71.38 ±5.73	71.54 ±2.20	64.25 ±4.27	45.35 ±2.83	84.70 ±2.79	48.99 ±3.41	82.84 ±2.41	66.23 ±0.71	62.50 ±0.94	**88.74** ±2.69	68.65
	SCCNN Yang & Isufi (2023)	70.21 ±2.82	75.63 ±1.97	69.20 ±3.04	47.73 ±3.58	84.69 ±0.98	50.33 ±2.91	83.36 ±0.86	82.13 ±3.23	70.59 ±4.28	87.83 ±2.08	72.17
	SCN Roddenberry et al. (2021)	72.34 ±2.55	75.60 ±2.07	68.80 ±2.74	47.00 ±3.82	84.16 ±1.13	49.88 ±3.23	**85.44** ±0.68	81.39 ±3.09	71.91 ±4.17	88.70 ±1.62	72.52
	TopoSRL Madhu & Chepuri (2023)	74.18 ±3.52	75.02 ±2.29	69.57 ±1.49	46.18 ±3.51	83.27 ±2.81	46.27 ±5.03	81.28 ±3.11	84.66 ±2.40	72.66 ±3.47	87.34 ±1.08	72.04
	SANN Gurugubelli & Chepuri (2024)	74.29 ±1.63	**77.62** ±2.28	72.70 ±2.16	48.97 ±1.02	84.01 ±2.06	**51.34** ±1.72	84.73 ±0.93	82.46 ±2.79	73.19 ±2.11	87.26 ±0.98	**73.66**
	RePHINE Verma et al. (2024)	75.45 ±2.88	72.10 ±2.35	**74.00** ±2.65	47.22 ±3.35	82.18 ±2.02	49.76 ±3.23	83.96 ±0.75	85.93 ±1.97	73.93 ±2.85	88.61 ±0.89	73.31
	TopoMamba Montagna et al. (2025)	75.66 ±1.86	76.90 ±2.63	71.89 ±0.86	**49.40** ±1.39	82.74 ±0.43	48.22 ±1.01	81.87 ±0.42	**86.50** ±1.03	**73.96** ±1.62	88.44 ±0.48	73.56
	CCMamba*-S(simplex)	76.60 ±2.48	69.18 ±3.15	71.20 ±2.91	42.27 ±4.28	**85.51** ±1.24	48.90 ±3.53	81.12 ±1.82	80.71 ±1.56	73.27 ±2.63	87.61 ±2.75	71.64
	CCMamba*-K(simplex)	**77.02** ±5.30	74.70 ±4.47	68.24 ±3.90	48.91 ±3.11	83.91 ±0.47	48.42 ±0.69	85.34 ±0.48	85.70 ±1.01	73.16 ±2.09	88.29 ±0.27	73.37
Cellular	CCNN Hajij et al. (2020)	74.47 ±2.17	72.04 ±2.76	72.00 ±1.92	49.33 ±3.27	81.54 ±1.33	**51.07** ±2.97	84.08 ±0.83	87.44 ±2.28	74.07 ±3.32	88.09 ±2.15	73.41
	CCXN	76.60 ±1.92	73.48 ±2.57	70.80 ±2.37	46.93 ±3.83	81.15 ±1.46	48.24 ±3.69	83.28 ±1.08	87.30 ±2.44	73.95 ±3.63	88.34 ±1.87	73.01
	CWN Bodnar et al. (2021a)	82.47 ±1.34	74.19 ±2.27	72.40 ±2.17	49.07 ±3.42	82.55 ±1.27	50.99 ±3.14	84.00 ±0.73	85.23 ±2.77	73.47 ±3.83	87.44 ±2.37	74.18
	TopoTune Papillon et al. (2025)	78.72 ±6.49	77.06 ±2.84	70.00 ±2.56	48.00 ±3.63	82.28 ±1.51	49.48 ±3.37	80.84 ±0.90	84.49 ±0.66	73.86 ±1.63	72.39 ±2.35	71.71
	CCMamba*-S(cellular)	85.11 ±2.40	78.14 ±1.68	**74.81** ±1.85	**50.13** ±3.02	**85.49** ±1.72	50.34 ±2.74	84.57 ±3.10	89.22 ±1.50	76.95 ±2.90	**89.29** ±1.47	**76.41**
	CCMamba*-K(cellular)	**87.23** ±5.75	**78.49** ±2.76	72.40 ±2.22	49.07 ±2.34	85.23 ±0.10	46.92 ±0.62	**84.68** ±0.92	**89.96** ±1.86	**77.79** ±2.12	88.66 ±0.63	76.04

MUTAG, corresponding to a **6.72%** relative improvement of CCMamba-S from graph to cellular structure on MUTAG indicates the benefit of higher-order relations.

Table 2: Test accuracy (%) of CCMamba and baseline methods on hypergraph benchmarks. Best results (**bold**) and second-best scores (underlined).

	Hypergraph Benchmarks									Citation			Coauthorship	
	20NewsW100	Mushroom	House	Walmart	Bail	Credit	German	Actor	Pokec	Cora	Citeseer	Pubmed	Cora	Dblp
UniGCN Huang & Yang (2021)	58.00 ±1.24	95.31 ±0.34	47.12 ±3.57	29.52 ±0.29	73.07 ±0.36	80.33 ±0.56	69.12 ±1.04	63.70 ±0.36	57.91 ±0.44	77.90 ±0.58	73.07 ±3.37	86.13 ±2.70	79.69 ±2.03	90.30 ±3.47
HyperGAT Ding et al. (2020)	62.47 ±0.48	95.79 ±0.26	47.05 ±3.64	29.66 ±0.49	78.99 ±1.32	80.69 ±0.44	39.71 ±0.56	63.76 ±0.25	57.92 ±0.40	77.62 ±1.12	72.37 ±2.08	86.24 ±3.44	82.91 ±1.77	90.95 ±2.64
UniGIN Huang & Yang (2021)	58.89 ±1.08	95.97 ±0.28	49.04 ±4.68	38.91 ±2.05	76.88 ±1.42	80.67 ±0.53	67.52 ±1.91	63.87 ±0.26	57.97 ±0.69	78.49 ±2.06	71.59 ±2.54	86.49 ±1.25	81.21 ±2.37	90.77 ±2.92
UniSAGE Huang & Yang (2021)	65.64 ±2.22	95.83 ±0.20	48.96 ±4.75	40.58 ±3.33	79.16 ±0.80	80.54 ±0.48	67.52 ±1.71	80.47 ±7.99	58.75 ±0.49	78.17 ±3.70	71.47 ±2.09	86.80 ±1.34	81.18 ±2.17	91.04 ±1.24
EDGNN Jo et al. (2021)	34.14 ±0.84	95.36 ±0.60	47.19 ±3.48	39.73 ±0.22	71.13 ±8.59	80.66 ±0.49	66.72 ±2.18	85.25 ±0.41	59.17 ±0.53	77.99 ±0.82	73.67 ±2.05	78.12 ±0.95	81.83 ±1.46	90.32 ±0.28
AllSet Chien et al. (2021)	80.36 ±0.38	99.56 ±0.31	47.06 ±3.50	29.92 ±0.29	95.21 ±0.52	80.13 ±0.54	69.36 ±1.69	84.20 ±0.33	58.61 ±0.56	76.81 ±3.77	71.76 ±1.95	87.14 ±4.32	81.98 ±2.32	91.40 ±2.93
DPHGNN Saxena et al. (2024)	80.45 ±0.18	99.91 ±0.03	50.46 ±4.84	42.31 ±0.11	93.50 ±0.69	80.76 ±0.57	69.60 ±0.98	84.68 ±0.04	58.90 ±1.15	81.73 ±1.53	71.98 ±1.54	86.75 ±0.68	77.20 ±0.57	90.45 ±0.15
EDHNN Wang et al. (2023)	80.62 ±0.04	99.88 ±0.02	51.81 ±1.64	33.74 ±1.30	94.24 ±0.46	80.83 ±0.65	70.27 ±1.24	84.84 ±0.16	59.07 ±0.93	81.49 ±1.30	71.45 ±1.06	86.77 ±0.48	77.70 ±1.11	91.30 ±0.16
HJRL Yan et al. (2024)	80.88 ±0.22	99.94 ±0.04	48.50 ±3.49	29.69 ±0.20	94.31 ±0.40	80.88 ±0.60	69.47 ±0.94	85.30 ±0.52	59.82 ±0.21	81.63 ±2.37	71.62 ±1.81	86.61 ±0.34	77.31 ±0.97	90.83 ±0.15
SheafHyperGNN Duta et al. (2023)	80.55 ±0.09	99.86 ±0.05	51.91 ±1.78	33.92 ±1.36	94.19 ±0.52	80.79 ±0.58	70.19 ±1.17	84.91 ±0.29	59.04 ±0.74	81.54 ±1.38	71.92 ±1.33	86.85 ±0.39	77.63 ±1.05	91.26 ±0.21
TMPHN Wang et al. (2024b)	80.78 ±0.31	99.90 ±0.06	48.23 ±3.41	29.75 ±0.24	94.27 ±0.45	80.85 ±0.54	69.33 ±1.05	85.09 ±0.51	59.76 ±0.26	81.58 ±2.41	71.55 ±1.74	86.57 ±0.42	77.14 ±1.03	90.92 ±0.24
TFHNN Tang et al. (2025)	80.66 ±0.21	99.91 ±0.03	48.37 ±3.56	29.81 ±0.17	94.35 ±0.36	80.84 ±0.53	69.87 ±0.82	85.19 ±0.21	59.58 ±0.65	79.03 ±1.03	72.10 ±1.72	86.96 ±0.75	76.71 ±0.35	90.49 ±0.13
CCMamba-S(hypergraph)	81.47 ±0.31	**99.98** ±0.03	54.49 ±4.66	**42.78** ±1.47	**95.27** ±0.18	**81.35** ±0.42	71.60 ±1.45	**86.64** ±0.17	60.05 ±1.01	**83.65** ±1.58	73.79 ±3.18	87.28 ±1.61	**84.34** ±2.40	**91.84** ±1.58
CCMamba-K(hypergraph)	**82.09** ±0.63	99.78 ±0.09	**54.80** ±4.77	41.35 ±1.62	92.39 ±0.78	80.92 ±0.40	**72.80** ±2.71	85.14 ±0.33	**60.40** ±0.58	79.04 ±1.00	**73.91** ±2.19	**87.32** ±0.57	75.04 ±0.97	85.34 ±0.26

CCMamba on Hypergraph Benchmarks For hypergraph-specific evaluation, we use native hypergraph benchmarks, including citation networks and co-authorship networks Huang & Yang (2021) and Deep Hypergraph Benchmarks DHG-Bench Li et al. (2026). Compared to the hypergraph baseline model, Table 2 shows that CCMamba achieves competitive accuracy on hypergraph benchmarks. On general hypergraph datasets, CCMamba-K improves over the strongest baselines by 5.57% on House (54.80% vs. 51.91%) and 3.60% on German (72.80% vs. 70.27%), while CCMamba-S reaches 86.64% on Actor, yielding a 1.57% relative gain over HJRL. On citation networks, CCMamba-K obtains the best results on CiteSeer (73.91%) and PubMed (87.32%). On co-authorship benchmarks, CCMamba-S achieves 84.34% on Cora and 91.84% on DBLP, including a 1.73% relative improvement on Cora over the strongest baseline. These results show competitive mean performance across heterogeneous hypergraph benchmarks.

CCMamba on Native Higher-Order Benchmarks. We also evaluate CCMamba on MANTRA 2-$\mathcal{M}$ Ballester et al. (2025), Manifold40 Hu et al. (2022), both constructed from combinatorial triangulations of 2-manifolds, and Pm25cc, a combinatorial-complex benchmark Battiloro et al. (2025). On the MANTRA name and genus classification tasks, the simplicial variant of CCMamba-K achieves the highest accuracies among the evaluated methods, at 91.34% and 93.02%, respectively. On Pm25cc, CCMamba-S obtains the numerically lowest MAE of 1.327, comparable to CAN at 1.330. These results show that CCMamba demonstrates applicability to native higher-order structures beyond graph-lifted benchmarks and remains competitive across both simplicial and combinatorial-complex domains.

Table 3: Test performance of CCMamba on native higher-order benchmarks.

	MANTRA 2-$\mathcal{M}$ name	MANTRA 2-$\mathcal{M}$ genus	Pm25cc	Manifold40	
	Accuracy	Accuracy	MAE	Accuracy	AUROC
SCN Roddenberry et al. (2021)	31.08 ±2.20	35.96 ±5.45	1.337 ±0.052	**65.50** ±2.24	96.52 ±0.36
SCCN Yang et al. (2022)	32.53 ±1.87	31.08 ±2.20	1.370 ±0.055	64.80 ±1.75	**96.61** ±0.46
SCCNN Yang & Isufi (2023)	61.40 ±3.58	31.60 ±3.78	1.383 ±0.021	63.08 ±2.37	96.27 ±0.40
CCMamba-S(simplex)	91.20 ±2.49	87.80 ±4.15	**1.332** ±0.051	61.50 ±5.40	96.33 ±1.14
CCMamba-K(simplex)	**91.34** ±1.92	**93.02** ±2.04	1.3525 ±0.042	65.42 ±1.37	96.49 ±0.36
CAN Barsbey et al. (2025)	62.40 ±13.23	55.80 ±7.03	1.330 ±0.052	63.60 ±2.16	97.01 ±0.17
CCXN Hajij et al. (2020)	71.20 ±16.98	77.04 ±5.61	1.374 ±0.050	62.10 ±3.27	96.30 ±0.51
CWN Bodnar et al. (2021a)	31.40 ±3.58	31.60 ±3.78	1.349 ±0.056	**66.20** ±5.63	96.49 ±0.57
CCCN Papillon et al. (2025)	43.60 ±19.86	40.20 ±9.93	1.361 ±0.060	62.60 ±3.27	96.65 ±0.30
TopoTune Papillon et al. (2025)	32.47 ±2.61	31.30 ±2.49	1.368 ±0.058	54.70 ±2.14	94.60 ±0.51
CCMamba-S(cellular)	81.60 ±4.10	79.40 ±6.39	**1.327** ±0.054	62.00 ±2.32	**97.14** ±0.58
CCMamba-K(cellular)	**88.24** ±6.35	**87.19** ±4.58	1.362 ±0.067	64.21 ±2.07	96.22 ±0.63

5.2 ABLATION STUDY (RQ2)

Ablation of CCMamba-S Components We ablate the main components of CCMamba-S in the cellular setting. Removing these components generally reduces predictive accuracy. **(1) Forward-only scan.** Replacing bidirectional scanning with a single forward scan lowers accuracy by 6.81% on MUTAG and 3.37 on PROTEINS. **(2) w/o BiMamba.** Removing the BiMamba module reduces 5.95 and 3.80%, respectively. **(3) w/o inter-rank.** Without cross-rank interaction, MUTAG accuracy falls by 14.47%, the largest drop observed on this dataset. **(4) w/o residual+FFN.** Removing the refinement block costs 8.93 and 3.01% on MUTAG and PROTEINS. **(5) w/o selection.** Disabling the selective parameterization reduces PROTEINS accuracy by 4.16%, the largest drop among the evaluated CCMamba-S components on this dataset.

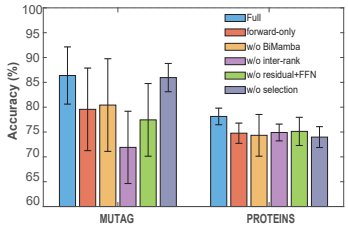


Figure 4: Components ablation

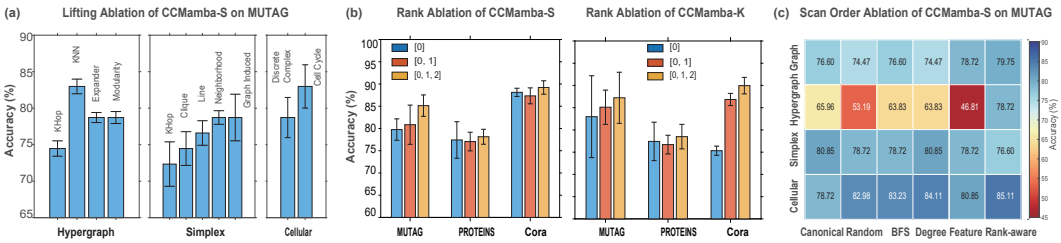


Figure 5: Ablation Studies. (a) Lifting function (graph to hypergraph, simplex and cellular) on CCMamba-S, (b)Input rank on CCMamba-S, CCMamba-K, (c)Input cell order of CCMamba-S.

Structural Ablations of CCMamba We further examine the effects of lifting strategy, input ranks, and cell ordering. **(a) Lifting function.** On MUTAG, the best-performing lifting depends on the target domain. KNN lifting performs best on hypergraphs, improving accuracy from 74.47% (KHop) to 82.98% (+11.43% relative); neighbourhood Complex and Graph-Induced lifting both reach 78.72% on simplicial complexes (+8.82%); and Cell Cycle lifting raises cellular accuracy from 78.72% to 82.98% (+5.41%). **(b) Input ranks.** Incorporating higher-order ranks benefits both CCMamba-S and CCMamba-K in the cellular domain: ranks $[0, 1]$ yield a +6.71% gain over $[0]$ alone on MUTAG, and ranks $[0, 1, 2]$ achieve the best results on Cora (89.22%) and CiteSeer (76.95%). **(c) Scan order.** For CCMamba-S, rank-aware ordering performs best on graphs, hypergraphs, and cellular complexes (79.75%, 78.72%, and 85.11%, respectively). In the simplicial setting, canonical or degree-based ordering performs better. Moreover, we conduct a matched-mixer ablation that controls for incidence lifting and structural support, comparing CCMamba-K with message passing, gated message passing, softmax attention, and kernel-attention alternatives (Appendix D.7).

5.3 Parameter Sensitivity Analysis (RQ3)

Figure 6 examines the sensitivity of CCMamba along four axes. **(1) Internal dimension.** For CCMamba-S, this corresponds to the SSM state dimension d_{state}, whereas for CCMamba-K it corresponds to the kernel feature dimension r. On MUTAG, both variants perform best at dimension=16. **(2) Maximum cell length** S**.** MUTAG attains its best accuracy at $S = 15$, whereas IMDB-B peaks at $S = 10$ and degrades at $S = 30$, suggesting a trade-off between incorporating broader higher-order structure and introducing less informative lifted cells. **(3) Model depth.** As the number of layers increases from 2 to 64 on PubMed, HyperGCN and SCCN deteriorate sharply, whereas both CCMamba variants maintain stable accuracy, suggesting strong resilience to over-smoothing. **(4) Scale.** The memory footprint of attention-based models grows rapidly with the number of heads, while Mamba scales more efficiently across graph, hypergraph, simplicial, and cellular domains.

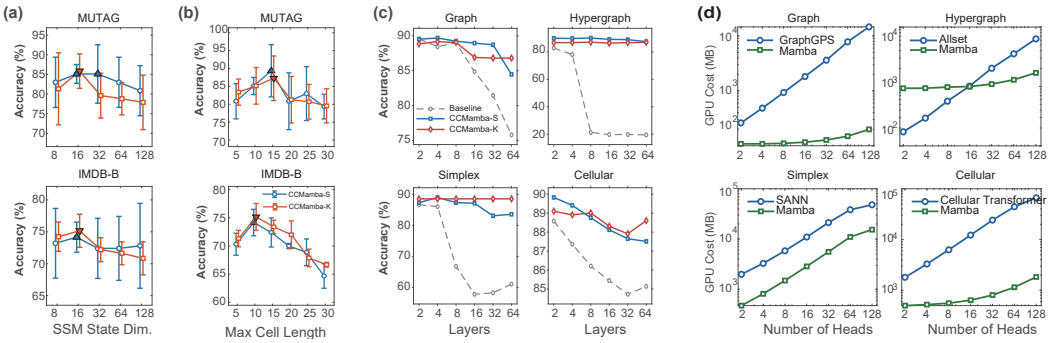


Figure 6: Hyperparameter Studies. (a) Latent dimension of SSM state, (b) maximum cell length, (c) Depth of model layer, and (d) Different heads of CCMamba-S and attention's variants.

