TOPOLOGICAL NEURAL OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce **Topological Neural Operators** (TNOs), a principled framework for operator learning on cell complexes that lifts neural operators (NOs) from functions on points and/or edges to discrete topological domains. TNOs represent data as features defined on cells of varying dimension and model their interactions through *Discrete Exterior Calculus*, enabling explicit cross-dimensional coupling via gradient-, curl-, and divergence-type operators. The key design principle is to decouple *where* information flows, as governed by fixed topological operators, from *how* it is transformed (which is learned), yielding models that respect the geometric support of physical quantities and expose conservation and compatibility structure. We further propose **Hierarchical TNOs** (HTNOs), which incorporate coarse complexes to propagate long-range and topology-dependent information. Our framework recovers standard point- and graph-based NOs as restricted cases, providing a unified perspective on operator learning across discretizations. Across a range of PDE benchmarks, including problems on irregular geometries, TNOs and HTNOs improve accuracy and physical residuals. Controlled studies further isolate the benefits of incorporating native higher-rank and topological structure.

1 INTRODUCTION

Modeling and predicting complex physical systems is a central challenge in science and engineering. Many systems, from fluid dynamics to electromagnetism and elasticity, are governed by *partial differential equations* (PDEs), whose solutions under varying parameters, geometries, and boundary conditions underpin applications such as digital twins, design optimization, and real-time simulation (Azizzadenesheli et al., 2024). Classical numerical methods (e.g., finite elements, spectral solvers) are accurate but computationally costly, often requiring precise and fine discretizations.

Operator learning addresses this by learning a solution operator between infinite-dimensional function spaces that captures how PDEs depend on inputs such as coefficients, forcing terms, and geometry (Boullé & Townsend, 2024; Kovachki et al., 2024; Jiao et al., 2025). *Neural operators* (NOs) (Kovachki et al., 2023) have emerged as a powerful instantiation, achieving strong performance and generalization across discretizations and resolutions via variants such as Fourier Neural Operators (FNOs) (Li et al., 2021; Liu et al., 2023) and their geometric or transformer-based extensions (Hao et al., 2023; Alkin et al., 2024). These methods enable orders-of-magnitude speedups over classical solvers while maintaining high fidelity, establishing a scalable paradigm for the physical sciences.

Despite these successes, existing NOs are fundamentally *point-centric*: they represent all physical quantities as functions defined on nodes, with interactions modeled via learned kernels or graph-based message passing. This abstraction neglects a key structural aspect of continuum physics: *physical quantities have geometric type* (Hirani, 2003). Potentials live on vertices, circulations on edges, fluxes on faces, and densities in volumes. Their interactions are governed by gradient, curl, and divergence operators, which couple quantities across dimensions. In compatible numerical discretizations, these relationships are encoded directly through incidence structures, ensuring conservation laws and identities (e.g., $\mathrm{div}\,\mathrm{curl} = 0$) hold by construction[1]. By collapsing all quantities to nodes, current neural operators obscure this structure, limiting their ability to model multi-physics systems, enforce conservation and compatibility, and capture topological effects such as cycles and harmonic modes.

As a remedy, we introduce **Topological Neural Operators (TNOs)**, a principled framework for operator learning on *cell complexes*, the natural discrete domain for multi-dimensional physical

[1]The boundary of a boundary is zero: a boundary shape (like a closed surface) has no boundary of its own.

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

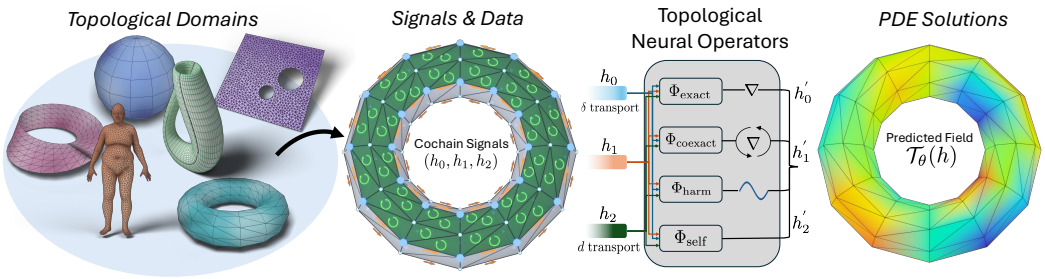


Figure 1: **Topological Neural Operators** operate on cell complexes (i) whose physical signals are cochains at multiple ranks (ii). A TNO layer (iii) couples ranks through fixed DEC operators (d^k, δ^k) and learns the rank-wise channel mixing in four blocks, signifying gradient, curl, harmonic, and self maps, producing a predicted PDE field on the complex (iv).

systems (see fig. 1). Our key design principle is to separate where information flows, governed by fixed topological operators, from how it is transformed, which is learned. Rather than representing data solely on points (or edges), TNOs operate on *cochains*, assigning features to cells of different dimensions (vertices, edges, faces, and higher-dimensional elements), and model their interactions using *Discrete Exterior Calculus* (DEC) (Hirani, 2003). This design explicitly encodes cross-dimensional structure: information flows between cells through incidence-based operators, while learnable components act on feature transformations. As a result, TNOs (i) respect the geometric support of physical quantities, (ii) incorporate conservation and compatibility structure into the architecture, and (iii) enable explicit modeling of multi-degree PDE systems.

This motivates a minimal extension of the NO paradigm: keep operator learning, but replace rank-0 point fields by cochain fields supported on cells of different dimensions. TNOs recover standard point-based NOs when restricted to 0-cochains, while allowing vertex-, edge-, face-, and volume-supported fields to interact through DEC operators. To capture long-range and topology-dependent effects, we further introduce **Hierarchical TNOs** (HTNOs), which propagate information across coarse complexes while preserving the cochain structure. In brief, **our contributions are:**

- A **framework for operator learning on cell complexes**, extending neural operators to cochain-valued fields and explicitly modeling cross-dimensional physical interactions via DEC.
- A **unifying perspective** encapsulating existing NOs as special cases of the proposed framework.
- **Hierarchical TNOs (HTNOs)** for efficient multi-scale propagation of global and topological information.
- **Comprehensive ablation studies** validating the design choices, including PDEs with native multi-rank structure, which show that TNOs effectively isolate physical signals across ranks.

We validate our approach across synthetic and real PDE benchmarks. (H)TNOs achieve improved accuracy on unseen meshes, while exhibiting stronger physical consistency in structured settings. Code and simulated data will be made publicly available.

2 Background and the Language of Discrete Topology

Physical quantities are dimensional: scalars live on vertices, circulations on edges, fluxes on faces, densities in volumes. The laws relating them are statements about boundaries, orientations, and dimensional couplings; this is the language of discrete exterior calculus we use throughout. We provide the necessary background and refer the reader to (Crane et al., 2013; Crane, 2018; Hirani, 2003; Desbrun et al., 2005; Meyer et al., 2003; Desbrun et al., 2006) for a thorough treatment.

2.1 Topological Structures

Definition 2.1 (Regular Cell Complex)**.** *A* regular cell complex *(RCC) is a finite set K together with a rank function* $\mathrm{rk} : K \to \mathbb{Z}_{\geq 0}$ *and* face relation $\sigma \prec \tau$*, defined when* $\mathrm{rk}(\tau) - \mathrm{rk}(\sigma) = 1$*, such that for any* $\tau, \rho \in K$ *with* $\mathrm{rk}(\tau) - \mathrm{rk}(\rho) = 2$*, exactly two elements* $\sigma \in K$ *satisfy both* $\rho \prec \sigma$ *and* $\sigma \prec \tau$*.*

Elements of K are called *cells*; those of rank k form the set K_k, with $n_k = |K_k|$ and $N = \max_\tau \mathrm{rk}(\tau)$. Rank-0 cells are vertices, rank-1 edges, rank-2 faces, and rank-3 volumes. In this representation, physical quantities such as pressure, circulation, magnetic flux, and charge density

are associated with cells of rank $0, 1, 2, 3$ respectively (see tab. 4). Every k-cell is a first-class computational entity able to exchange information with any incident cell. As we show in sec. 3, the PDEs governing Maxwell, Navier–Stokes, and elasticity are precisely statements about how quantities at different ranks interact, a structure graphs cannot represent and RCCs encode natively.

The face relation $\sigma \prec \tau$ records immediate incidence and determines differential operators. For every k-cell τ and $(k-1)$-cell σ one assigns an orientation sign $[\tau : \sigma] \in \{-1, 0, +1\}$, with $[\tau : \sigma] \neq 0$ iff $\sigma \prec \tau$. Assembling these into the *signed incidence matrix* $B_k \in \mathbb{R}^{n_{k-1} \times n_k}$ with $(B_k)_{\sigma\tau} = [\tau : \sigma]$, *the diamond condition* (Basak, 2010; Aschbacher, 1996; Savoy, 2022) forces

$$B_k B_{k+1} = 0 \quad \forall\, k, \tag{1}$$

the discrete condition that **the boundary of a boundary is empty**, which underlies $\mathrm{curl}(\mathrm{grad}) = 0$ and $\mathrm{div}(\mathrm{curl}) = 0$, and is the structural foundation of the physical laws and topology that follow.

A *k-cochain* $u^k : K_k \to \mathbb{R}^{d_k}$ assigns a feature vector to each k-cell; the cochain space is $C^k(K; \mathbb{R}^{d_k}) \cong \mathbb{R}^{n_k \times d_k}$. The degree k is geometrically meaningful: it records the type of object on which the quantity lives, and assigning a field to the wrong degree breaks the structural laws governing it (see sec. A.2). The full multirank signal space is $C^\bullet = \bigoplus_{k=0}^{N} C^k$.

2.2 Discrete Exterior Calculus

From the boundary matrices, one derives three families of operators that serve as the computational primitives of the Discrete Exterior Calculus (DEC) framework (Crane, 2018; Desbrun et al., 2006).

Discrete exterior derivative. The *discrete exterior derivative* $d^k = B_{k+1}^\top : C^k \to C^{k+1}$ lifts a k-cochain to a $(k+1)$-cochain by accumulating signed values over incidence relations. Concretely, d^0 is the discrete gradient (oriented differences across edges), d^1 is the discrete curl (oriented sums around faces), and d^2 is the discrete divergence. The identity $d^{k+1} \circ d^k = 0$ follows directly from eq. (1) and is the discrete counterpart of $\mathrm{curl}(\mathrm{grad}) = 0$ and $\mathrm{div}(\mathrm{curl}) = 0$. The resulting sequence $C^0 \xrightarrow{d^0} \cdots \xrightarrow{d^{N-1}} C^N$ is called the *discrete de Rham complex* of K (see eq. (15)).

Hodge operators and codifferential. Equipping each C^k with a positive-definite *Hodge star*[2] $M_k \in \mathbb{R}^{n_k \times n_k}$ encoding geometric data such as edge lengths, face areas, and cell volumes, gives the formal M_k-adjoint of d^{k-1}, the *codifferential*, mapping downward in degree:

$$\delta^k = M_{k-1}^{-1} B_k M_k : C^k \to C^{k-1}. \tag{2}$$

While d^k is purely topological (determined by B_{k+1} alone), δ^k depends on geometry encoded in M_k.

Hodge Laplacian. Together, d^k and δ^k yield the *k-Hodge Laplacian*

$$\Delta_k = \underbrace{\delta^{k+1} \circ d^k}_{\Delta_k^\uparrow} + \underbrace{d^{k-1} \circ \delta^k}_{\Delta_k^\downarrow} : C^k \to C^k, \tag{3}$$

which is self-adjoint and positive semidefinite with respect to M_k. The decomposition into $\Delta_k^\uparrow$ and $\Delta_k^\downarrow$ is physically meaningful: $\Delta_k^\uparrow$ couples u^k upward to $(k+1)$-cells via d^k (curl-like coupling), while $\Delta_k^\downarrow$ couples u^k downward to $(k-1)$-cells via δ^k (divergence-like coupling). At $k = 0$, $\Delta_0 = \delta^1 d^0$ recovers the weighted graph Laplacian; at $k = 1$, Δ_1 simultaneously involves edges, vertices, and faces, and is thus already an operator with no graph-level analog (Crane, 2018).

Hodge Decomposition (HD). The three operator families organize every cochain space into an orthogonal decomposition with respect to the M_k-inner product:

$$C^k(K; \mathbb{R}) = \underbrace{\mathrm{im}(d^{k-1})}_{\text{exact}} \oplus \underbrace{\ker(\Delta_k)}_{\text{harmonic}} \oplus \underbrace{\mathrm{im}(\delta^{k+1})}_{\text{coexact}}, \tag{4}$$

Exact cochains are *potential-driven*; coexact cochains are *divergence-free*; harmonic cochains are *topologically constrained*: they represent the cohomology of the de Rham complex, so

$$\dim \ker(\Delta_k) = \beta_k \tag{5}$$

[2] See Section A.2 for the precise definition.

where β_k is the k-th Betti number, counting k-dimensional holes (Hirani, 2003; Arnold et al., 2006).

The Hodge Decomposition is central to our TNO for two reasons: it characterizes what physical solutions look like (incompressible flows are coexact; conservative fields are exact; topological modes are harmonic), and it provides the DEC-compatible channels through which conservation and compatibility constraints can be preserved, as we show in sec. 4.1.

The complex K thus encodes everything a learning framework needs: the domain $\{K_k\}_{k=0}^N$ stratified by dimension, the signal spaces $\{C^k(K;\mathbb{R}^{d_k})\}$ where physical fields live, the differential operators $d^k = B_{k+1}^\top$, $\delta^k = M_{k-1}^{-1} B_k M_k$, and $\Delta_k = \delta^{k+1} d^k + d^{k-1}\delta^k$ that move information between dimensions, and the identity $B_k B_{k+1} = 0$ encoding the algebraic backbone of discrete physical laws.

3 Discrete PDEs on Cell Complexes

The operators d^k, δ^k, Δ_k appear directly in the governing equations of many physical systems, coupling fields across dimensions. Let K be a finite RCC of dimension N with Hodge stars $\{M_k\}_{k=0}^N$ as in sec. 2. The de Rham complex provides a language for physical PDEs on bounded domains: a field is stored on the cells where it naturally lives (tab. 4), while d^k, δ^k, and Δ_k describe how that field changes across neighboring dimensions (Desbrun et al., 2006; Arnold et al., 2006).

A *discrete PDE on* $(K, \{M_k\})$ *at degree* k is an equation

$$\mathcal{F}\big(u^k,\ d^k u^k,\ \delta^k u^k,\ \Delta_k u^k,\ a,\ f\big) = 0 \quad \text{in } C^k(K;\mathbb{R}^d), \tag{PDE}$$

where $u^k \in C^k(K;\mathbb{R}^d)$ is the unknown field and $f \in C^k(K;\mathbb{R}^d)$ is the source. The coefficient field $a \in C^\bullet(K;\mathbb{R}^p)$ contains the material or constitutive data that define the particular PDE instance, such as diffusion coefficients, conductivity, permeability, permittivity, or density. It may live on one degree or across several degrees, depending on the physics. The map $\mathcal{F}$ is the PDE residual; solving the PDE means finding u^k so that this residual vanishes.

Boundary conditions are imposed on a subcomplex $\partial K \subseteq K$. Essential conditions prescribe the boundary trace $u^k|_{\partial K}$, while natural conditions prescribe flux-type data and enter through the Hodge-adjoint codifferential δ^k, hence through restricted incidence and Hodge-star operators.

Many physical systems involve more than one degree at once: Maxwell equations couple a 1-form with a 2-form, Darcy couples a 0-form with a 1-form, Navier–Stokes in vorticity form couples 1- and 2-forms. A *multi-degree discrete PDE* is a system

$$\mathcal{F}_k\big(\{u^j\}_{j=0}^N,\ \{d^j u^j\}_j,\ \{\delta^j u^j\}_j,\ a,\ f\big) = 0 \quad \text{in } C^k(K;\mathbb{R}^{d_k}), \quad k = 0,\dots,N, \tag{mPDE}$$

with unknown $u \in C^\bullet(K;\mathbb{R}^{d_\bullet})$. The equation at degree k can depend on $d^{k-1}u^{k-1}$ (information from below, via $\Delta_k^\downarrow$-type coupling), on $\delta^{k+1}u^{k+1}$ (from above, via $\Delta_k^\uparrow$-type coupling), and on $\Delta_k u^k$. The coupling pattern is fixed by the cellular structure of K; the weights, by $\{M_k\}$. A learnable map that processes degrees independently, without d and δ connecting them, cannot represent this.

For time-dependent problems, the unknown is $u : [0,T] \to C^\bullet(K;\mathbb{R}^{d_\bullet})$ satisfying

$$\partial_t u^k(t) = \mathcal{A}_k\big(\{u^j(t)\}_{j=0}^N,\ \{d^j u^j(t)\}_j,\ \{\delta^j u^j(t)\}_j,\ a,\ f(t)\big), \qquad u(0) = u_0, \tag{Evol}$$

encompassing parabolic/hyperbolic problems, and fully coupled multi-physics systems. Discretizing in time yields one-step maps $u(t) \mapsto u(t+\Delta t)$, the natural targets for time-stepping operators.

Examples. In sec. A.3.1, we illustrate (Evol) with two standard systems whose fields live on different cochain degrees and couple through d and δ: discrete Maxwell- and wave-type couplings. In both cases, the dynamics are incidence-driven: fields on different cochain degrees are coupled by d and δ, and the associated constraints are preserved by the algebraic identities of the complex.

4 Topological Neural Operators (TNOs)

Each PDE admits a *solution operator* sending input data (initial conditions, sources) to an output field. For the wave system, a map $C^0 \times C^1 \to C^0 \times C^1$ advances (u,p) in time; for Maxwell, a map $C^1 \times C^2 \to C^1 \times C^2$ advances (E,B). Approximating such an operator $\mathcal{G}$ from data is the subject of operator learning (Kovachki et al., 2023). Here $\mathcal{G}$ acts on tuples of cochains organized by degree, with its action mediated by d, δ, Δ on K. The following defines our class of learnable maps:

Definition 4.1 (Topological Neural Operators)**.** *Let K range over a family of finite cell complexes of dimension N with boundary matrices $\{B_k\}$ and specified geometric data. For $i = 1, \dots, m$ and $j = 1, \dots, n$, let $k_i, \ell_j \in \{0, \dots, N\}$ and $d_i, r_j \geq 1$, and define*

$$\mathcal{X}(K) = \prod_{i=1}^{m} C^{k_i}(K; \mathbb{R}^{d_i}), \qquad \mathcal{Y}(K) = \prod_{j=1}^{n} C^{\ell_j}(K; \mathbb{R}^{r_j}). \tag{6}$$

A Topological Neural Operator (TNO) *is a parameterized family of neural networks $\{\mathcal{T}_\theta^K : \mathcal{X}(K) \to \mathcal{Y}(K)\}_{K, \theta \in \Theta}$ approximating a target operator $\mathcal{G} : \mathcal{X}(K) \to \mathcal{Y}(K)$ and satisfying:*

- ***(P1) Cellular intrinsicness.*** *$\mathcal{T}_\theta^K$ uses the cellular structure and geometry of K, via incidence-supported transports, same-degree spectral maps, trace / restriction maps, and cellwise transformations shared across cells of the same degree. The construction commutes with cell relabeling. Hodge stars and Laplacians optionally supply metric and spectral structure.*
- ***(P2) Multi-degree coupling.*** *Interactions between cochain degrees, when present, follow incidence relations determined by the boundary operators or compositions of these relations. Learned feature transformations are assigned to degree-to-degree routes and shared across cells within each route. Single-degree realizations are also allowed.*
- ***(P3) Discretization transferability.*** *The weights θ are shared across supported discretizations, with a parameter space Θ independent of the number of cells. Cellular and geometric operators are assembled on each complex using the same construction rules.*

Remarks. (**P1**) makes the construction independent of cell ordering; geometric data may include coordinates or Hodge stars. (**P2**) is the key higher-order feature: signals may live on vertices, edges, faces, and higher cells, with $\mathcal{T}_\theta$ coupling these degrees. Such cross-degree coupling is not merely an architectural extension, but is imposed by the physics (e.g., Maxwell/wave PDEs in sec. A.3.1). (**P3**) shares parameters across discretizations. Agreement under prolongation and restriction, or a continuum limit, requires additional assumptions (Kovachki et al., 2023); see sec. C.4.

The choice of degrees $\{k_i\}$ and $\{\ell_j\}$ in dfn. 4.1 determines the operator type: degree-preserving ($C^k \to C^k$) for single-field PDEs; gradient-type ($C^0 \to C^1$) for potential-to-flux maps like Darcy or Fourier; divergence-type ($C^1 \to C^0$) for flux-to-source maps; and multi-degree ($C^1 \times C^2 \to C^1 \times C^2$) for coupled systems like Maxwell. We next detail how $\mathcal{T}_\theta$ is computed (*learned*).

4.1 Realization as a Topological Neural Network

Dfn. 4.1 specifies what a TNO must be (P1–P3); it does not describe how to compute one. We now present a possible architecture that realizes these axioms as a topological neural network (TNN) (Hajij et al., 2023b) using the boundary maps and chosen Hodge stars $\{M_k\}$ that define d, δ, and Δ.

Architecture. A TNN realizing $\mathcal{T}_\theta^K : \mathcal{X}(K) \to \mathcal{Y}(K)$ on an RCC K as in dfn. 4.1 has the form

$$\mathcal{T}_\theta^K = \mathrm{Dec}_\theta \circ \mathcal{L}_\theta^{(L-1)} \circ \cdots \circ \mathcal{L}_\theta^{(0)} \circ \mathrm{Enc}_\theta. \tag{7}$$

The encoder lifts the input cochains on K to hidden cochains $h_0^k \in C^k(K; \mathbb{R}^{d_h})$ at every rank $k = 0, \dots, N$. The incidence structure of K supplies the discrete operators used in the network. The topological layers $\mathcal{L}_\theta^{(\ell)}$ update these cochains while preserving their cochain-valued structure. The decoder reads the target degrees from the final hidden cochains. Material parameters and other coefficient fields enter as input cochains, which keep the architecture intrinsic to $(K, \{M_k\})$ in the sense of (P1). Next, we explain how the individual topological layers $\mathcal{L}_\theta^{(\ell)}$ are realized.