5.4 Efficiency Study (RQ4)

Figure 7 compares the training time and peak GPU memory of CCMamba-S and CCMamba-K with domain-specific baselines across graph, hypergraph, simplicial, and cellular representations. CCMamba-S is generally faster due to fused selective scans; on the simplicial representation, it reduces training time by 87.9% and peak memory by 72.4% relative to SimplexGAT. CCMamba-K instead performs incidence-supported factorized kernel aggregation. On the same representation, it requires $218.23\,s$ and $3{,}338.2\,MB$, compared with $671.68\,s$ and $7{,}440\,MB$ for SimplexGAT. It also uses less peak memory than the evaluated hypergraph and cellular baselines, although its runtime advantage is not uniform across domains.

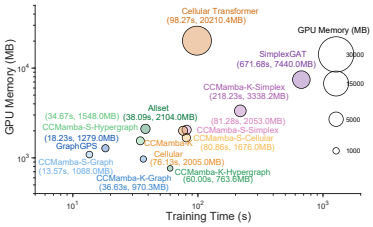


Figure 7: GPU Memory (MB) and Training Time (s) on Proteins.

6 Discussion

CCMamba extends structured mixing from sequence-induced causal support to multi-rank incidence structures. A limitation is the design tension between sequential efficiency and structural symmetry. CCMamba-S retains efficient selective scanning but depends on a non-canonical rank-aware ordering, whereas CCMamba-K removes this dependence through incidence-defined structural mixing at the cost of the fused scan. Designing selective state-space operators that respect topological symmetry while retaining efficient sequential implementations remains an important direction. Our evaluation also relies partly on higher-order structures obtained through lifting across graph, hypergraph, simplicial, and cellular domains. Although we additionally include native higher-order benchmarks, broader native combinatorial-complex datasets are still needed to assess generalization across more diverse incidence structures. The theoretical characterization of CCMamba-K is established under sufficient assumptions on the kernel, aggregation, and update functions. These assumptions describe an idealized setting rather than every practical implementation. An open question is how far they can be relaxed while preserving the same structural guarantees.

REFERENCES

Amirreza Akbari, Amauri H Souza, and Vikas K Garg. The logical expressiveness of topological neural networks. In *International Conference on Learning Representations*, 2026.

Safa Ben Atitallah, Chaima Ben Rabah, Maha Driss, Wadii Boulila, and Anis Koubaa. Exploring graph mamba: a comprehensive survey on state-space models for graph learning. *arXiv preprint arXiv:2412.18322*, 2024.

Rubén Ballester, Ernst Röell, Daniel Bin Schmid, Mathieu Alain, Sergio Escalera, Carles Casacuberta, and Bastian Rieck. MANTRA: The manifold triangulations assemblage. In *The Thirteenth International Conference on Learning Representations*, 2025.

Melih Barsbey, Rubén Ballester, Andac Demir, Carles Casacuberta, Pablo Hernández-García, David Pujol-Perich, Sarper Yurtseven, Sergio Escalera, Claudio Battiloro, Mustafa Hajij, and Tolga Birdal. Higher-order molecular learning: The cellular transformer. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

Claudio Battiloro, Indro Spinelli, Lev Telyatnikov, Michael M Bronstein, Simone Scardapane, and Paolo Di Lorenzo. From latent graph to latent topology inference: Differentiable cell complex module. In *International Conference on Learning Representations*, 2024.

Claudio Battiloro, Ege Karaismailoglu, Mauricio Tec, George Dasoulas, Michelle Audirac, and Francesca Dominici. E(n) equivariant topological neural networks. In *International Conference on Learning Representations*, 2025.

Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 119–130, 2024.

Guillermo Bernárdez, Lev Telyatnikov, Marco Montagna, Federica Baccini, Mathilde Papillon, Miquel Ferriol-Galmés, Mustafa Hajij, Theodore Papamarkou, Maria Sofia Bucarelli, Olga Zaghen, et al. Icml topological deep learning challenge 2024: Beyond the graph domain. *arXiv preprint arXiv:2409.05211*, 2024.

Maciej Besta, Florian Scheidl, Lukas Gianinazzi, Grzegorz Kwasniewski, Shachar Klaiman, Jürgen Müller, and Torsten Hoefler. Demystifying higher-order graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.

Aviv Bick, Kevin Li, Eric Xing, J Zico Kolter, and Albert Gu. Transformers to ssms: Distilling quadratic knowledge to subquadratic models. *Advances in Neural Information Processing Systems*, 37:31788–31812, 2024.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2022.

Martin Carrasco, Guillermo Bernardez, Marco Montagna, Nina Miolane, and Lev Telyatnikov. Hopse: Scalable higher-order positional and structural encoder for combinatorial representations. *arXiv preprint arXiv:2505.15405*, 2025.

Jiawen Chen, Qi Shao, Duxin Chen, and Wenwu Yu. Decoupling spatio-temporal prediction: When lightweight large models meet adaptive hypergraphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 167–178, 2025.

Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. *Advances in Neural Information Processing Systems*, 37:86145–86179, 2024.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2021.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *The Twelfth International Conference on Learning Representations*, 2024.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. Be more with less: Hypergraph attention networks for inductive text classification. In *EMNLP*, pp. 4927–4936, 2020.

Kaize Ding, Albert Jiongqian Liang, Bryan Perozzi, Ting Chen, Ruoxi Wang, Lichan Hong, Ed H Chi, Huan Liu, and Derek Zhiyuan Cheng. Hyperformer: Learning expressive sparse feature representations via hypergraph transformer. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 2062–2066, 2023.

Wenjie Du, Jiahui Zhang, Xuqiang Li, Sihan Wang, Zhengyang Zhou, Jun Xia, Ye Wei, Yang Wang, et al. I2mole: Interaction-aware invariant molecular learning for generalizable property prediction. In *The Fourteenth International Conference on Learning Representations*, 2026.

Iulia Duta, Giulia Cassarà, Fabrizio Silvestri, and Pietro Liò. Sheaf hypergraph networks. *Advances in Neural Information Processing Systems*, 36:12087–12099, 2023.

Floor Eijkelboom, Rob Hesselink, and Erik J Bekkers. E (n) equivariant message passing simplicial networks. In *International Conference on Machine Learning*, pp. 9071–9081. PMLR, 2023.

Yam Eitan, Yoav Gelberg, Guy Bar-Shalom, Fabrizio Frasca, Michael M Bronstein, and Haggai Maron. Topological blindspots: Understanding and extending topological deep learning through the lens of expressivity. In *International Conference on Learning Representations*, 2025.

Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. How powerful are k-hop message passing graph neural networks. *Advances in neural information processing systems*, 35: 4776–4790, 2022.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.

Jorge Luiz Franco, Gabriel Duarte, Alexander V Nikitin, Moacir A Ponti, Diego Mesquita, and Amauri H Souza. Differentiable lifting for topological neural networks. In *International Conference on Learning Representations*, 2026.

Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. Understanding and extending subgraph gnns by rethinking their symmetries. *Advances in Neural Information Processing Systems*, 35:31376–31390, 2022.

Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2020.

Zeyuan Guo, Enmao Diao, Cheng Yang, and Chuan Shi. Graph tokenization for bridging graphs and transformers. In *The Fourteenth International Conference on Learning Representations*, 2026.

Sravanthi Gurugubelli and Sundeep Prabhakar Chepuri. Sann: Simple yet powerful simplicial-aware neural networks. In *International Conference on Learning Representations*, 2024.

Hajij, Istvan, and Zamzmi. Cell complex neural networks. In *Topological Data Analysis and Beyond Workshop at NeurIPS 2020*, 2020.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K Dey, Soham Mukherjee, Shreyas N Samaga, et al. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, AIdo Guzman-Saenz, ToIga Birdal, and Michael T Schaub. Combinatorial complexes: Bridging the gap between cell complexes and hypergraphs. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 799–803. IEEE, 2023.

Mustafa Hajij, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John L. Davenport, Sheik Dawood, Balaji Cherukuri, Joseph G. Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, and Tolga Birdal. Copresheaf topological neural networks: A generalized deep learning framework. In *Neural Information Processing Systems*, 2025.

Xin He, Yili Wang, Wenqi Fan, Xu Shen, Xin Juan, Rui Miao, and Xin Wang. Mamba-based graph convolutional networks: tackling over-smoothing with selective state space. In *International Joint Conference on Artificial Intelligence*, 2025.

Xin He, Yili Wang, Yiwei Dai, and Xin Wang. Dual mamba for node-specific representation learning: Tackling over-smoothing with selective state space modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21672–21680, 2026.

Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Jun-Xiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R Martin. Subdivision-based mesh convolution networks. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.

Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. In *International Joint Conference on Artificial Intelligence*, 2021.

Yiming Huang, Yujie Zeng, Qiang Wu, and Linyuan Lü. Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 12653–12661, 2024a.

Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. Boosting the cycle counting power of graph neural networks with i^2-GNNs. In *The Eleventh International Conference on Learning Representations*, 2023.

Yinan Huang, Siqi Miao, and Pan Li. What can we learn from state space models for machine learning on graphs? *arXiv preprint arXiv:2406.05815*, 2024b.

Sukjun Hwang, Aakash Lahoti, Ratish Puduppully, Tri Dao, and Albert Gu. Hydra: Bidirectional state space models through generalized matrix mixers. *Advances in Neural Information Processing Systems*, 37:110876–110908, 2024.

Jaehyeong Jo, Jinheon Baek, Seul Lee, Dongki Kim, Minki Kang, and Sung Ju Hwang. Edge representation learning with hypergraphs. *Advances in Neural Information Processing Systems*, 34:7534–7546, 2021.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pp. 15546–15566. PMLR, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Maximilian Krahn, Lennart Bastian, Vikas K Garg, Björn Schuller, and Tolga Birdal. Collapsed effective operators for higher-order structures. In *Forty-third International Conference on Machine Learning*, 2026.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026.

Manuel Lecha, Andrea Cavallo, Francesca Dominici, Elvin Isufi, and Claudio Battiloro. Higher-order topological directionality and directed simplicial neural networks. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

See Hian Lee, Feng Ji, and Wee Peng Tay. Sgat: Simplicial graph attention network. In *International Joint Conference on Artificial Intelligence*, 2022.

Fan Li, Xiaoyang Wang, Wenjie Zhang, Ying Zhang, and Xuemin Lin. DHG-bench: A comprehensive benchmark for deep hypergraph learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems*, 37:127030–127058, 2024.

Zhao Li, Xin Wang, Jun Zhao, Wenbin Guo, and Jianxin Li. Hycube: Efficient knowledge hypergraph 3d circular convolutional embedding. *IEEE Transactions on Knowledge and Data Engineering*, 37 (4):1902–1914, 2025.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Ephy R Love, Benjamin Filippenko, Vasileios Maroulas, and Gunnar Carlsson. Topological convolutional layers for deep learning. *Journal of Machine Learning Research*, 24(59):1–35, 2023.

Hiren Madhu and Sundeep Prabhakar Chepuri. Toposrl: topology preserving self-supervised simplicial representation learning. *Advances in Neural Information Processing Systems*, 36:64306–64317, 2023.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

Takahiro Matsushita and Shun Wakatsuki. Dominance complexes, neighborhood complexes and combinatorial alexander duals. *Journal of Combinatorial Theory, Series A*, 211:105978, 2025.

Marco Montagna, Simone Scardapane, and Lev Telyatnikov. Topological deep learning with state-space models: A mamba approach for simplicial complexes. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2025.

Christopher Morris, Nils Morten Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *International Conference on Machine Learning 2020 Workshop on Graph Representation Learning and Beyond*, 2020a.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. *Advances in Neural Information Processing Systems*, 33: 21824–21840, 2020b.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

Mark EJ Newman. Spectral methods for community detection and graph partitioning. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 88(4):042822, 2013.

Theodore Papamarkou, Tolga Birdal, Michael Bronstein, Gunnar Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, Paolo Di Lorenzo, et al. Position paper: Challenges and opportunities in topological deep learning. *arXiv preprint arXiv:2402.08871*, 24:25, 2024a.

Theodore Papamarkou, Tolga Birdal, Michael M. Bronstein, Gunnar E. Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Lio, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T Schaub, Petar Veličković, Bei Wang, Yusu Wang, Guowei Wei, and Ghada Zamzmi. Position: Topological deep learning is the new frontier for relational learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *International Conference on Machine Learning*, volume 235, pp. 39529–39555, 2024b.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey of message-passing topological neural networks. *arXiv preprint arXiv:2304.10031*, 2023.

Mathilde Papillon, Guillermo Bernardez, Claudio Battiloro, and Nina Miolane. Topotune: A framework for generalized combinatorial complex neural networks. In *International Conference on Machine Learning*, 2025.

Raphael Pellegrin, Lukas Fesser, and Melanie Weber. Higher-order learning with graph neural networks via hypergraph encodings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Phu Pham, Quang-Thinh Bui, Ngoc Thanh Nguyen, Robert Kozma, Philip S. Yu, and Bay Vo. Topological data analysis in graph neural networks: Surveys and perspectives. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9758–9776, 2025.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023.

Ladislav Rampasek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021.

Siddhant Saxena, Shounak Ghatak, Raghu Kolla, Debashis Mukherjee, and Tanmoy Chakraborty. Dphgnn: A dual perspective hypergraph neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2548–2559, 2024.

Yehjin Shin, Seojin Kim, and Noseong Park. Graph signal processing meets mamba2: Adaptive filter bank via delta modulation. In *International Conference on Learning Representations*, 2026.

Diaaeldin Taha, James Chapman, Marzieh Eidi, Karel Devriendt, and Guido Montufar. Demystifying topological message-passing with relational structures: A case study on oversquashing in simplicial message-passing. In *International Conference on Learning Representations*, 2025.

Bohan Tang, Zexi Liu, Keyue Jiang, Siheng Chen, and Xiaowen Dong. Training-free message passing for learning on hypergraphs. In *International Conference on Learning Representations*, volume 2025, pp. 69491–69519, 2025.

Lev Telyatnikov, Guillermo Bernardez, Marco Montagna, Pavlo Vasylenko, Ghada Zamzmi, Mustafa Hajij, Michael T Schaub, Nina Miolane, Simone Scardapane, and Theodore Papamarkou. Topobenchmarkx: A framework for benchmarking topological deep learning. *arXiv preprint arXiv:2406.06642*, 2024.

Quang Truong and Peter Chin. Weisfeiler and lehman go paths: Learning topological features via path complexes. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 15382–15391, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Yogesh Verma, Amauri H Souza, and Vikas Garg. Topological neural networks go persistent, equivariant, and continuous. In *International Conference on Machine Learning*, pp. 49388–49407. PMLR, 2024.

Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024a.

Fuli Wang, Karelia Pena-Pena, Wei Qian, and Gonzalo R Arce. T-hypergnns: Hypergraph neural networks via tensor representations. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5044–5058, 2024b.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations (ICLR)*, 2023.

Yanbo Wang and Muhan Zhang. An empirical study of realized GNN expressiveness. In *Forty-first International Conference on Machine Learning*, 2024.

Tao Wen, Elynn Chen, and Yuzhou Chen. Tensor-view topological graph neural network. In *International Conference on Artificial Intelligence and Statistics*, pp. 4330–4338. PMLR, 2024.

Hanrui Wu, Andy Yip, Jinyi Long, Jia Zhang, and Michael K Ng. Simplicial complex neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):561–575, 2023a.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in attention-based graph neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, pp. 1509–1520. Curran Associates, Inc., 2019.

Yuguang Yan, Yuanlin Chen, Shibo Wang, Hanrui Wu, and Ruichu Cai. Hypergraph joint representation learning for hypervertices and hyperedges via cross expansion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 9232–9240, 2024.

Maosheng Yang and Elvin Isufi. Convolutional learning on simplicial complexes. *arXiv preprint arXiv:2301.11163*, 2023.

Ruochen Yang, Frederic Sala, and Paul Bogdan. Efficient representation learning for higher-order data with simplicial complexes. In *Learning on Graphs Conference*, pp. 13–1. PMLR, 2022.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pp. 40–48. PMLR, 2016.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, and Jianxin Li. Dg-mamba: Robust and efficient dynamic graph structure learning with selective state space models. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 22272–22280, 2025.

Bohang Zhang, Guhao Feng, Yiheng Du, Di He, and Liwei Wang. A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests. In *International Conference on Machine Learning*, pp. 41019–41077. PMLR, 2023.

Detian Zhang, Chengqiang Zhang, Yanghui Rao, Li Qing, and Chunjiang Zhu. Improved expressivity of hypergraph neural networks through high-dimensional generalized weisfeiler-leman algorithms. In *International Conference on Machine Learning*, 2025.

Kaicheng Zhang, Piero Deidda, Desmond Higham, and Francesco Tudisco. Are we measuring oversmoothing in graph neural networks correctly? In *The Fourteenth International Conference on Learning Representations*, 2026.

Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any GNN with local structure awareness. In *International Conference on Learning Representations*, 2022a.

Lingxiao Zhao, Neil Shah, and Leman Akoglu. A practical, progressively-expressive gnn. *Advances in Neural Information Processing Systems*, 35:34106–34120, 2022b.

APPENDIX CONTENTS

A RELATED WORK **17**

B Theoretical Proof **18**

B.1 Permutation equivariance on Rank-Aware Sequence Construction 19

B.2 Permutation equivariance on SSM layer of CCMamba-S 20

B.3 Permutation equivariance on incidence-masked layer of CCMamba-K 21

B.4 Expressive Power of CCMamba 23

C Experiments setups **27**

C.1 Graph Datasets 27

C.2 Higher-Order Graph Benchmarks 27

C.3 Baselines Models 29

C.4 Hyperparameters of Models 29

D Additional Experiments **30**

D.1 Lifting Operation Ablation of CCMamba 30

D.2 Ablation on Sequence Construction Strategies on CCMamba-S 32

D.3 Expressivity of CCMamba-K 32

D.4 Permutation Equivariance and Order Sensitivity. 33

D.5 Deep-layered CCMamba-S vs. Over-Smoothing Problem 34

D.6 Ablation of Aggregate Normalization and LayerNorm 34

D.7 Mixer Ablation with Matched Structural Inputs 34

D.8 Ablation of Kernel feature maps and dimension sensitivity. 35

D.9 Efficiency Analysis 35

E Limitations and Future works **36**

F LLM USAGE STATEMENT **36**

A RELATED WORK

Trends of topological deep learning To capture complex interactions beyond pairwise relations, higher-order models such as hypergraph Feng et al. (2019); Ding et al. (2020); Li et al. (2025), simplicial, and cellular neural networks Roddenberry et al. (2021); Lecha et al. (2025); Wu et al. (2023a); Bodnar et al. (2021a) have been extensively developed. Combinatorial complexes (CCs) Hajij et al. (2022; 2023) offer a unified, hierarchical framework to generalize these diverse geometries Papillon et al. (2023). Despite this unifying potential, most existing works remain fragmented. They either focus on specific structures (e.g., hypergraphs Huang & Yang (2021); Gao et al. (2020) and simplicial, cellular complexes Taha et al. (2025); Huang et al. (2024a); Besta et al. (2025); Carrasco et al. (2025)) or mapping CCs into strictly Hasse graphs (e.g., TopoTune Papillon et al. (2025), TC Akbari et al. (2026)). Despite this unifying perspective, TopoTune remains tied to specific higher-order domains or simplifies CCs through restricted constructions such as Hasse-graph-based representations, employing GNN aggregation locally and lacking reliance on long-range information. As noted in recent position and survey papers on topological deep learning Montagna et al. (2025); Papamarkou et al. (2024a;b), an open challenge is to design models that preserve the multi-rank relational structure of higher-order data while remaining general across topological domains. Our framework bridges this gap by directly manipulating the association matrix of CC, providing a rank-aware unified backbone that can be generalized to graphs, hypergraphs, simplicial complexes, and cellular complexes without structural simplification, and learning long-range dependencies through a state-space model.