4.2 Topological layers

Each topological layer updates the hidden cochains on K while preserving their geometric type. Let $\mathbf{H} = (H^0, \dots, H^N)$ with $H^k \in \mathbb{R}^{n_k \times d_h}$. The layer applies a learned nonlinear map ϕ_θ to a fixed block-structured operator that encodes the discrete exterior calculus of K. As shown on the right, $\mathbf{H}_{\mathrm{out}}$ is defined through incidence matrices and Hodge stars of K. Its tridiagonal

$$\mathbf{H}_{\mathrm{out}} = \phi_\theta\left(\begin{bmatrix} \Delta_0 & \delta^1 & 0 & \cdots \\ d^0 & \Delta_1 & \delta^2 & \cdots \\ 0 & d^1 & \Delta_2 & \ddots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix} \mathbf{H} W\right)$$

sparsity is exact: equations at degree k couple only ranks $k \pm 1$. The trainable objects are the channel-mixing matrices W and transformations ϕ_θ. In the rigid DEC version, the transport maps themselves are fixed; in the copresheaf variant (Hajij et al., 2025) below, the fiber maps carried along incidences may be learned. See Appendix C.5.

For a given output rank k, the rank-k block update is (out-of-range terms omitted):

$$H^k_{\text{out}} = \phi_\theta\Big(d^{k-1}H^{k-1}W^{\downarrow}_k,\ \Delta^{\uparrow}_k H^k W^{\Delta,\uparrow}_k,\ \Delta^{\downarrow}_k H^k W^{\Delta,\downarrow}_k,\ H^k,\ \delta^{k+1}H^{k+1}W^{\uparrow}_k\Big). \tag{8}$$

Fig. 2 illustrates the template nature of TNOs: Maxwell fixes the active degrees and DEC routes, while the network learns how to mix the routed edge and face cochains (see App. B). The construction admits a continuous *rigidity spectrum*. At the rigid extreme, the transport operators are exactly the DEC operators of K. At the flexible extreme, each incidence $y \prec x$ carries a learnable fiber map $\rho_{y\to x}$, yielding twisted operators d_ρ, δ_ρ, $\Delta_{\rho,k}$ (copresheaf realization).

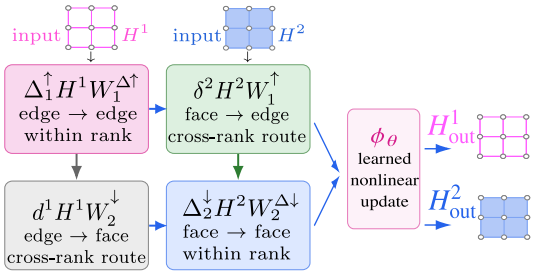


Figure 2: Visual TNO layer induced by Maxwell-type routing. Cross-rank transport is carried by d^1 and δ^2, while same-rank propagation is carried by the Hodge-Laplacian channels $\Delta^{\uparrow}$ and $\Delta^{\downarrow}$. The learned weights $W^\bullet$ and nonlinear update ϕ_θ determine how strongly the routed features are mixed.

Operator-learning interpretation. The distinction from a generic CCNN layer (Hajij et al., 2023b) is that the support of spatial transport is prescribed by the DEC operators. Cross-rank coupling is carried by d and δ, while same-rank propagation is carried by $\Delta^{\uparrow}$ and $\Delta^{\downarrow}$. Thus, the architecture does not learn where information can flow; it learns how transported features are mixed through $W^\bullet$ and ϕ_θ. This is what allows the same weights to be used on different meshes. Under compatible refinements and stable Hodge discretizations, the DEC operators d, δ, Δ approximate their continuum counterparts in the FEEC sense (Arnold, 2018). Since the same weights θ are shared across discretizations with support maps recomputed from each cell complex, the resulting family $\{\mathcal{T}^K_\theta\}_K$ is compatible with FEEC refinement, although it is not guaranteed to converge under it.

Linear realization of TNO. The simplest special case takes ϕ_θ to be linear and uses the Hodge decomposition channels directly. The rank-k linear TNO layer is

$$\mathcal{T}_{\theta,k}(h) = \delta^{k+1}h^{k+1}\,W^{\uparrow}_k + d^{k-1}h^{k-1}\,W^{\downarrow}_k + P^{\text{harm}}_k h^k\,W^{\text{harm}}_k + h^k\,W^{\text{self}}_k. \tag{9}$$

The four terms correspond to coexact, exact, harmonic, and local channels: $\mathrm{im}(\delta^{k+1})$, $\mathrm{im}(d^{k-1})$, $\ker(\Delta_k)$, and a residual self-channel, respectively. Here P^{harm}_k denotes the M_k-orthogonal projection onto the harmonic subspace $\ker(\Delta_k)$. Thus, the layer separates the three topological components of a rank-k signal while keeping transport fixed by the DEC operators. The following propositions show how FNO and GNO arise as rank-0 specializations of the TNO framework.

Proposition 4.2 (FNO as a rank-0 spectral TNO (Li et al., 2021))**.** *Let K_N be a uniform periodic cubical complex with $N_1 \times \cdots \times N_d$ vertices, and let $u \in C^0(K_N; \mathbb{R}^{c_{\text{in}}})$. Let $\mathcal{F}_N$ denote the discrete Fourier transform on 0-cochains. For a fixed set of resolved Fourier modes and shared multipliers giving real outputs, the linear FNO layer*

$$u \longmapsto Wu + \mathcal{F}_N^{-1} R_\theta \mathcal{F}_N u \tag{10}$$

is realized by a rank-0 spectral TNO layer whose same-rank spectral channel is diagonal in the Fourier basis of K_N. Proof is provided in sec. C.2.

Proposition 4.3 (GNO as a rank-0 graph-quadrature TNO (Li et al., 2020))**.** *On a directed graph complex whose vertices are sample points and whose rank-1 cells represent non-self quadrature interactions, the GNO update*

$$u_i \mapsto Wu_i + \sum_{j\in\mathcal{N}(i)} \kappa_\theta(x_i, x_j)u_j\omega_j, \tag{11}$$

is recovered as a rank-0 TNO with a learned incidence-supported kernel message $h^1(e_{ij}) = \kappa_\theta(x_i, x_j)u_j\omega_j$ on each directed rank-1 cell $e_{ij} : v_j \to v_i$, followed by target-incidence aggregation at each vertex. Self-interactions are cellwise maps. Proof is provided in sec. C.3.

Hierarchical Realization of TNO (HTNO). The TNO construction extends naturally to a hierarchy of cell complexes. Representing such a hierarchy as a single domain $\widetilde{K}$ requires cells that group fine cells without assuming strict boundary structure. To achieve this, we construct $\widetilde{K}$ as a combinatorial complex (CC) (Hajij et al., 2023b;a), which subsumes graphs, RCCs, and hypergraphs (see sec. A). Let $K_0 \leftarrow K_1 \leftarrow \cdots \leftarrow K_L$, where $K_{\ell+1}$ is a coarsening of K_ℓ and $K_0 = K$. Each level carries cochain spaces $C^k(K_\ell)$ and discrete exterior derivatives $d_\ell^k : C^k(K_\ell) \to C^{k+1}(K_\ell)$. Levels are connected by degree-preserving transfer maps

$$R_\ell^k : C^k(K_\ell) \to C^k(K_{\ell+1}), \qquad \Pi_\ell^k : C^k(K_{\ell+1}) \to C^k(K_\ell). \tag{12}$$

Ideally, these transfers commute with the coboundary:

$$d_{\ell+1}^k R_\ell^k = R_\ell^{k+1} d_\ell^k, \qquad d_\ell^k \Pi_\ell^k = \Pi_\ell^{k+1} d_{\ell+1}^k. \tag{13}$$

This is the analog of the commuting-diagram condition in FEEC multigrid (Arnold et al., 2006; Arnold, 2018). When (13) holds, exact components are preserved across levels, so transfer respects the de Rham complex (15) rather than mixing cochain types arbitrarily. Coexact and harmonic components are controlled through the corresponding Hodge-adjoint and Hodge-Laplacian structure induced by Hodge stars. Here, R_ℓ^0, Π_ℓ^0 are mean-pooling and broadcast over a partition that does not necessarily obey (13). Higher-rank coarse features are re-derived from the incidence of $K_{\ell+1}$, and the condition serves as the FEEC ideal rather than a constraint we enforce (sec. D.1).

An HTNO arranges TNO blocks as a learned V-cycle for de Rham complexes (Hiptmair, 1998; Trottenberg et al., 2001). TNO layers on K_ℓ are Hodge-compatible smoothers (Hiptmair, 2002), with coarse corrections transferred via R_ℓ^k, Π_ℓ^k built from a partition of K_ℓ, which can be Voronoi, k-means, or learned end-to-end. Yet this maintains the operator class: (P1)–(P3) hold level-wise so it is still a TNO at every level. Fine blocks capture local cross-degree physics and coarse blocks propagate long-range information. Sec. D defines the V-cycle equation and implementation.

5 Experiments

We evaluate TNO and HTNO on a broad span of steady-state PDEs over irregular 2D and 3D meshes: Poisson's equation with Gaussian and multiscale-sinusoidal sources, hyper-elastic deformations of variable domains, compressible flow past 2D airfoils spanning subsonic, transonic, and supersonic regimes, and large-scale 3D wing-surface aerodynamics under compressible RANS in the transonic regime. These come from three established public benchmarks (Mousavi et al., 2025; Wen et al., 2025; Paischer et al., 2025), with mesh sizes ranging from $\sim$1K to 65K nodes. We complement these with two controlled synthetic studies of our own: Oriented-Darcy, with per-face random tensor orientation, which isolates how TNO natively ingests higher rank signals (sec. 5.3), and a component ablation on synthetic topologies (Darcy; gauge-augmented multi-rank Curl–curl) that disentangles the harmonic-basis input and copresheaf transport (sec. 5.4). Dataset descriptions and the full training protocol are deferred to sec. F. On the same synthetic families, sec. G reports PDE residuals and conservation errors (sec. G.3), zero-shot generalization to finer resolutions (sec. G.4), and ablations of spectral MPNN inputs and cross-rank coupling (secs. G.5 and G.6), and computational complexity (sec. G.7). Implementations are in JAX and run on GH200s (96GB).

5.1 Steady-State Benchmarks on Irregular Geometry

Poisson-Gauss, Airfoil, Elasticity (tab. 1). We evaluate on three established steady-state benchmarks assembled by Mousavi et al. (2025): Poisson-Gauss (PG, fixed point cloud; from Herde et al. (2024)), Airfoil Flow (AF, variable geometry), and Elasticity (variable; both from Li et al. (2023b)). We compare against the seven baselines reported in Mousavi et al. (2025): RIGNO-18, RIGNO-12, MeshGraphNet (MGN) (Pfaff et al., 2021), Geo-FNO (Li et al., 2023b), FNO DSE (Lingsch et al., 2024), GINO (Li et al., 2023c), and UPT (Alkin et al., 2024) (tab. 1). Both TNO and HTNO outperform all seven on PG and Elasticity; TNO is competitive with RIGNO-18 on AF, which fixes the far-field at M_∞=0.8, α=0° and varies only the airfoil geometry (Li et al., 2023b). The GAOT airfoils (tab. 2a) genuinely sweep $M \in [0.5, 1.4]$ and $\alpha \in [0.5°, 5.0°]$ per sample (Wen et al., 2025), crossing the shock-formation boundary (see sec. G.2), and thus pose a greater challenge (next).

Compressible airfoils and Poisson-with-sines (tab. 2a). We additionally evaluate on the four single-file steady-state datasets (Wen et al., 2025): three compressible-airfoil sets (NACA0012/2412,

Table 1: Established steady-state benchmarks used in Mousavi et al. (2025). We depict median test relative L^1 (%) error. **Bold**: best; underline: second. † official implementations (see sec. E.3).

	Ours		*Baselines*							
Dataset	HTNO	TNO	RIGNO-18	RIGNO-12	GAOT†	MGN	Geo-FNO	FNO DSE	GINO	UPT
Poisson-Gauss	1.30	**1.03**	2.26	2.52	1.39	30.9	8.16	2.27	7.57	48.4
Airfoil	1.21	1.11	**1.00**	1.09	3.12	10.1	4.48	1.99	2.00	45.7
Elasticity	**1.70**	2.73	4.31	4.63	4.07	11.9	5.53	4.81	4.38	12.6

Table 2: Additional steady-state benchmarks: median test relative L^1 (%). †Official implementation; **bold**: best, underline: second. EmmiWing is given per-variable (p_s, $\tau_{x,y,z}$) and aggregate (Agg).

(a) Compressible airfoils and PCS (Wen et al., 2025)

	Ours		Baselines	
Dataset	HTNO	TNO	RIGNO†	GAOT†
NACA0012	4.53	**3.77**	5.09	12.13
NACA2412	4.73	**3.92**	5.73	12.25
RAE2822	4.55	**3.92**	4.81	12.98
PCS	**0.52**	3.09	7.11	0.72

(b) EmmiWing (Paischer et al., 2025).

Model	p_s	τ_x	τ_y	τ_z	Agg.
PointNet	0.29	4.24	5.75	5.90	2.62
Transformer	0.30	4.21	5.57	6.13	2.60
RIGNO-18†	0.29	4.46	6.02	6.30	2.78
Transolver†	0.28	4.14	5.52	5.57	2.55
HTNO	**0.24**	**3.94**	**5.32**	**5.37**	**2.41**

RAE2822) on 8K-node meshes, spanning subsonic to supersonic regimes, and Poisson-with-sines on a fixed 16K-node mesh (PCS). On all three airfoils, both TNO and HTNO clearly outperform RIGNO and GAOT. This can be attributed to the elliptic characteristics that dominate steady compressible flow at subsonic and transonic conditions (smooth pressure/velocity outside shocks), which align naturally with the Hodge / harmonic-basis inductive biases that characterize TNOs. PCS is the converse setting: a fixed mesh with a smooth, multiscale-sinusoidal Poisson source, for which GAOT's structured-latent encoder is well matched, beating RIGNO by an order of magnitude. Nevertheless, the HTNO, which combines local and global interactions, yields strong performance (**0.52** vs. 0.72).

5.2 Large-Scale Surface PDEs: EmmiWing

EmmiWing (Paischer et al., 2025) comprises 3D wing surfaces (raw meshes 244K–426K nodes; variable per sample) with four output channels (surface pressure p_s plus three wall-shear-stress components $\tau_{x,y,z}$; official $25{,}674/999/2{,}992$ split). Following the AB-UPT training regime (Alkin et al., 2025), we train on 65K shared FPS query points per sample with $16{,}384$-node random subsampling per step, and evaluate uniformly on the 65K FPS; see sec. F.3 for the full protocol. HTNO leads aggregate L^1 at 2.41%, beating Transolver (Wu et al., 2024) (2.55%), our matched-budget self-attention Transformer (2.60%), PointNet (2.62%), and RIGNO-18 (2.78%); HTNO additionally converges in $\sim 6\times$ faster than RIGNO-18 on PG and Elasticity (fig. 14). TNO is subsumed by the HTNO, which models long-range interactions without per-sample Hodge decompositions.

5.3 Oriented-Darcy: Per-Face Random Tensor Orientation

Setup. TNOs naturally accommodate higher-rank signals. We test this property on Oriented-Darcy, an anisotropic Darcy flow on triangulated square domains $[-1, 1]^2$ with one or two circular holes:

$$-\nabla\cdot\big(\kappa(x)\,\nabla u\big) = f, \qquad u|_{\partial\Omega_{\text{outer}}} = 0, \quad u|_{\partial\Omega_{\text{hole},k}} = g_k \sim \mathcal{U}[0.5, 1.5]. \tag{14}$$

The piecewise-constant tensor $\kappa(x)$ has eigenvalues $(\kappa_\parallel, \kappa_\perp) = (4, 1)$. Its principal axis $\phi_t \sim \mathcal{U}[0, \pi)$ is drawn independently for each face and sample. We store both $\cos(2\phi_t)$ and $\sin(2\phi_t)$ on faces which together determine the tensor $\kappa(x)$. We compare TNO with a width-matched and a parameter-matched MPNN. Each model receives the same vertex inputs and one of three orientation inputs: vertex (no orientation information beyond the vertex inputs), projected (incident-face averages projected to vertices), or native rank-2 (the stored face field). TNO ingests the latter as a 2-cochain; the MPNN receives its incident-face mean as an edge feature. The dataset comprises 100 meshes $\times$ 50 samples with $4{,}000/500/500$ train/val/test samples, whole meshes held out ($80/10/10$), so test geometries are unseen in training. Simulation and training details are in sec. F.8.

Results. Tab. 3a contrasts the processing of inputs across architectures. Projecting orientation to vertices lowers L^1 error by 2.38 percentage points (pp) for the MPNN and 2.33 pp for TNO; these averages therefore retain useful information. Ingesting the face field natively lowers TNO's error

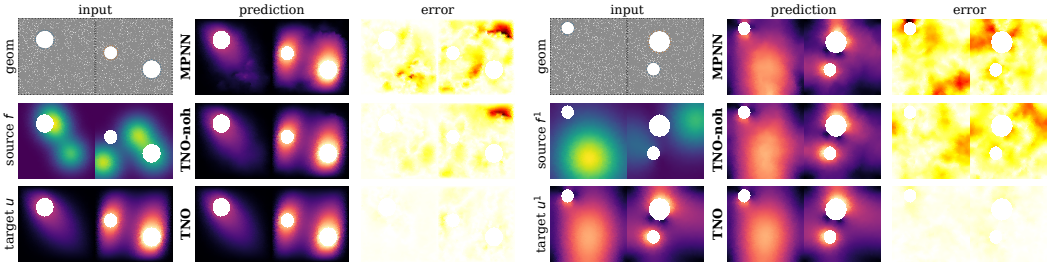


Figure 3: Qualitative ablation across three models: Vanilla MPNN vs. TNO (no harmonics) vs. TNO with harmonics / copresheaves on Darcy (left) and Curl–curl (right; magnitude of the Whitney-reconstructed edge field u^1), for which we depict model inputs and targets, predictions, and errors.

Table 3: Median test error (%), mean of 3 seeds. Left (a): Oriented-Darcy by orientation input, L^1/L^2; right (b): ablation on Darcy and Curl–curl, L^1 by rank (secs. 5.3 and 5.4). Width-matched MPNN: w=192; parameter-matched widths in secs. F.7 and F.8; (a) seed standard deviations are at most 0.24 pp (sec. F.8). Lower is better; **best**/second-best per column are emphasized.

Model	Orientation input	L^1	L^2
MPNN, width-matched	vertex	7.67	7.87
	projected	5.29	5.10
	native rank-2	5.03	4.72
MPNN, param.-matched	vertex	7.66	7.92
	projected	5.34	5.22
	native rank-2	4.99	4.76
TNO	vertex	6.65	7.38
	projected	4.32	4.66
	native rank-2	**1.68**	**1.89**

		Curl–curl		
Variant	Darcy	C^0	C^1	C^2
MPNN, width-matched	9.19	21.46	23.00	10.22
MPNN, param.-matched	8.59	22.57	22.03	9.80
TNO	7.86	19.18	18.47	9.98
TNO + harm. basis	2.29	2.23	2.87	2.27
TNO + copresheaf	7.47	17.60	15.96	6.87
TNO + copresheaf + harm.	**1.89**	**1.60**	**2.37**	**1.87**

by 2.64 pp, whereas passing the same field to the MPNN as edge features is nearly uninformative, demonstrating the necessity of native rank input ingestion for this operator learning task.

5.4 Ablations

Setup. We isolate the contribution of TNO's harmonic-basis input and copresheaf transport on two PDEs whose flux carries non-trivial curl on denser meshes than sec. 5.3: Darcy ($-\nabla\cdot(\kappa\nabla u) = f$) and Curl–curl, a gauge-augmented system with outputs on C^0, C^1, and C^2 (see secs. F.5 and F.6). Per-sample variability spans the sources, coefficients, and per-hole Dirichlet or circulation values. Datasets are simulated as in sec. 5.3 (tab. 8). All variants share a backbone (hidden dim 192), with models without copresheaf transport adjusted to match their counterparts in parameters.

Results. Disabling the harmonic basis loses 5.6 pp on Darcy and 13.6 pp on the curl–curl C^1 output (see tab. 3b). Copresheaf transport and the harmonic basis are complementary. With harmonic on, copresheaves help; with harmonic off, they help, but less (Darcy 7.86 $\rightarrow$ 7.47). The complete TNO reduces error by 79–93% on both PDEs relative to the width-matched MPNN (see fig. 3). This suggests harmonic/copresheaf components are an effective inductive bias (sec. G.5), particularly for physical systems with non-trivial curl.

6 Conclusion

We introduced **TNOs**, a topological framework for operator learning on cell complexes. Fields are represented as cochains on vertices, edges, faces, and volumes, and their interactions are routed by DEC operators. Thus topology determines where information flows, while learning determines how features are transformed. This yields architectures suited to physical coupling and irregular geometries, with weights shared across meshes, with existing neural operators arising as special cases. Empirically, TNOs and HTNOs perform strongly across benchmarks on irregular geometries, reducing errors relative to message-passing baselines. This motivates scientific foundation models that learn operators on structured spaces, unifying geometry, physics, and computation.

Limitations and future work. Our work leaves ample room for future work: scaling to dynamic, adaptive, and very large domains; developing approximation, stability, and convergence theory, clarifying the role of harmonic and copresheaf-based channels; and extending TNOs to multiscale systems, evolving manifolds, inverse problems, and gauge- or symmetry-equivariant operator families.

Ethics and reproducibility. Our experiments use public PDE benchmarks and synthetic simulations, thus no human subjects or personal data are involved. TNOs are learned surrogates, so their predictions should be checked against numerical solvers before use in engineering decisions, especially on unseen geometries or regimes. Dataset splits, simulation details, and training protocols are documented in sec. 5, sec. F, and sec. E; implementations will be published upon acceptance.

AI use statement. In this work, we used generative AI tools for language editing and for code editing, debugging, validating synthetic data simulation parameters, and visualization. All AI-assisted text and code were reviewed by the authors. The authors take full responsibility for the final content.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *Adv. Neural Inform. Process. Syst.*, 2024.

Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. AB-UPT: Scaling neural CFD surrogates for high-fidelity automotive aerodynamics simulations via anchored-branched universal physics transformers. *Trans. Mach. Lear. Resea.*, 2025. arXiv:2502.09692.

Douglas N. Arnold. *Finite Element Exterior Calculus*, volume 93 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2018. doi: 10.1137/1.9781611975543.

Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Multigrid in $H(\mathrm{div})$ and $H(\mathrm{curl})$. *Numerische Mathematik*, 85(2):197–217, 2000.

Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.

Douglas N. Arnold, Richard S. Falk, and Ragnar Winther. Finite element exterior calculus: from Hodge theory to numerical stability. *Bulletin of the American Mathematical Society*, 47(2):281–354, 2010.

Michael Aschbacher. Combinatorial cell complexes. In *Progress in Algebraic Combinatorics*, volume 24 of *Advanced Studies in Pure Mathematics*, pp. 1–80. Mathematical Society of Japan, 1996. doi: 10.2969/aspm/02410001.

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024.

Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. DOLFINx: The next generation FEniCS problem solving environment. Zenodo, 2023.

Francesca Bartolucci, Emmanuel de Bézenac, Bogdan Raonic, Roberto Molinaro, Siddhartha Mishra, and Rima Alaifari. Representation equivalent neural operators: a framework for alias-free operator learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/dc35c593e61f6df62db541b976d09dcf-Abstract-Conference.html.

Tathagata Basak. Combinatorial cell complexes and poincaré duality. *Geometriae Dedicata*, 147(1): 357–387, 2010.

Nathan Bell and Anil N. Hirani. PyDEC: Software and algorithms for discretization of exterior calculus. *ACM Transactions on Mathematical Software*, 39(1):3:1–3:41, 2012. doi: 10.1145/2382585.2382588.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric PDEs. *The SMAI Journal of Computational Mathematics*, 7: 121–157, 2021.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and lehman go cellular: CW networks. In *Adv. Neural Inform. Process. Syst.*, 2021a.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido Montúfar, Pietro Liò, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *Int. Conf. Mach. Lear.*, 2021b.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022.

Alain Bossavit. Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism. *IEE Proceedings A (Physical Science, Measurement and Instrumentation, Management and Education, Reviews)*, 135(8):493–500, 1988.

Alain Bossavit. *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*. Academic Press, 1998.

Nicolas Boullé and Alex Townsend. A mathematical guide to operator learning. In *Handbook of Numerical Analysis*, volume 25, pp. 83–125. Elsevier, 2024.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, second edition, 2000. doi: 10.1137/1.9780898719505.

Andrey Bryutkin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Hamlet: Graph transformer neural operator for partial differential equations. In *International Conference on Machine Learning*, pp. 4624–4641. PMLR, 2024.

Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *Journal of Machine Learning Research*, 26(300):1–52, 2025.

Gengxiang Chen, Xu Liu, Qinglu Meng, Lu Chen, Changqing Liu, and Yingguang Li. Learning neural operators on riemannian manifolds. *National Science Open*, 3(6):20240001, 2024.

Chaoran Cheng and Jian Peng. Equivariant neural operator learning with graphon convolution. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 61960–61984, 2023.

Pengcheng Cheng. Gauge-equivariant intrinsic neural operators for geometry-consistent learning of elliptic pde maps. *arXiv preprint arXiv:2603.14734*, 2026.

Jae Choi, Yuzhou Chen, Huikyo Lee, Hyun Kim, and Yulia R. Gel. SNN-PDE: Learning dynamic PDEs from data with simplicial neural networks. In *AAAI*, pp. 11561–11569, 2024.

Keenan Crane. Discrete differential geometry: An applied introduction. `https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf`, 2018. Lecture notes, Carnegie Mellon University.

Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, pp. 1–126. 2013.

Mathieu Desbrun, Anil N Hirani, Melvin Leok, and Jerrold E Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.

Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete differential forms for computational modeling. In *ACM SIGGRAPH 2006 Courses*, pp. 39–54. 2006.

Sharif Elcott, Yiying Tong, Eva Kanso, Peter Schröder, and Mathieu Desbrun. Stable, circulation-preserving, simplicial fluids. *ACM Transactions on Graphics*, 26(1):4, 2007. doi: 10.1145/1189762.1189766.

Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed deep neural operator networks. In *Machine learning in modeling and simulation: methods and applications*, pp. 219–254. Springer, 2023.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Aldo Guzmán-Sáenz, Tolga Birdal, and Michael T. Schaub. Combinatorial complexes: bridging the gap between cell complexes and hypergraphs. In *Asilomar Conference on Signals, Systems, and Computers*, pp. 799–803. IEEE, 2023a.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2023b.

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 *Adv. Neural Inform. Process. Syst.*, 2025.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. arXiv:2012.06333.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Ralf Hiptmair. Multigrid method for Maxwell's equations. *SIAM Journal on Numerical Analysis*, 36 (1):204–225, 1998.

Ralf Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11:237–339, 2002.

Anil Nirmal Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, 2003.

James M Hyman, Mikhail Shashkov, and Stanly Steinberg. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. *Journal of Computational Physics*, 132(1):130–148, 1997.

Anran Jiao, Haiyang He, Rishikesh Ranade, Jay Pathak, and Lu Lu. One-shot learning for solution operators of partial differential equations. *Nature Communications*, 16(1):8386, 2025.

Nikola B Kovachki, Samuel Lanthaler, and Andrew M Stuart. Operator learning: Algorithms and analysis. *Handbook of Numerical Analysis*, 25:419–467, 2024.

Nikolas Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.*, 24(89):1–97, 2023.

Samuel Leventhal, Attila Gyulassy, Valerio Pascucci, and Mark Heimann. Modeling hierarchical topological structure in scientific images with graph neural networks. In *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 2995–2999. IEEE, 2023.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020. arXiv:2003.03485.

Zongyi Li, Nikolas Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *ICLR*, 2021.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *J. Mach. Learn. Res.*, 24(388):1–26, 2023b.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Adv. Neural Inform. Process. Syst.*, 2023c.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024.

Yunfeng Liao, Yangxin Wu, and Xiucheng Li. Boundary-value PDEs meet higher-order differential topology-aware GNNs. In *Advances in Neural Information Processing Systems*, 2025.

Lek-Heng Lim. Hodge laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020.

Levi E. Lingsch, Mike Yan Michelis, Emmanuel De Bezenac, Sirani M. Perera, Robert K. Katzschmann, and Siddhartha Mishra. Beyond regular grids: Fourier-based neural operators on arbitrary domains. In *Int. Conf. Mach. Lear.*, 2024.

Konstantin Lipnikov, Gianmarco Manzini, and Mikhail Shashkov. Mimetic finite difference method. *Journal of Computational Physics*, 257:1163–1227, 2014.

Ning Liu, Siavash Jafarzadeh, and Yue Yu. Domain agnostic fourier neural operators. *Advances in neural information processing systems*, 36:47438–47450, 2023.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pp. 35–57. Springer, 2003.

Mamdouh S. Mohamed, Anil N. Hirani, and Ravi Samtaney. Discrete exterior calculus discretization of incompressible Navier–Stokes equations over surface simplicial meshes. *Journal of Computational Physics*, 312:175–191, 2016. doi: 10.1016/j.jcp.2016.02.028.

Sepehr Mousavi, Shizheng Wen, Levi Lingsch, Maximilian Herde, Bogdan Raonic, and Siddhartha Mishra. Rigno: A graph-based framework for robust and accurate operator learning for pdes on arbitrary domains. In *Adv. Neural Inform. Process. Syst.*, 2025.

Jean-Claude Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

Fabian Paischer, Leo Cotteleer, Yann Dreze, Richard Kurle, Dylan Rubini, Maurits Bleeker, Tobias Kronlachner, and Johannes Brandstetter. Going with the speed of sound: Pushing neural surrogates into highly-turbulent transonic regimes. In *NeurIPS 2025 Workshop on Machine Learning and the Physical Sciences*, 2025. arXiv:2511.21474.

Theodore Papamarkou, Tolga Birdal, Michael M. Bronstein, Gunnar E. Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, 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 *Int. Conf. Mach. Lear.*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39529–39555, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *ICLR*, 2021.

Lenka Ptáčková and Luiz Velho. A simple and complete discrete exterior calculus on general polygonal meshes. *Computer Aided Geometric Design*, 88:102002, 2021.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017.

Blaine Quackenbush and Paul J Atzberger. Geometric neural operators (gnps) for data-driven deep learning in non-euclidean settings. *Machine Learning: Science and Technology*, 5(4):045033, 2024.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Adv. Neural Inform. Process. Syst.*, 2023.

Mansi Sakarvadia, Kareem Hegazy, Amin Totounferoush, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W. Mahoney. The false promise of zero-shot super-resolution in machine-learned operators. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/71d1fdbaa2c57c308e3abb27cbb491c8-Abstract-Conference.html.

Maxime Savoy. Combinatorial cell complexes: Duality, reconstruction and causal cobordisms. *arXiv preprint arXiv:2201.12846*, 2022.

Michael T Schaub, Austin R Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

Lennon J. Shikhman and Shane Gilbertie. Cellular sheaf neural operators for structure-preserving surrogate modeling of constrained PDEs. *arXiv preprint arXiv:2606.00937*, 2026.

Dmitriy Smirnov and Justin Solomon. HodgeNet: Learning spectral geometry on triangle meshes. *ACM Transactions on Graphics (SIGGRAPH)*, 40(4), 2021.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023.

Nathaniel Trask, Andy Huang, and Xiaozhe Hu. Enforcing exact physics in scientific machine learning: A data-driven exterior calculus on graphs. *Journal of Computational Physics*, 456: 110969, 2022.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.

Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schüller. *Multigrid*. Academic Press, London, 2001.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

Francesco Viganò, Tolga Birdal, Michael T Schaub, and Mauricio Barahona. Root-to-leaf path random walks, normalized hodge laplacians, and cheeger inequalities on simplicial complexes. *arXiv preprint arXiv:2604.27241*, 2026.

Hrishikesh Viswanath, Md Ashiqur Rahman, Abhijeet Vyas, Andrey Shor, Beatriz Medeiros, Stephanie Hernandez, Suhas Eswarappa Prameela, and Aniket Bera. Neural operator: Is data all you need to model the world? an insight into the paradigm of data-driven scientific ML. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3682604.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains. In *Adv. Neural Inform. Process. Syst.*, 2025. arXiv:2505.18781.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Int. Conf. Mach. Lear.*, 2024.

Maosheng Yang, Geert Leus, and Elvin Isufi. Hodge-aware convolutional learning on simplicial complexes. *Transactions on Machine Learning Research*, 2025.

Minglang Yin, Nicolas Charon, Ryan Brody, Lu Lu, Natalia Trayanova, and Mauro Maggioni. A scalable framework for learning the geometry-dependent solution operators of partial differential equations. *Nature Computational Science*, 4(12):928–940, 2024.

Seungwoo Yoo, Kyeongmin Yeo, Jisung Hwang, and Minhyuk Sung. Neural Green's functions. In *Adv. Neural Inform. Process. Syst.*, 2025.

Dongzhe Zheng, Tao Zhong, and Christine Allen-Blanchette. Topology-preserving neural operator learning via Hodge decomposition. In *Int. Conf. Mach. Lear.*, 2026. arXiv:2605.13834.

Weiheng Zhong and Hadi Meidani. Physics-informed geometry-aware neural operator. *Computer Methods in Applied Mechanics and Engineering*, 434:117540, 2025.

TOPOLOGICAL NEURAL OPERATORS

Appendix

TABLE OF CONTENTS

A Background **16**

B PDE Templates as Typed Cochain Systems **25**

C Proofs **28**

D Hierarchical TNOs: Multigrid Motivation and Architecture **33**

E Implementation Details **34**

F Experimental Details **37**

G Extended Experimental Results **41**

H Additional Related Work **46**

A BACKGROUND

A.1 COMBINATORIAL COMPLEXES

Combinatorial complexes generalize simplicial and cell complexes as well as hypergraphs, acting as a unifying topological structure:

Definition A.1 (Combinatorial complex (Hajij et al., 2023b;a))**.** *A* combinatorial complex *(CC) is a triple* $(S, \mathcal{X}, \mathrm{rk})$ *consisting of a set* S *of entities, a subset* $\mathcal{X} \subseteq \mathcal{P}(S) \setminus \{\emptyset\}$ *of cells, and a rank function* $\mathrm{rk} : \mathcal{X} \to \mathbb{Z}_{\geq 0}$ *satisfying* $\mathrm{rk}(\{v\}) = 0$ *for all* $v \in S$ *and* $\mathrm{rk}(x) \leq \mathrm{rk}(y)$ *whenever* $x \subseteq y$*.*

Lifting an RCC K to a CC $\widetilde{K}$ retains the cochains of K but may carry additional cells, e.g., the higher-order groupings that form the coarse levels of an HTNO (sec. D.1). The lift does not change the target operator $\mathcal{G}$ or the input/output spaces $\mathcal{X}(K), \mathcal{Y}(K)$; it only enlarges the domain on which the hierarchy acts. See fig. 4.

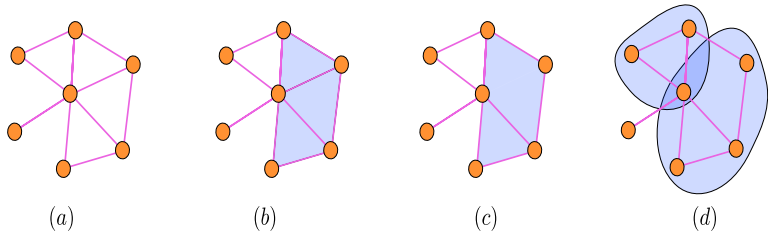


Figure 4: Progression of topological domains with increasing modeling flexibility. **(a)** A *graph* represents only vertices and pairwise edges. **(b)** A *simplicial complex* adds higher-order simplices, but each simplex is determined by its lower-dimensional faces. **(c)** A *cell complex* allows more general cells, such as polygonal regions, whose boundaries need not be simplices. **(d)** A *combinatorial complex* further relaxes the construction by allowing independently specified, possibly overlapping higher-order relations. Each step increases the flexibility of the domain on which cochains and topological neural operators can be defined.

A.2 Discrete Differential Forms and Physical Cochains

We recall the discrete differential-form viewpoint used throughout the paper (Hirani, 2003; Desbrun et al., 2005; 2006; Crane et al., 2013; Crane, 2018), together with the finite-element-exterior-calculus perspective of Arnold et al. (2006). Our goal is not to introduce differential forms for their own sake, but to explain why physical fields should not all be represented as vertex signals. Many physical quantities are not point samples. A potential is naturally measured at points, a circulation along curves, a flux through surfaces, and a density over volumes. Discrete exterior calculus (DEC) keeps this measurement type in the discrete model.

Throughout, K denotes an oriented regular cell complex. We write K_k for the set of k-cells: vertices for $k = 0$, edges for $k = 1$, faces for $k = 2$, and volumes for $k = 3$. Examples of regular cell complexes are shown in fig. 5.

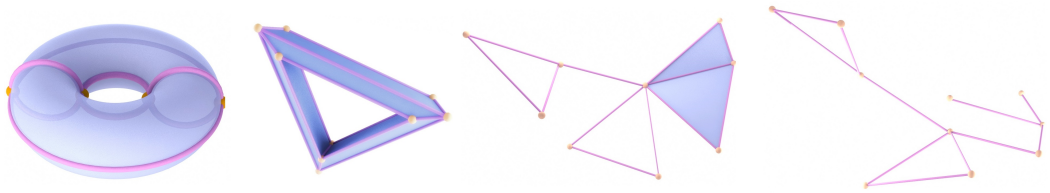


Figure 5: Four examples of regular cell complexes.

A *discrete k-form*, or k-cochain, is a field supported on oriented k-cells:

$$C^k(K;\mathbb{R}^d) = \{u^k : K_k \to \mathbb{R}^d\} \cong \mathbb{R}^{|K_k|\times d}.$$

Thus 0-cochains live on vertices, 1-cochains on edges, 2-cochains on faces, and 3-cochains on volumes. The degree k is therefore not merely an index. It records the geometric type of the quantity.

The useful intuition is that cochains are *cellular measurements*. A 0-cochain stores point measurements, a 1-cochain oriented line measurements, 2-cochains oriented surface measurements, 3-cochain volume measurements, and so forth. This is the discrete analog of the integral viewpoint in exterior calculus: one does not approximate every physical field by point values; one stores each quantity on the cells over which they are naturally measured.

This distinction is important. A flux through a face is not the same kind of object as a scalar value at a vertex. A circulation along an edge cannot be represented faithfully as a node feature. If such fields are collapsed to vertices, their orientation, incidence, and conservation relations become hidden or must be relearned from data (see sec. 5.3). DEC avoids this collapse by making the support dimension part of the representation.

Chains, cochains, and discrete integration. Let K be an oriented cell complex and let K_k denote its set of oriented k-cells (Hirani, 2003). A k-chain is a finite formal linear combination of k-cells,

$$c = \sum_{\sigma\in K_k} c_\sigma\, \sigma, \qquad c_\sigma \in \mathbb{R}.$$

The coefficients c_σ record how the cells are assembled. A positive coefficient uses the chosen orientation of σ, while a negative coefficient uses the opposite orientation. Thus, a chain is a discrete geometric domain: a 0-chain is a signed collection of vertices, a 1-chain is an oriented path made of edges, a 2-chain is an oriented surface patch made of faces, and so on.

In this paragraph, we take cochains to be scalar-valued, since this is the setting in which the integration pairing and Stokes' theorem are most directly stated. A k-cochain is a linear functional on k-chains. Equivalently, it is specified by assigning one scalar to each oriented k-cell,

$$u^k : K_k \to \mathbb{R}.$$

Its value on a chain $c = \sum_{\sigma\in K_k} c_\sigma\sigma$ is defined by linearity:

$$\langle u^k, c\rangle = \sum_{\sigma\in K_k} c_\sigma\, u^k(\sigma).$$

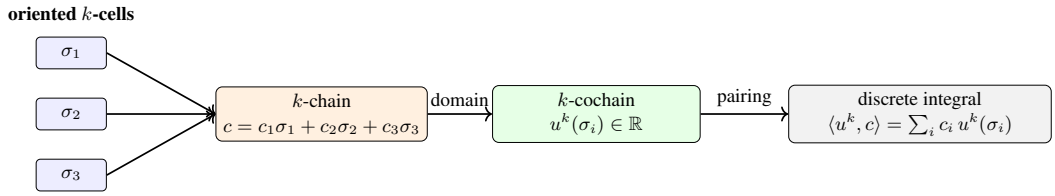


Figure 6: Chains and cochains as discrete integration. A k-chain is a signed combination of oriented k-cells and represents a discrete k-dimensional domain. A k-cochain assigns measurements to oriented k-cells. Their pairing sums the cochain values over the chain with the chain coefficients, giving the discrete analogue of integrating a k-form over a k-dimensional region.

This pairing is the discrete analog of integration. If ω^k is a smooth k-form and R is a k-dimensional oriented region, then the continuum measurement is

$$\int_R \omega^k.$$

In the discrete setting, the region R is replaced by a k-chain c, the form ω^k is replaced by a k-cochain u^k, and the integral is replaced by

$$\int_R \omega^k \quad \rightsquigarrow \quad \langle u^k, c\rangle.$$

This is the sense in which cochains are discrete integrands. A 1-cochain can be summed along an oriented path. A 2-cochain can be summed over an oriented surface patch. A 3-cochain can be summed over a collection of volume cells. The degree k specifies both where the quantity is stored and what kind of discrete region it can be measured on. See Figure 6.

For neural-network features we often use vector-valued cochains $u^k : K_k \to \mathbb{R}^{d_k}$. These should be understood as d_k scalar cochains stored in parallel, with the same pairing applied componentwise. The scalar case is the one used to state the discrete Stokes relation; the vector-valued case is its channel-wise extension.

This interpretation prepares the definition of the exterior derivative. Once cochains are viewed as discrete integrands, Stokes' theorem becomes a statement about how the measurement of a derivative on a cell is related to the measurement of the original cochain on the oriented boundary of that cell.

From cellular measurements to the discrete exterior derivative. A cochain stores measurements on cells. The degree records the type of measurement: values on vertices, line measurements on edges, surface measurements on faces, and volume measurements on 3-cells. This is the primal placement convention used throughout the paper: a physical quantity is placed on the cell type over which it is naturally measured.

Table 4: Physical quantities and their natural cochain degrees under the primal placement convention (Desbrun et al., 2006; Arnold et al., 2006). A k-form is discretized as a k-cochain by integration over oriented k-cells.

Physical quantity	Geometric type	Degree k	Primal cell
Temperature, pressure, potential	0-form	0	Vertices
Electric field E, circulation, line flow, Darcy flux	1-form	1	Edges
Magnetic flux B, vorticity flux	2-form	2	Faces
Volumetric charge, mass, source density	3-form	3	Volumes

Table 4 should be read as a statement about measurement, not only about notation. A scalar potential is sampled at points. A circulation or line integral is measured along an oriented curve, and in the discrete complex, this means an edge. A flux is measured through an oriented surface, and in the discrete complex, this means a face. A density or source term is measured over a volume. This is why the cochain degree is part of the field's physical meaning.

The table uses the primal convention. In an n-dimensional domain, the Hodge star identifies primal k-cochains with dual $(n-k)$-cochains. Consequently, the same physical vector quantity may be stored on different cells depending on whether one uses the primal or dual complex. Unless stated otherwise, this paper uses the primal convention in Table 4.

The question is then how to differentiate such quantities without forgetting where they are measured. The classical answer is that differentiation and boundary measurement are tied together. In one dimension, the fundamental theorem of calculus says that the integral of a derivative over an interval is determined by the values at the two boundary points:

$$\int_a^b f'(t)\,dt = f(b) - f(a).$$

If the interval is oriented from a to b, its boundary is the formal signed sum

$$\partial[a,b] = [b] - [a].$$

Thus the derivative measured on the interval is obtained from the original function measured on the oriented boundary. The endpoint at the head of the interval contributes with sign $+1$, and the endpoint at the tail contributes with sign -1.

The same boundary principle appears in higher dimensions. For a vector field in the plane, Green's theorem relates circulation around the boundary of a region to a curl-type quantity inside the region. In three dimensions, Stokes' theorem relates circulation around the boundary curve of a surface to curl through the surface. Gauss' theorem relates flux through the boundary surface of a volume to divergence inside the volume. Exterior calculus writes these boundary laws as one statement:

$$\int_R d\omega = \int_{\partial R} \omega.$$

Here ω is measured on k-dimensional objects, $d\omega$ is measured on $(k+1)$-dimensional objects, R is a $(k+1)$-dimensional region, and ∂R is its oriented boundary. The identity states that the derivative of a quantity over a region is determined by the original quantity on the boundary of that region.