Scalability of higher-order graph learning Computational efficiency represents a formidable bottleneck in scaling Topological Deep Learning to large-scale datasets. Current frameworks frequently rely on attention-based neighbourhood fusion Rampasek et al. (2022); Deng et al. (2024); Barsbey et al. (2025); Papillon et al. (2025); Gurugubelli & Chepuri (2024); Bick et al. (2024), which dense self-attention typically incurs quadratic complexity in the number of cells. $\mathcal{O}(N^2)$ concerning the neighbourhood size. Convolution-based operators in frameworks like TopoBenchmarkX Telyatnikov et al. (2024) and SMCN Eitan et al. (2025) alleviate complexity suffer from over-smoothing as model depth increases. problems Wu et al. (2023b); Zhang et al. (2026). Alternatively, Selective State-Space Models (SSMs) Dao & Gu (2024); Hwang et al. (2024); Yuan et al. (2025) have emerged as a scalable alternative to Transformers, achieving linear-time inference $\mathcal{O}(N)$ for long-range sequences. While recent works have adapted SSMs to graph sequences Atitallah et al. (2024); Wang et al. (2024a); He et al. (2025) and temporal graphs Li et al. (2024); Shin et al. (2026), they remain confined to pairwise structures He et al. (2026); Huang et al. (2024b). CCMamba addresses this limitation through a shared incidence-lifting framework with two complementary mixers, CCMamba-S processes rank-wise representations using selective SSM scans, whereas CCMamba-K replaces sequential recurrence with incidence-structured factorized kernel mixing.

Expressive power analysis of graph-mamba methods. Weisfeiler-Lehman tests characterize the discriminative power of message passing: 1-WL bounds ordinary GNNs Xu et al. (2018), and analogous refinements have been defined for hypergraphs via bipartite incidence Zhang et al. (2025); Huang & Yang (2021), for simplicial complexes via boundary and co-boundary operators Bodnar et al. (2021b); Eitan et al. (2025), and for cellular complexes via dimension-wise rules Bodnar et al. (2021a); Truong & Chin (2024); Papillon et al. (2025). Each of these bounds rests on a precondition: the model's output must be a well-defined function of the isomorphism class, which on a complex means equivariant layers and an invariant readout Eitan et al. (2025); Xu et al. (2018). Message passing satisfies it by construction, since every layer aggregates a multiset of neighbors. A selective scan does not. It reads an ordered sequence, and a domain with no canonical ordering forces that order to be manufactured by linearization in existing Graph Mamba variants Wang et al. (2024a); Behrouz & Hashemi (2024); He et al. (2025), making the output a function of the chosen order as well as the complex, so the multiset argument no longer applies as stated. The alternatives, canonicalizing the order or randomizing it and settling for invariance in expectation, either change the bound being proved or weaken it to a probabilistic statement. We instead recover the precondition structurally, deriving mixing coefficients from incidence relations rather than from scan position, and characterize the resulting layer against 1-CCWL.

Relation to linear attention and state-space models. CCMamba is also related to kernelized linear attention Dao & Gu (2024); Rampasek et al. (2022); Katharopoulos et al. (2020); Bick et al.

(2024) and recent Mamba-style models on graphs. Linear and kernel attention replace quadratic softmax interactions with factorized query–key aggregation, while Graph Mamba methods obtain efficient global propagation by imposing a sequence order on graph elements. Our two instantiations occupy different positions in this design space. CCMamba-S retains selective state-space recurrence and therefore inherits dependence on the chosen within-rank ordering, whereas CCMamba-K retains the factorized kernel interaction but replaces position-dependent sequence support with index-free, incidence-induced structural support over multiple cell ranks. The latter construction therefore separates the use of efficient kernel factorization from the choice of an imposed linearization, and admits permutation-equivariance and a conditional 1-CCWL characterization under the assumptions stated in our analysis.

Distinction of CCMamba from existing works. Table A.1 compares CCMamba with representative attention- and state-space-based mixing mechanisms. The key distinction is how the mixing support is defined. Standard attention uses global token interactions, whereas sequential Graph Mamba and CCMamba-S rely on an imposed scan order. CCMamba instead extends structured mixing to combinatorial complexes through rank-aware incidence lifting and rank-specific mixing. CCMamba-S retains bidirectional selective SSM scans within each rank. This preserves the recurrent structure and efficient fused sequence processing of selective SSMs, while incidence lifting provides cross-rank information before the scan. However, the resulting mixing still depends on the chosen within-rank ordering and is therefore not permutation equivariant in general. Unlike CCMamba-S, CCMamba-K introduces incidence-supported factorized kernel mixing to capture non-local structural dependencies. CCMamba-K is not itself a selective SSM block, which is the order-free structured-mixing counterpart induced by our support decomposition.

CCMamba-K shares the factorized content interaction $\psi(q_\sigma)^\top \psi(k_\tau)$ with kernelized linear attention, but the kernel factorization itself is not the source of its distinction. The key difference lies in how the mixing operator is structurally defined. Standard linear attention typically aggregates over a global token set, whereas CCMamba-K couples factorized content interactions with rank-aware incidence support and index-free structural coefficients $a_{\sigma\tau}$. Its mixing matrix therefore takes the form $M = L_{\mathrm{CC}} \circ (\Psi_Q \Psi_K^\top)$, where L_{CC} encodes the incidence-defined support and structural modulation of the combinatorial complex. Together with the uncontracted structural aggregate z_σ, this yields a permutation-equivariant, multiplicity-aware mixer without imposing a cell linearization. Thus, CCMamba-K is best viewed as an incidence-supported structural kernel mixer for multi-rank combinatorial complexes rather than as standard global linear attention.

Table A.1: Design-space comparison of CCMamba with representative sequence- and attention-based mixing mechanisms. N denotes the total number of cells, N_k the number of rank-k cells, M the number of nonzero incidence pairs, d the hidden dimension, d_s the SSM state dimension, and r the kernel feature dimension. Permutation equivariance for attention-based methods assumes no order-dependent positional encoding or non-equivariant masking.

	Softmax Attention	Linear Attention	Kernel Attention	Sequential Graph Mamba	CCMamba-S Forward	CCMamba-S BiMamba	CCMamba-K Structural
Mixing mechanism							
Mixing support	global	global	global	scan sequence	rank sequence	rank sequence	$\mathcal{N}(\sigma)$
Structural support	none	none	none	graph ordering	incidence lifting	incidence lifting	incidence mask
Pairwise weighting	$\exp(q^\top k)$	$\psi(q)^\top \psi(k)$	$\psi(q)^\top \psi(k)$	selective SSM	selective SSM	selective SSM	$a_{\sigma\tau}\ \psi(q_\sigma)^\top \psi(k_\tau)$
Normalization	softmax	normalized	normalized	none	none	none	none
State recurrence	×	×	×	✓	✓	✓ (bi)	×
Input-dependent mixing	✓	✓	✓	✓	✓	✓	✓
Structural properties							
Native domain	sequence/set	sequence/set	sequence/set	graph	CC	CC	CC
Explicit multi-rank modeling	×	×	×	×	✓	✓	✓
Depends on imposed order	no	no	no	yes	yes	yes	no
Permutation equivariant	yes	yes	yes	not in general	no	no	yes
Per-layer receptive field	global	global	global	sequence-global	rank-global	rank-global	$\mathcal{N}(\sigma)$

B THEORETICAL PROOF

A selective scan is not permutation-equivariant under arbitrary reorderings of cells within a rank in general; this is the obstruction formalized by Proposition 1. We therefore first characterize how the incidence-lifted representation transforms under relabeling, and then determine which intra-rank

mixers preserve this structure. Under independent relabelings of 0-, 1-, and 2-cells by permutation matrices P_0, P_1, and P_2, the lifted features transform as $X'_k = P_k X_k$ and the incidence operators as $I'_k = P_{k-1} I_k P_k^\top$, so rank-preserving relabelings permute the cells together with their incidence relations rather than altering the underlying combinatorial structure.

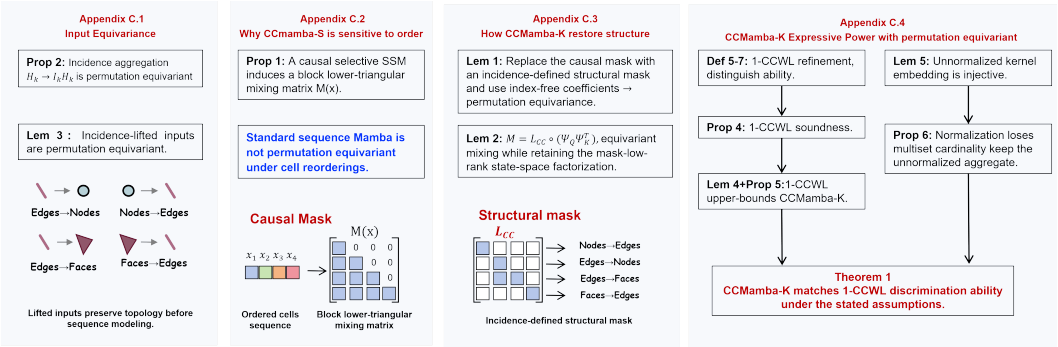


Figure B.1: Roadmap of the proof.

Figure B.1 summarizes the argument. We first show that any causal scan induces a block lower-triangular mixing matrix and therefore cannot be equivariant. We then replace sequence-induced causal support with incidence-supported, index-free structural coefficients and show that the resulting CCMamba-K layer is permutation-equivariant. Its interaction matrix admits a support-weighted factorization in which structural coefficients modulate factorized kernel interactions, while its computational cost is governed by sparse incidence support rather than sequential recurrence. Under the additional assumptions stated in Section B.4, the resulting architecture also admits the corresponding Combinatorial Complex Weisfeiler–Leman (CCWL) expressivity characterization. CCMamba-K realizes this permutation-equivariant structural mixer, whereas CCMamba-S provides a complementary sequence-based realization that retains selective scanning under a rank-aware ordering but does not inherit the permutation-equivariance guarantee.

B.1 Permutation equivariance on Rank-Aware Sequence Construction

Definition 5 (Rank-Aware Ordering). *Let $\mathcal{C}_k = \{\sigma \in \mathcal{C} : \mathrm{rk}(\sigma) = k\}$ denote the set of rank-k cells. For each $\sigma \in \mathcal{C}_k$, we define its rank-aware structural signature as*

$$\mathbf{s}_k(\sigma) = \big(d_k^-(\sigma), d_k^+(\sigma)\big), \tag{11}$$

where

$$d_k^-(\sigma) = \sum_{\tau \in \mathcal{C}_{k-1}} |[I_k]_{\tau,\sigma}|\,, \qquad d_k^+(\sigma) = \sum_{\gamma \in \mathcal{C}_{k+1}} |[I_{k+1}]_{\sigma,\gamma}|\,. \tag{12}$$

Here, $d_k^-(\sigma)$ and $d_k^+(\sigma)$ denote the numbers of incident lower- and higher-rank cells, respectively, with missing terms at the boundary ranks set to zero. Cells within each rank are lexicographically sorted according to $\mathbf{s}_k(\sigma)$, with a fixed deterministic rule for resolving ties. The resulting permutation π_k defines the ordered rank-k sequence $X_k^{\mathrm{ord}} = P_{\pi_k} X_k$, which is subsequently processed by the rank-specific selective state-space encoder.

To obtain rank-aware inputs for sequence modeling, we construct rank-specific incidence-lifted representations across different cell ranks from the rank-aware aggregation and Proposition 2.

Proposition 2. *Let $I_k \in \mathbb{R}^{n_{k-1} \times n_k}$ be the incidence matrix between $(k-1)$-cells and k-cells, and let $H_k \in \mathbb{R}^{n_k \times d}$ be the rank-k feature matrix. Let P_{k-1} and P_k be permutation matrices acting on the $(k-1)$-cells and k-cells, respectively. Under a relabeling of cells, suppose $I'_k = P_{k-1} I_k P_k^\top$, $H'_k = P_k H_k$. Then $I'_k H'_k = P_{k-1}(I_k H_k)$. Thus, $H_k \mapsto I_k H_k$ is permutation equivariant.*

Proof for Proposition 2. Assume a relabeling of $(k-1)$-cells and k-cells, parameterized by the permutation matrices P_{k-1} and P_k, respectively. Under this transformation, the incidence matrix and the rank-k feature matrix update as follows: $I'_k = P_{k-1} I_k P_k^\top$, $H'_k = P_k H_k$. Evaluating the incidence-aggregated representation yields: $I'_k H'_k = (P_{k-1} I_k P_k^\top)(P_k H_k)$. By definition, the

permutation matrix P_k is orthogonal, which guarantees $P_k^\top P_k = \mathbf{I}$. Substituting this identity naturally simplifies the expression $I'_k H'_k = P_{k-1} I_k (P_k^\top P_k) H_k = P_{k-1} I_k \mathbf{I} H_k = P_{k-1}(I_k H_k)$. Consequently, the aggregation map $H_k \mapsto I_k H_k$ is permutation equivariant with respect to any relabeling of the underlying complex. □

Lemma 3 states that incidence lifting is equivariant to rank-wise relabeling: under permutations P_k of the k-cells, the lifted features transform as $X'_k = P_k X_k$, so cross-rank relations are carried along with the relabeling rather than destroyed by it. The claim concerns the lifting operator alone. The selective scan applied to these representations consumes them under a rank-aware ordering, and the lemma says nothing about its behavior under reorderings within a rank; layer equivariance is established separately for CCMamba-K in Lemma 1 and Lemma 2.

Lemma 3. *Let $H_k \in \mathbb{R}^{n_k \times d_k}$ be the rank-k feature matrix, and I_k be the incidence matrix. The incidence-lifted representations X_0, X_1, X_2 for rank-0 , rank-1 , and rank-2 cells, respectively :*

$$X_0 = [H_0 \,\|\, I_1 H_1], \quad X_1 = [H_1 \,\|\, I_1^\top H_0 \,\|\, I_2 H_2], \quad X_2 = [H_2 \,\|\, I_2^\top H_1],$$

where $\|$ denotes channel-wise concatenation. Let P_0, P_1, P_2 be permutation matrices acting on 0-, 1-, and 2-cells, respectively. Suppose the incidence permuted matrices satisfy $I'_1 = P_0 I_1 P_1^\top$, $I'_2 = P_1 I_2 P_2^\top$, and $H'_k = P_k H_k$. Then the relabeled inputs satisfy: $X'_0 = P_0 X_0$, $X'_1 = P_1 X_1$, $X'_2 = P_2 X_2$. Hence, the incidence-lifted inputs are rank-wise permutation equivariant.

Proof of Lemma 3. Assume a relabeling of 0-, 1-, and 2-cells, defined by the permutation matrices P_0, P_1, and P_2, respectively. The incidence and feature matrices transform as follows:

$$I'_1 = P_0 I_1 P_1^\top, \qquad I'_2 = P_1 I_2 P_2^\top, \qquad H'_k = P_k H_k, \quad \forall k \in \{0,1,2\}.$$

We first prove that each incidence-based aggregation term transforms equivariantly. For the forward incidence aggregations, by Proposition 2, we have:

$$I'_1 H'_1 = (P_0 I_1 P_1^\top)(P_1 H_1) = P_0 I_1 H_1, I'_2 H'_2 = (P_1 I_2 P_2^\top)(P_2 H_2) = P_1 I_2 H_2.$$

Similarly, for the transpose-incidence aggregations, the transposed incidence matrices are:

$$(I'_1)^\top = P_1 I_1^\top P_0^\top, \qquad (I'_2)^\top = P_2 I_2^\top P_1^\top.$$

Consequently, their corresponding aggregations yield:

$$(I'_1)^\top H'_0 = (P_1 I_1^\top P_0^\top)(P_0 H_0) = P_1 I_1^\top H_0, (I'_2)^\top H'_1 = (P_2 I_2^\top P_1^\top)(P_1 H_1) = P_2 I_2^\top H_1.$$

Since channel-wise concatenation with rank-wise permutations, we obtain the incidence-lifted inputs for the node, edge, and face features X'_0, X'_1, and X'_2, respectively:

$$\begin{aligned}
X'_0 &= [H'_0 \,\|\, I'_1 H'_1] = [P_0 H_0 \,\|\, P_0 (I_1 H_1)] = P_0 [H_0 \,\|\, I_1 H_1] = P_0 X_0, \\
X'_1 &= [H'_1 \,\|\, (I'_1)^\top H'_0 \,\|\, I'_2 H'_2] = [P_1 H_1 \,\|\, P_1 (I_1^\top H_0) \,\|\, P_1 (I_2 H_2)] = P_1 [H_1 \,\|\, I_1^\top H_0 \,\|\, I_2 H_2] = P_1 X_1, \\
X'_2 &= [H'_2 \,\|\, (I'_2)^\top H'_1] = [P_2 H_2 \,\|\, P_2 (I_2^\top H_1)] = P_2 [H_2 \,\|\, I_2^\top H_1] = P_2 X_2.
\end{aligned} \tag{13}$$

It concludes that $X'_k = P_k X_k$, $\forall k \in \{0,1,2\}$. Thus, the incidence-lifted representations are rank-wise permutation equivariant. □

B.2 Permutation equivariance on SSM layer of CCMamba-S

Proof for Proposition 1. We prove the claim in two steps. First, we make the block lower-triangular structure of $\mathbf{M}(\mathbf{x})$ fully explicit. Second, we exhibit a minimal counterexample at $T = 2$ that shows equation 8 fails even for the simplest non-trivial permutation.

Since $\mathbf{M}_{ji} = 0$ for $i > j$, the mixing matrix admits the explicit block lower-triangular form

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix}
\mathbf{C}_0^\top \bar{\mathbf{B}}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{C}_1^\top \bar{\mathbf{A}}_1 \bar{\mathbf{B}}_0 & \mathbf{C}_1^\top \bar{\mathbf{B}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{C}_2^\top \bar{\mathbf{A}}_2 \bar{\mathbf{A}}_1 \bar{\mathbf{B}}_0 & \mathbf{C}_2^\top \bar{\mathbf{A}}_2 \bar{\mathbf{B}}_1 & \mathbf{C}_2^\top \bar{\mathbf{B}}_2 & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{C}_{T-1}^\top \prod_{t=1}^{T-1} \bar{\mathbf{A}}_t \bar{\mathbf{B}}_0 & \mathbf{C}_{T-1}^\top \prod_{t=2}^{T-1} \bar{\mathbf{A}}_t \bar{\mathbf{B}}_1 & \mathbf{C}_{T-1}^\top \prod_{t=3}^{T-1} \bar{\mathbf{A}}_t \bar{\mathbf{B}}_2 & \cdots & \mathbf{C}_{T-1}^\top \bar{\mathbf{B}}_{T-1}
\end{bmatrix}. \tag{14}$$

Two structural facts follow directly from equation 14: ❶ The nonzero pattern is strictly lower-triangular by blocks, then output position j depends only on input positions $i \le j$. ❷ Each entry $\mathbf{M}_{ji}$ is a product of consecutive transition operators indexed by the interval $[i+1, j]$, and the coefficient depends not only on the identities of i and j, but on the entire ordered chain $\bar{\mathbf{A}}_{i+1}, \bar{\mathbf{A}}_{i+2}, \ldots, \bar{\mathbf{A}}_j$ traversed between them.