This is the viewpoint DEC keeps. Since a cochain already stores integrated measurements on cells, the discrete derivative is defined by the same boundary relation. A vertex quantity differentiates to an edge quantity by taking signed endpoint differences. An edge quantity differentiates to a face quantity by taking signed circulation around the face boundary. A face quantity differentiates to a volume quantity by taking signed flux through the volume boundary. The common operation is evaluation on an oriented boundary.

On a cell complex, the role of the region R is played by a $(k+1)$-cell τ, and the role of ∂R is played by the oriented cellular boundary of τ. Let

$$B_{k+1} : C_{k+1}(K) \to C_k(K)$$

be the oriented boundary matrix. Its columns are indexed by $(k+1)$-cells and its rows are indexed by k-cells. For $\tau \in K_{k+1}$, the column indexed by τ records

$$\partial\tau = \sum_{\sigma \in K_k} [\tau : \sigma]\,\sigma, \qquad [\tau : \sigma] \in \{-1, 0, +1\}.$$

The coefficient $[\tau : \sigma]$ is nonzero exactly when σ is a boundary face of τ. The sign records orientation: $+1$ if the chosen orientation of σ agrees with the orientation induced from τ, and -1 if it disagrees.

Now let $u^k \in C^k(K)$ be a k-cochain. Its discrete exterior derivative $d^k u^k$ is the $(k+1)$-cochain obtained by measuring u^k on the oriented boundary of each $(k+1)$-cell:

$$\langle d^k u^k, \tau \rangle = \langle u^k, \partial\tau \rangle, \qquad \tau \in K_{k+1}.$$

In matrix form,

$$d^k : C^k(K) \to C^{k+1}(K), \qquad d^k = B_{k+1}^\top.$$

Equivalently,

$$(d^k u^k)(\tau) = \sum_{\sigma \prec \tau} [\tau : \sigma]\, u^k(\sigma).$$

Thus d^k takes measurements on k-cells and produces measurements on $(k+1)$-cells by signed summation over oriented boundaries. Its nonzero pattern comes from incidence, and its signs come from orientation. This is why the discrete exterior derivative is a topological operator (Hirani, 2003; Desbrun et al., 2005).

For TNOs, this is the basic upward cross-rank route. Vertex cochains are coupled to edge cochains, edge cochains to face cochains, and face cochains to volume cochains through the same boundary rule that appears in the underlying differential equations. The cell complex fixes the route; the learnable part is the channel mixing applied to the routed cochain features.

The discrete de Rham complex. Chaining the exterior derivatives yields the sequence of cochain spaces

$$0 \to C^0(K) \xrightarrow{d^0} C^1(K) \xrightarrow{d^1} C^2(K) \xrightarrow{d^2} C^3(K) \to 0, \qquad d^{k+1} \circ d^k = 0, \tag{15}$$

called the *discrete de Rham complex* of K (Desbrun et al., 2006; Arnold et al., 2006). The word "complex" is thus used in two senses: the cell complex K is the domain, while (15) is the sequence of cochain spaces on it, together with the maps between them. Because $d^{k+1} \circ d^k = 0$ (see below), $\operatorname{im}(d^{k-1}) \subseteq \ker(d^k)$, and the quotient $H^k(K) = \ker(d^k)/\operatorname{im}(d^{k-1})$ is the cellular cohomology of K, whose dimension is the k-th Betti number β_k. The Hodge decomposition (eq. (4)) picks out the harmonic cochains $\ker(\Delta_k)$ as the M_k-orthogonal representatives of $H^k(K)$, which is why $\dim\ker(\Delta_k) = \beta_k$. The commuting condition (13) on HTNO transfers and the multigrid correspondence of sec. D are formulated relative to this complex; sec. D.1 states to what extent our implementation satisfies it.

These maps are the cellular analogues of the familiar vector-calculus operators:

$$d^0 = \text{gradient-type map}, \qquad d^1 = \text{curl-type map}, \qquad d^2 = \text{divergence-type map}.$$

Here "type" means that the maps act between cellular measurement spaces. The map d^0 takes vertex measurements to edge measurements. The map d^1 takes edge measurements to face measurements. The map d^2 takes face measurements to volume measurements.

For $x \in C^0(K)$ and an oriented edge $e = [i, j]$,

$$(d^0 x)([i, j]) = x_j - x_i.$$

This is the fundamental theorem of calculus on a single edge: the measurement of the derivative along the edge is the signed difference of the endpoint values. Hence $d^0 x$ is an edge cochain.

For $y \in C^1(K)$ and an oriented face $f \in K_2$,

$$(d^1 y)(f) = \sum_{e \prec f} [f : e]\, y(e).$$

The value on the face is the signed circulation of the edge measurements around its boundary. This is the cellular version of Green's/Stokes' theorem. For an oriented triangle $[i, j, k]$, the induced boundary is

$$\partial[i, j, k] = [j, k] - [i, k] + [i, j].$$

With stored edge orientations $[i, j]$, $[i, k]$, and $[j, k]$, this gives

$$(d^1 y)([i, j, k]) = y_{ij} - y_{ik} + y_{jk}.$$

If the edge is stored as $[k, i]$ rather than $[i, k]$, the same circulation is written as

$$(d^1 y)([i, j, k]) = y_{ij} + y_{jk} + y_{ki}.$$

The two expressions represent the same oriented boundary measurement. The signs make the result independent of the bookkeeping choice for edge orientations.

For $z \in C^2(K)$ and an oriented volume cell $c \in K_3$,

$$(d^2 z)(c) = \sum_{f \prec c} [c : f]\, z(f).$$

The value on the volume is the signed accumulation of the face measurements on its boundary. This is the cellular version of Gauss' theorem: it measures net boundary flux.

The following example makes the construction explicit (Bell & Hirani, 2012). Consider the oriented complex in fig. 7 with

$$K_0 = ([1], [2], [3], [4], [5]),$$
$$K_1 = ([1,2], [1,3], [2,3], [2,4], [3,4], [4,5]), \qquad K_2 = ([1,2,3], [2,3,4]).$$

The matrix d^0 has one row per edge and one column per vertex. The row indexed by $[i, j]$ has a -1 in column i, a $+1$ in column j, and zeros elsewhere. Multiplying d^0 by a vertex cochain therefore computes all oriented edge differences at once.

The matrix d^1 has one row per face and one column per edge. Each row is the oriented boundary of one face written in the chosen edge basis. For the first face,

$$\partial[1,2,3] = [2,3] - [1,3] + [1,2],$$

so the first row of d^1, using the edge ordering above, is

$$[\,1,\ -1,\ 1,\ 0,\ 0,\ 0\,].$$

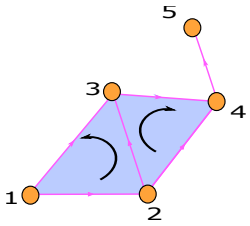
For the second face,

$$\partial[2,3,4] = [3,4] - [2,4] + [2,3],$$

so the second row is

$$[\,0,\ 0,\ 1,\ -1,\ 1,\ 0\,].$$

Multiplying d^1 by an edge cochain y returns one circulation value per filled face.



$$d^0 = B_1^\top : C^0(K) \to C^1(K), \qquad d^1 = B_2^\top : C^1(K) \to C^2(K)$$

$$d^0 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \qquad d^1 = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 \end{bmatrix}.$$

Figure 7: Coboundary matrices for the oriented complex. The matrix d^0 maps vertex cochains to edge cochains: each row computes one oriented difference, e.g. $[-1, 1, 0, 0, 0]$ gives $x_2 - x_1$ on $[1, 2]$. The matrix d^1 maps edge cochains to face cochains: each row computes one signed circulation. For the face $[1, 2, 3]$, the boundary relation $\partial[1, 2, 3] = [2, 3] - [1, 3] + [1, 2]$ gives $(d^1 y)([1, 2, 3]) = y_{12} - y_{13} + y_{23}$; similarly, $(d^1 y)([2, 3, 4]) = y_{23} - y_{24} + y_{34}$. Hence, these matrices encode oriented incidence, not just adjacency.

The example shows the general rule. The operator d^k evaluates a k-cochain on the oriented boundary of each $(k + 1)$-cell. This is the discrete Stokes map. It is also the basic upward route used by TNO layers for cross-rank coupling.

Boundaries cancel in pairs. The boundary-based definition of d^k has an immediate consequence: applying two such maps in a row gives zero. Algebraically,

$$B_k B_{k+1} = 0, \qquad \text{equivalently} \qquad d^{k+1} d^k = 0.$$

The first identity says that the boundary of a cell has no remaining boundary. The second identity is the corresponding statement on cochains:

$$C^k(K) \xrightarrow{d^k} C^{k+1}(K) \xrightarrow{d^{k+1}} C^{k+2}(K), \qquad d^{k+1} d^k = 0.$$

The reason is entirely local. Take a $(k + 2)$-cell and look at the k-cells that appear after taking its boundary twice. Each such k-cell appears exactly twice, once from each adjacent $(k + 1)$-face, and the two induced orientations are opposite, meaning these two contributions cancel. Thus, the identity is not a limiting statement and does not depend on mesh refinement, training, or numerical accuracy. It holds exactly on the oriented cell complex.

In low degrees, this gives the familiar compatibility laws

$$d^1 d^0 = 0 \qquad \Longleftrightarrow \qquad \mathrm{curl}(\mathrm{grad}) = 0,$$

and

$$d^2 d^1 = 0 \qquad \Longleftrightarrow \qquad \mathrm{div}(\mathrm{curl}) = 0.$$

The first says that a field built from vertex differences has zero signed circulation around every filled face. Around a triangle, the edge differences telescope:

$$(x_j - x_i) + (x_k - x_j) + (x_i - x_k) = 0.$$

The second says that a face field built from edge circulations has zero signed accumulation around every volume. Each interior edge contributes twice, with opposite signs, when the face circulations around the volume are summed.

Topology fixes incidence; geometry weights it. The matrices B_k and $d^k = B_{k+1}^\top$ know which cells touch and how their orientations agree. They do not know lengths, areas, volumes, angles, or material coefficients. Geometry enters through Hodge-star, or mass, matrices

$$M_k \in \mathbb{R}^{|K_k| \times |K_k|}, \qquad M_k \succ 0,$$

which define inner products on cochains:

$$\langle u^k, v^k \rangle_{M_k} = (u^k)^\top M_k v^k.$$

The matrix M_k stores metric and material information at degree k: edge lengths, face areas, cell volumes, anisotropic weights, permittivity, permeability, conductivity, or other coefficient fields (Hirani, 2003; Bell & Hirani, 2012). In primal-dual DEC, M_k can be interpreted as the map relating a primal k-cochain to a dual $(n-k)$-cochain.

The codifferential is the metric adjoint of d. The codifferential maps downward in degree,

$$\delta^k : C^k(K) \to C^{k-1}(K),$$

and is defined as the adjoint of d^{k-1} with respect to the Hodge inner products:

$$\langle d^{k-1} u^{k-1}, v^k \rangle_{M_k} = \langle u^{k-1}, \delta^k v^k \rangle_{M_{k-1}}.$$

In matrix form, up to the global sign convention chosen for the Hodge star,

$$\delta^k = M_{k-1}^{-1} B_k M_k.$$

Thus d moves cochains upward by taking signed boundary sums, while δ moves cochains downward by the metric adjoint operation (Hirani, 2003; Desbrun et al., 2005). This is why δ is divergence-like: it measures how a k-cell quantity accumulates onto incident $(k-1)$-cells after applying the appropriate geometric weights.

Concretely, at $k = 2$ the codifferential $\delta^2 = M_1^{-1} B_2 M_2 = M_1^{-1} (d^1)^\top M_2$ sends a face cochain B to the edge cochain

$$(\delta^2 B)_e = (M_1^{-1})_{ee} \sum_{f \succ e} [f : e]\, (M_2)_{ff}\, B_f, \tag{16}$$

illustrated in Fig. 8.

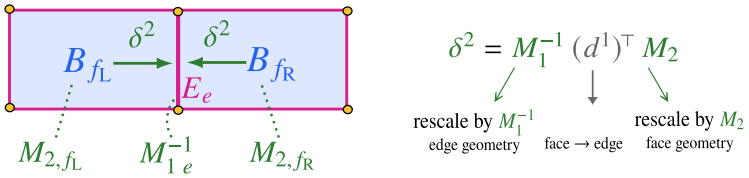


Figure 8: Codifferential as face-to-edge transport (Eq. (16)): neighbouring faces $f_L, f_R \in K_2$ sharing edge $e \in K_1$ contribute via the incidence sign $[f : e]$, weighted by the Hodge factors M_2 and M_1^{-1}.

This clarifies the role of the DEC maps inside a TNO. The incidence matrices B_k define the cellular routes along which information can be transported: they specify which cells are incident, how ranks are coupled, and how orientations enter the coupling. The exterior derivative $d^k = B_{k+1}^\top$ moves a

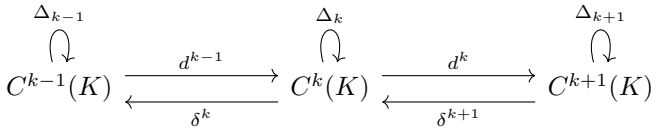
k-cochain upward to $(k+1)$-cells by signed aggregation over boundaries. On the other hand, the codifferential $\delta^k = M_{k-1}^{-1} B_k M_k$

moves a k-cochain downward to $(k-1)$-cells by the metric-weighted adjoint route. Thus d^k and δ^k are not generic message-passing maps: they are structured operators for moving, coupling, and aggregating cochain features across adjacent ranks. The Hodge matrices M_k determine how geometry and material coefficients weight these transports.

The Hodge Laplacian has two same-rank channels. The k-Hodge Laplacian decomposes as

$$\Delta_k = \underbrace{\delta^{k+1} d^k}_{\Delta_k^{\uparrow}} + \underbrace{d^{k-1} \delta^k}_{\Delta_k^{\downarrow}} : C^k(K) \to C^k(K).$$

Both terms start and end in $C^k(K)$, but they route the signal through different adjacent ranks. The upper term sends a k-cochain through $(k+1)$-cells and back, while the lower term sends it through $(k-1)$-cells and back. See Figure 9. Thus Δ_k is not only a same-rank operator; it records which neighboring dimension mediates the interaction (Hirani, 2003; Crane, 2018; Lim, 2020).



$$\Delta_k = \underbrace{\delta^{k+1} d^k}_{\Delta_k^{\uparrow}} + \underbrace{d^{k-1} \delta^k}_{\Delta_k^{\downarrow}}.$$

Figure 9: Upper and lower Hodge–Laplacian channels around $C^k(K)$. The differential d raises degree and moves rightward, while the codifferential δ lowers degree and moves leftward. The upper channel $\delta^{k+1} d^k$ propagates k-cochains through $(k+1)$-cells; the lower channel $d^{k-1}\delta^k$ propagates them through $(k-1)$-cells. The self-loops indicate same-rank Hodge Laplacian propagation on each cochain space.

For 0-cochains, the lower channel is absent:

$$\Delta_0 = \delta^1 d^0,$$

which is the usual graph or grid Laplacian on vertex fields. For 1-cochains, both channels are present:

$$\Delta_1 = \underbrace{\delta^2 d^1}_{\text{face-mediated}} + \underbrace{d^0 \delta^1}_{\text{vertex-mediated}}.$$

The first term sends edge values around faces and back to edges, coupling edges that bound common 2-cells. This is the circulation, or curl-type, route. The second term sends edge values to vertices and back to edges, coupling edges that meet at common 0-cells. This is the accumulation, or divergence-type, route. See Figure 10. Hence, an edge field has two distinct edge-to-edge propagation mechanisms: one mediated by faces and one mediated by vertices. Treating edge fields as ordinary graph features collapses this distinction into a single adjacency-based route.

Hodge decomposition separates physical modes. With the Hodge inner product, the k-cochain space decomposes as

$$C^k(K;\mathbb{R}) = \underbrace{\operatorname{im}(d^{k-1})}_{\text{exact}} \oplus \underbrace{\ker(\Delta_k)}_{\text{harmonic}} \oplus \underbrace{\operatorname{im}(\delta^{k+1})}_{\text{coexact}}.$$

Exact cochains come from lower-degree potentials. Coexact cochains come from higher-degree flux or circulation potentials. Harmonic cochains satisfy

$$d^k h = 0, \qquad \delta^k h = 0, \qquad \Delta_k h = 0,$$

and encode global topological degrees of freedom. In the scalar case,

$$\dim \ker(\Delta_k) = \beta_k,$$

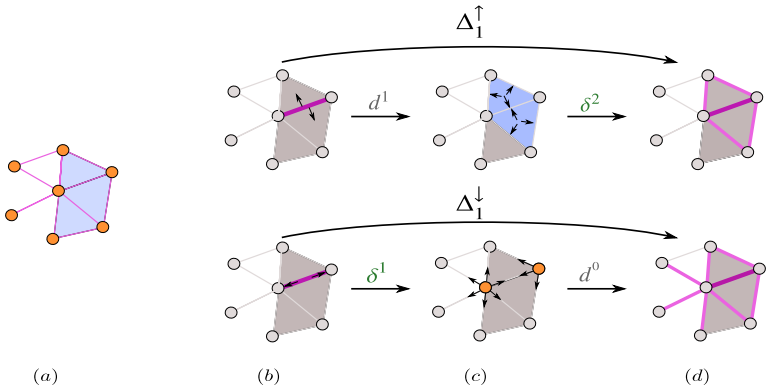


Figure 10: Upper and lower Hodge–Laplacian channels on edge cochains. Starting from an edge-supported signal $u \in C^1(K)$, the upper channel first applies $d^1 : C^1(K) \to C^2(K)$, which computes signed circulation around each face, and then applies $\delta^2 : C^2(K) \to C^1(K)$, which returns this face quantity to incident edges through the metric-weighted adjoint route. This gives $\Delta_1^{\uparrow} u = \delta^2 d^1 u$, a face-mediated edge-to-edge coupling. The lower channel first applies $\delta^1 : C^1(K) \to C^0(K)$, which measures the metric-weighted accumulation of the edge field at vertices, and then applies $d^0 : C^0(K) \to C^1(K)$, which sends the resulting vertex potential back to edges by oriented differences. This gives$\Delta_1^{\downarrow} u = d^0 \delta^1 u$, a vertex-mediated edge-to-edge coupling. Thus $\Delta_1^{\uparrow}$ and $\Delta_1^{\downarrow}$ are two distinct same-rank propagation mechanisms: one couples edges through shared faces, the other through shared vertices. This distinction is the degree-1 DEC analogue of separating curl-type and divergence-type effects, and is precisely what is collapsed when edge fields are treated as ordinary graph features.

the k-th Betti number (Hirani, 2003; Arnold et al., 2006). Thus harmonic components are not local noise: they are the cochain modes supported by the topology of the domain.

This is one of the main reasons DEC is useful for operator learning. Many physical fields naturally live in one of these components. Conservative fields are exact. Divergence-free fields are coexact, up to harmonic components. Topological modes, such as circulation around non-contractible cycles, appear in the harmonic subspace. A model that does not expose these channels must infer them indirectly from data.

A.3 Continuum Physics and the mPDE Formulation

The governing equations of continuum physics can be phrased in terms of how different geometric dimensions interact. An electric field $\mathbf{E}$, integrated along oriented edges, and a magnetic flux $\mathbf{B}$, integrated over oriented faces, are in this case naturally 1-cochain and 2-cochains, respectively. Faraday's law, $\partial_t B = -d^1 E$, couples them through the exterior derivative d^1, which maps edge-valued data to face-valued data by accumulating signed values around each face. In physical terms, the circulation of $\mathbf{E}$ around a face boundary equals the negative rate of change of the magnetic flux through that face (Bossavit, 1998; Hiptmair, 2002). Likewise, the incompressibility constraint $\delta^1 v = 0$ expresses divergence-freedom as a structural property rather than a penalty to enforce (Elcott et al., 2007; Mohamed et al., 2016). More broadly, many physical laws are organized by the identity $d^{k+1} \circ d^k = 0$, which encodes, in a single algebraic statement, facts such as the absence of magnetic monopoles, the exactness of conservative fields, and the compatibility of flows (Arnold et al., 2006). All are consequences of the same geometric principle: the boundary of a boundary is empty. The natural discrete setting for this is a cell complex K consisting of vertices, edges, faces, and higher-dimensional cells with orientations: the minimal structure on which Stokes' theorem holds exactly. The theorem forces $B_k B_{k+1} = 0$, from which the exterior derivative $d^k = B_{k+1}^{\top}$, the codifferentials δ^k, and the Hodge Laplacians Δ_k all follow, forming the discrete exterior calculus on K. On K, the governing laws of a broad class of physical systems define an operator

$$G : \bigoplus_i C^{k_i}(K; \mathbb{R}^{d_i}) \longrightarrow \bigoplus_j C^{\ell_j}(K; \mathbb{R}^{r_j}),$$

implicitly characterized by

$$\mathcal{F}_k\big(\sigma,\ u^k,\ d^{k-1}u^{k-1},\ \delta^{k+1}u^{k+1},\ \Delta_k u^k,\ a,\ f\big) = 0, \qquad \forall\, k,\ \forall\, \sigma \in K_k, \tag{mPDE$'$}$$

where u^k are unknown cochains, a encodes material parameters at their natural degree, f a source, and $\mathcal{F}_k$ a possibly nonlinear local functional. The structure of (mPDE) is what makes the problem hard for existing architectures: the equation at degree k couples u^k to cochains one degree below via $d^{k-1}u^{k-1}$ and one degree above via $\delta^{k+1}u^{k+1}$ — cross-dimensional coupling that neither a grid nor a graph can express. Maxwell, incompressible Navier–Stokes, and linear elasticity are all instances of (mPDE); single-degree equations are the degenerate exception.

A.3.1 Example PDEs

We illustrate (Evol) with two standard systems whose fields live on different cochain degrees and couple through d and δ.