For a selective SSM, the parameters $(\bar{\mathbf{A}}_t, \bar{\mathbf{B}}_t, \mathbf{C}_t)$ are input-dependent: $\bar{\mathbf{A}}_t = \bar{\mathbf{A}}(x_t)$, $\bar{\mathbf{B}}_t = \bar{\mathbf{B}}(x_t)$, $\mathbf{C}_t = \mathbf{C}(x_t)$. Hence permuting $\mathbf{x}$ permutes the underlying operator sequence, and both (a) and (b) are affected. At $T = 2$, It suffices to show $\mathbf{M}(\mathbf{P}\mathbf{x})\mathbf{P} = \mathbf{P}\mathbf{M}(\mathbf{x})$ fails for a single choice of $\mathbf{x}$, $\mathbf{P}$, and model parameters. Take $T = 2$, $\mathbf{x} = [x_0, x_1]^\top$ with $x_0 \neq x_1$, and let $\mathbf{P}$ be the swap $\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Then $\mathbf{P}\mathbf{x} = [x_1, x_0]^\top$. By equation 14, the mixing matrix on the original input is

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(x_0)^\top \bar{\mathbf{B}}(x_0) & \mathbf{0} \\ \mathbf{C}(x_1)^\top \bar{\mathbf{A}}(x_1)\, \bar{\mathbf{B}}(x_0) & \mathbf{C}(x_1)^\top \bar{\mathbf{B}}(x_1) \end{bmatrix}, \tag{15}$$

and on the permuted input

$$\mathbf{M}(\mathbf{P}\mathbf{x}) = \begin{bmatrix} \mathbf{C}(x_1)^\top \bar{\mathbf{B}}(x_1) & \mathbf{0} \\ \mathbf{C}(x_0)^\top \bar{\mathbf{A}}(x_0)\, \bar{\mathbf{B}}(x_1) & \mathbf{C}(x_0)^\top \bar{\mathbf{B}}(x_0) \end{bmatrix}. \tag{16}$$

Multiplying on the right by $\mathbf{P}$ swaps the columns:

$$\mathbf{M}(\mathbf{P}\mathbf{x})\mathbf{P} = \begin{bmatrix} \mathbf{0} & \mathbf{C}(x_1)^\top \bar{\mathbf{B}}(x_1) \\ \mathbf{C}(x_0)^\top \bar{\mathbf{B}}(x_0) & \mathbf{C}(x_0)^\top \bar{\mathbf{A}}(x_0)\, \bar{\mathbf{B}}(x_1) \end{bmatrix}. \tag{17}$$

Multiplying $\mathbf{M}(\mathbf{x})$ on the left by $\mathbf{P}$ swaps the rows:

$$\mathbf{P}\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(x_1)^\top \bar{\mathbf{A}}(x_1)\, \bar{\mathbf{B}}(x_0) & \mathbf{C}(x_1)^\top \bar{\mathbf{B}}(x_1) \\ \mathbf{C}(x_0)^\top \bar{\mathbf{B}}(x_0) & \mathbf{0} \end{bmatrix}. \tag{18}$$

Comparing equation 17 and equation 18 block by block, the anti-diagonal blocks agree, while the diagonal blocks do not: the $(0,0)$ block is $\mathbf{0}$ on the left but $\mathbf{C}(x_1)^\top \bar{\mathbf{A}}(x_1)\bar{\mathbf{B}}(x_0)$ on the right, and the $(1,1)$ block is $\mathbf{C}(x_0)^\top \bar{\mathbf{A}}(x_0)\bar{\mathbf{B}}(x_1)$ on the left but $\mathbf{0}$ on the right. Both would have to vanish for $\mathbf{M}(\mathbf{P}\mathbf{x})\mathbf{P} = \mathbf{P}\mathbf{M}(\mathbf{x})$ to hold, which requires $\bar{\mathbf{B}}(x_0)$ to lie in the kernel of $\mathbf{C}(x_1)^\top \bar{\mathbf{A}}(x_1)$ and $\bar{\mathbf{B}}(x_1)$ in the kernel of $\mathbf{C}(x_0)^\top \bar{\mathbf{A}}(x_0)$, which is a degenerate configuration that no selective parameterization enforces. Since $\mathbf{M}$ depends on its input, failure of the matrix identity $\mathbf{M}(\mathbf{P}\mathbf{x})\mathbf{P} = \mathbf{P}\mathbf{M}(\mathbf{x})$ does not by itself imply that the induced map $f(\mathbf{x}) = \mathbf{M}(\mathbf{x})\mathbf{x}$ fails to be permutation-equivariant. It is therefore sufficient to exhibit a valid causal selective-scan parameterization and an input for which equivariance fails.

Consider the scalar case with $\mathbf{C}(x) = \bar{\mathbf{B}}(x) = 1$ and $\bar{\mathbf{A}}(x) = a \neq 0$. Then $f(\mathbf{x}) = (x_0,\, ax_0 + x_1)$. For the swap permutation $\mathbf{P}$, $f(\mathbf{P}\mathbf{x}) = (x_1,\, ax_1 + x_0)$, $\mathbf{P}f(\mathbf{x}) = (ax_0 + x_1,\, x_0)$. For generic nonzero inputs these two expressions are different. Hence there exist valid causal selective-scan parameters and inputs for which $f(\mathbf{P}\mathbf{x}) \neq \mathbf{P}f(\mathbf{x})$. Therefore, causal selective scans are not permutation-equivariant in general with respect to arbitrary token reorderings. This establishes that permutation equivariance is not guaranteed by the causal selective-scan architecture itself. □

B.3 Permutation equivariance on incidence-masked layer of CCMamba-K

Proof for Lemma 1. Write $\mathcal{F}(\mathbf{X}; \mathcal{CC})_\sigma = \left[\psi(q_\sigma)^\top \mathbf{S}_\sigma \,\middle\|\, \mathbf{z}_\sigma\right]$ for the layer of Definition 4, where $q_\sigma = \mathbf{W}_Q X_\sigma$, $k_\tau = \mathbf{W}_K X_\tau$, $v_\tau = \mathbf{W}_V X_\tau$, and $\mathbf{S}_\sigma, \mathbf{z}_\sigma$ are given by equation 9. Let $\pi : \mathcal{CC}_1 \to \mathcal{CC}_2$ be a rank- and incidence-preserving bijection with permutation matrix $\mathbf{P}_\pi$, so that the relabeled input satisfies $(\mathbf{P}_\pi \mathbf{X})_{\pi(\sigma)} = X_\sigma$ for every $\sigma \in \mathcal{CC}_1$.

Since $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are shared linear maps applied identically to every cell regardless of rank labeling, $q'_{\pi(\sigma)} = \mathbf{W}_Q(\mathbf{P}_\pi \mathbf{X})_{\pi(\sigma)} = \mathbf{W}_Q X_\sigma = q_\sigma$, and likewise $k'_{\pi(\tau)} = k_\tau$, $v'_{\pi(\tau)} = v_\tau$, for all σ, τ. Because π preserves rank and incidence, $\pi(\mathcal{N}(\sigma)) = \mathcal{N}(\pi(\sigma))$, and the pairwise invariants of Definition 2 are preserved: $\zeta(\pi(\sigma)) = \zeta(\sigma)$ and $\iota(\pi(\sigma), \pi(\tau)) = \iota(\sigma, \tau)$. Since $\{a_{\sigma\tau}\}$ is index-free structural (Definition 3), $a_{\sigma\tau}$ is a function of $\zeta(\sigma), \zeta(\tau), \iota(\sigma, \tau)$ and the features

X_σ, X_τ alone. Combined with $q'_{\pi(\sigma)} = q_\sigma, k'_{\pi(\tau)} = k_\tau$, $v'_{\pi(\tau)} = v_\tau$, this gives $a'_{\pi(\sigma)\pi(\tau)} = \alpha\big(\zeta(\pi(\sigma)), \zeta(\pi(\tau)), \iota(\pi(\sigma), \pi(\tau)), X'_{\pi(\sigma)}, X'_{\pi(\tau)}\big) = \alpha\big(\zeta(\sigma), \zeta(\tau), \iota(\sigma, \tau), X_\sigma, X_\tau\big) = a_{\sigma\tau}$.

Further, $\tau' = \pi(\tau)$ as a bijection of $\mathcal{N}(\sigma)$ onto $\mathcal{N}(\pi(\sigma))$, then

$$\mathbf{S}'_{\pi(\sigma)} = \sum_{\tau' \in \mathcal{N}(\pi(\sigma))} a'_{\pi(\sigma)\tau'}\, \psi(k'_{\tau'}) \otimes v'_{\tau'} = \sum_{\tau \in \mathcal{N}(\sigma)} a_{\sigma\tau}\, \psi(k_\tau) \otimes v_\tau = \mathbf{S}_\sigma, \tag{19}$$

and by the identical argument $\mathbf{z}'_{\pi(\sigma)} = \mathbf{z}_\sigma$. Since $q'_{\pi(\sigma)} = q_\sigma$ and $\mathbf{S}'_{\pi(\sigma)} = \mathbf{S}_\sigma$, $\mathbf{z}'_{\pi(\sigma)} = \mathbf{z}_\sigma$,

$$\mathcal{F}(\mathbf{P}_\pi \mathbf{X}; \mathcal{CC}_2)_{\pi(\sigma)} = \big[\psi(q'_{\pi(\sigma)})^\top \mathbf{S}'_{\pi(\sigma)} \,\big\|\, \mathbf{z}'_{\pi(\sigma)}\big] = \big[\psi(q_\sigma)^\top \mathbf{S}_\sigma \,\big\|\, \mathbf{z}_\sigma\big] = \mathcal{F}(\mathbf{X}; \mathcal{CC}_1)_\sigma. \tag{20}$$

Since this holds for every $\sigma \in \mathcal{CC}_1$ and π is a bijection, stacking over all cells gives $\mathcal{F}(\mathbf{P}_\pi \mathbf{X}; \mathcal{CC}_2) = \mathbf{P}_\pi\, \mathcal{F}(\mathbf{X}; \mathcal{CC}_1)$. Taking $\mathcal{CC}_1 = \mathcal{CC}_2 = \mathcal{CC}$ and π an automorphism of $\mathcal{CC}$ shows the layer is equivariant to automorphisms. □

Remark 1 (Structural neighbourhood of CCMamba-K)**.** *For practical CCMamba-K, $\mathcal{N}(\sigma)$ consists of the immediate adjacent-rank incidence neighbours of σ. For a rank-k cell, $\mathcal{N}(\sigma) = \mathcal{N}_{k-1\to k}(\sigma) \cup \mathcal{N}_{k+1\to k}(\sigma)$, with out-of-range ranks omitted. In a 2-dimensional complex, this gives*

$$\begin{aligned} \mathcal{N}(v) &= \mathcal{N}_{1\to 0}(v), \\ \mathcal{N}(e) &= \mathcal{N}_{0\to 1}(e) \cup \mathcal{N}_{2\to 1}(e), \\ \mathcal{N}(f) &= \mathcal{N}_{1\to 2}(f). \end{aligned}$$

We do not additionally include same-rank adjacency induced by shared boundary or co-boundary cells, nor explicit multi-hop relations. Multi-hop information is obtained implicitly by stacking CCMamba layers.

Proof for Lemma 2. Fix a cell $\sigma \in \mathcal{C}$. By equation 9, $\mathbf{S}_\sigma = \sum_{\tau \in \mathcal{N}(\sigma)} a_{\sigma\tau}\, \psi(k_\tau) \otimes v_\tau$, and Definition 3 sets $a_{\sigma\tau} = 0$ for $\tau \notin \mathcal{N}(\sigma)$, so the sum extends to all of $\mathcal{C}$ without changing its value: $\mathbf{S}_\sigma = \sum_{\tau \in \mathcal{C}} a_{\sigma\tau}\, \psi(k_\tau) \otimes v_\tau$. By equation 10, the first block of the output at σ is $\psi(q_\sigma)^\top \mathbf{S}_\sigma$, which bilinearity expands as

$$\psi(q_\sigma)^\top \mathbf{S}_\sigma = \sum_{\tau \in \mathcal{C}} a_{\sigma\tau}\, \big(\psi(q_\sigma)^\top \psi(k_\tau)\big)\, v_\tau. \tag{21}$$

Setting $\mathbf{M}_{\sigma\tau} := a_{\sigma\tau}\, \psi(q_\sigma)^\top \psi(k_\tau)$ for every pair $\sigma, \tau \in \mathcal{C}$, this becomes $\psi(q_\sigma)^\top \mathbf{S}_\sigma = \sum_{\tau \in \mathcal{C}} \mathbf{M}_{\sigma\tau}\, v_\tau$, so the first output block equals $\mathbf{MV}$, where $\mathbf{M} = [\mathbf{M}_{\sigma\tau}]$ and $\mathbf{V}$ stacks the v_τ row-wise. This is the counterpart on $\mathcal{CC}$ of the sequence mixing $y = \mathbf{M}(\mathbf{x})\mathbf{x}$ in equation 3.

Let $\mathbf{\Psi}_Q, \mathbf{\Psi}_K$ stack $\psi(q_\sigma)^\top$ and $\psi(k_\tau)^\top$ row-wise over $\mathcal{C}$, so that $(\mathbf{\Psi}_Q \mathbf{\Psi}_K^\top)_{\sigma\tau} = \psi(q_\sigma)^\top \psi(k_\tau)$, and let $[\mathbf{L}_{\mathcal{CC}}]_{\sigma\tau} = a_{\sigma\tau}$, the index-free structural coefficients of Definition 3, supported on $\{(\sigma, \tau) : \tau \in \mathcal{N}(\sigma)\}$. $\mathbf{M}_{\sigma\tau} = [\mathbf{L}_{\mathcal{CC}}]_{\sigma\tau} \cdot (\mathbf{\Psi}_Q \mathbf{\Psi}_K^\top)_{\sigma\tau}$, hence $\mathbf{M} = \mathbf{L}_{\mathcal{CC}} \circ \big(\mathbf{\Psi}_Q \mathbf{\Psi}_K^\top\big)$, with $\circ$ the Hadamard product. The two factors play the roles they do in equation 3. The factor $\mathbf{\Psi}_Q \mathbf{\Psi}_K^\top$ has rank at most $r := \dim \mathcal{H}_\psi$ and depends only on the features through q, k, not on $\mathcal{CC}$; it is the analogue of the low-rank factor $\mathbf{C}_j^\top \mathbf{B}_i$. All dependence on the complex sits in $\mathbf{L}_{\mathcal{CC}}$, whose nonzero pattern is the incidence structure of $\mathcal{CC}$ rather than the causal support $\{(j, i) : i \le j\}$. Thus, $\mathbf{M}$ admits an exact support-weighted factorization, $\mathbf{M} = \mathbf{L}_{\mathcal{CC}}(\mathbf{X}) \circ (\Psi_Q \Psi_K^\top)$, where $\mathbf{L}_{\mathcal{CC}}(\mathbf{X})$ has incidence-defined support and $\Psi_Q \Psi_K^\top$ provides the factorized kernel interaction. Although $\Psi_Q \Psi_K^\top$ has rank at most the kernel feature dimension, the Hadamard product $\mathbf{M}$ itself need not be low rank. □

Remark 2 (What restores equivariance, and what the kernel is for)**.** *The equivariance result in Lemma 1 relies on two structural conditions. First, the interaction support is given by the incidence-induced neighbourhood $\mathcal{N}(\sigma)$, which transforms under rank- and incidence-preserving relabelings, $\mathcal{N}(\pi\sigma) = \pi(\mathcal{N}(\sigma))$, rather than by a position-dependent causal prefix. Second, the coefficients $a_{\sigma\tau}$ are index-free and therefore satisfy $a'_{\pi\sigma,\pi\tau} = a_{\sigma\tau}$. Apart from being applied pointwise and shared across cells, the equivariance proof does not require injectivity or other special properties of the feature map ψ. Thus, permutation equivariance is a consequence of equivariant support and index-free coefficients, rather than of kernelization itself. For example, an attention layer with equivariant logits and softmax normalization over the same incidence neighbourhoods would also be permutation-equivariant.*

The kernel plays a different role. A finite-dimensional feature map expresses the content interaction as $\psi(q_\sigma)^\top \psi(k_\tau)$ and permits the value-weighted neighbourhood information to be accumulated into the fixed-size aggregate $\mathbf{S}_\sigma$. Consequently, the first output block admits the exact support-weighted factorization of Lemma 2. For fixed feature dimensions, this aggregation can be evaluated over the nonzero incidence relations without constructing a dense pairwise interaction matrix. Standard softmax attention does not admit the same exact finite-dimensional bilinear factorization, although sparse softmax attention over the same incidence support can likewise be evaluated without dense all-pairs attention.

Remark 3 (Implementation of structural coefficients $a_{\sigma,\tau}$.)**.** *For each typed relation matrix R associated with a neighbourhood route, we instantiate the index-free coefficient as $a_{\sigma\tau} = g^{\mathrm{str}}_{\sigma\tau} g^{\Delta}_{\sigma\tau} \mathbf{1}[\tau \in \mathcal{N}(\sigma)]$, where σ denotes the destination cell, τ the source cell, and $w_{\sigma\tau} = |R_{\sigma\tau}|$ is the magnitude of their relation weight. For the corresponding relation graph, we define the destination and source degrees as $d_\sigma = \sum_\tau |R_{\sigma\tau}|$, $d_\tau = \sum_\sigma |R_{\sigma\tau}|$, respectively. The two gates are*

$$g^{\mathrm{str}}_{\sigma\tau} = \mathrm{sigmoid}\Big(\mathrm{MLP}_{\mathrm{str}}\left[w_{\sigma\tau}, (d_\tau + 1)^{-1/2}, (d_\sigma + 1)^{-1/2}\right]\Big), \tag{22}$$

$$g^{\Delta}_{\sigma\tau} = \mathrm{sigmoid}(\mathrm{MLP}_{\Delta}\left[h_\sigma, w_{\sigma\tau}\right]). \tag{23}$$

The support indicator $\mathbf{1}[\tau \in \mathcal{N}_R(\sigma)]$ determines which source–destination pairs may interact. The structural gate $g^{\mathrm{str}}_{\sigma\tau}$ modulates each admissible interaction using only relation and degree information, whereas $g^{\Delta}_{\sigma\tau}$ further conditions this coefficient on the current representation h_σ of the destination cell. The source feature h_τ does not enter these gates directly; source–target content interaction is introduced separately by the kernel term $\psi(q_\sigma)^\top \psi(k_\tau)$.

The MLP parameters are shared across cell pairs within each relation type . Under a rank- and incidence-preserving relabeling, a typed relation matrix transforms as $R' = P_{\mathrm{dst}} R P_{\mathrm{src}}^\top$, so relation weights and the corresponding source and destination degrees are permuted consistently, while $h'_{\pi\sigma} = h_\sigma$ for transformed cell features. Consequently, $a'_{\pi\sigma,\pi\tau} = a_{\sigma\tau}$, and the implemented coefficients satisfy the index-free condition in Definition 3.

B.4 Expressive Power of CCMamba

We characterize the discriminative power of CCMamba-K in terms of the combinatorial-complex Weisfeiler–Lehman refinement. The section is organized so that each statement uses only the ones preceding it: Definitions 6–8 fix the refinement, Proposition 3 records its soundness, Lemma 4 and Proposition 4 establish the *upper* bound, Lemma 5 and Proposition 5 supply the ingredients of the *lower* bound, and Theorem 1 combines the two.

Definition 6 (**Labeled combinatorial complex**)**.** *A labeled combinatorial complex is a combinatorial complex $\mathcal{CC} = (\mathcal{S}, \mathcal{C}, \mathrm{rk})$ with a colouring $\ell : \mathcal{C} \to \mathbb{N}$. For a cell $\sigma \in \mathcal{C}$ we call $\ell(\sigma)$ its label.*

Definition 7 (1-CCWL refinement)**.** *Let $\mathcal{N}_{r\to k}(\sigma)$ denote the set of rank-r source cells incident to a rank-k target cell σ, and $\mathcal{R}(\sigma) \subseteq \{0, 1, 2\}$ its set of incident ranks. We initialize $\ell^{(0)}_\sigma = \mathrm{HASH}_0(\mathrm{rk}(\sigma), x_\sigma)$, where HASH_0 is injective; for featureless complexes, x_σ is taken to be constant within each rank. The Combinatorial Complex Weisfeiler–Lehman (CCWL) refinement iterates*

$$\ell^{(t+1)}_\sigma = \mathrm{HASH}\left(\ell^{(t)}_\sigma, \left(\{\{\ell^{(t)}_\tau\}\}_{\tau \in \mathcal{N}_{r\to k}(\sigma)}\right)_{r \in \mathcal{R}(\sigma)}\right), \qquad k = \mathrm{rk}(\sigma), \tag{24}$$

where HASH *is injective on its countable domain and $\{\{\cdot\}\}$ denotes a multiset.*

Instantiating equation 24 on a 2-dimensional complex recovers the familiar node/edge/face form: with E_v the edges incident to node v, V_e and F_e the nodes and faces incident to edge e, and E_f the edges incident to face f,

$$\begin{aligned}
\ell^{(t+1)}_v &= \mathrm{HASH}\big(\ell^{(t)}_v,\ \{\{\ell^{(t)}_e\}\}_{e\in E_v}\big), \quad E_v = \mathcal{N}_{1\to 0}(v) \\
\ell^{(t+1)}_e &= \mathrm{HASH}\big(\ell^{(t)}_e,\ \{\{\ell^{(t)}_v\}\}_{v\in V_e},\ \{\{\ell^{(t)}_f\}\}_{f\in F_e}\big), \quad V_e = \mathcal{N}_{0\to 1}(e), F_e = \mathcal{N}_{2\to 1}(e) \\
\ell^{(t+1)}_f &= \mathrm{HASH}\big(\ell^{(t)}_f,\ \{\{\ell^{(t)}_e\}\}_{e\in E_f}\big). E_f = \mathcal{N}_{1\to 2}(f)
\end{aligned} \tag{25}$$

Definition 8 (**Distinguishability**)**.** 1-*CCWL* distinguishes $\mathcal{CC}_1$ *and* $\mathcal{CC}_2$ *if there is $t \geq 0$ with*

$$\{\{\ell^{(t)}_{\mathcal{CC}_1,\sigma} : \sigma \in \mathcal{CC}_1\}\} \ \neq\ \{\{\ell^{(t)}_{\mathcal{CC}_2,\sigma} : \sigma \in \mathcal{CC}_2\}\}. \tag{26}$$

Proposition 3. *If 1-CCWL distinguishes two attributed combinatorial complexes $\mathcal{CC}_1$ and $\mathcal{CC}_2$, then they are non-isomorphic under any rank-, incidence-, and feature-preserving bijection.*

Proof. We prove the contrapositive. Let $\varphi : \mathcal{CC}_1 \to \mathcal{CC}_2$ be a rank-, incidence-, and feature-preserving bijection, so that $\mathrm{rk}(\varphi(\sigma)) = \mathrm{rk}(\sigma)$, $x_{\varphi(\sigma)} = x_\sigma$, and $\varphi(\mathcal{N}_{r\to k}(\sigma)) = \mathcal{N}_{r\to k}(\varphi(\sigma))$ for all σ and r, where $k = \mathrm{rk}(\sigma)$. We claim

$$\ell^{(t)}_{\mathcal{CC}_2,\varphi(\sigma)} = \ell^{(t)}_{\mathcal{CC}_1,\sigma} \qquad \text{for all } t \geq 0,\ \sigma \in \mathcal{CC}_1. \tag{27}$$

For $t = 0$, this holds because φ preserves rank and initial features. Assuming equation 27 at round t, incidence preservation makes φ restrict to a bijection $\mathcal{N}_{r\to k}(\sigma) \to \mathcal{N}_{r\to k}(\varphi(\sigma))$ for each r, and the inductive hypothesis equates the labels of corresponding cells. Hence, the neighbour multisets in equation 24 coincide, and equation 27 holds at round $t+1$. Since φ is a bijection, the global label multisets agree at every round t. □

The upper bound needs only that a layer is a function of the cell's own representation together with its neighbour multisets not that it is injective.

Lemma 4 (A CCMamba layer is a multiset function)**.** *Let $\mathcal{A}$ denote one CCMamba-K layer followed by ϕ_2 and ϕ_1. There is a map Λ, independent of the complex and of any cell indexing, with*

$$h^{(t+1)}_\sigma = \Lambda\Big(h^{(t)}_\sigma,\ \big(\{\!\{h^{(t)}_\tau\}\!\}_{\tau\in\mathcal{N}_{r\to k}(\sigma).}\big)_{r\in\mathcal{R}(\sigma)}\Big), \qquad k = \mathrm{rk}(\sigma). \tag{28}$$

Consequently, if $\ell^{(j)}_{\mathcal{CC}_1,\sigma_1} = \ell^{(j)}_{\mathcal{CC}_2,\sigma_2}$ for all $j \leq t$, then $h^{(t)}_{\mathcal{CC}_1,\sigma_1} = h^{(t)}_{\mathcal{CC}_2,\sigma_2}$.

Proof. By equation 4, the lifted input $X^{(t)}_k$ is assembled from H_k and the incidence products $I_k^\top H_{k-1}$, $I_{k+1}H_{k+1}$, each a sum over an incidence-induced neighbourhood and hence a function of the corresponding multiset. By equation 9, the state $(\mathbf{S}_\sigma, \mathbf{z}_\sigma)$ is likewise a sum over $\mathcal{N}(\sigma)$; Combiner in equation 6 sums over $\mathcal{N}_{r\to k}(\sigma).$; and ϕ_2, ϕ_1 act on the resulting aggregates, applied identically at every cell. No step reads a cell index, by Lemma 1. Composing these maps yields Λ, which depends on the layer parameters alone.

Assume the initial colour $\ell^{(0)}_\sigma$ encodes $\mathrm{rk}(\sigma)$ together with the input feature, so that $\ell^{(0)}_{\sigma_1} = \ell^{(0)}_{\sigma_2}$ implies $h^{(0)}_{\sigma_1} = h^{(0)}_{\sigma_2}$ and $\mathcal{R}(\sigma_1) = \mathcal{R}(\sigma_2)$. We show by induction on t that $\ell^{(t)}_{\sigma_1} = \ell^{(t)}_{\sigma_2}$ implies $h^{(t)}_{\sigma_1} = h^{(t)}_{\sigma_2}$; the base case is the above. Suppose the claim holds at t and let $\ell^{(t+1)}_{\sigma_1} = \ell^{(t+1)}_{\sigma_2}$. Since HASH is injective, equation 24 gives $\ell^{(t)}_{\sigma_1} = \ell^{(t)}_{\sigma_2}$ and, for each r, $\{\!\{\ell^{(t)}_\tau\}\!\}_{\tau\in\mathcal{N}_{r\to k}(\sigma_1)} = \{\!\{\ell^{(t)}_\tau\}\!\}_{\tau\in\mathcal{N}_{r\to k}(\sigma_2)}$. Fix a bijection $\pi_r : \mathcal{N}_{r\to k}(\sigma_1) \to \mathcal{N}_{r\to k}(\sigma_2)$ with $\ell^{(t)}_\tau = \ell^{(t)}_{\pi_r(\tau)}$, which the multiset equality provides. Applying the inductive hypothesis to each pair $(\tau, \pi_r(\tau))$ gives $h^{(t)}_\tau = h^{(t)}_{\pi_r(\tau)}$, so the feature multisets coincide as well. Both arguments of Λ therefore agree at σ_1 and σ_2, and equation 28 yields $h^{(t+1)}_{\sigma_1} = h^{(t+1)}_{\sigma_2}$. Since colour refinement is monotone, $\ell^{(t)}_{\sigma_1} = \ell^{(t)}_{\sigma_2}$ implies $\ell^{(j)}_{\sigma_1} = \ell^{(j)}_{\sigma_2}$ for every $j \leq t$, as $\ell^{(j)}$ is recoverable from $\ell^{(t)}$ through the injective hashes, then the hypothesis of the lemma may equivalently be stated at round t alone. □

Proposition 4 (1-CCWL upper bound for CCMamba-K)**.** *Assume that each CCMamba-K layer satisfies the conditions of Lemma 4, and that* READOUT *is permutation-invariant. For any two combinatorial complexes $\mathcal{CC}_1$ and $\mathcal{CC}_2$, if 1-CCWL does not distinguish them, then $\mathcal{A}(\mathcal{CC}_1) = \mathcal{A}(\mathcal{CC}_2)$. Equivalently, if $\mathcal{A}(\mathcal{CC}_1) \neq \mathcal{A}(\mathcal{CC}_2)$, then 1-CCWL distinguishes $\mathcal{CC}_1$ and $\mathcal{CC}_2$.*

Proof. Suppose 1-CCWL does not distinguish $\mathcal{CC}_1$ and $\mathcal{CC}_2$. Then, at every refinement round, the two complexes have the same multiset of 1-CCWL colours. By Lemma 4, cells with the same 1-CCWL colour receive the same CCMamba-K representation. Hence, at every network depth T,

$$\{\!\{h^{(T)}_{\mathcal{CC}_1,\sigma} : \sigma \in \mathcal{CC}_1\}\!\} = \{\!\{h^{(T)}_{\mathcal{CC}_2,\sigma} : \sigma \in \mathcal{CC}_2\}\!\}. \tag{29}$$

Since READOUT is permutation-invariant, equal multisets of final cell representations yield equal outputs $\mathcal{A}(\mathcal{CC}_1) = \mathcal{A}(\mathcal{CC}_2)$. The stated implication follows by contraposition. □

Proposition 5. *Let $\tilde{y}_\sigma = \psi(q_\sigma)^\top \mathbf{S}_\sigma / \psi(q_\sigma)^\top \mathbf{z}_\sigma$ be the normalized variant of equation 10, with $a_{\sigma\tau} \equiv 1$ on $\mathcal{N}(\sigma)$. Then $\tilde{y}_\sigma$ is a weighted mean of $\{v_\tau\}_{\tau \in \mathcal{N}(\sigma)}$ and is invariant to uniform duplication of the neighbourhood multiset: $\mathcal{M}$ and $m\mathcal{M}$ yield the same output for any $m \in \mathbb{N}$. Consequently, normalization collapses multiset distinctions that 1-CCWL preserves.*

Proof. Take σ_1, σ_2 of equal rank with $\mathcal{M}_1 = \{\{u\}\}$, $\mathcal{M}_2 = \{\{u, u\}\}$, and write $k_u = \mathbf{W}_K u$, $v_u = \mathbf{W}_V u$. Then $\mathbf{S}_{\sigma_2} = 2\mathbf{S}_{\sigma_1}$ and $\mathbf{z}_{\sigma_2} = 2\mathbf{z}_{\sigma_1}$, and the factor cancels: $\tilde{y}_{\sigma_1} = \tilde{y}_{\sigma_2} = v_u$ for every $\psi, \mathbf{W}_K, \mathbf{W}_V$. The same cancellation holds for $\mathcal{M} \mapsto m\mathcal{M}$, $m \in \mathbb{N}$. Since equation 24 assigns σ_1, σ_2 distinct colours, the claim follows. □

Proposition 5 motivates retaining the unnormalized aggregate $\mathbf{z}_\sigma$ in equation 10, and the same argument constrains $\mathrm{Combine}_r$ in equation 6 for both instantiations. We next characterize when this aggregate uniquely represents a neighbourhood multiset.

Definition 9 (Integrally strictly positive definite kernel)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact. A bounded measurable symmetric kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is* integrally strictly positive definite *(ISPD) if for every finite signed Borel measure $\nu \neq 0$ on $\mathcal{X}$, $\iint_{\mathcal{X}\times\mathcal{X}} \kappa(x, y)\, \mathrm{d}\nu(x)\, \mathrm{d}\nu(y) > 0$. The Gaussian kernel $\kappa(x, y) = \exp(-\|x - y\|^2/2s^2)$ and the Laplacian kernel are ISPD.*

For an unnormalized sum, injectivity must also preserve multiplicities, which correspond to the masses of the associated finite measure.

Together with Lemmas 1 and 2, this yields the expressivity characterization.

Lemma 5 (Injectivity of unnormalized kernel embeddings)**.** *Let κ be ISPD with canonical feature map $\psi : \mathcal{X} \to \mathcal{H}_\kappa$. For a finite multiset $\mathcal{M} = \{\{x_1, \ldots, x_n\}\}$ define $\Xi(\mathcal{M}) = \sum_{j=1}^n \psi(x_j)$. Then Ξ is injective on finite multisets; hence it preserves both multiplicities and cardinality.*

Proof for Lemma 5. Identify a multiset $\mathcal{M} = \{\{x_1, \ldots, x_n\}\}$ with the finite counting measure $\mu_\mathcal{M} = \sum_{j=1}^n \delta_{x_j}$; this identification is a bijection between finite multisets over $\mathcal{X}$ and finite positive integer combinations of Dirac measures, and $\Psi(\mathcal{M}) = \int_\mathcal{X} \varphi\, \mathrm{d}\mu_\mathcal{M}$ is the kernel embedding of $\mu_\mathcal{M}$.

Let $\mathcal{M}_1 \neq \mathcal{M}_2$ and set $\nu = \mu_{\mathcal{M}_1} - \mu_{\mathcal{M}_2}$. Because the two multisets differ in at least one element or in one multiplicity, ν is a nonzero finite signed measure. By the reproducing property and Fubini's theorem,

$$\big\|\Psi(\mathcal{M}_1) - \Psi(\mathcal{M}_2)\big\|^2_{\mathcal{H}_\kappa} = \Big\langle \int \varphi\, \mathrm{d}\nu,\ \int \varphi\, \mathrm{d}\nu \Big\rangle = \iint \kappa(x, y)\, \mathrm{d}\nu(x)\, \mathrm{d}\nu(y)\ >\ 0, \tag{30}$$

where the final inequality is Definition 9. Hence $\Psi(\mathcal{M}_1) \neq \Psi(\mathcal{M}_2)$. Cardinality is recovered as a special case: if $\mathcal{M}_2$ is obtained from $\mathcal{M}_1$ by duplicating elements, the associated counting measures differ, so their embeddings differ. □

In Theorem 1, we require injectivity after the key projection, i.e., of $\Xi_K(\mathcal{M}) = \sum_{x\in\mathcal{M}} \psi(W_K x)$. This condition is satisfied, for example, when W_K is injective on the attainable representation set and ψ is the canonical feature map of an ISPD kernel. For clarity, let $\bar{h}_\sigma^{(l+1)}$ denote the idealized pre-refinement representation immediately after ϕ_1, before the LayerNorm–FFN post-processing in Eq. (7). The lower-bound construction below concerns $\bar{h}_\sigma^{(l)}$, whereas the upper bound remains valid after any shared cell-wise post-processing.

Theorem 1 (Expressivity of CCMamba-K)**.** *Consider an* idealized *CCMamba-K network: layers of Definition 4 with unsigned incidence matrices, followed by ϕ_2, ϕ_1, and a complex-level* READOUT*. Let the initial colour $\ell_\sigma^{(0)}$ encode $(\mathrm{rk}(\sigma), x_\sigma)$, and let $h_\sigma^{(0)}$ be an injective function of $\ell_\sigma^{(0)}$. Consider:* ❶ *$\Xi_K(\mathcal{M}) = \sum_{x\in\mathcal{M}} \psi(W_K x)$ is injective on bounded-size multisets, and X_σ determines $(\mathrm{rk}(\sigma), h_\sigma)$.* ❷ *$a_{\sigma\tau} \equiv 1$, and the aggregate z_σ is retained unnormalized.* ❸ *$\mathcal{N}(\sigma) = \bigsqcup_{r\in\mathcal{R}(\sigma)} \mathcal{N}_{r\to k}(\sigma)$., and ϕ_1 receives $h_{\mathrm{intra},\sigma}$, i.e. $\bar{h}_\sigma^{(l+1)} = \phi_1\left(\bar{h}_\sigma^{(l)}, h_{\mathrm{intra},\sigma}^{(l+1)}, \phi_2(\cdot)\right)$.* ❹ *$\phi_1$ is injective, and* READOUT *is injective on bounded-size multisets. Then:*

(i) ***Upper bound.*** *Under $a_{\sigma\tau} \equiv 1$ and the first part of* ❸*, with a permutation-invariant* READOUT*, for every $T \geq 0$ there is a map Ψ_T, independent of the complex, with*

$h_\sigma^{(T)} = \Psi_T\big(\ell_\sigma^{(3T)}\big)$. *Hence, if* 1*-CCWL does not distinguish* $\mathcal{CC}_1$ *and* $\mathcal{CC}_2$*, no CCMamba-K network does.*

(ii) ***Lower bound.*** *Under* ❶–❹*, for every* $T \geq 0$ *there is a map* D_T *such that* $\ell_\sigma^{(T)} = D_T\Big(\bar{h}_\sigma^{(T)}\Big)$. *Hence, if* 1*-CCWL distinguishes* $\mathcal{CC}_1$ *and* $\mathcal{CC}_2$ *within* T *rounds, the corresponding idealized depth-*T *CCMamba-K network distinguishes them.*

Under assumptions ❶–❹*, the idealized CCMamba-K specialization has the same distinguishing power as 1-CCWL, up to the constant-factor refinement depth in the upper-bound construction.*

Proof of Theorem 1. Since $\mathcal{CC}$ is finite, all neighbourhood multisets have size at most some $n_{\max} < \infty$, and ❶ is used only on such multisets. We write $A \Rightarrow B$ if $B = g(A)$ for a map g that depends only on the network parameters, not on the complex or on cell indices. By Lemma 1 (with $a_{\sigma\tau} \equiv 1$) and Lemma 3, no step of a layer reads a cell index, so all maps below are of this kind. Since HASH in equation 25 is injective, for $k = \mathrm{rk}(\sigma)$,

$$\ell_\sigma^{(s+1)} \;\Leftrightarrow\; \Big(\ell_\sigma^{(s)}, \big(\{\{\ell_\tau^{(s)}\}\}_{\tau \in \mathcal{N}_{r\to k}(\sigma).}\big)_{r \in \mathcal{R}(\sigma)}\Big), \tag{31}$$

and consequently $\ell_\sigma^{(s')} \Rightarrow \ell_\sigma^{(s)}$ for $s \leq s'$, and $\ell_\sigma^{(s)} \Rightarrow \mathrm{rk}(\sigma)$ for all s.

(i) Upper bound. The case $T = 0$ holds by initialization. Assume $h_\tau^{(T)} = \Psi_T(\ell_\tau^{(3T)})$ for all τ, and set $s = 3T$. ▶ Lifting. Because the incidence matrices are unsigned, the entries of $I_k^\top H_{k-1}$ and $I_{k+1}H_{k+1}$ at σ in equation 4 are sums of h_τ over $\mathcal{N}_{k-1\to k}(\sigma)$ and $\mathcal{N}_{k+1\to k}(\sigma)$. By equation 31, $\ell_\sigma^{(s+1)} \Rightarrow X_\sigma$. ▶ Kernel mixing. The query q_σ depends only on X_σ. By ❸ and $a_{\sigma\tau} \equiv 1$, the quantities S_σ and z_σ in equation 9 split into per-rank sums over $\mathcal{N}_{r\to k}(\sigma)$., each a function of $\{\{X_\tau\}\}$ over that neighbourhood. Hence $\ell_\sigma^{(s+2)} \Rightarrow h_{\mathrm{intra},\sigma}^{(T+1)}$. ▶ Intra-rank Fusion. $\mathrm{Combiner}$ in equation 6 sums $h_{\mathrm{intra},\tau}^{(T+1)}$ over $\mathcal{N}_{r\to k}(\sigma)$., so $\ell_\sigma^{(s+3)} \Rightarrow \mathrm{Combiner}(\sigma)$. The maps ϕ_2, ϕ_1 and the refinement equation 7 act cell-wise, since LayerNorm and FFN operate on each cell's feature vector separately. Hence $\ell_\sigma^{(s+3)} \Rightarrow h_\sigma^{(T+1)}$, which closes the induction. If 1-CCWL does not distinguish the two complexes, their colour multisets agree at round $3T$. Applying Ψ_T cell-wise shows that their feature multisets agree, and the permutation-invariant $\mathrm{READOUT}$ returns equal outputs.