Example 1 (Discrete Maxwell-type coupling)**.** Maxwell's equations describe how electric and magnetic fields evolve together: a changing magnetic field induces an electric field, and a changing electric field, together with current, induces a magnetic field. On an oriented 3-complex K, the electric field is naturally represented as an edge cochain $E \in C^1$, while the magnetic flux is represented as a face cochain $B \in C^2$ (Bossavit, 1998; Hiptmair, 2002). The core discrete coupling has the form

$$\partial_t B = -d^1 E, \qquad \partial_t E = \delta^2 B - J,$$

with constraints

$$d^2 B = 0, \qquad \delta^1 E = -\rho.$$

Thus E drives B through d^1, which measures circulation around faces, while B drives E through δ^2, which aggregates adjacent face fluxes onto edges. The essential point is not the specific physical normalization, but the cross-degree structure: fields living on different ranks of the complex interact through the exterior derivative and its adjoint.

Example 2 (Discrete wave-type coupling)**.** The wave equation describes how a disturbance propagates through an elastic medium: a displacement at one point pulls on its neighbors, which pull on theirs, and the wave travels outward. Let $u \in C^0$ be a scalar displacement on vertices, and let $p \in C^1$ be an edge variable encoding local stretch or momentum-like flux. A first-order discrete wave system can be written schematically as

$$\partial_t u = \delta^1 p, \qquad \partial_t p = -c^2 d^0 u,$$

where c is the wave speed. Here d^0 sends vertex displacements to edge differences, while δ^1 aggregates edge quantities back to vertices. Eliminating p gives

$$\partial_t^2 u = -c^2 \delta^1 d^0 u = -c^2 \Delta_0 u,$$

showing that the wave operator itself is built from the same cross-degree maps.

B PDE Templates as Typed Cochain Systems

The previous sections define the main objects of the framework: a physical state is a tuple of cochains, the operators d, δ, Δ provide the allowed transport between and within degrees, and a TNO layer learns how to mix the transported features without learning the incidence support itself. The purpose of this section is to make this correspondence explicit for standard PDE systems.

A useful example to explain this correspondence is Maxwell's equations illustrated in Figure 11. In a compatible discretization, the electric field is not naturally a scalar value at vertices; it is a circulation along oriented edges. Likewise, the magnetic field is a flux through oriented faces (Bossavit, 1998; Hiptmair, 2002). Thus

$$E \in C^1(K), \qquad B \in C^2(K).$$

The discrete curl is the coboundary

$$d^1 : C^1(K) \to C^2(K),$$

which sends edge circulations to face fluxes. Its metric adjoint,

$$\delta^2 : C^2(K) \to C^1(K),$$

Table 5: PDE templates as typed cochain systems. The DEC column shows the structural skeleton of the equation, while the TNO column shows the corresponding learned nonlinear cochain map.

PDE / system	DEC core structure	TNO generalization	Practical challenge / insight
Poisson / diffusion	$\Delta_0 u = f, \quad \Delta_0 = \delta^1 d^0$	$H^0_{\text{out}} = \phi_\theta(H^0, \Delta_0 H^0 W_\Delta, f)$	Learns a Green-type solution operator on irregular meshes.
Heat equation	$\partial_t u + \Delta_0 u = f$	$H^0_{t+\Delta t} = \phi_\theta(H^0_t, \Delta_0 H^0_t W_\Delta, f_t, t)$	Time enters as a feature, recurrent state, or learned time-step map.
Advection–diffusion reaction	$\partial_t c + \mathcal{L}_v c = -\kappa \Delta_0 c + R(c)$	$H^0_{\text{out}} = \phi_\theta(H^0, \mathcal{L}_v H^0, \Delta_0 H^0 W_\Delta, R_\theta(H^0))$	Needs stable advection, upwinding, and control of stiff reactions.
Darcy / mixed elliptic	$q = -\mathcal{K}_a\, d^0 p, \quad \delta^1 q = -f$	$H^{0:1}_{\text{out}} = \phi_\theta(\mathcal{B}_{01} H^{0:1} W, a, f)$	Pressure and flux are kept as different cochain types; conservation is native.
Conservation law	$\partial_t \rho - \delta^1 q = s$	$H^{0:1}_{\text{out}} = \phi_\theta(H^0, H^1, \delta^1 H^1, s)$	Telescoping conservation follows from incidence algebra.
Wave, first-order	$\partial_t z = \mathcal{A}_{01} z, \quad z = (u, p)$	$H^{0:1}_{t+\Delta t} = \phi_\theta(H^{0:1}_t, \mathcal{A}_{01} H^{0:1}_t)$	Same cross-rank structure as Darcy, but used for evolution; energy matters.
Maxwell	$\partial_t z = \mathcal{A}_{12} z - (J, 0), \quad z = (E, B)$	$H^{1:2}_{\text{out}} = \phi_\theta(\mathcal{B}_{12} H^{1:2} W, J)$	Separates electric circulation from magnetic flux; gauge and Gauss constraints matter.
Incompressible Navier–Stokes	$\partial_t u + \mathcal{L}_u u = -d^0 p - \nu \Delta_1 u, \quad \delta^1 u = 0$	$H^1_{\text{out}} = \Pi_{\delta=0} \phi_\theta(H^1, \mathcal{L}_{H^1} H^1, \Delta_1 H^1 W_\Delta, \delta^2 H^2 W_\delta)$	Nonlinear advection must be paired with a hard divergence-free constraint.
Compressible Euler / NS	Continuity, momentum, and energy equations with fluxes	$H^{0:2}_{\text{out}} = \phi_\theta(H^{0:2}, dH, \delta H, \mathcal{F}_\theta(H^{0:2}))$	Requires shock-aware fluxes, positivity, and entropy consistency.
Elasticity	$-\operatorname{div} \sigma = f, \quad \sigma = \mathbb{C} : \varepsilon(u), \quad \varepsilon_e(u) = t_e^\top (d^0 u)_e, \quad u \in C^0(K; \mathbb{R}^n)$	$H^0_{\text{out}} = \phi_\theta(H^0, \varepsilon(H^0) W_\varepsilon, \Delta_0 H^0 W_\Delta, \mathbb{C})$	Needs vector-valued cochains: edge elongations $t_e^\top (d^0 u)_e$ (t_e the unit edge tangent) fix $\operatorname{sym} \nabla u$ per simplex; a full treatment uses the elasticity complex. The constitutive law is load-bearing.
Shallow water	$\partial_t h - \delta^1(hu) = 0$, momentum balance	$H^{0:1}_{\text{out}} = \phi_\theta(H^0, H^1, \delta^1(H^0 H^1), d^0 H^0, b)$	Couples height, flux, bathymetry, wetting/drying, and positivity.
Vorticity–stream	$\partial_t \omega + \{\psi, \omega\} = -\nu \Delta_2 \omega, \quad \Delta_2 \psi = \omega$	$H^2_{\text{out}} = \phi_\theta(H^2, \text{Poisson}_\theta(H^2), \mathcal{L}_{H^1} H^2, \Delta_2 H^2 W_\Delta)$	The Poisson / Biot–Savart solve is the global component.
Mean-field games Optimal transport	Continuity coupled with Hamilton–Jacobi–Bellman	$(\rho_{\text{out}}, \varphi_{\text{out}}) = \phi_\theta(\rho, \varphi, d^0 \varphi, \Delta_0 \varphi, \delta^1(\rho\, d^0 \varphi))$	Requires monotonicity, Wasserstein geometry, and stable density–potential coupling.
General k-form Hodge flow	$\partial_t u^k + \Delta_k u^k = f^k, \quad \Delta_k = \Delta_k^\uparrow + \Delta_k^\downarrow$	$H^k_{\text{out}} = \phi_\theta(H^k, \Delta_k^\uparrow H^k W_\uparrow, \Delta_k^\downarrow H^k W_\downarrow, f^k)$	Universal form-valued PDE template; only the active operator block changes.

$$\mathcal{B}_{01} = \begin{bmatrix} \Delta_0 & \delta^1 \\ d^0 & \Delta_1^\downarrow \end{bmatrix}, \qquad \mathcal{B}_{12} = \begin{bmatrix} \Delta_1^\uparrow & \delta^2 \\ d^1 & \Delta_2^\downarrow \end{bmatrix}, \qquad \mathcal{A}_{01} = \begin{bmatrix} 0 & \delta^1 \\ -c^2 d^0 & 0 \end{bmatrix}, \qquad \mathcal{A}_{12} = \begin{bmatrix} 0 & \delta^2 \\ -d^1 & 0 \end{bmatrix}.$$

sends face fluxes back to edge-supported quantities. Ignoring material weights for notational clarity, the semi-discrete Maxwell system has the typed form

$$\partial_t B = -d^1 E, \qquad \partial_t E = \delta^2 B - J,$$

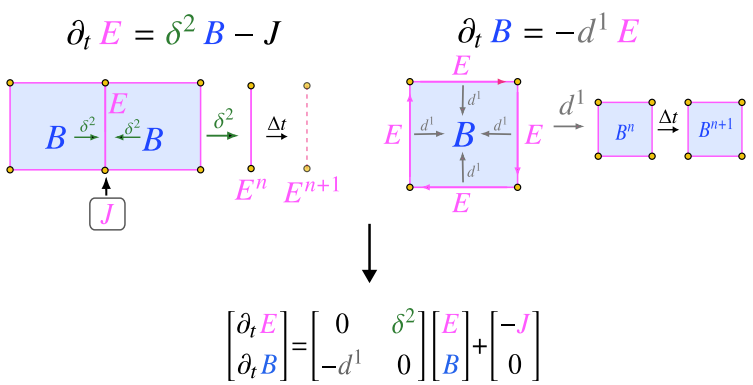


Figure 11: Maxwell coupling as cross-rank cochain transport. The electric field E lives on edges and the magnetic flux B lives on faces. Faraday's law uses d^1 to map edge circulation to face updates, while the Ampère–Maxwell law uses δ^2 to map neighboring face fluxes back to edge updates, with source J. Together, these two typed updates form the block Maxwell system.

$$\begin{bmatrix}\partial_t E\\ \partial_t B\end{bmatrix} = \begin{bmatrix}0 & \delta^2\\ -d^1 & 0\end{bmatrix}\begin{bmatrix}E\\ B\end{bmatrix} + \begin{bmatrix}-J\\ 0\end{bmatrix} \qquad \begin{bmatrix}H^1_{\text{out}}\\ H^2_{\text{out}}\end{bmatrix} = \phi_\theta\left(\begin{bmatrix}\Delta_1^{\uparrow} & \delta^2\\ d^1 & \Delta_2^{\downarrow}\end{bmatrix}\begin{bmatrix}H^1\\ H^2\end{bmatrix} W, J\right)$$

(a) Typed Maxwell coupling. (b) Induced TNO layer.

Figure 12: From Maxwell coupling to a TNO template. **(a)** The Maxwell system is a typed cochain system: the electric field E lives on edges, the magnetic flux B lives on faces, and the PDE couples them through the cross-rank DEC maps $d^1 : C^1 \to C^2$ and $\delta^2 : C^2 \to C^1$, with source J acting on the edge equation. **(b)** The corresponding TNO layer keeps this typed routing. The off-diagonal blocks d^1 and δ^2 transport information between edge and face cochains, while the diagonal Hodge-Laplacian channels $\Delta_1^{\uparrow}$ and $\Delta_2^{\downarrow}$ perform same-rank propagation. Channel maps act on the feature dimension after DEC transport and are suppressed in the visual notation for readability. Learning enters through the channel weights $W^{\bullet}$ and the nonlinear update ϕ_θ, which determine how the routed features are mixed rather than where information can flow.

or, equivalently,

$$\partial_t \begin{bmatrix}E\\ B\end{bmatrix} = \begin{bmatrix}0 & \delta^2\\ -d^1 & 0\end{bmatrix}\begin{bmatrix}E\\ B\end{bmatrix} - \begin{bmatrix}J\\ 0\end{bmatrix}.$$

This is precisely the kind of coupling encoded by the TNO block: the equation does not merely say that two feature arrays interact; it specifies that one field lives on edges, the other on faces, and that the interaction must pass through the incidence operators of the complex. The corresponding learned cochain update has the same structural support,

$$\begin{bmatrix}H^1_{\text{out}}\\ H^2_{\text{out}}\end{bmatrix} = \phi_\theta\left(\begin{bmatrix}\Delta_1^{\uparrow} & \delta^2\\ d^1 & \Delta_2^{\downarrow}\end{bmatrix}\begin{bmatrix}H^1\\ H^2\end{bmatrix} W, J\right).$$

Here H^1 and H^2 are hidden edge and face cochains. The block matrix fixes where information can flow: edge-to-face through d^1, face-to-edge through δ^2, and same-rank propagation through the Hodge Laplacian channels. The source $J \in C^1(K)$ is an external edge-supported input to the update, not part of the DEC transport block. The trainable part, W and ϕ_θ, learns how these transported features are combined. The signs and physical constants in the Maxwell operator may be fixed explicitly or absorbed into the learned channel maps; the displayed TNO block is meant to preserve the typed incidence support of the PDE. See Figure 12.

This example also shows the exact-sequence structure. Namely, the identity $d^2 d^1 = 0$ implies

$$\partial_t(d^2 B) = -d^2 d^1 E = 0.$$

Hence, the magnetic Gauss constraint is preserved by the algebra of the complex. Similarly, $\delta^1 \delta^2 = 0$, together with the discrete charge-continuity law, gives the corresponding compatibility condition for

the electric equation. These identities are inherited from the incidence structure, not learned as soft penalties.

Darcy (following sec. 5). The Darcy / mixed-elliptic system studied in sec. 5 constructs a typed routing across all three cochain ranks of a 2D complex. The pressure is defined as a scalar at the vertices, and the Darcy flux is a line integral along oriented edges, and the diffusivity is face-supported (a scalar on faces in the isotropic case, a rank-2 tensor on faces in sec. 5.3):

$$p \in C^0(K), \qquad q \in C^1(K), \qquad a \in C^2(K),$$

with constitutive and balance equations

$$q = -\mathcal{K}_a d^0 p, \qquad \delta^1 q = -f.$$

Here $\mathcal{K}_a : C^1(K) \to C^1(K)$ is the Darcy constitutive map induced by the face-supported coefficient a, mapping edge gradients to edge fluxes (Hyman et al., 1997; Lipnikov et al., 2014). A TNO block on this system carries hidden pressure and flux cochains H^0, H^1 and routes them through the $\mathcal{B}_{01}$ block of Table 5; the material field a enters at its native rank as a face-valued input channel that conditions the constitutive map. This is the synthetic mesh family we examine in sec. 5.3. Incident-cell averaging loses individual face orientations, although the projected channels retain useful local information. The rank-2 path receives the full face field, which determines the tensor in each triangle. The same incidence algebra makes the residual $\delta^1 q + f$ a vertex-supported quantity computed exactly by the complex. Thus mass balance can be enforced, penalized, or projected at the correct cochain degree rather than approximated through an untyped graph message.

Together, Maxwell and Darcy cover two qualitatively distinct roles a typed coupling can play: evolutionary cross-rank dynamics with gauge structure (Maxwell), and an elliptic constraint with native discrete conservation (Darcy). Poisson activates a single 0-cochain and the scalar Hodge Laplacian Δ_0. Fluids, elasticity, conservation laws, and transport systems introduce problem-specific nonlinearities, such as advection, constitutive maps, learned fluxes, projections, or positivity constraints. In all cases, the same principle remains: the PDE determines the typed transport, while the TNO learns the nonlinear cochain map supported on that transport.

Table 5 summarizes this correspondence and should be read as an unpacking of the multi-degree PDE form and the block TNO layer, rather than as a collection of separate architectures.

C Proofs

C.1 Admissible TNO operations

We make the construction in dfn. 4.1 explicit. A realization specifies the family of complexes, the geometric data it uses, and the following operations. The same rules apply to preprocessing, encoding, hidden layers, and decoding. Sums, channel concatenations at the same degree, and finite compositions of admissible operations are allowed.

Cellwise maps and cellular transport. A cellwise map has the form

$$z_\sigma^k = \phi_{\theta,k}(h_\sigma^k, a_\sigma),$$

where a_σ denotes cell data and the function is shared over cells of degree k. Fixed cellular transports include B_k, $B_k^\top$, and their compositions. The coboundary is $d^k = B_{k+1}^\top$; supplying Hodge stars additionally permits δ and Δ. No Hodge operator or decomposition is required. A learned incidence transport can take the form

$$m_\sigma^k = \sum_{\tau \in \mathcal{N}_{kj}(\sigma)} w_{\sigma\tau}\, \psi_{\theta,kj}(h_\sigma^k, h_\tau^j, a_{\sigma\tau}),$$

where $\mathcal{N}_{kj}$ is an incidence relation determined by the boundary operators or a composition of such relations. Thus cross-degree transport follows (P2), with a shared parameterization for each degree-to-degree route. The weights $w_{\sigma\tau}$ can include orientation signs, Hodge coefficients, or quadrature weights. The message function is shared across the relation; learned scalar weights, including normalized attention weights, may also be generated by shared functions. This permits geometry-dependent kernels without independently learning a matrix entry for each incidence. An auxiliary

interaction graph may be constructed from coordinates by a stated rule, such as a fixed physical radius, provided that the construction respects cell relabeling.

Spectral operations. Let V_k^K contain a selected basis of rank-k modes and let A_k^K be its specified analysis map, with $A_k^K V_k^K = I$. Both the basis construction and its normalization are part of the realization. If a Hodge star M_k is supplied and the basis is M_k-orthonormal, $A_k^K = (V_k^K)^* M_k$. A spectral operation has the form

$$h^k \longmapsto V_k^K \Phi_{\theta,k}(A_k^K h^k; \lambda^K),$$

where λ^K denotes mode descriptors. We allow shared maps applied to each mode, shared kernels between modes, and shared maps on a fixed-size retained coefficient array. Modewise maps may depend on the mode descriptor. A kernel between modes has entries generated by the same function of the source and target descriptors, with a specified summation or normalization rule; it does not assign independent parameters to every mode pair. Thus a fixed set of Fourier modes may have distinct learned channel matrices, while the number of learned parameters remains independent of the mesh size. When a Hodge Laplacian is used, its harmonic projector is obtained by retaining the harmonic subspace and taking Φ to be the identity.

Bases are computed from the complex and its specified geometry, independently of target outputs; they are not free per-mesh parameters. Modewise maps must share the same parameterization as the number of available modes changes. Fixed-size coefficient maps must specify the retained modes and their ordering on every supported complex. These requirements exclude an unrestricted learned matrix whose dimensions grow with the mesh.

For a cell permutation π_k, an attributed basis and its analysis map transform as $V_k^{\pi K} = \pi_k V_k^K$ and $A_k^{\pi K} = A_k^K \pi_k^{-1}$. Recomputed bases must yield the same output under relabeling: either a geometric convention fixes signs and repeated eigenspaces, or the coefficient operation is equivariant to the remaining basis changes. Independence of arbitrary spectral frames is an additional property when the frame is supplied as domain data.

Transfers and hierarchies. An auxiliary or coarse complex and its transfer maps are constructed by a specified geometric or incidence rule. A transfer has a declared degree and a declared relation between source and target cells; its entries are fixed geometric weights or are generated by shared functions of the associated data. Learned coarsening is allowed under the same rules. If the number of levels varies, the realization specifies how parameters are reused so that the parameter space stays fixed. An auxiliary CC may supply extra incidence relations, but $d^2 = 0$ is asserted only for operators belonging to an actual chain complex.

What this guarantees. Each operation has a specified action on cochains, and the complete construction shares parameters and commutes with cell relabeling. New realizations can combine these operations without using the particular DEC layer of eq. (8). In particular, the class includes some ordinary message-passing models; its additional scope is the treatment of fields and interactions at arbitrary degrees.

Orientation equivariance can be imposed separately: reversing cell orientations and transforming incidence matrices and oriented cochains consistently must then produce the corresponding sign changes in the output. Generic cellwise nonlinearities need not satisfy this property. Likewise, learned incidence maps need not obey $d_\rho^2 = 0$ or make δ_ρ the Hodge adjoint of d_ρ. Such identities, conservation properties, and consistency under refinement require additional assumptions on the realization.

C.2 Proof of prop. 4.2 [FNO-Recovery]

Proof. The vertices of K_N carry 0-cochains, and its cubical incidences specify the periodic grid connectivity. The incidence-derived vertex operator $B_1 B_1^\top$ is translation invariant and diagonalized by the discrete Fourier basis. This supplies a spectral realization on the same cellular connectivity, without requiring Hodge stars or a Hodge decomposition.

Fix a finite frequency set $\Lambda = -\Lambda \subset \mathbb{Z}^d$ and supported grids with $N_r > 2\max_{\xi\in\Lambda} |\xi_r|$ for every coordinate r. Let $R_\theta(\xi)$ vanish outside Λ and satisfy $R_\theta(-\xi) = \overline{R_\theta(\xi)}$. Use the coordinate-specified Fourier modes indexed by Λ as the selected rank-0 basis. Analysis is the normalized discrete Fourier

transform restricted to these modes, and synthesis is the inverse transform with zero coefficients outside Λ. The grid-size assumption ensures that the selected frequencies are distinct on every supported grid, so $A_0^{K_N} V_0^{K_N} = I$. Choose the coefficient map

$$(\Phi_\theta \widehat{u})(\xi) = R_\theta(\xi)\widehat{u}(\xi), \qquad \xi \in \Lambda.$$

This is a fixed-size spectral map, with the same frequency labels and parameters on every grid. The symmetry of Λ and the conjugate-symmetry condition on R_θ give real outputs for real inputs. Adding the shared cellwise map Wu gives

$$Wu + V_0^{K_N} \Phi_\theta A_0^{K_N} u = Wu + \mathcal{F}_N^{-1} R_\theta \mathcal{F}_N u.$$

To check (P1), let P permute the vertices and their coordinates. The Fourier analysis matrix becomes $\mathcal{F}_N P^{-1}$ and synthesis becomes $P\mathcal{F}_N^{-1}$. Therefore

$$\big(P\mathcal{F}_N^{-1}\big) R_\theta \big(\mathcal{F}_N P^{-1}\big)(Pu) = P\mathcal{F}_N^{-1} R_\theta \mathcal{F}_N u.$$

and hence the cellwise term also commutes with P. All features remain at rank 0, so the cross-degree condition (P2) is trivially fulfilled. The fixed frequency set and shared matrices W and $R_\theta(\xi)$ give (P3). Shared cellwise biases, activations, lifting, and readout can be added and layers composed to recover the corresponding nonlinear FNO network. □

This inclusion uses the coordinate-specified Fourier basis and frequency-specific multipliers. The cubical connectivity provides a compatible cellular setting, but does not require the Fourier channel to have nearest-neighbor support. This multiplier also supports duplicity: distinct Fourier modes can share the same eigenvalue but have different learned matrices. In this way, we recover a same-degree spectral realization through the framework in dfn. 4.1.

C.3 Proof of prop. 4.3 [GNO-Recovery]

Proof. Treat each directed non-self interaction as an oriented edge. For $e : v_j \to v_i$, define endpoint selection matrices S and T by $S_{ej} = 1$ and $T_{ei} = 1$, with other entries zero. The graph boundary matrix is $B_1 = T^\top - S^\top$; equivalently, $S = (-B_1^\top)_+$ and $T = (B_1^\top)_+$, where $(\cdot)_+$ denotes the entrywise positive part. Thus both routes are determined by boundary incidence. Assign the edge message

$$m_e = \omega_j \kappa_\theta(x_i, x_j) u_j.$$

This is a vertex-to-edge transport: S selects the source value, and the same kernel function generates the channel map on every edge. A second incidence transport sums m_e into its target vertex using $T^\top$. Consequently,

$$(T^\top m)_i = \sum_{j \in \mathcal{N}(i),\, j \neq i} \omega_j \kappa_\theta(x_i, x_j) u_j.$$

Any $j = i$ term is a cellwise kernel map. Adding this term and Wu_i gives exactly the update in prop. 4.3.

For vertex and induced edge permutations P and Q, the endpoint matrices become $S' = QSP^{-1}$ and $T' = QTP^{-1}$. The relabeled edge data generate $m' = Qm$, hence $T'^\top m' = PT^\top m$. The remaining maps are shared and cellwise, satisfying (P1). The two boundary-supported routes $0 \to 1 \to 0$ satisfy (P2). The matrices W and kernel parameters θ can be shared across graphs, as they are independent of the number of vertices and edges, proving (P3). Shared cellwise biases and activations may be added, and layers composed, to recover the corresponding nonlinear GNO network. □

The edge messages are auxiliary to the rank-0 input/output map. The two incidence transports are independent and need not be adjoints. In the signed route notation of sec. C.5, using $B_1^\top$ and B_1 as the underlying transports, choose source-to-edge transport $-\omega_j \kappa_\theta(x_i, x_j)$, target-to-edge transport zero, edge-to-target transport I, and edge-to-source transport zero. The source incidence sign cancels the minus sign. This realizes the GNO kernel through learned incidence maps without requiring Hodge stars or identifying the kernel with a fixed DEC Laplacian.

C.4 DISCRETIZATION CONSISTENCY

The definition concerns a family of discrete maps. The recovery results above are exact identities on each supported discretization; neither parameter sharing nor these identities alone establishes a continuum limit. Agreement under refinement is also an additional property: for specified prolongation $P_{K\to K'}$ and restriction $R_{K'\to K}$, it asks that $R_{K'\to K}\mathcal{T}_\theta^{K'}P_{K\to K'}x \approx \mathcal{T}_\theta^K x$ in a stated norm and input class, with error controlled by discretization assumptions.

For a family K_h discretizing a fixed domain, let X and Y be continuum field spaces with the required degrees, $\mathcal{S}_h^X : X \to \mathcal{X}(K_h)$ an input discretization, and $\mathcal{I}_h^Y : \mathcal{Y}(K_h) \to Y$ an output reconstruction. A realization is *consistent* with a continuum map $\mathcal{T}_\theta : X \to Y$ on an admissible class $\mathcal{A} \subset X$ if, for each fixed θ,

$$\sup_{u\in\mathcal{A}} \left\| \mathcal{I}_h^Y \mathcal{T}_\theta^{K_h} \mathcal{S}_h^X u - \mathcal{T}_\theta u \right\|_Y \longrightarrow 0 \qquad (h \to 0). \tag{17}$$

The field class, norm, discretization, and reconstruction must be specified. This is an additional property, separate from approximation of the target PDE operator. It is also distinct from transfer to domains with different topology. The distinction between accepting different resolutions and converging to a continuum operator is discussed by Kovachki et al. (2023).

Fourier transfer. For the linear FNO layer in prop. 4.2, Fourier prolongation P and restriction R give $RT_{\theta,N'}P = T_{\theta,N}$ on a common resolved Fourier subspace, when transforms use compatible normalization. The multiplier at each shared frequency is identical on both grids. Nonlinear activations can introduce unresolved frequencies, so this exact identity does not automatically extend to nonlinear stacks. A continuum convergence claim additionally requires control of sampling and aliasing errors and of the nonlinear maps.

Kernel quadrature. For a GNO integral channel, let $\{V_j\}$ partition a finite-measure domain D, with $x_j \in V_j$, $\omega_j = \mu(V_j)$, and $\operatorname{diam}(V_j) \le h$. Fix the coefficient field entering the kernel. Suppose $y \mapsto \kappa_\theta(x, y)u(y)$ is uniformly L-Lipschitz in y, over output locations x and inputs in the stated class. Then

$$\sup_i \left\| \sum_j \omega_j \kappa_\theta(x_i, x_j)u(x_j) - \int_D \kappa_\theta(x_i, y)u(y)\, d\mu(y) \right\| \le L\mu(D)h.$$

The error over each V_j is at most $Lh\mu(V_j)$; summing gives the bound. A Lipschitz activation multiplies the bound by its Lipschitz constant when the other terms are unchanged. Neighborhood sums require the kernel to vanish outside the retained interactions or a separate truncation estimate. This proves consistency of the integral channel at sampled output points under the stated assumptions. A result of the form eq. (17) also requires a consistent output reconstruction and stability when layers are composed.

C.5 CELL-WISE REALIZATION AND COPRESHEAF VARIANT

The block update in sec. 4.2 can be written cell by cell. This form makes explicit that the DEC realization is local, but with locality defined by DEC rather than by an arbitrary graph adjacency. The relevant neighborhoods are induced by the routes d, δ, and the two Hodge–Laplacian channels $\Delta^\uparrow, \Delta^\downarrow$.

Let K_k denote the k-cells of K. For a hidden k-cochain $h^k \in C^k(K; \mathbb{R}^{d_k})$, write

$$h^k_\sigma := h^k(\sigma), \qquad \sigma \in K_k.$$

For a target k-cell σ, define the four DEC neighborhoods

$$\mathcal{N}^k_{\mathrm{ex}}(\sigma) = \{\rho \in K_{k-1} : \rho \prec \sigma\}, \qquad \mathcal{N}^k_{\mathrm{coex}}(\sigma) = \{\eta \in K_{k+1} : \sigma \prec \eta\},$$

and

$$\mathcal{N}^k_{\Delta\uparrow}(\sigma) = \{\tau \in K_k : (\Delta^\uparrow_k)_{\sigma\tau} \neq 0\}, \qquad \mathcal{N}^k_{\Delta\downarrow}(\sigma) = \{\tau \in K_k : (\Delta^\downarrow_k)_{\sigma\tau} \neq 0\}.$$

The first two neighborhoods are cross-rank. The exact route $\mathcal{N}^k_{\mathrm{ex}}$ receives from $(k-1)$-cells through d^{k-1}, while the coexact route $\mathcal{N}^k_{\mathrm{coex}}$ receives from $(k+1)$-cells through δ^{k+1}. The last two neighborhoods are same-rank. The upper Laplacian route couples k-cells through common

$(k+1)$-cells, while the lower Laplacian route couples k-cells through common $(k-1)$-cells. For $k=1$, these are edge-to-edge couplings through shared faces and shared vertices, respectively.

We now attach a copresheaf to each route. For each route

$$r \in \{\mathrm{ex}, \mathrm{coex}, \Delta^{\uparrow}, \Delta^{\downarrow}\},$$

the neighborhood function $\mathcal{N}_r^k$ induces a directed graph whose edges are

$$y \to \sigma \qquad \text{whenever} \qquad y \in \mathcal{N}_r^k(\sigma).$$

A route-dependent copresheaf assigns a feature space F_x to every cell x appearing in this directed graph and a linear transport map

$$\rho^r_{y\to\sigma} : F_y \to F_\sigma$$

to every directed neighborhood edge $y \to \sigma$. In implementation, one often takes $F_\sigma \simeq \mathbb{R}^{d_k}$ for all k-cells σ, but the notation allows different ranks, cell types, or incidences to carry different feature spaces.

The DEC scalar coefficient along each route is kept separate from the copresheaf map. Define

$$a^{k,\mathrm{ex}}_{\sigma\rho} = [\sigma : \rho], \qquad a^{k,\mathrm{coex}}_{\sigma\eta} = (\delta^{k+1})_{\sigma\eta},$$

and

$$a^{k,\Delta^{\uparrow}}_{\sigma\tau} = (\Delta^{\uparrow}_k)_{\sigma\tau}, \qquad a^{k,\Delta^{\downarrow}}_{\sigma\tau} = (\Delta^{\downarrow}_k)_{\sigma\tau}.$$

The coefficient $a^{k,\mathrm{ex}}$ is purely incidence and orientation. The coefficients involving δ and Δ carry the metric or material weights induced by the Hodge stars. The learned copresheaf map $\rho^r_{y\to\sigma}$ then transports the source feature into the target feature space. Equivalently, the scalar DEC coefficient could be absorbed into $\rho^r_{y\to\sigma}$, but separating them makes clear what is fixed by DEC and what is learned.

The cell-wise update has four routed copresheaf channels:

$$m^{k,\mathrm{ex}}(\sigma) = \bigoplus_{\rho\in\mathcal{N}^k_{\mathrm{ex}}(\sigma)} \psi^{\mathrm{ex}}_\theta\big(h^k_\sigma,\ a^{k,\mathrm{ex}}_{\sigma\rho}\,\rho^{\mathrm{ex}}_{\rho\to\sigma} h^{k-1}_\rho\big)\,.$$

$$m^{k,\mathrm{coex}}(\sigma) = \bigoplus_{\eta\in\mathcal{N}^k_{\mathrm{coex}}(\sigma)} \psi^{\mathrm{coex}}_\theta\big(h^k_\sigma,\ a^{k,\mathrm{coex}}_{\sigma\eta}\,\rho^{\mathrm{coex}}_{\eta\to\sigma} h^{k+1}_\eta\big)\,.$$

$$m^{k,\Delta^{\uparrow}}(\sigma) = \bigoplus_{\tau\in\mathcal{N}^k_{\Delta^{\uparrow}}(\sigma)} \psi^{\Delta^{\uparrow}}_\theta\Big(h^k_\sigma,\ a^{k,\Delta^{\uparrow}}_{\sigma\tau}\,\rho^{\Delta^{\uparrow}}_{\tau\to\sigma} h^k_\tau\Big)\,.$$

$$m^{k,\Delta^{\downarrow}}(\sigma) = \bigoplus_{\tau\in\mathcal{N}^k_{\Delta^{\downarrow}}(\sigma)} \psi^{\Delta^{\downarrow}}_\theta\Big(h^k_\sigma,\ a^{k,\Delta^{\downarrow}}_{\sigma\tau}\,\rho^{\Delta^{\downarrow}}_{\tau\to\sigma} h^k_\tau\Big)\,.$$

Here $\bigoplus$ is a permutation-invariant aggregation, such as a sum, mean, or attention-weighted sum. Each message first transports the source feature through the copresheaf map $\rho^r_{y\to\sigma}$, then weights it by the DEC coefficient of the corresponding route, and finally applies the learnable message function ψ^r_θ.

The output feature at σ is

$$h^k_{\mathrm{out}}(\sigma) = \phi_\theta\Big(h^k_\sigma,\ m^{k,\mathrm{ex}}(\sigma),\ m^{k,\mathrm{coex}}(\sigma),\ m^{k,\Delta^{\uparrow}}(\sigma),\ m^{k,\Delta^{\downarrow}}(\sigma)\Big)\,.$$

The functions ψ^r_θ form route-wise messages, while ϕ_θ mixes the self feature with the four routed channels. The DEC operators determine which cells can communicate and provide the orientation/Hodge coefficients. The copresheaf maps determine how features are transported between the local feature spaces attached to the source and target cells.

In operator form, the route-wise copresheaf transports are

$$(d^{k-1}_\rho h^{k-1})_\sigma = \sum_{\rho\prec\sigma} [\sigma:\rho]\,\rho^{\mathrm{ex}}_{\rho\to\sigma} h^{k-1}_\rho,$$

$$(\delta_\rho^{k+1} h^{k+1})_\sigma = \sum_{\eta \succ \sigma} (\delta^{k+1})_{\sigma\eta}\, \rho^{\mathrm{coex}}_{\eta\to\sigma} h^{k+1}_\eta,$$

and

$$(\Delta^{\uparrow}_{k,\rho} h^k)_\sigma = \sum_{\tau \in \mathcal{N}^k_{\Delta^\uparrow}(\sigma)} (\Delta^{\uparrow}_k)_{\sigma\tau}\, \rho^{\Delta^\uparrow}_{\tau\to\sigma} h^k_\tau,$$

$$(\Delta^{\downarrow}_{k,\rho} h^k)_\sigma = \sum_{\tau \in \mathcal{N}^k_{\Delta^\downarrow}(\sigma)} (\Delta^{\downarrow}_k)_{\sigma\tau}\, \rho^{\Delta^\downarrow}_{\tau\to\sigma} h^k_\tau.$$

Thus the copresheaf variant replaces the fixed DEC transports by route-wise transported DEC operators:

$$d^{k-1} \rightsquigarrow d^{k-1}_\rho, \qquad \delta^{k+1} \rightsquigarrow \delta^{k+1}_\rho, \qquad \Delta^{\uparrow}_k, \Delta^{\downarrow}_k \rightsquigarrow \Delta^{\uparrow}_{k,\rho}, \Delta^{\downarrow}_{k,\rho}.$$

The fixed DEC layer is recovered by taking the copresheaf maps to be identity maps, or shared channel maps independent of the incident pair. Then the only route coefficients are the signed incidence coefficients and the Hodge-weighted DEC coefficients. The copresheaf version keeps the same DEC supports but allows the feature transport itself to vary by route, rank, orientation, geometry, cell type, or learned attributes. This gives a controlled extension of the DEC layer: the topology fixes the admissible routes, while the copresheaf maps learn how features are translated along those routes.

The notation identifies the DEC route whose support is used. Independently learned route maps do not in general satisfy $d^{k+1}_\rho d^k_\rho = 0$, make δ_ρ the Hodge adjoint of d_ρ, or make the separately weighted Laplacian routes equal to compositions of d_ρ and δ_ρ. These properties require compatibility constraints beyond the TNO definition. Orthogonal fiber maps preserve the norm within each transported fiber, but do not by themselves impose these identities.

D Hierarchical TNOs: Multigrid Motivation and Architecture

HTNO borrows its structure from multigrid methods for de Rham complexes (Hiptmair, 1998; Arnold et al., 2000). For curl-type problems, point smoothers are ineffective on the exact components of a k-cochain (the near-kernel of the operator); Hodge-compatible smoothers therefore also relax on the neighboring degrees $C^{k\pm1}$, reached through d and δ, and treat the harmonic part separately. The TNO layer has the same typed routes (eq. (9)), with learned channel maps in place of fixed relaxation parameters, and HTNO adds a coarse-level correction. This is a structural analogy that motivates the architecture; a trained HTNO does not inherit the convergence guarantees of multigrid.

D.1 HTNO as a learned two-grid method

HTNO adds a coarse complex K^c and degree-wise transfer maps

$$R_k : C^k(K) \to C^k(K^c), \qquad \Pi_k : C^k(K^c) \to C^k(K).$$

For de Rham systems, the ideal structure-preserving condition is that transfer commute with the coboundary:

$$d^k_{K^c} R_k = R_{k+1} d^k_K, \qquad d^k_K \Pi_k = \Pi_{k+1} d^k_{K^c}.$$

This is the standard commuting-diagram condition in FEEC multigrid (Arnold et al., 2000; 2006; Arnold, 2018). When these identities hold, taking a coboundary and transferring levels agrees with transferring first and then taking the coboundary. In particular, exact cochains remain exact across levels, and the discrete differential structure is preserved by restriction and prolongation. Without such compatibility, a coarse correction can turn a gradient-like error into a non-gradient error after prolongation, weakening the decomposition on which Hodge-type smoothers rely (Hiptmair, 1998; Arnold et al., 2000).

In the present HTNO implementation, these commuting identities are not imposed as hard algebraic constraints. Rather, they serve as the FEEC ideal motivating the hierarchy. The TNO blocks on each level use the DEC operators of that level, so typed cochain transport is preserved level-wise. The inter-level maps used in our experiments are induced by a partition of the fine vertices and should therefore be viewed as cochain-compatible coarse correction maps, not as exact commuting-diagram transfers.

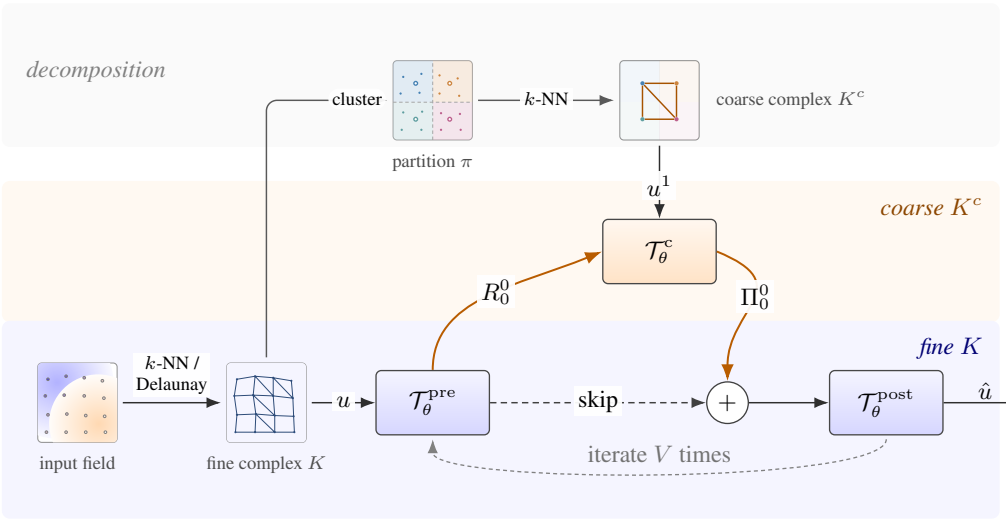


Figure 13: HTNO as a learned two-grid V-cycle. **Bottom:** the input field on a point cloud is lifted to a fine cell complex K (k-NN or Delaunay), giving an input cochain $u \in C^\bullet(K)$ that drives the V-cycle. The pre-smoother $\mathcal{T}_\theta^{\text{pre}}$ runs on K; its output is restricted by R_0^0 to the coarse complex, where $\mathcal{T}_\theta^c$ computes a correction; the correction is prolonged back by Π_0^0 and added to the pre-smoothed state via the skip; $\mathcal{T}_\theta^{\text{post}}$ closes the cycle, producing $\hat{u}$. The whole block can be iterated V times with shared weights. **Top:** the coarse complex is built once from K. Fine vertices are clustered (fixed k-means or learned soft Voronoi) into a partition π, and k-NN on the resulting centroids yields the coarse complex K^c, on which the coarse cochain u^c lives.

With this interpretation, an HTNO block is a learned two-grid V-cycle (fig. 13) (Trottenberg et al., 2001; Briggs et al., 2000) that pre-smooths on K with Hodge-structured TNO layers (Hiptmair, 1998; 2002; Arnold et al., 2000), restricts to K^c, computes an approximate correction on K^c with coarse-level TNO layers, prolongs the correction back to K, additively adds it to the pre-smoothed state, and post-smooths on K. With $L = 2$ this reads

$$\mathrm{HTNO}_\theta(h) \;=\; \mathcal{T}_\theta^{\text{post}}\Big(\,\mathcal{T}_\theta^{\text{pre}}(h) \;+\; \Pi_0^0\,\mathcal{T}_\theta^c\big(R_0^0\,\mathcal{T}_\theta^{\text{pre}}(h)\big)\,\Big). \tag{18}$$

Our implementation realizes R_0^0, Π_0^0 as mean-pool and broadcast over a partition $\pi : V(K) \to \{1, \dots, N_c\}$ of the fine vertices. The partition is either *fixed* (for example, Voronoi or k-means cells on the input coordinates), or *learned end-to-end* as a soft Voronoi diagram: N_c trainable query vectors attend over the warm-up features to predict centroid positions, soft assignments are obtained from $W = \mathrm{softmax}(-\|p - c\|^2/T)$ (or a Gaussian RBF kernel), and auxiliary spread and entropy losses on W keep the centroids from collapsing. The coarse complex K^c is built by k-NN over the centroids, inheriting its own incidence matrices and Hodge stars.

Higher-rank transfers ($k \geq 1$) would require structured prolongation and restriction maps, for example Whitney/Nédélec-style constructions, to satisfy the commuting identities exactly (Nédélec, 1980; Bossavit, 1998; Arnold et al., 2006). We do not impose such maps in the present implementation. Instead, we re-derive rank-1 and rank-2 features on K^c from the pooled vertex field using the coarse incidence structure. Thus $\mathcal{T}_\theta^c$ runs the multi-rank TNO update of eq. (9) on K^c when higher-rank features are present, and a rank-0 message-passing block otherwise. The fine–coarse–fine block used in our experiments is the $V = 1$ instance of eq. (18); iterating with shared weights for $V > 1$, or recursing on deeper hierarchies, gives learned multilevel V-cycles.

E Implementation Details

E.1 MPNN ablation baseline

The MPNN used in the ablations of sec. 5.4 is a standard residual message-passing network on rank-0 (vertex) features only, sharing the Delaunay graph and the node/edge inputs of TNO but with no

copresheaf, harmonic, or higher-rank channels. Inputs $x_v \in \mathbb{R}^{F_{\text{in}}}$ and edge features $e_{ij} \in \mathbb{R}^{F_e}$ are lifted to width w by a $\text{Dense} \to \text{swish} \to \text{LayerNorm}$ encoder applied independently on vertices and edges. Each of the L=12 residual layers, with $\bar{h} = \text{LN}(h)$, computes

$$m_{ij} = \text{swish}\big(W_m\,[\,\bar{h}_i \,\|\, \bar{h}_j \,\|\, e_{ij}\,]\big), \qquad a_i = \sum_{j:\, j\to i} m_{ij}, \tag{19}$$

$$h_i \;\leftarrow\; h_i \;+\; \text{LN}\Big(\text{swish}\big(W_2\,\text{swish}(W_1\,[\,\bar{h}_i \,\|\, a_i\,])\big)\Big), \tag{20}$$

with sum aggregation on the destination index. A final $\text{LN} \to \text{Dense}$ decoder produces the vertex output; on curl–curl, two-layer heads read edge outputs from $[\,h_i \,\|\, h_j \,\|\, e_{ij}\,]$ and face outputs from the three oriented incident edge latents. The matched-width baseline uses w=192, the parameter-matched baseline is widened to match or exceed the parameter count of the full TNO (cf. sec. F.7); all other optimizer/training settings are inherited from the ablation backbone.