(ii) Lower bound. The case $T = 0$ holds by initialization. Assume $\ell_\tau^{(T)} = D_T(\bar{h}_\tau^{(T)})$ for all τ. By ❸ and ❹, $\bar{h}_\sigma^{(T+1)} \Rightarrow \big(\bar{h}_\sigma^{(T)}, h_{\mathrm{intra},\sigma}^{(T+1)}\big)$. In particular, $h_\sigma^{(T+1)}$ determines $\ell_\sigma^{(T)}$ and the block z_σ of equation 10. By ❷, $z_\sigma = \Xi_K\big(\{\{X_\tau\}\}_{\tau\in\mathcal{N}(\sigma)}\big)$, so ❶ gives

$$z_\sigma \Rightarrow \{\{X_\tau\}\}_{\tau\in\mathcal{N}(\sigma)} \Rightarrow \{\{h_\tau^{(T)}\}\}_{\tau\in\mathcal{N}(\sigma)} \Rightarrow \{\{\ell_\tau^{(T)}\}\}_{\tau\in\mathcal{N}(\sigma)}, \tag{32}$$

where the last step applies D_T element-wise. Each $\ell_\tau^{(T)}$ determines $\mathrm{rk}(\tau)$, and by ❸ $\mathcal{N}(\sigma)$ is the disjoint union of the $\mathcal{N}_{r\to k}(\sigma)$. Splitting by rank therefore recovers every per-rank multiset, and equation 31 gives $h_\sigma^{(T+1)} \Rightarrow \ell_\sigma^{(T+1)}$. Now suppose 1-CCWL separates the complexes at round T. If their multisets of $h^{(T)}$ were equal, applying D_T cell-wise would make their colour multisets equal, a contradiction. Hence the feature multisets differ, and by ❹ so do the outputs of $\mathrm{READOUT}$. Combining (i) and (ii) yields the final proof. □

C EXPERIMENTS SETUPS

C.1 GRAPH DATASETS

Graph Datasets We evaluate CCMamba on a collection of real-world graph benchmarks covering both graph-level and node-level classification tasks. For **graph classification**, we consider four standard TUDatasets[1] Morris et al. (2020a): MUTAG and PROTEINS, which contain molecular and protein graphs, respectively, and the social-network benchmarks IMDB-BINARY and IMDB-MULTI. We report the mean and standard deviation of test accuracy following a fixed train/validation/test splitting protocol and report the mean and standard deviation of test accuracy over five random seeds $\{0, 3, 5, 7, 9\}$. For **node classification**, we use Amazon-Ratings, Roman-Empire, and Minesweeper from the Heterophilic Graph Dataset Platonov et al. (2023), together with the citation-network benchmarks Cora, CiteSeer, and PubMed Yang et al. (2016). Test accuracy is used as the evaluation metric for all node-classification datasets. To evaluate CCMamba across different higher-order domains, we further transform each base graph into hypergraph, simplicial-complex, and cellular-complex representations using the structural lifting operators implemented in TopoBench[2] Telyatnikov et al. (2024). Within each higher-order domain, CCMamba and the corresponding baseline methods are evaluated on the same lifted structures, so that performance differences are not attributable to different lifting procedures. Dataset statistics are summarized in Table C.1, followed by brief descriptions of the individual benchmarks.

Table C.1: The statistics of graph datasets.

Datasets	Nodes	Edges	Features	Classes	Tasks
MUTAG	3371	7442	7	2	graph classification
PROTEINS	43471	81044	3	2	graph classification
IMDB-BINARY	19770	193100	0	2	graph classification
IMDB-MULTI	19500	98910	0	3	graph classification
AMAZON-RATINGS	24492	93050	300	5	node classification
ROMAN-EMPIRE	22662	32927	300	18	node classification
MINESWEEPER	10000	39402	7	2	node classification
Cora	2708	10556	1433	7	node classification
CiteSeer	3327	9104	3703	6	node classification
PubMed	19717	88648	500	3	node classification

- **MUTAG** dataset is a collection of nitroaromatic compounds, where nodes represent atoms labelled by atomic type, and edges denote the bridges between atoms.
- **PROTEINS**: A protein classification benchmark (enzymatic or non-enzymatic) where nodes represent amino acids and edges connect residues within a 6 Å spatial threshold.
- **IMDB-B** & **IMDB-M**: Movie collaboration datasets consisting of actor ego-networks derived from IMDb. Nodes represent actors, and edges connect individuals who co-appear in the same film. The graph-level task is to classify these ego-networks into specific movie genres: Action or Romance for IMDB-B, and Comedy, Romance, or Sci-Fi for IMDB-M.
- **AMAZON-RATINGS**: This dataset consists of Amazon product co-purchase network, where nodes represent products and edges denote items purchased together.
- **ROMAN-E**: A heterophilic word-dependency graph derived from the English Wikipedia article on the Roman Empire. Nodes represent words, with edges explicitly encoding both sequential adjacency and syntactic dependencies within sentences.
- **MINESWEEPER**: The node of this dataset corresponds to a grid square in the Minesweeper game. Each square connects to its eight neighbours, forming a set of edges.
- **Cora**, **Citeseer**, and **PubMed**: These three datasets are datasets for node classification. Each node represents a paper, while edges are citation relations between papers.

C.2 HIGHER-ORDER GRAPH BENCHMARKS

Deep Hypergraph Benchmarks Li et al. (2026) Beyond these academic-network benchmarks, we further include datasets from DHG-Bench [3] , which provides a broad evaluation suite for hypergraph node classification. DHG-Bench spans multiple domains and structural regimes, including homophilic, heterophilic, traditional, and fairness-oriented hypergraphs. The homophilic group contains datasets such as Cora, Pubmed, DBLP, and Walmart, where nodes connected by the same hyperedge tend to

[1] https://github.com/chrsmrrs/tudataset

[2] https://github.com/geometric-intelligence/TopoBenchmark

[3] https://github.com/Coco-Hut/DHG-Bench

share similar labels. The heterophilic group includes datasets such as Amazon-Ratings, and Pokec, where hyperedge membership does not necessarily imply label agreement. The traditional group covers classic hypergraph learning scenarios from text, tabular data, 3D vision, and human action recognition, including datasets such as 20NewsW100, Mushroom. The fairness group includes German, Bail, and Credit, which are useful for evaluating model behavior on datasets involving sensitive attributes. These hypergraph datasets vary substantially in scale, feature dimensionality, number of classes, hyperedge size, and hyperedge homophily. For example, the benchmark ranges from small datasets such as Zoo to large-scale networks such as Pokec, and includes both highly homophilic datasets such as Pubmed and more heterophilic datasets such as Actor. Dataset statistics are summarized in Table C.2.

Table C.2: Statistics of hypergraph node classification benchmarks. $|\mathcal{V}|$: nodes, $|\mathcal{E}|$: hyperedges, avg $|e|$: average hyperedge size, H_{edge}: hyperedge homophily ratio. Best viewed in color.

Category	Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Feat	Classes	avg $\lvert e\rvert$	H_{edge}
Traditional	20Newsgroups	16,242	100	100	4	654.5	—
	Mushroom	8,124	298	22	2	136.3	—
	House	1,290	340	100	2	34.7	0.51
Homophilic	Cora (co-citation)	2,708	1,579	1,433	7	3.0	0.90
	Pubmed (co-citation)	19,717	7,963	500	3	4.4	0.95
	Cora (co-authorship)	2,708	1,072	1,433	7	4.3	0.80
	DBLP (co-authorship)	41,302	22,363	1,425	6	4.5	0.87
	Walmart	88,860	69,906	100	11	6.6	0.53
Heterophilic	Actor	7,600	33,544	932	5	—	0.37
	Amazon-Ratings(hypergraph)	24,492	42,256	300	10	—	0.41
	Pokec	1,632,803	~800K	65	2	—	0.44
Fair	German	1,000	~800	27	2	—	—
	Bail	18,876	~16,000	18	2	—	—
	Credit	30,000	~25,000	13	2	—	—

Simplicial and Combinatorial complex Benchmarks. MANTRA [4] and Manifold40 [5] are native simplicial-complex benchmarks: their higher-order structure is given by triangulations or triangular meshes, where 0-simplices are vertices, 1-simplices are edges, and 2-simplices are triangular faces. Pm25cc is a native combinatorial-complex whose higher-rank cells are not obtained from mesh triangles or graph clique lifting, but are provided directly as geospatial cells with incidence relations.

MANTRA 2-$\mathcal{M}$ datasets. We further evaluate CC-Mamba on the MANTRA dataset Ballester et al. (2025), which consists of combinatorial triangulations of 2-manifolds and 3-manifolds. These datasets provide intrinsically higher-order structures, explicitly encoding topological invariants that strictly evade standard graph representations, this dataset statistics are detailed in Table C.3. The two classification tasks:

Table C.3: Details of 2-manifolds

|V|	4	5	6	7	8	9	10	Total
2-$\mathcal{M}$	1	1	1	1	43	655	42,426	43,128

- **MANTRA Genus**: Predicting the underlying surface genus.
- **MANTRA Name**: Classifying the specific homeomorphism type of the manifold.

Manifold40. Manifold40 Hu et al. (2022) is the watertight, 2-manifold remesh of ModelNet40 Krahn et al. (2026), comprising 12,311 shapes across 40 object categories, split into 9,843 training and 2,468 test samples (Table C.4) following the standard protocol. Each mesh is naturally represented as a simplicial 2-complex: vertices are 0-simplices, edges are 1-simplices, and triangular faces are 2-simplices. Hence, no graph-to-complex lifting is required; the boundary incidence structure is read directly from the mesh connectivity. We use the simplicial structure for 40-way shape classification.

Pm25cc. Battiloro et al. (2025) Pm25cc [6] is a native geospatial combinatorial-complex benchmark built from a nationwide network of $PM_{2.5}$ air-quality monitoring stations. Unlike graph benchmarks that are subsequently lifted to higher-order domains, Pm25cc is provided directly as a combinatorial

[4] https://github.com/aidos-lab/mantra-benchmarks

[5] https://github.com/lzhengning/SubdivNet

[6] https://github.com/NSAPH-Projects/topological-equivariant-networks

Table C.4: Summary of Pm25cc and Manifold40 benchmarks.

Dataset	Native Domain	Task	Level	#Samples	Target	Metric
Pm25cc	Combinatorial complex	Regression	Node	3,946	$PM_{2.5}$ concentration	MAE
Manifold40	Simplicial complex	Classification	Complex/mesh	12,311	40 object classes	Accuracy / AUROC

complex: 0-cells correspond to monitoring stations, higher-rank cells encode spatially organized regional interactions, and the incidence structure is loaded from the original geospatial complex rather than constructed by a post-hoc clique or k-hop lifting. Each station carries continuous air-quality and geographic attributes, and the task is node-level regression for $PM_{2.5}$ concentration prediction. This benchmark is therefore included as a native combinatorial-complex regression task, complementing the simplicial-complex classification benchmarks.

C.3 Baselines Models

We compare against a broad range of baselines spanning multiple topological domains: Graph neural networks, Hypergraph neural networks, Simplicial complex networks and Cellular complexes networks. All models are evaluated on the same lifted higher-order structures (hypergraphs, simplicial complexes, or cellular complexes). The details of baseline methods as below.

- **Graph Neural Networks: GCN** Kipf (2016): Graph Convolutional Networks adopt the spectral graph convolutions. **GAT** Veličković et al. (2018): Graph Attention Networks that aggregate neighbor features. **GIN** Xu et al. (2018): Graph Isomorphism Networks distinguish node and graph classification tasks. **MbaGCN** He et al. (2025): Graph Convolutional Networks based on the selective state space models.
- **Hypergraph Neural Networks: HyperGCN** Yadati et al. (2019) updates node through weighted hyperedge feature aggregation and **HyperGAT** Ding et al. (2020) extend graph attention to hypergraphs. **UniGNN** Huang & Yang (2021) (including UniGNNII, UniSAGE, UniGIN) unifies graph and hypergraph message passing. **AllSet** Chien et al. (2021) integrates deep sets and set Transformers. **EDGNN** Jo et al. (2021) jointly encodes node features, edge labels during neighbourhood aggregation.
- **Simplicial Neural Networks: SCCN,SCCNN** Yang et al. (2022) learn representations over relaxed simplex complexes. **SCN** Roddenberry et al. (2021); Lecha et al. (2025) introduce simplicial convolutions on simplex complexes. **SANN** Gurugubelli & Chepuri (2024) focuses on computational efficiency via pre-aggregation, while **SGAT** Lee et al. (2022) utilizes multi-head attention for simplicial interactions. **TopoSRL** Madhu & Chepuri (2023) learns topology-preserving representations for simplicial complexes. **RePHINE** Verma et al. (2024) enhances persistent synchronization expressiveness through color filtering of vertices and edges. **TopoMamba** Montagna et al. (2025) is Simplicial selective state space models.
- **Cellular Neural Networks: CCNN/CXN** Hajij et al. (2020) generalize convolutions to cell complexes. **CWN** Bodnar et al. (2021a) enhances expressivity through cellular Weisfeiler-Lehman message passing. **CCCN** Papillon et al. (2025) leverages cellular complexes neural networks to capture arbitrary higher-order interactions and hierarchical relationships.

C.4 Hyperparameters of Models

Hyperparameters for all methods are selected using the validation set under the same model-selection protocol. All baseline results reported in experimental tables are reproduced by us rather than copied from prior literature. The complete configurations [7] and scripts [8] for reproducing both baseline and CCMamba results are available at `https://anonymous.4open.science/r/CCMamba`.

Hyperparameter settings on graph datasets. All graph-domain experiments are trained for 500 epochs and repeated over five random seeds $\{1, 3, 5, 7, 9\}$. Unless otherwise stated, CCMamba-S uses $d_{\text{state}} = 16$ and an expansion ratio of 2. CCMamba-K uses kernel dimension 64, structural hidden

[7] https://anonymous.4open.science/r/CCMamba/configs

[8] https://anonymous.4open.science/r/CCMamba/scripts

dimension 128, residual scale 0.5, dropout 0.3, and learning rate 5×10^{-4}. For graph lifting, we use HypergraphKHopLifting, SimplicialCliqueLifting, and CellCycleLifting for hypergraph, simplicial-complex, and cellular-complex representations, respectively, with ProjectionSum feature lifting where applicable. The maximum cell length is 15 for MUTAG and PROTEINS and 10 for the remaining graph datasets. For MUTAG, PROTEINS, IMDB-BINARY, and IMDB-MULTI, we use a single fixed $50\%/25\%/25\%$ train/validation/test split over graphs. The split is shared by all methods and kept fixed across the five training seeds, so the reported variation reflects training stochasticity rather than changes in the data partition. No cross-validation is used. For Roman-Empire, Amazon-Ratings, Minesweeper, Cora, CiteSeer, and PubMed, we use a fixed $50\%/25\%/25\%$ train/validation/test split over nodes, again shared across all methods and seeds. All results in Tables 1 and 2 are obtained under the same data partitions, evaluation metrics, and five-seed protocol.

Hyperparameter settings on native hypergraph datasets. On native hypergraph benchmarks, CCMamba-S and CCMamba-K use the original node–hyperedge incidence structures without additional lifting or cell-length truncation. Both variants use hidden dimension 128, 2 layers, and are trained for 500 epochs over five random seeds $\{1, 3, 5, 7, 9\}$. CCMamba-S uses $d_{\text{state}} = 8$, expansion ratio 2, dropout 0.3, and learning rate 10^{-3}, while CCMamba-K uses kernel dimension 64, structural hidden dimension 128, residual scale 0.5, and learning rate 5×10^{-4}. We follow the dataset-specific split configurations provided for the coauthorship, cocitation, AllSet, and DHG benchmarks. Most datasets use a random $50\%/25\%/25\%$ train/validation/test split controlled by a dedicated data-split seed, while a small number follow their dataset-specific split handlers. For each dataset, the same partition is used by CCMamba and all corresponding baselines. The exact split configuration for every dataset is provided in the released configuration files.

Hyperparameter settings on complex-domain datasets. For MANTRA, Pm25cc, and Manifold40, we directly use their native higher-order structures without additional lifting or cell-length truncation. MANTRA and Manifold40 are represented as simplicial complexes, while Pm25cc is used as a native combinatorial-complex benchmark. Both CCMamba variants use hidden dimension 128, 2 layers, dropout 0.3, and learning rate 5×10^{-4}, and are trained for 500 epochs over five random seeds $\{1, 3, 5, 7, 9\}$. CCMamba-S uses $d_{\text{state}} = 8$ and expansion ratio 2, while CCMamba-K uses kernel dimension 64, structural hidden dimension 128, and residual scale 0.5. The batch size is 1 for MANTRA and Pm25cc and 4 for Manifold40. For Pm25cc, we use the dataset-generated $70\%/15\%/15\%$ train/validation/test node masks with mask_seed=0, shared across all runs. For MANTRA, the evaluated subset is partitioned using a fixed random $50\%/25\%/25\%$ train/validation/test split; the subset identity is controlled separately from the training seed. For Manifold40, we follow the official train/test division and construct a stratified validation subset from the training portion. The resulting data partitions are shared across methods for each benchmark.

D Additional Experiments

D.1 Lifting Operation Ablation of CCMamba

To obtain higher-order topological inputs, we apply several graph lifting operations, summarized in Table D.1. We categorize them into three groups: connectivity-based lifting, which constructs cells from graph motifs such as cycles, cliques, and local neighbourhoods; feature-based lifting, which forms higher-order relations using node attribute similarity; and global-structure-based lifting, which incorporates non-local connectivity or community information. These lifting choices determine the cells and incidence relations processed by CCMamba.

Table D.2 reports the effect of different lifting strategies on CCMamba. The results show that the choice of lifting has a clear impact on performance, and the best strategy depends on both the target topological domain. For the cellular variant, cycle lifting achieves strong results on several benchmarks, the model with cycle lifting improves over the Discrete Configuration baseline. This suggests that cycle-based cells can be useful for molecular graphs, where cyclic substructures may provide informative higher-order patterns. For the hypergraph variant, KNN lifting performs competitively, reaching 82.98% on MUTAG and 74.80% on IMDB-B, indicating that feature similarity based neighbourhoods can complement the connectivity. For the simplex variant, neighbourhood Complex and Graph-Induced lifting generally provide more stable performance than Clique Lifting.

Table D.1: Ablation study of lifting operations on CCMamba.