E.2 Complexity of a TNO layer

Let $n_k = |K_k|$ be the number of rank-k cells, $n_\bullet = \sum_k n_k$, and $h = d_h$. Let

$$s_\bullet = \sum_{k=1}^{N} \text{nnz}(B_k), \qquad \lambda_\bullet = \sum_{k=0}^{N} \Big(\text{nnz}(\Delta_k^{\uparrow}) + \text{nnz}(\Delta_k^{\downarrow})\Big),$$

with "nnz" counting the non-zero entries for the sparse operators at runtime. A residual TNO layer has two costs. First, the DEC transports d^{k-1}, δ^{k+1}, and $\Delta_k^{\uparrow,\downarrow}$ are sparse matrix–dense matrix multiplies, costing $O(h(s_\bullet + \lambda_\bullet))$. Second, the learned maps are shared channel mixes applied cellwise, costing $O(n_\bullet h^2)$. Thus

$$T_{\text{TNO}}^{(1)} = O\big(h(s_\bullet + \lambda_\bullet) + n_\bullet h^2\big)\,. \tag{21}$$

For bounded-degree cell complexes, $s_\bullet + \lambda_\bullet = O(n_\bullet) = O(n_0)$, so a *rigid* DEC TNO layer is $O(n_0 h^2)$.

For the MPNN in eqs. (19) and (20), let n_0 be the number of vertices and m the number of directed edges; edge features are lifted to width w by the encoder, so they enter the layer at width w. The edge message W_m costs $O(mw^2)$, aggregation costs $O(mw)$, and the two-layer node update costs $O(n_0 w^2)$. Hence

$$T_{\text{MPNN}}^{(1)} = O\big((m + n_0)w^2\big)\,, \tag{22}$$

which is $O(n_0 w^2)$ under bounded degree.

Both layers are quadratic in their channel width: $O(n_0 w^2)$ for MPNN and $O(n_0 h^2)$ for rigid DEC TNO, but the cost is accrued in different places. An MPNN spends $O(w^2)$ *per edge* in the message MLP; a rigid DEC TNO spends only $O(h)$ per incidence in transport and $O(h^2)$ *per cell* across all ranks in channel mixing. The extra factor $n_\bullet$ instead of n_0 is the cost of carrying edge, face, and higher-rank cochains as native state variables, which is what prescribes typed cross-rank routing. If instead the copresheaf realization is used with incidence-specific dense fiber maps $\rho_{y\to x} \in \mathbb{R}^{h\times h}$ (as in the CopresheafGIN variant of Hajij et al. (2025); their GCN/Sage variants restrict $\rho_{y\to x}$ to be diagonal, retaining the $O(s_\bullet h)$ cost), transport raises to $O(s_\bullet h^2)$, matching the per-edge $O(w^2)$ scaling of MPNN message MLPs at common width.

E.3 Baseline Implementations: RIGNO, GAOT, HOGNN, SNN-PDE, HSD

For RIGNO (Mousavi et al., 2025), GAOT (Wen et al., 2025), and HOGNN (Liao et al., 2025), we use the authors' official implementations. These are adopted from published optimizer settings; we tune only the epoch budget and per-dataset width on the validation split; test numbers come from the best-validation checkpoint consistent with our models' evaluation protocol.

GAOT (Wen et al., 2025) is evaluated on both the RIGNO tab. 1 and GAOT tab. 2a benchmarks as it is the strongest published unstructured-mesh operator outside the methods evaluated by RIGNO tab. 1, and it shares the same problem setting. We follow the upstream optimizer (AdamW with cosine-decay, peak 10^{-3}, weight decay 10^{-5}, gradient clipping 1.0) and its effective batch of 64 (data-parallel over four GH200s). We extend the epoch budget to 2,000 with patience 300 (the

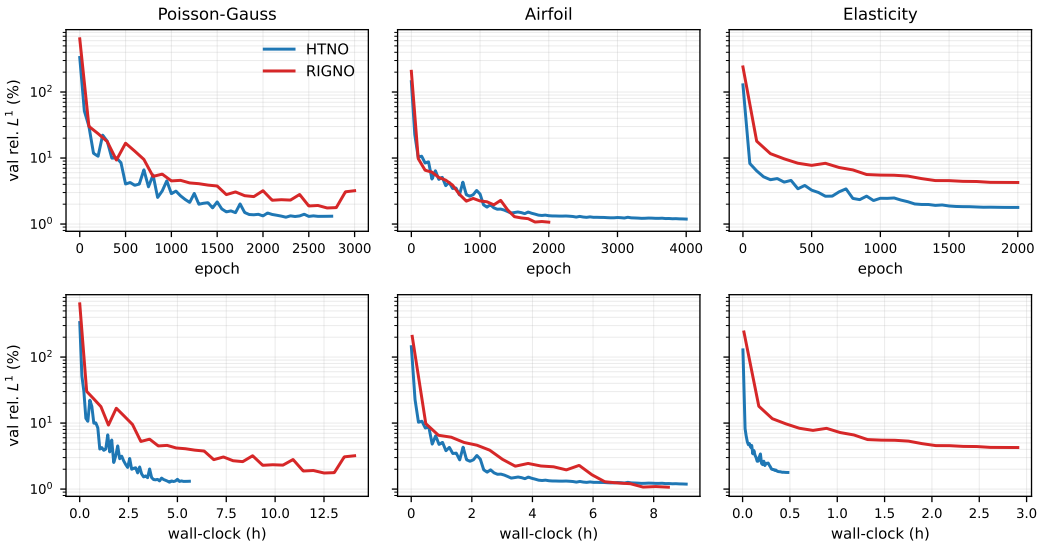


Figure 14: Validation relative L^1 vs. training epochs (top row) and cumulative wall-clock hours (bottom row) for HTNO and RIGNO on the three RIGNO steady-state benchmarks. Both models were trained under their respective official optimizer/batch settings; final test numbers correspond to the entries in tab. 1.

upstream configurations use 100 and 1,000 for PG and Elasticity, respectively); due to best-validation checkpoint selection, this remains a safety margin to assure convergence.

We note that the public GAOT release does not match the paper's Table 1 numbers on the GAOT suite itself (e.g. NACA0012 12.07% vs. paper 6.81%); this has been observed in issue #1 of the public repository that not all results are reproducible: for instance, they use 2048 train samples instead of 1024 (RIGNO / our protocol) on Poisson Gauss. GAOT entries are therefore a head-to-head comparison under identical experimental conditions, not a reproduction of the paper's reported results.

RIGNO (Mousavi et al., 2025) convergence observations. In contrast to the RIGNO multi-level physical/regional graph, HTNO arranges TNO blocks as a V-cycle over a sequence of coarsened cell complexes $K_0 \leftarrow \cdots \leftarrow K_L$ connected by degree-preserving restriction and prolongation maps R_ℓ^k, Π_ℓ^k (sec. 4): fine-level blocks capture short-range cross-degree physics, while coarse-level blocks, acting on the same cochain types after transfer, propagate long-range information. In addition to performance improvements across benchmarks (see sec. 5), we observe notably faster convergence. Fig. 14 depicts two qualitative observations from training: on these three benchmarks, HTNO reaches lower validation error in notably fewer epochs and in less wall-clock time than RIGNO, except Airfoil, where the two are comparable. Field-level prediction comparisons for these benchmarks are given in sec. G.1.

HOGNN (Liao et al., 2025) is a closely related baseline, in a spirit that shares motivations with TNO. Both lift signals beyond the vertices using DEC operators, providing an informative head-to-head comparison. We run it on every public 2D benchmark with a matching TNO or HTNO entry: PG, AF, Elasticity, NACA0012/2412, RAE2822, and PCS. We use the authors' optimizer configuration (AdamW with cosine-decay, peak 5×10^{-4}, weight decay 10^{-4}, gradient clipping 1.0), 300 epochs, patience 200, and the identical Delaunay graph construction as for TNO, HTNO, and RIGNO; per-dataset tuning is restricted to hidden width. EmmiWing is excluded because the public release is 2D only, and implementing the HOGNN DEC operator stack on 3D surfaces is outside our scope.

Results. Tab. 6 lists test relative L^1 error for HOGNN on every benchmark with a matching TNO or HTNO entry. HOGNN does not match the leading method in any of the cells evaluated. It lifts the problem onto higher-rank cells but communicates only locally, without coarsening or a global operator. It was designed for boundary-value electrostatics on small domains, where a local receptive field can suffice.

Table 6: HOGNN, SNN-PDE and HSD across all 2D benchmarks: test relative L^1 (%); lower is better, **bold**: best per column. "Best of ours" is the smaller of TNO and HTNO from the cited table. "—": not run, since SNN-PDE needs one fixed mesh (sec. E.3).

	tab. 1			tab. 2a			
Method	Poisson-Gauss	Airfoil	Elasticity	NACA0012	NACA2412	RAE2822	PCS
Best of ours	**1.03**	**1.11**	**1.70**	**3.77**	**3.92**	**3.92**	**0.52**
HOGNN	33.78	15.64	21.24	9.36	10.33	9.28	50.44
SNN-PDE (paper width)	8.75	—	—	—	—	—	28.55
SNN-PDE (param.-matched)	5.69	—	—	—	—	—	12.71
HSD (release width)	11.23	22.50	84.71	77.88	78.81	80.19	55.51
HSD (param.-matched)	20.29	19.44	84.94	75.17	77.65	78.73	51.38

SNN-PDE. SNN-PDE (Choi et al., 2024) is the closest simplicial architecture to TNO (sec. H). We follow the published equations and, where possible, the publicly released implementation. Its node embedding holds one learned vector per vertex, so it needs a fixed mesh, which only PG and PCS have. We train two versions of the models. One at the paper's width and one parameter-matched, $2{,}000$ epochs, patience 200 (tab. 6). One parameter-matched PCS run takes over five days, so we stop it at epoch $1{,}721$ and report its best checkpoint.

HSD. HSD (Zheng et al., 2026) is a concurrent neural operator built on a Hodge decomposition (sec. H). We port the release's Table-1 model (a truncated Hodge-spectral branch with $k{=}128$ modes per rank and an FNO branch on a grid lift) with its loss and optimizer, $2{,}000$ epochs, patience 200. HSD learns in the eigenbasis of a single complex. The datasets PG and PCS share a single mesh, but Airfoil, Elasticity, NACA0012/2412, and RAE2822 benchmarks require building an eigenbasis per sample, which is consistent for all models.

HSD's weights are thus bound to eigenmode indices, which works when the underlying discrete topology is fixed. With one eigenbasis per sample, however, the k-th mode of one mesh need not be the k-th mode of a different mesh (sec. G.5). On Airfoil, whose meshes differ little between samples, most modes keep their identity. On Elasticity only the lowest modes do, and HSD's validation error rises from the first epochs while its training error keeps falling. The NACA and RAE2822 point sets are resampled per sample, so no mode keeps its identity, and HSD cannot encode the changing basis.

F Experimental Details

We provide additional implementation details for our models, and baselines. TNO and HTNO are implemented in JAX (Bradbury et al., 2018). We use the authors' official implementations (see sec. E), which are often run in PyTorch (Paszke et al., 2019); we configure these with the same dataloaders our JAX models use to guarantee consistency. Implementations will be published upon acceptance. All models are run on a cluster of nodes consisting of 4 NVIDIA GH200s (96GB VRAM). The maximum compute budget on larger datasets (EmmiWing, PCS) is 4 GPUs simultaneously. Other steady-state benchmarks are run with 1 GPU per model. Throughout, reported errors are medians of the per-sample relative error over the test set, matching Mousavi et al. (2025); Wen et al. (2025).

F.1 Poisson-Gauss, Airfoil, Elasticity

Datasets. We use three datasets from the RIGNO benchmark (Mousavi et al., 2025):

1. **Poisson-Gauss** (PG, Poisson with Gaussian sources on a random point cloud, $9{,}216$ nodes, $1024/128/256$ train/val/test, fixed positions);
2. **Airfoil Flow** (AF, compressible Euler flow with varying airfoil geometry at fixed far-field $M_\infty{=}0.8$, $\alpha{=}0°$ (Li et al., 2023b, §4.4), $200{\times}50$ C-grid of quadrilateral elements ($\sim$11K nodes) redistributed by Mousavi et al. (2025) with $2048/128/256$ train/val/test, variable positions);
3. **Elasticity** (hyper-elastic deformation of varying domains, $\sim$1K nodes, $1024/256/512$, variable). We note that AF is, on closer inspection, an effectively single-condition transonic benchmark; we audit its regime spread against the GAOT airfoils in sec. G.2.

Baselines. This section contains additional details regarding baselines and training protocols. We evaluate RIGNO (Mousavi et al., 2025), GAOT (Wen et al., 2025) and HOGNN (Liao et al., 2025) on the same splits using the authors' official implementations (sec. E.3).

Training protocol. For TNO, we report the best result from per-dataset tuning around a harmonic baseline (12 layers, hidden 128, $r{=}8$ harmonic modes), varying depth, width, learning rate, neighborhood size, and Delaunay graph construction. HTNO uses dynamic k-means clustering ($K{=}32$–128 regions), 18 layers ($6{+}6{+}6$ fine/coarse/fine), hidden 128–256 depending on dataset scale. Training uses AdamW with cosine-decay (peak $5{\times}10^{-4}$), gradient clipping at 1.0, and early stopping with patience 200–500. Models train for 2,000–4,000 epochs depending on convergence. The same protocol applies to the other benchmarks below.

F.2 Compressible Airfoils and Poisson-with-sines

Datasets. From the GAOT release (Wen et al., 2025) we use NACA0012/2412 and RAE2822 ($\sim$8K-node airfoils, spanning subsonic, transonic, and supersonic regimes) and Poisson-C-Sines (PCS, fixed 16,431-node mesh).

Baselines. HTNO and TNO are trained as in the sec. F.1 protocol; RIGNO and GAOT are evaluated on identical splits using the authors' official implementations (sec. E.3).

Splits. The GAOT public release contains per-dataset NetCDF files but does not publish the train/val/test indices, making it challenging to reproduce GAOT's Table 1. We thus reconstruct splits based on the sample counts declared in the paper. We follow the upstream data pipeline's default contiguous-block convention,

$$\text{train} = [0,\, N_{\text{tr}}), \quad \text{val} = [N_{\text{tr}},\, N_{\text{tr}} + N_{\text{val}}), \quad \text{test} = [N_{\text{tot}} - N_{\text{te}},\, N_{\text{tot}}),$$

i.e. the test split is the last N_{te} samples of the source file, with a gap between val and test on the larger airfoil files. Counts per dataset are listed in tab. 7.

Table 7: Reconstructed GAOT-suite splits. N_{tot} is the sample count of the public release; the $5000/128/256$ airfoil train/val/test split matches Wen et al. (2025, §D.2.1, p.36), drawn from a pool of 5384 samples per airfoil; PCS uses the maximal contiguous train slice the file admits.

Dataset	N_{tot}	N_{tr}	N_{val}	N_{te}
NACA0012	43,490	5000	128	256
NACA2412	47,925	5000	128	256
RAE2822	48,375	5000	128	256
Poisson-C-Sines	5,000	4616	128	256

Bluff-Body. The Bluff-Body benchmark from the same release is omitted: the public collection contains only 7 of the 10 paper pretraining shapes (Square, Cone, and Rectangle-L are absent, and the paper's 5-Ellipse pretraining set is replaced in the release by Ellipse-1/2 plus other shapes designated in the paper itself as fine-tuning geometries); we thus omit this evaluation due to incompleteness.

F.3 Large-Scale Surface PDEs: EmmiWing

Dataset and resolution protocol. EmmiWing (Paischer et al., 2025) provides 29,727 steady-state compressible-RANS wing surfaces with raw meshes of 244K–426K nodes (variable across geometries). We use the official convex-hull-peeling split (ID-random, OOD-extrapolation, OOD-interpolation; $25{,}674/999/2{,}992$ train/val/test, with 62 erroneous cases excluded). Following AB-UPT (Alkin et al., 2025), we adopt a 65,536/16,384-point sampling protocol: each raw mesh is once FPS-resampled to 65K shared query points, and a fresh 16,384-node subset is drawn each training step. This follows established conventions in the literature: in Paischer et al. (2025), (PointNet, Transolver, ViT) also train at 16,384 surface points; AB-UPT (Alkin et al., 2025) uses the $65{,}536{\to}16{,}384$ supernode/anchor pattern that we follow most closely.

Deviations from the published protocol. We evaluate on the 65K FPS subset, where Paischer et al. (2025); Alkin et al. (2025) evaluate on the full 244K–426K raw mesh via chunked inference.

Aggregate-L^1 numbers are therefore not directly comparable to the published Table 2 of Paischer et al. (2025); the comparisons in tab. 2b are between the five models trained and evaluated under the same regime. Closing this gap would require equipping the topological backbone with a neural-field-style decoder that decouples prediction resolution from the fixed shared mesh, which we leave as future work.

Training protocol. In addition to HTNO and RIGNO-18, we report Transolver (Wu et al., 2024) (the slice-attention transformer designed for unstructured meshes), a pre-norm self-attention Transformer (Vaswani et al., 2017) with sinusoidal coordinate-based positional encoding, and a PointNet-style baseline (Qi et al., 2017) (per-point MLPs with global max-pool, implemented as pre-norm residual blocks). All models are trained under matched compute budgets with AdamW, peak learning rate 5×10^{-4} on a cosine-decay schedule, gradient clipping at 1.0, and early stopping on validation L^1.

Per-channel results. Pressure is recovered to high fidelity by every model ($\leq 0.30\%$); the wall-shear-stress channels dominate the spread. HTNO leads on every channel; Transolver is the consistent runner-up, 0.2–0.3 pp behind on each τ component (see tab. 2b).

F.4 Synthetic-Topology Datasets

The three synthetic datasets share the mesh family of sec. 5 but serve distinct studies (tab. 8): Oriented-Darcy (sec. 5.3) carries its tensor orientation on faces, while Darcy and Curl–curl (sec. 5.4) isolate the harmonic basis and copresheaf transport on a scalar and a multi-rank problem. All three use 100 meshes $\times$ 50 samples with $4{,}000/500/500$ train/val/test samples, whole meshes held out ($80/10/10$), so test geometries are unseen in training. Every re-meshed test set of sec. G.4 likewise holds 500 samples on 25 further held-out meshes. We provide further details in the next sections.

Table 8: The three synthetic-topology datasets: Darcy (sec. F.5), Curl–curl (sec. F.6), and Oriented-Darcy (sec. F.8).

	Darcy	Curl–curl	Oriented-Darcy
PDE	$-\nabla\cdot(\kappa\nabla u)=f$	$(\Delta_1+\varepsilon)\,A=b$	$-\nabla\cdot(\kappa(x)\nabla u)=f$
Holes per mesh	1–2	1–3	1–2
Nodes per mesh	1,016 (median)	1,018 (median)	513 (median)
Inputs	7 vertex channels	vertex, edge, and face sources	5 vertex, 2 face channels
Targets	$u\in C^0$	$\delta^1 A\in C^0$, $A\in C^1$, $d^1 A\in C^2$	$u\in C^0$
Varies per sample	tensor, source, hole values	sources, hole circulations	face orientations, source, hole values
Harmonic sector of label	none (scalar label)	$\approx$30% of the C^1 energy	none (scalar label)
Meshes $\times$ samples	100×50	100×50	100×50
Train/val/test samples	$4{,}000/500/500$	$4{,}000/500/500$	$4{,}000/500/500$
Train/val/test meshes	$80/10/10$	$80/10/10$	$80/10/10$
Solver	FEniCSx, P_1 Lagrange	DEC	FEniCSx, P_1, DG0 κ
Ground-truth residual	1.5×10^{-5} (energy-dual)	1.9×10^{-5} ($\mathbf{K}A{=}b$)	9.8×10^{-8}

F.5 Darcy

PDE. We solve $-\nabla\cdot(\kappa\nabla u)=f$ with anisotropic κ, $u|_{\partial\Omega_{\text{outer}}}{=}0$, and $u|_{\partial\Omega_{\text{hole},k}}{=}g_k$.

Meshes. The dataset family used in sec. 5.4 is a 2D planar mesh distribution with a square outer boundary and one or two interior holes ($\sim$1,000 nodes per mesh) under standard boundary jitter and adaptive mesh-density variation.

Inputs and targets. Models receive $c{=}7$ vertex channels $[f, \mathbb{1}_{\partial}, g, \kappa_{\parallel}, \kappa_{\perp}, 0, \varphi]$: $\kappa_{\parallel}$, $\kappa_{\perp}$, and a zero Robin coefficient are carried to every vertex, and the gradient of the linear field $\varphi(x){=}x_1\cos\theta{+}x_2\sin\theta$ is the principal axis of κ. There are no edge or face inputs; the target is $u\in C^0$.

Sampling. Per sample, the Darcy tensor ($\kappa_\parallel/\kappa_\perp \sim \log\mathcal{U}[2,8]$ at $\det\kappa{=}4$, orientation $\theta \sim \mathcal{U}[0,\pi)$, constant in space), a signed 1–3-bump Gaussian source, and the per-hole Dirichlet BC values ($\sim \mathcal{U}[0.5,1.5]$) vary.

Solver. The variational form is assembled in FEniCSx (Baratta et al., 2023) with P_1 Lagrange elements; every mesh belongs to exactly one split.

Significance. Darcy is a scalar elliptic benchmark: its label $u \in C^0$ has no harmonic sector, and the tensor, source, and boundary values change per sample, so a model must condition on them.