Variant	Lifting Operation	Description
Graph→Hypergraph	KHop Lifting Telyatnikov et al. (2024)	Treats each node and its k-hop neighbourhood as a hyperedge.
	KNN Lifting Gao et al. (2020); Chen et al. (2025)	Forms hyperedges from each node and its k nearest neighbors.
	ExpanderGraph Bernárdez et al. (2024)	Constructs a sparse expander graph as the lifted hypergraph structure.
	Modularity Maximization Newman (2013)	Detects communities by modularity and forms hyperedges within communities.
Graph→Simplex	KHop Lifting Telyatnikov et al. (2024)	Treats each k-hop neighbourhood as a simplex with its faces.
	Clique Lifting Bodnar et al. (2021a)	Promotes graph cliques to simplices and includes their faces.
	Line Lifting Telyatnikov et al. (2024)	Builds the clique complex of the line graph, treating graph edges as vertices.
	neighbourhood Complex Lifting Matsushita & Wakatsuki (2025)	Forms simplices from vertex sets sharing a common neighbor.
	Graph Induced Lifting Telyatnikov et al. (2024)	Promotes connected subgraphs to simplices.
Graph→Cellular	Discrete Configuration Complex Lifting Bernárdez et al. (2024)	Constructs cells from collision-free k-node configurations of the graph.
	Cell Cycle Lifting Hajij et al. (2022); Bodnar et al. (2021a)	Attaches a 2-cell to each detected graph cycle.

Table D.2: Ablation study of Lifting Operation on CCMamba-S

		MUTAG	PROTEINS	IMDB-B	IMDB-M	ROMAN-E	AMAZON-R	Minesweeper	Cora	Citeseer	PubMed
Hypergraph	KHop Lifting	74.47 ±1.05	**77.42** ±0.01	72.80 ±0.91	34.67 ±1.96	79.60 ±1.21	50.14 ±0.34	**82.36** ±1.20	79.26 ±0.89	76.95 ±0.73	89.19 ±0.51
	KNN Lifting	**82.98** ±1.01	**77.42** ±0.23	**74.80** ±1.01	35.73 ±3.24	**79.67** ±1.21	**51.18** ±2.01	82.28 ±1.54	**79.32** ±0.94	**77.07** ±0.55	89.21 ±0.94
	ExpanderGraph	78.72 ±0.69	77.06 ±2.62	74.40 ±1.68	**36.00** ±1.37	79.62 ±0.58	50.76 ±1.02	81.56 ±1.36	**79.32** ±0.94	75.15 ±2.06	89.13 ±1.18
	Modularity Maximization	78.72 ±0.82	76.70 ±0.16	71.60 ±0.25	34.67 ±2.16	79.17 ±0.31	50.81 ±0.44	81.04 ±0.37	78.14 ±0.58	74.31 ±0.73	**89.49** ±1.09
Simplex	KHop Lifting	72.34 ±3.04	76.34 ±1.61	70.00 ±1.37	30.67 ±0.64	**87.27** ±1.05	48.78 ±3.60	80.76 ±3.17	61.15 ±0.95	**67.35** ±1.43	87.91 ±1.20
	Clique Lifting	74.47 ±2.31	73.84 ±1.18	71.60 ±2.03	32.34 ±1.21	86.87 ±2.01	**48.86** ±3.78	80.84 ±4.65	**62.48** ±2.77	67.23 ±1.52	**88.13** ±1.60
	Line Lifting	76.60 ±1.68	75.27 ±1.47	**74.00** ±1.69	**34.84** ±1.69	86.67 ±2.60	48.69 ±4.26	80.88 ±2.08	61.45 ±1.69	**67.35** ±1.44	87.67 ±1.40
	neighbourhood Complex Lifting	**78.72** ±0.93	**76.70** ±2.51	73.20 ±2.10	34.00 ±2.10	86.89 ±1.68	48.83 ±0.66	80.84 ±2.66	62.04 ±2.71	67.11 ±1.45	87.67 ±2.91
	Graph Induced Lifting	**78.72** ±3.20	**76.70** ±1.65	72.00 ±2.44	32.76 ±2.44	87.13 ±1.39	48.82 ±2.72	**81.72** ±1.24	61.89 ±2.15	67.23 ±2.75	87.85 ±3.40
Cellular	Discrete Complex Lifting	78.72 ±2.75	78.14 ±0.64	71.20 ±0.42	52.53 ±0.10	85.37 ±3.06	**48.80** ±3.26	83.56 ±1.90	88.63 ±3.29	74.50 ±2.80	88.66 ±2.96
	CCMamba*-S(cellular)	**85.11** ±2.40	78.14 ±1.68	**74.81** ±1.85	**50.13** ±3.02	**85.49** ±1.72	**50.34** ±2.74	**84.57** ±3.10	**89.22** ±1.50	**76.95** ±2.90	**89.29** ±1.47

These results indicate that lifting is not merely a preprocessing step, but an important modeling choice that determines which higher-order structures are exposed to CCMamba.

Algorithm D.1 summarizes the graph-to-complex lifting procedure used in CCMamba. Nodes and edges are first represented as rank-0 and rank-1 cells, after which higher-order cells are extracted from structural patterns such as cliques or cycles. Incidence matrices between adjacent ranks are then constructed to encode cross-rank relations. The combinatorial complex and incidence operators serve as the structural input for subsequent rank-aware aggregation and state-space modeling.

Algorithm D.1 Lifting a Graph to a Combinatorial Complex

Input: Graph $G = (V, E)$, features $\mathbf{X}$, maximum rank K, and maximum cell length L
Output: Combinatorial complex $\mathcal{CC}$ and incidence operators $\mathcal{I}$
Initialize $\mathcal{CC} \leftarrow \emptyset$ and $\mathcal{I} \leftarrow \emptyset$
Phase 1: Structural Feature Mapping
    for each node $v \in V$ **do**
        Extract node feature $\mathbf{x}_v$ from $\mathbf{X}$
        Add v to $\mathcal{CC}$ as a 0-cell with feature $\mathbf{x}_v$
    end for
    for each edge $e = (u, v) \in E$ **do**
        Add e to $\mathcal{CC}$ as a 1-cell with edge features
    end for
Phase 2: Higher-Order Structure Extraction and Lifting
    for $k = 2$ **to** K **do**
        $\mathcal{S}_k \leftarrow$ identify candidate k-cells of size at most L (e.g., cliques or cycles)
        Add cells in $\mathcal{S}_k$ to $\mathcal{CC}$ as rank-k cells
    end for
Phase 3: Incidence Construction
    for $r = 1$ **to** K **do**
        $I_r \leftarrow \text{IncidenceMatrix}(r-1, r) \in \mathbb{R}^{n_{r-1} \times n_r}$
        $\mathcal{I} \leftarrow \mathcal{I} \cup \{I_r\}$
    end for
return $\mathcal{CC}, \mathcal{I}$

D.2 Ablation on Sequence Construction Strategies on CCMamba-S

To investigate the impact of sequence construction order strategies on topological messaging in Mamba, we conducted ablation studies on six sequencing strategies applied to CCMamba, covering four topological domains: graph, hypergraph, simplicial complex, and cellular complex. we evaluate the following strategies: (1) Canonical, it preserves the original cell indexing order and serves as the default baseline. (2) Random, it applies a uniformly random permutation at each forward pass to assess order sensitivity; (3) BFS, it derives a breadth-first traversal order from the Hodge Laplacian by treating its off-diagonal nonzero entries as adjacency relations, ensuring that topologically adjacent cells appear contiguously in the input sequence. (4) Degree, it sorts cells in descending order of their Laplacian diagonal entries, prioritizing highly connected cells so that structurally central information enters the SSM state early. (5) Feature Norm, which sorts cells by the ℓ_2 norm of their feature vectors in descending order, placing the most semantically salient cells at the front of the sequence. (6) Rank-aware order are designed by the definition 5.

Table D.3: **Ordering sensitivity of CCMamba.** Test accuracy (%, mean $\pm$ std) under six cell-ordering strategies across four complex domains and ten benchmarks.

Domain	Strategy	Dataset									
		MUTAG	PROTEINS	IMDB-B	IMDB-M	ROMAN-E	AMAZON-R	Minesweeper	Cora	Citeseer	PubMed
Graph	Canonical	76.60 ±6.80	61.29 ±1.51	73.60 ±2.80	**53.07** ±3.12	**86.15** ±0.35	44.65 ±0.88	**86.08** ±0.72	88.18 ±1.51	**76.11** ±2.39	89.37 ±0.55
	Random	74.47 ±10.02	61.29 ±7.78	**74.00** ±3.55	52.00 ±3.51	85.92 ±0.36	44.86 ±0.70	85.80 ±0.54	**88.92** ±1.90	75.39 ±1.87	89.39 ±0.49
	BFS	76.60 ±9.91	74.19 ±7.96	72.40 ±2.72	52.53 ±3.42	85.81 ±0.19	44.29 ±0.54	**86.08** ±0.38	**88.92** ±1.68	**76.11** ±2.26	89.49 ±0.47
	Degree	74.47 ±8.30	67.38 ±4.88	73.20 ±3.09	51.73 ±3.61	86.04 ±0.29	44.88 ±1.19	85.76 ±0.33	**88.92** ±1.89	75.27 ±1.97	89.33 ±0.39
	Feature	78.72 ±9.93	61.29 ±5.48	71.60 ±2.58	49.87 ±2.17	85.88 ±0.31	45.01 ±1.02	85.84 ±0.40	**88.92** ±1.66	75.27 ±1.63	89.45 ±0.57
	Rank-aware	**79.75** ±2.40	**77.42** ±4.12	73.20 ±3.56	52.81 ±3.78	83.46 ±2.40	**52.92** ±2.85	85.52 ±1.10	88.12 ±0.93	**76.11** ±2.60	**89.51** ±1.87
Hypergraph	Canonical	65.96 ±15.68	75.63 ±9.74	70.00 ±2.17	33.87 ±1.37	**86.53** ±0.61	45.27 ±0.54	82.40 ±0.89	88.48 ±0.39	75.03 ±0.55	89.17 ±0.41
	Random	53.19 ±10.09	61.29 ±2.52	72.80 ±2.89	33.87 ±1.37	79.35 ±0.55	**50.78** ±0.82	81.88 ±0.56	**89.22** ±1.19	**76.71** ±1.12	89.88 ±0.37
	BFS	63.83 ±15.04	61.29 ±2.52	72.80 ±3.01	33.87 ±1.37	79.40 ±0.49	46.56 ±1.20	81.36 ±0.15	88.77 ±1.16	74.67 ±0.57	89.09 ±0.39
	Degree	63.83 ±15.04	61.29 ±2.52	**74.80** ±3.31	33.87 ±1.37	81.38 ±0.76	45.91 ±0.63	82.00 ±0.71	88.92 ±0.82	75.75 ±1.44	89.39 ±0.42
	Feature	46.81 ±7.46	61.29 ±2.52	71.20 ±2.30	33.87 ±1.37	80.07 ±0.43	50.35 ±0.60	81.52 ±0.41	88.18 ±0.65	75.75 ±1.46	**89.94** ±0.41
	Rank-aware	**78.72** ±2.10	**76.67** ±2.31	72.40 ±2.66	**46.40** ±3.92	78.52 ±1.83	49.53 ±2.23	**83.48** ±1.05	87.59 ±3.16	75.03 ±4.16	89.53 ±2.63
Simplex	Canonical	**80.85** ±5.12	**79.57** ±2.36	72.40 ±2.44	**53.33** ±3.56	85.47 ±0.20	48.70 ±0.77	80.84 ±0.90	87.44 ±1.92	75.03 ±1.31	88.46 ±0.49
	Random	78.72 ±5.08	78.85 ±2.34	73.20 ±2.85	50.93 ±2.85	**85.56** ±0.41	48.60 ±0.72	80.84 ±0.90	87.44 ±1.90	**75.15** ±1.31	**88.50** ±0.55
	BFS	78.72 ±4.51	**79.57** ±2.63	73.20 ±2.34	50.13 ±1.79	85.49 ±0.30	48.47 ±0.73	80.84 ±0.90	**88.18** ±2.09	**75.15** ±1.32	88.44 ±0.50
	Degree	**80.85** ±4.66	77.06 ±1.91	**73.60** ±3.50	50.13 ±3.20	**85.56** ±0.28	48.46 ±0.67	80.84 ±0.90	87.44 ±1.90	**75.15** ±1.01	88.48 ±0.49
	Feature	78.72 ±4.90	77.78 ±3.83	73.20 ±3.77	50.40 ±2.52	85.51 ±0.23	48.55 ±0.70	80.84 ±0.90	87.59 ±1.94	**75.15** ±1.28	88.46 ±0.39
	Rank-aware	76.60 ±2.48	69.18 ±3.15	71.20 ±2.91	42.27 ±4.28	85.51 ±1.24	**48.90** ±3.53	**81.12** ±1.82	80.71 ±1.56	73.27 ±2.63	87.61 ±2.75
Cellular	Canonical	78.72 ±6.45	77.78 ±2.82	72.40 ±1.93	51.73 ±3.12	85.30 ±0.56	49.22 ±0.89	83.12 ±0.56	**89.22** ±1.97	76.95 ±2.06	88.97 ±0.64
	Random	82.98 ±6.31	74.91 ±2.10	71.20 ±0.95	50.40 ±2.72	77.29 ±0.40	**51.46** ±1.81	84.00 ±0.60	87.89 ±0.76	77.07 ±1.73	89.15 ±0.76
	BFS	**83.23** ±9.44	75.63 ±2.20	72.80 ±1.74	46.93 ±1.94	75.29 ±2.68	47.92 ±1.63	83.26 ±0.74	88.77 ±1.51	**77.19** ±1.91	87.99 ±2.34
	Degree	84.11 ±1.79	75.99 ±2.40	73.20 ±1.88	54.40 ±4.58	85.02 ±0.66	51.05 ±0.36	83.04 ±0.81	88.04 ±1.11	76.95 ±1.79	88.64 ±0.62
	Feature	80.85 ±9.91	76.34 ±2.75	73.20 ±2.49	**55.73** ±5.78	77.30 ±0.42	49.53 ±0.60	83.84 ±0.84	88.97 ±1.02	76.95 ±1.96	89.13 ±0.69
	Rank-aware	85.11 ±2.40	**78.14** ±1.68	**74.81** ±1.85	50.13 ±3.02	**85.49** ±1.72	50.34 ±2.74	**84.57** ±3.10	**89.22** ±1.50	76.95 ±2.90	**89.29** ±1.47

Table D.3 reports the proposed rank-aware strategy generally achieves competitive performance across most settings and tends to exhibit relatively stable behavior when the underlying structure changes. In contrast, heuristic-based strategies such as BFS, Degree, and Feature Norm can perform well on specific datasets, but their relative performance varies noticeably across different structural domains. These results suggest that rank-aware structural ordering provides a useful inductive bias across several domains, although no single serialization is uniformly optimal.

D.3 Expressivity of CCMamba-K

Following Wang & Zhang (2024), Table D.4 compares both variants against subgraph GNNs Bevilacqua et al. (2022); Frasca et al. (2022); Zhao et al. (2022a); Huang et al. (2023), higher-order WL variants Morris et al. (2020b); Zhang et al. (2023), and structural encodings Bouritsas et al. (2022); Ying et al. (2021). CCMamba-K (GIN-style graph) applies CCMamba-K to graph-domain inputs, where the local graph message-passing operator follows a GIN-style aggregation scheme. Specifically, node features are updated by combining the central node representation with summed neighbor messages, followed by an MLP transformation. On top of this GIN-style graph backbone, CCMamba-K introduces incidence-supported factorized kernel mixing to capture higher-order incidence-structured dependencies beyond standard pairwise graph interactions.

CCMamba-S leads on seven of ten datasets, most clearly on IMDB-M (52.81 vs. 48.96) and AMAZON-R (52.92 vs. 49.72); its margins on Cora, PROTEINS, and PubMed fall within one standard deviation and indicate parity. CCMamba-K shows a different profile: it tops MUTAG and reaches 90.36% on ROMAN-E, over ten% above every baseline. On the featureless IMDB datasets it falls below the baselines. The two variants therefore exhibit complementary performance profiles.

Table D.4: Comparison with expressive GNN baselines on graph-domain benchmarks. Test accuracy (Mean±std %).

Model	MUTAG	PROTEINS	IMDB-B	IMDB-M	ROMAN-E	AMAZON-R	Minesweeper	Cora	Citeseer	PubMed
GIN Xu et al. (2018)	78.72 ±2.61	76.70 ±1.87	71.20 ±1.86	46.40 ±4.07	79.92 ±2.20	49.22 ±3.58	83.60 ±0.37	87.59 ±1.92	73.11 ±4.08	**89.51** ±0.94
DS-GNN Bevilacqua et al. (2022)	82.13 ±7.15	71.76 ±2.88	68.32 ±2.69	45.97 ±3.45	75.96 ±0.43	47.34 ±0.57	85.41 ±0.58	78.23 ±1.73	70.28 ±1.17	88.40 ±0.37
SUN Frasca et al. (2022)	79.15 ±6.96	70.97 ±1.13	68.08 ±4.13	46.93 ±3.26	76.16 ±0.51	48.99 ±1.21	85.14 ±0.73	85.29 ±1.42	74.14 ±1.54	88.72 ±0.51
PPGN Maron et al. (2019)	77.45 ±7.15	72.97 ±2.04	68.40 ±1.74	47.31 ±1.72	78.93 ±0.81	48.35 ±0.54	85.81 ±0.55	86.12 ±0.84	74.17 ±1.07	88.53 ±0.49
GNN-AK Zhao et al. (2022a)	80.42 ±6.28	73.91 ±3.21	70.56 ±0.96	48.59 ±2.59	78.48 ±0.45	48.72 ±0.88	85.75 ±0.83	85.91 ±1.29	73.42 ±2.17	88.67 ±0.30
KP-GNN Feng et al. (2022)	79.58 ±7.76	71.69 ±3.06	69.12 ±2.32	48.27 ±2.26	78.79 ±0.81	48.54 ±1.16	**85.87** ±0.55	86.09 ±0.73	74.24 ±1.15	88.50 ±0.50
KC-SetGNN Zhao et al. (2022b)	81.27 ±7.58	70.75 ±1.40	66.56 ±1.76	48.96 ±3.08	75.01 ±0.29	49.72 ±0.60	85.40 ±0.58	86.11 ±1.69	73.64 ±1.55	88.50 ±0.45
I2-GNN Huang et al. (2023)	80.85 ±8.24	72.69 ±3.00	69.44 ±2.39	48.69 ±3.38	78.90 ±0.51	48.67 ±0.31	85.74 ±0.58	86.15 ±0.85	74.17 ±1.07	88.52 ±0.47
GSN Bouritsas et al. (2022)	79.58 ±6.66	74.77 ±5.00	71.28 ±2.14	47.94 ±3.50	76.08 ±0.58	47.12 ±0.63	85.43 ±0.60	78.17 ±1.49	70.71 ±2.51	88.56 ±0.39
Graphormer Ying et al. (2021)	79.58 ±4.90	71.90 ±2.16	68.48 ±4.06	48.00 ±4.57	76.38 ±0.27	49.25 ±0.73	85.06 ±0.40	85.41 ±1.63	73.95 ±1.83	88.61 ±0.60
SparseWL Morris et al. (2020b)	82.98 ±2.22	73.48 ±2.28	69.84 ±3.32	46.93 ±2.10	78.99 ±0.78	48.41 ±0.52	85.76 ±0.54	85.85 ±0.86	74.21 ±1.10	88.54 ±0.50
SWL Zhang et al. (2023)	80.43 ±6.96	71.69 ±3.10	69.52 ±4.58	48.16 ±2.81	78.80 ±0.58	48.40 ±0.67	85.72 ±0.58	85.85 ±0.86	74.19 ±1.10	88.54 ±0.48
CCMamba-S(graph)	79.75 ±2.40	**77.42** ±4.12	**73.20** ±3.56	**52.81** ±3.78	83.46 ±2.40	**52.92** ±2.85	85.52 ±1.10	**88.12** ±0.93	**76.11** ±2.60	**89.51** ±1.87
CCMamba-K(graph)	80.85 ±3.36	74.62 ±1.80	68.32 ±1.68	49.97 ±3.50	82.38 ±0.87	45.37 ±0.60	85.14 ±1.08	81.51 ±1.37	71.43 ±1.66	89.05 ±0.40
CCMamba-K (GIN-style graph)	**84.98** ±4.43	74.55 ±3.18	54.40 ±3.56	41.60 ±3.85	**90.36** ±0.49	47.66 ±0.94	85.04 ±2.42	80.95 ±1.67	73.47 ±1.84	89.27 ±0.51

Table D.5 evaluates structural expressiveness on graph instances that are indistinguishable by graph 1-WL. For BREC and SR25, accuracy denotes the fraction of non-isomorphic pairs successfully separated; for CSL, it denotes 10-way classification accuracy, with 10% corresponding to chance.

As expected from its graph 1-WL expressiveness, GIN Xu et al. (2018) fails to separate the BREC and SR25 instances and performs at chance level on CSL. CCMamba-K separates 6.25% of the BREC pairs and 9.20% of the SR25 pairs. These results indicate that the lifted incidence representation exposes structural information that is unavailable to graph 1-WL and can be exploited by CCMamba-K. Importantly, this does not imply expressiveness beyond 1-CCWL: under the conditions of our theoretical analysis, CCMamba-K remains upper-bounded by 1-CCWL on the resulting combinatorial complex. The gains are also modest—dedicated higher-order architectures achieve substantially higher separation rates on BREC, and the CSL result remains within one standard deviation of chance—so we interpret the experiment as evidence for the benefit of incidence-aware higher-order representation rather than uniformly stronger expressive power.

Table D.5: Results on graph-domain benchmarks.

Model	BREC	SR25	CSL
GIN	0.00±0.00	0.00±0.00	10.00±0.00
CCMamba-K(graph)	**6.25±1.27**	**9.20±1.79**	**11.60±2.89**

D.4 Permutation Equivariance and Order Sensitivity.

We evaluate the sensitivity of CCMamba-S to the ordering of cells under rank-preserving permutations. For each complex, we independently permute the cells within each rank and transform the corresponding incidence operators. The relative equivariance error at rank k is defined as

$$\epsilon_{\text{eq}}^{(k)} = \frac{\left\| F_k(\mathbf{P}\mathbf{X}, \mathbf{P}\mathcal{I}\mathbf{P}^\top) - \mathbf{P}_k F_k(\mathbf{X}, \mathcal{I}) \right\|_F}{\left\| F_k(\mathbf{X}, \mathcal{I}) \right\|_F + \varepsilon}, \tag{33}$$

where $\mathbf{P} = \mathrm{diag}(\mathbf{P}_0, \ldots, \mathbf{P}_K)$ applies independent permutations within each rank. We additionally measure the change in the final model prediction under the same permutations. All results are averaged over 100 random rank-preserving permutations.

Table D.6: Permutation sensitivity on MUTAG under 100 rank-preserving random permutations. Lower values indicate stronger permutation equivariance.

Model	Rank-0 ϵ_{eq}	Rank-1 ϵ_{eq}	Prediction gap
GAT	$1.92\times10^{-7} \pm 6.64\times10^{-8}$	–	$3.71\times10^{-7} \pm 2.51\times10^{-7}$
CCMamba-S (Forward)	1.07 ± 0.18	0.58 ± 0.16	1.10 ± 0.90
CCMamba-S (BiMamba)	1.22 ± 0.16	0.80 ± 0.13	3.59 ± 0.94
CCMamba-K	$1.24\times10^{-6} \pm 1.37\times10^{-7}$	$8.70\times10^{-6} \pm 1.05\times10^{-7}$	$2.01\times10^{-6} \pm 1.77\times10^{-7}$

Table D.6 shows the two CCMamba variants exhibit a clear qualitative separation under rank-preserving relabelings. CCMamba-S incurs $O(1)$ equivariance errors at both ranks, demonstrating that its outputs depend strongly on the imposed within-rank ordering. Notably, replacing the forward scan with bidirectional scanning does not remove this dependence, confirming that bidirectionality mitigates directional scan asymmetry but does not establish permutation equivariance. In contrast, CCMamba-K reduces the representation-level equivariance error to the 10^{-6}–10^{-5} range and the

prediction discrepancy to approximately 10^{-6}, several orders of magnitude below those of CCMamba-S. Its behavior is therefore consistent with the permutation-equivariant construction of the incidence-defined kernel mixer, up to finite-precision numerical effects.

D.5 Deep-layered CCMamba-S vs. Over-Smoothing Problem

Convolutional operators oversmooth with depth Yang et al. (2022); Wu et al. (2023b); Zhang et al. (2026), a bottleneck shared by GCN Kipf (2016) and its higher-order counterparts. Figure D.1 evaluates CCMamba variants demonstrate remarkable resilience that against over-smoothing as depth increases from 2 to 64 layers. Conventional GCNs and higher-order convolutional models (HyperGCN, SCCN, CCCN) suffer significant performance degradation with increased depth. For example, on Cora, HyperGCN collapses from 84.34% to 12.70% at 8 layers, while SCCN drops to 41.65% at 32 layers (41.60% reduction). GCN also exhibits a consistent decline, falling to 73.26% at 64 layers (15.79% reduction). In contrast, CCMamba remains remarkably stable. On Cora, CCMamba (graph) maintains consistent accuracy from 2 to 32 layers (87.44% vs. 87.74%), showing only marginal decay even at extreme depths. This stability stems from the selective state-space mechanism, which replaces passive local averaging with input-dependent transitions. By adaptively gating information across topological structures, CCMamba preserves discriminative signals even in deep architectures, effectively mitigating the over-smoothing problem common in topological deep learning.

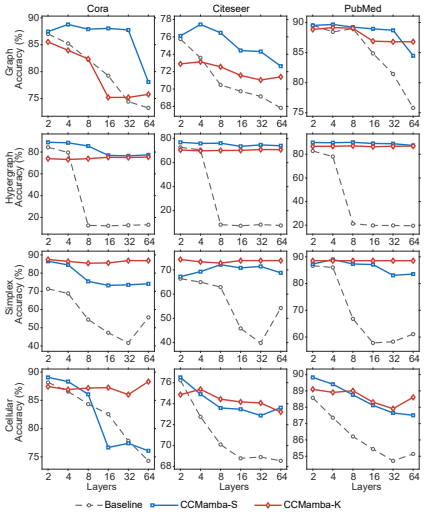


Figure D.1: Test accuracy (%) of CCMamba-S with different layer depths (2, 4, 8, 16, 32, 64) in node-level task.

D.6 Ablation of Aggregate Normalization and LayerNorm

Table D.7 separates the two choices. Proposition 5 shows that normalized aggregation is invariant to uniform duplication and Lemma 5 that the unnormalized kernel embedding retains multiplicity, so Theorem 1 requires the uncontracted z_σ for the stated 1-CCWL condition.

Table D.7: Aggregate normalization and LayerNorm ablation on CCMamba-K.

Variant	Graph			Hypergraph			Simplex			Cellular		
	PROTEINS	ROMAN-E	Cora	PROTEINS	ROMAN-E	Cora	PROTEINS	ROMAN-E	Cora	PROTEINS	ROMAN-E	Cora
Unnorm. z_σ + LN	**77.42** ±3.53	**89.36** ±0.43	**87.82** ±1.36	77.78 ±2.23	**66.48** ±0.36	**74.74** ±1.07	79.57 ±2.28	**87.86** ±0.80	**88.92** ±1.33	77.36 ±2.13	**85.57** ±0.23	**88.63** ±1.57
Unnorm. z_σ + No LN	74.19 ±6.28	88.12 ±0.58	79.76 ±1.27	61.29 ±2.52	64.45 ±0.28	65.07 ±0.96	77.78 ±3.71	87.26 ±0.98	88.33 ±1.42	70.25 ±6.02	85.55 ±0.43	88.48 ±1.55
Norm. $\tilde{y}_\sigma$ + LN	**77.42** ±3.32	88.00 ±0.32	86.41 ±1.60	**78.49** ±2.20	66.47 ±0.36	**74.74** ±1.09	**80.65** ±5.34	87.06 ±0.67	88.77 ±1.50	**78.85** ±2.69	84.08 ±0.46	88.33 ±1.38
Norm. $\tilde{y}_\sigma$ + No LN	73.48 ±7.36	86.83 ±0.48	80.35 ±1.22	75.27 ±7.93	66.03 ±0.22	74.30 ±1.25	76.34 ±2.29	87.47 ±1.16	88.33 ±1.12	61.29 ±2.52	84.80 ±0.77	88.07 ±1.47
Remove z_σ + LN	75.63 ±3.50	88.26 ±0.63	86.26 ±1.87	77.42 ±3.62	**66.48** ±0.36	**74.74** ±1.05	76.34 ±2.20	87.33 ±1.05	88.77 ±1.65	76.21 ±3.47	83.64 ±0.34	88.33 ±1.40
Remove z_σ + No LN	61.29 ±2.52	86.89 ±0.59	80.95 ±1.14	75.63 ±9.49	66.04 ±0.22	74.30 ±1.37	75.99 ±1.99	86.62 ±0.99	88.18 ±1.47	67.74 ±5.01	84.61 ±0.53	88.07 ±1.54

The unnormalized variant with LayerNorm is best or tied-best on eight of twelve columns, but its margin over the normalized variant is narrow—both reach 74.74% on hypergraph Cora, and normalization leads on simplicial and cellular PROTEINS. Retaining z_σ is thus necessary for the guarantee without being decisive on benchmarks that separate few pairs by multiplicity alone; against the ablated variant it is worth 1.8 and 3.2% on graph and simplicial PROTEINS. LayerNorm matters more than either choice. Every variant loses accuracy without it, with the largest drops on PROTEINS(14.3%) for the ablated variant on graph, 17.6 for the normalized variant on cellular. Unnormalized aggregation leaves the state magnitude scaling with neighbourhood size, so the theoretical condition is practical only when LayerNorm keeps that magnitude in range.

D.7 Mixer Ablation with Matched Structural Inputs

To further distinguish CCMamba-K from an incidence-based attention layer, we additionally compare it against message passing, softmax attention, and kernel-attention mixers under the same incidence lifting, support, backbone, and training protocol. Results are reported as the mean test accuracy with standard deviation across random seeds. Table D.8 reports CCMamba-K achieves the top performance among all compared mixers, reaching 87.23%, 78.49%, 89.96%, and 77.79% on MUTAG, PROTEINS, Cora, and CiteSeer, respectively. Compared to the best-performing baseline mixer, CCMamba-K provides absolute accuracy gains of 2.38%, 0.79%, 2.22%, and 2.29% across these datasets.

Table D.8: Matched mixer ablation.

Model	MUTAG	PROTEINS	Cora	CiteSeer
Incidence-MP	82.70 ±4.31	76.41 ±2.22	87.09 ±0.44	73.93 ±1.43
Gated-Incidence-MP	84.30 ±3.16	77.70 ±3.09	86.53 ±1.12	75.50 ±2.13
Incidence-Softmax	84.15 ±3.81	73.05 ±2.58	87.74 ±1.03	75.41 ±2.33
Incidence-Kernel-Attn	84.85 ±3.42	75.99 ±2.18	86.77 ±1.44	74.31 ±2.43
CCMamba-K	**87.23** ±5.75	**78.49** ±2.76	**89.96** ±1.86	**77.79** ±2.12

Remark 4. *This ablation controls for incidence lifting, structural support, backbone, and training protocol, thereby isolating the choice of mixer. The improvement of performance suggests that the gains of CCMamba-K are not solely attributable to higher-order incidence inputs, but also to its combination of structural gating and factorized kernel mixing. These results are reported over five random seeds and should not be interpreted as claims of statistical significance.*

D.8 Ablation of Kernel feature maps and dimension sensitivity.

Figure D.2 shows that kernelized feature maps can improve CCMamba-K performance. PORF achieves the best result on MUTAG (82.98%), while Gaussian RFF performs best on IMDB-BINARY (72.48%), exceeding the strongest non-kernel mapping by 0.43 and 1.36 %, respectively. The optimal feature dimension is also dataset-dependent: PORF peaks at $d_\psi = 64$ on MUTAG, whereas Gaussian RFF peaks at $d_\psi = 256$ on IMDB-BINARY. Performance does not increase monotonically with d_ψ, indicating that larger kernel dimensions are not necessarily beneficial.

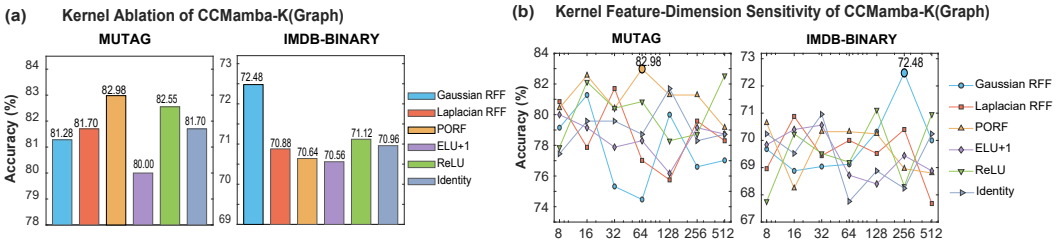


Figure D.2: **Kernel ablation of CCMamba-K.** (a) Performance of different kernel feature maps. (b) Sensitivity to the kernel feature dimension.

D.9 Efficiency Analysis

Let n_k denote the number of rank-k cells and $N = \sum_{k=0}^{2} n_k$ the total number of cells. We define $M_{\mathrm{inc}} = \mathrm{nnz}(I_1) + \mathrm{nnz}(I_2)$ as the number of nonzero incidence pairs, and $M_{\mathcal{N}} = \sum_{\sigma \in \mathcal{C}} |\mathcal{N}(\sigma)|$ as the number of directed relations used by the structural mixer. For the adjacent-rank incidence neighbourhoods used in our implementation, $M_{\mathcal{N}} = \mathcal{O}(M_{\mathrm{inc}})$. We further denote by d the hidden dimension, d_s the SSM state dimension, and $r = \dim \mathcal{H}_\psi$ the kernel feature dimension.

CCMamba-S. The sparse incidence operations $I_1 H_1$, $I_1^\top H_0$, $I_2 H_2$, and $I_2^\top H_1$ require $\mathcal{O}(M_{\mathrm{inc}} d)$ operations. Rank-specific linear projections and MLPs contribute $\mathcal{O}(Nd^2)$, while the rank-wise BiMamba blocks cost $\mathcal{O}\big(N(d^2 + d\,d_s)\big)$, where the forward and backward scans introduce only a constant factor of two. Inter-rank fusion adds another $\mathcal{O}(M_{\mathrm{inc}} d)$ sparse aggregation cost and $\mathcal{O}(Nd^2)$ projection cost. Therefore, the overall complexity of one CCMamba-S layer is $\mathcal{O}\big(M_{\mathrm{inc}} d + N(d^2 + d\,d_s)\big)$, and an L-layer network scales linearly with L.

CCMamba-K. By Lemma 2, for each cell σ, CCMamba-K accumulates the structural state $\mathbf{S}_\sigma = \sum_{\tau \in \mathcal{N}(\sigma)} a_{\sigma\tau}\, \psi(k_\tau) \otimes v_\tau$. Each supported relation contributes one rank-one update $\psi(k_\tau) \otimes v_\tau$ at cost $\mathcal{O}(rd)$, resulting in an aggregation cost of $\mathcal{O}(M_{\mathcal{N}} rd)$. Computing the auxiliary aggregate z_σ requires an additional $\mathcal{O}(M_{\mathcal{N}} r)$ operations, while the contraction $\psi(q_\sigma)^\top \mathbf{S}_\sigma$ costs $\mathcal{O}(rd)$ per

cell. The projections W_Q, W_K, and W_V, together with evaluation of the feature map ψ, contribute $\mathcal{O}\big(N(d^2 + dr)\big)$. The structural coefficients $a_{\sigma\tau}$ are evaluated only on supported relations and add $\mathcal{O}(M_{\mathcal{N}} d)$ cost, which is subsumed by the aggregation term for $r \geq 1$. Hence, the total complexity of one CCMamba-K layer is $\mathcal{O}\big(M_{\mathcal{N}} rd + N(d^2 + dr)\big)$. Since $M_{\mathcal{N}} = \mathcal{O}(M_{\mathrm{inc}})$ for the incidence neighbourhoods used in our implementation, this reduces to $\mathcal{O}\big(M_{\mathrm{inc}} rd + N(d^2 + dr)\big)$.

E Limitations and Future works

Limitations. Selective SSMs rely on structured causal mixing that supports efficient recurrent evaluation along an ordered sequence Dao & Gu (2024); Lahoti et al. (2026); Yuan et al. (2025). Extending this mechanism to combinatorial complexes therefore requires adapting sequential state-space computation to rank-structured, permutation-sensitive inputs. CCMamba-S addresses this through rank-aware ordering and bidirectional scanning Hwang et al. (2024), while CCMamba-K instead performs permutation-equivariant mixing over incidence neighborhoods. These two variants thus represent complementary design choices: CCMamba-S emphasizes efficient sequential state-space computation, whereas CCMamba-K emphasizes equivariant higher-order interaction. Further work on canonicalization and structured masking Kaba et al. (2023); Katharopoulos et al. (2020); Huang et al. (2024b) may help unify these properties within a single formulation.

Most benchmarks are graphs converted into complexes using fixed lifting operations Telyatnikov et al. (2024); Papillon et al. (2025). The lifting procedure also determines computational cost: the number of incidence pairs may grow with clique or cycle size Bodnar et al. (2021a), while truncating large cells can remove useful higher-order structure. Although we also evaluate on native higher-order datasets, including MANTRA Ballester et al. (2025), Manifold40 Hu et al. (2022), and Pm25cc Battiloro et al. (2025), these datasets remain small, and some performance differences are within one standard deviation. Larger and more diverse native complexes are therefore needed. Our expressivity analysis focuses on CCMamba-K, whose permutation-equivariant formulation permits a direct connection to Weisfeiler–Lehman-style analysis Xu et al. (2018); Papamarkou et al. (2024a); Carrasco et al. (2025). Under the conditions stated in Theorem 1, the implemented layer admits a 1-CCWL upper bound, while the corresponding lower bound follows under standard injectivity assumptions on multiset embeddings, update functions, and readouts. Extending comparable characterizations to selective scan architectures remains an interesting theoretical direction.

Future Work. The lifting operation determines which higher-order cells and incidence relations are exposed to the model Battiloro et al. (2024); Franco et al. (2026), while existing benchmarks Telyatnikov et al. (2024); Ballester et al. (2025) differ in domains, liftings, splits, and evaluation protocols. Developing larger native combinatorial-complex benchmarks with standardized liftings and protocols would enable more systematic comparison across methods. Attention-based models Wu et al. (2023b), such as GraphGPS Rampasek et al. (2022), simplicial attention networks Lee et al. (2022), and cellular Transformer Barsbey et al. (2025), characterize message passing through learned pairwise weights between nodes, or between nodes and edges. Mamba-based graph models instead propagate information through selective recurrence, and have made progress on over-smoothing He et al. (2025; 2026); Wang et al. (2024a); Behrouz & Hashemi (2024). Their expressive power remains less understood Huang et al. (2024b); Montagna et al. (2025), and these questions have received little attention on higher-order domains. Graph foundation models commonly adopt Transformers to encode graph tokens Guo et al. (2026); Liu et al. (2025). Selective state-space models offer a linear-time alternative to transformer for long token sequences Dao & Gu (2024); Bick et al. (2024), yet their use as backbones for graph pretraining, and especially for higher-order structures, remains unexplored. Open questions include how to tokenize cells of different ranks, how to reconcile scan order at scale across higher-order graph domains.

F LLM USAGE STATEMENT

In preparing this manuscript, LLM was used as a general-purpose assistive tool. Its application was confined to the final stages of writing, specifically for minor polishing and refinement of the English prose to improve clarity and readability. The LLM did not contribute to the research ideation, experimental methodology, data analysis, or the generation of results and conclusions in this work.