F.6 CURL–CURL

PDE. We solve a gauge-augmented curl–curl system $(\Delta_1 + \varepsilon)\, A = b$ ($\Delta_1{=}d^0\delta^1{+}\delta^2 d^1$, $\varepsilon{=}10^{-4}$).

Meshes. The mesh family is that of sec. F.5, with one to three interior holes.

Inputs and targets. Vertex, edge, and face sources drive the three outputs $\delta^1 A \in C^0$, $A \in C^1$, and $d^1 A \in C^2$, with the harmonic part of the edge source carried into A explicitly. The edge source is a smooth random vector field integrated along each edge plus $\sum_k c_k g_k$, where g_k is the closed cochain that winds once around hole k (period 2π) and has zero period around the other holes.

Sampling. Per sample, the Gaussian sources and the per-hole circulations $c_k \sim \mathcal{U}[-0.5, 0.5]$ vary.

Solver. The system is assembled with DEC operators and solved to a relative residual below 10^{-12}; every mesh belongs to exactly one split.

Significance. Curl–curl is a multi-rank benchmark: its targets on C^0, C^1, C^2 are coupled, the circulations give the C^1 label a harmonic sector ($\approx$30% of its energy), and the vertex and face targets depend on the edge source (sec. G.6).

F.7 ABLATIONS

Training. All ablation variants share one backbone and one training recipe: hidden width 192, MLP messages, cells up to rank 1 on Darcy and rank 2 on curl–curl, dropout 0.20, AdamW with a cosine schedule (peak learning rate $5{\times}10^{-4}$, weight decay 10^{-4}), gradient clipping at 1.0, batch size 16, 300 epochs, rigid 2D augmentation, and seeds 42, 1, 2, each run on a single GH200.

Variants. The four TNO variants differ only in whether copresheaf transport (Cayley restriction map) and the harmonic-basis input are switched on. We use the orthogonal Cayley map by default. Diagonal, full, outer-product, and shared-local restriction maps performed similarly or worse on Darcy, and the resolution and cross-rank studies (secs. G.4 and G.6) use the diagonal map. Copresheaf transport adds parameters, so the two variants without it are widened to match the counterparts with it in the number of parameters.

Baselines. The MPNN baselines use the same backbone with copresheaf transport and the harmonic basis switched off, at width 192 (width-matched) and at width 364 (parameter-matched to the full TNO on curl–curl; on Darcy this gives the MPNN more parameters than TNO). The extended Darcy studies of sec. G use a slightly narrower parameter-matched MPNN (6.69M parameters, against 6.79M for the full TNO).

F.8 ORIENTED-DARCY

Dataset and equation. Oriented-Darcy uses meshes with approximately 500 vertices and one or two holes, and stores the coefficient orientation on faces: a non-topological operator such as an MPNN can only ingest this signal through incident-cell averages, whereas a TNO reads it natively as a 2-cochain. On each sample we solve

$$-\nabla\cdot\big(\kappa(x)\nabla u\big) = f, \qquad u|_{\partial\Omega_{\text{outer}}} = 0, \quad u|_{\partial\Omega_{\text{hole},k}} = g_k \sim \mathcal{U}[0.5, 1.5].$$

The source is a sum of 1–3 signed Gaussian bumps (amplitudes in $[1, 2.5]$, width 0.25). The piecewise-constant tensor on face t is

$$\kappa_t = R(\phi_t)\,\mathrm{diag}(4,1)R(\phi_t)^\top, \qquad \phi_t \sim \mathcal{U}[0,\pi),$$

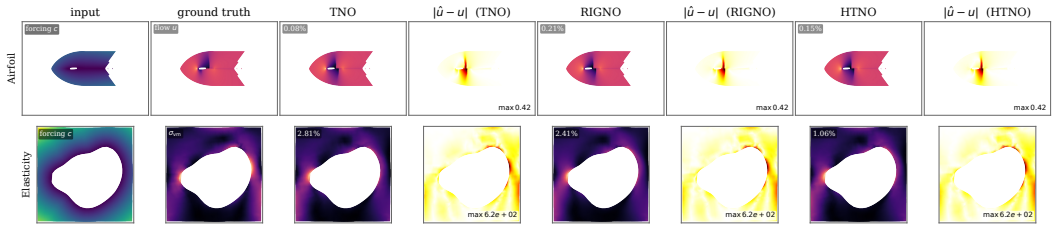


Figure 15: Qualitative comparison on the RIGNO-suite steady-state benchmarks (Airfoil, Elasticity). Each row shows one median-test-L^1 sample for the dataset; columns: input forcing, ground-truth field, and per-architecture prediction $\hat{u}$ followed by absolute error $|\hat{u} - u|$ for TNO, RIGNO, and HTNO. The three error panels in a row share a common color scale, with the shared $\max$ stamped on each panel so residual magnitudes are directly comparable across architectures.

with independent orientations across faces and samples. We solve the P_1 finite-element system in FEniCSx (Baratta et al., 2023), assigning the DG0 tensor through the cell index map.

Inputs. The five vertex channels are $[f, \nVdash_\partial, g, \kappa_\parallel, \kappa_\perp]$. Each face stores the two-channel field $c_f = [\cos(2\phi_t), \sin(2\phi_t)]$; edge inputs c_e are incident-face means of that field. Both face channels are needed: $\kappa_{xx} = 2.5 + 1.5c_{f,1}$, $\kappa_{yy} = 2.5 - 1.5c_{f,1}$, and $\kappa_{xy} = 1.5c_{f,2}$. The projected variant appends incident-edge and incident-face means of these channels to the vertex inputs. For the face-input variant, TNO reads c_f through its rank-2 encoder. The MPNN reads the same stored field after aggregation into bidirectional edge features. The vertex and projected variants do not read native edge or face inputs.

Training and validation. All variants train for 300 epochs with batch size 16, dropout 0.20, AdamW (peak learning rate 5×10^{-4}, weight decay 10^{-4}), gradient clipping at 1.0, and seeds 42, 1, and 2. TNO uses width 192, a harmonic basis, and Cayley copresheaf transport; it has about 10.42M parameters in each input variant. The width-matched MPNN uses width 192 (about 2.67M parameters), the parameter-matched MPNN width 380 (about 10.44M). Geometric augmentation is disabled because the loader does not rotate the orientation channels with the mesh. Every mesh belongs to exactly one split. Across the three seeds, the sample standard deviation of every entry of tab. 3a is at most 0.24 pp.

G Extended Experimental Results

This section presents qualitative comparisons on the steady-state benchmarks and explains the differences between the two airfoil datasets. We then examine whether the ablation models satisfy the discrete PDEs, and generalize zero-shot to other mesh resolutions. Additional experiments compare spectral inputs to the MPNN, study cross-rank coupling, and report computational cost.

Paired control solution. For the input-shift experiments below, each shifted sample is paired with an in-distribution control sample sharing its mesh and unchanged random draws. The *paired control solution* reference predicts the control sample's ground-truth field for the shifted sample, $\hat{u}_i^{\text{shift}} = u_i^{\text{ctrl}}$, and scores it against u_i^{shift} using the same relative L^1 metric and sample aggregation as the models. This reference requires no learned model: it retains the exact original solution while ignoring the imposed input change. Its error is zero on the control set and on unchanged target fields; it is unavailable when the shift changes the meshes. Beating it means predicting the shifted solution more accurately than retaining the exact unshifted solution. Exceeding its error does not by itself establish that a model ignores the change.

G.1 Qualitative comparisons on the steady-state suites

RIGNO-suite (Airfoil, Elasticity). Fig. 15 shows field-level predictions on a median-test-L^1 sample of each dataset for TNO, RIGNO, and HTNO. On AF, all three architectures recover the ground-truth density field to within $\sim$0.1–0.2% relative L^1, with residuals concentrated near the airfoil tip and along the wake; on Elasticity, the residuals localize on the loaded notch boundary. The visual closeness of the three error maps on AF mirrors the table-level finding (tab. 1) that TNO and

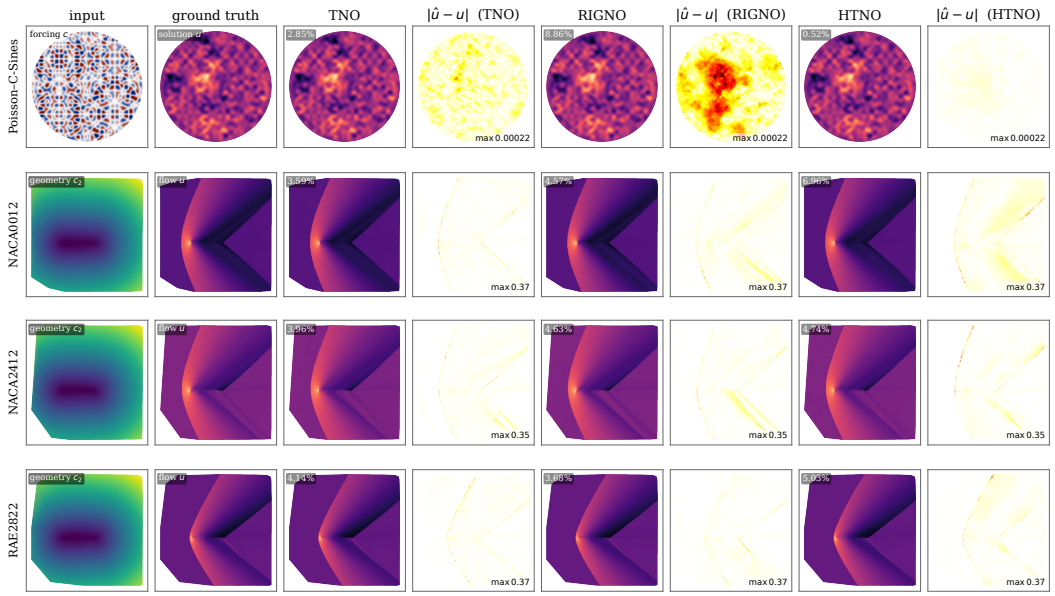


Figure 16: Qualitative comparison on the GAOT-suite benchmarks (Poisson-C-Sines, NACA0012, NACA2412, RAE2822). Each row shows one median-test-L^1 sample for the dataset; columns: model input, ground-truth field, and per-architecture prediction $\hat{u}$ followed by absolute error $|\hat{u} - u|$ for TNO, RIGNO, and HTNO. Per-row error panels share a common color scale, allowing residual magnitudes to be directly comparable across architectures. Triangulation on the airfoil rows uses a vertex-local long-edge filter to drop spurious convex-hull bridges across the wing void.

RIGNO-18 are largely indistinguishable on this benchmark. This can be attributed to the dataset's simple regime (see sec. G.2).

GAOT-suite (PCS, NACA0012/2412, RAE2822). Fig. 16 shows the analogous comparison on the four single-file steady-state datasets of Wen et al. (2025). PCS (top row) is dominated by a high-frequency multiscale sinusoidal forcing pattern, which HTNO can represent well. The three airfoil rows expose the regime-sweep difficulty of this benchmark: the operators concentrate residual along the boundary layer and—for the supersonic RAE2822 sample—along the captured shock, where RIGNO has the lowest error on this sample. Per-row error panels share a common color scale, so RIGNO and HTNO residuals are directly comparable.

G.2 Differences in airfoils across datasets

TNO marginally underperforms RIGNO-18 on AF and outperforms it by 19–32% relative test-L^1 on the GAOT airfoils. This is due to how the benchmarks differ in what their generators sweep across samples (tab. 9): Li et al. (2023b) fixes the AF far-field at M_∞=0.8, α=0° and varies only the airfoil geometry, exposing a single distance-to-boundary field as conditioning (RIGNO, Mousavi et al., 2025, adopts this dataset directly); Wen et al. (2025) draws each NACA0012/2412/RAE2822 sample at a different (M, α) with $M \in [0.5, 1.4]$ and $\alpha \in [0.5°, 5.0°]$ on per-sample meshes, exposing M and α as conditioning scalars.

Table 9: What varies across samples for the two airfoil benchmark families (published quantities).

Dataset family	M_∞	α	conditioning channels	varies per sample
AF (Li et al., 2023b)	0.8 (fixed)	0° (fixed)	distance field	airfoil geometry only
GAOT airfoils (Wen et al., 2025)	$[0.5, 1.4]$	$[0.5°, 5.0°]$	M, α (scalars)	$(M, \alpha$, mesh)

Table 10: Physical consistency on Darcy. We report the median and 95th percentile of the energy-dual residual, boundary flux error, and net mass defect, all normalized by the forcing. Values are mean ±std over 3 seeds. The ground-truth median residual is 1.5×10^{-5} and mass defect is 4.4×10^{-6}. **Bold**: best; underline: second; *parameter-matched.

Variant	$\mathrm{rel}_{\mathrm{dual}}$	$\mathrm{rel}_{\mathrm{dual}}$ (p95)	Flux error	Mass defect
MPNN (w=192)	12.44±0.87	43.19±2.59	5.75±0.21	4.15±0.33
MPNN*	11.51±1.49	41.66±4.54	5.46±0.76	4.09±0.49
TNO	7.53±0.06	26.26±0.50	4.45±0.08	2.70±0.21
TNO + harm. basis	<u>3.36</u>±0.03	<u>12.62</u>±0.54	<u>2.80</u>±0.39	<u>1.73</u>±0.22
TNO + copresheaf	7.61±0.05	26.59±0.59	4.63±0.15	2.97±0.23
TNO + copresheaf + harm.	**3.05**±0.06	**11.28**±0.29	**2.01**±0.08	**1.38**±0.03

G.3 Physical consistency

Setup. We ask whether the accuracy gains in tab. 3b also improve agreement with the discrete PDEs. We evaluate the same trained models on their test sets, reporting medians over samples and means over three training seeds.

Darcy

Setup. We measure the finite-element residual of the predicted pressure in the energy-dual norm,

$$\mathrm{rel}_{\mathrm{dual}} = \frac{\|\mathrm{Res}(\hat{u})\|_{H^{-1}}}{\|Mf\|_{H^{-1}}}, \qquad \|v\|^2_{H^{-1}} = v_I^\top L_{II}^{-1} v_I,$$

where L is the sample's anisotropic stiffness matrix, I denotes non-Dirichlet vertices, and Mf is the discrete source. We also measure net mass defect and boundary flux error, both normalized by the total forcing.

Results. The full TNO has the lowest residual, flux error, and mass defect (tab. 10); its energy-dual residual is about four times smaller than the parameter-matched MPNN's. Most of the improvement comes from the harmonic basis. Residuals can exceed one because the interior forcing used for normalization is small relative to the boundary-driven solution. Thus, better field accuracy here is accompanied by better physical consistency, although the predictions do not exactly solve the discrete equations.

Curl–curl

Setup. We ask whether the separately predicted vertex, edge, and face fields obey their DEC relations. The C^0 and C^2 defects compare $\delta^1\hat{A}$ and $d^1\hat{A}$ with the corresponding predicted fields. We also measure the system residual,

$$\frac{\|\mathbf{K}\hat{A} - b\|_2}{\|b\|_2},$$

where $\mathbf{K}A = b$ is the assembled system of sec. F.6. We separate edge error into harmonic and non-harmonic components using the $\star_1$-orthogonal harmonic projection (tab. 11).

Results. The full TNO has the lowest residual and cross-rank defects, with the harmonic basis providing the largest gain. Yet its 2.37% edge error still produces a 499% system residual: applying $\mathbf{K}$ amplifies field errors. Copresheaf transport improves consistency further, especially in the non-harmonic component. These gains indicate better approximate physical consistency, although the predictions still do not solve the discrete equations closely.

G.4 Resolution generalization

This study is distinct from the weight sharing across meshes of (P3), which every experiment in the main paper tests on unseen meshes at the training resolution. Here we test a harder property, zero-shot generalization to coarser or finer resolutions than seen in training, similar to a super-resolution

Table 11: Physical consistency on curl–curl. Errors in the harmonic and non-harmonic edge components, harmonic coefficients, system residual, and vertex and face consistency defects are in %. Values are mean ±std over 3 seeds. **Bold**: best; underline: second; *parameter-matched.

Variant	Harmonic	Non-harm.	Coefficients	Residual	C^0 defect	C^2 defect
MPNN*	13.81±0.58	24.72±0.33	14.24±0.38	2781±37	125.9±1.1	448±2
TNO	11.06±0.67	20.54±0.61	12.16±0.65	3111±397	101.3±10.2	549±73
TNO + harm. basis	2.38±0.26	2.99±0.01	2.55±0.25	617±33	26.2±0.6	74±5
TNO + copresheaf	9.80±0.07	17.72±0.21	10.81±0.03	2368±95	81.9±1.9	426±12
TNO + copresheaf + harm.	**2.22**±0.13	**2.36**±0.12	**2.31**±0.14	**499**±18	**20.3**±0.7	**64**±2

Table 12: Resolution generalization on Darcy: test relative L^1 error (%) at each target mesh size n, mean ±std over 3 seeds. The n=1,000 column uses new held-out meshes, so it differs from tab. 3b. Variants without copresheaf transport use the parameter-matched widths of that table; all variants with copresheaf transport use the diagonal restriction map. °Resolution included in two-resolution training; **bold**: best; underline: second; *parameter-matched.

Variant	n=500	n=1,000	n=2,000	n=4,000
Zero-shot (trained at n=1,000)				
MPNN*	12.58±0.15	9.94±1.31	43.97±7.83	217.64±71.26
TNO	15.74±0.67	8.93±0.27	19.58±0.12	38.21±0.98
TNO + harm. basis	13.98±0.06	2.47±0.05	17.60±0.39	28.80±2.82
TNO + copresheaf	11.75±0.65	8.37±0.19	18.64±0.69	61.75±9.76
TNO + copresheaf + harm.	11.36±0.67	2.13±0.06	13.21±1.33	26.08±3.81
TNO + copresheaf + harm. + integral	10.56±1.10	**1.48**±0.07	9.16±0.57	44.74±3.77
Two-resolution (trained at n=500 and n=1,000)				
MPNN*	6.26°±0.11	10.28°±0.11	35.00±0.34	83.01±5.03
TNO	5.77°±0.35	9.23°±0.19	20.33±0.12	36.37±0.42
TNO + harm. basis	2.93°±0.07	2.78°±0.11	13.55±0.21	33.75±2.16
TNO + copresheaf	4.93°±0.03	8.71°±0.20	19.12±0.29	37.27±1.97
TNO + copresheaf + harm.	2.42°±0.09	2.37°±0.09	13.60±1.07	31.21±4.46
TNO + copresheaf + harm. + integral	**1.71**°±0.10	1.60°±0.09	**8.92**±0.87	**21.34**±1.86

experiment. It is closer to discretization invariance, the stronger notion that requires convergence to a continuum operator under refinement (Kovachki et al., 2023), which we do not claim. We ask how well the present instantiations generalize across resolutions, and whether two-resolution training or mesh-aware operators help. Coarse discretizations can lose unresolved information, while operator implementations can introduce aliasing, so zero-shot generalization across resolutions is difficult even for neural operators designed for continuous functions (Bartolucci et al., 2023; Sakarvadia et al., 2026).

DARCY

Setup. We evaluate the same held-out layouts at n=500–4,000 nodes with fixed training normalization (sec. F.5). We compare training at n=1,000 with training at n=500 and 1,000 for the same number of optimizer steps. An integral variant adds quadrature-weighted aggregation over a fixed physical radius. All copresheaf variants in this study use diagonal restriction maps.

Results. Error grows on finer unseen meshes for every model (tab. 12). Two-resolution training improves performance mainly at the added coarse resolution; integral transport combined with it gives the best fine-mesh results. Even that variant rises from 1.60% at n=1,000 to 21.34% at n=4,000. The TNO variants stay below the training-mean baseline on the finest mesh, whereas the MPNN exceeds it. The large seed spread for some n=4,000 rows limits close comparisons. These modifications improve transfer without making this instantiation invariant to refinement.

CURL–CURL

Setup. We use matched held-out domains at n=500–4,000 nodes and score all three output ranks. Alongside the standard TNO and a parameter-matched MPNN, we test a mass-weighted spectrum with a linear filter that accounts for cell size, and an integral update over a fixed physical radius. The integral variants train at two resolutions; all TNO variants use diagonal restriction maps.

Results. Standard TNO loses much of its edge accuracy away from the training mesh (tab. 13). Mass weighting sharply improves edge transfer; with integral transport it gives the lowest edge error at every tested resolution. On the finest mesh, mass-weighted variants outperform the MPNN across all three fields, while integral transport especially helps the vertex output. Errors still rise under refinement, so improved cross-resolution accuracy does not establish discretization invariance.

Table 13: Resolution generalization on curl–curl: test relative L^1 error (%) for each output rank and target mesh size n. Errors are averaged over test samples, then reported as mean $\pm$std over 3 seeds. All TNO rows include the harmonic basis and diagonal copresheaf maps. The n=1,000 column uses new held-out meshes, so it differs from tab. 3b. °Resolution included in two-resolution training; **bold**: best; underline: second per rank and resolution; *parameter-matched.

Variant	n=500	n=1,000	n=2,000	n=4,000
C^0, zero-shot (trained at n=1,000)				
MPNN*	27.76±0.31	27.38±0.17	36.54±0.20	51.83±1.09
TNO	12.23±1.87	1.60±0.22	21.17±4.63	39.06±32.52
TNO + mass-weighted spectrum	10.38±2.12	1.84±0.42	12.24±3.60	45.78±3.29
C^0, two-resolution (trained at n=500 and n=1,000)				
MPNN*	19.83°±0.08	28.19°±0.12	39.47±0.58	54.51±1.75
TNO	1.61°±0.09	1.36°±0.09	24.60±6.03	78.96±54.37
TNO + integral	1.76°±0.08	1.53°±0.12	20.76±6.19	**22.63**±4.94
TNO + integral + mass-weighted spectrum	**1.46**°±0.13	**1.34**°±0.17	**7.15**±0.91	32.59±3.57
C^1, zero-shot (trained at n=1,000)				
MPNN*	34.58±1.04	24.81±0.26	38.81±1.63	65.24±8.35
TNO	35.60±6.02	2.47±0.14	36.89±1.57	78.03±2.32
TNO + mass-weighted spectrum	6.64±0.60	2.06±0.41	6.13±1.64	15.59±1.07
C^1, two-resolution (trained at n=500 and n=1,000)				
MPNN*	19.13°±0.11	24.85°±0.09	37.31±0.25	58.20±2.71
TNO	3.16°±0.04	2.88°±0.05	35.04±1.98	89.07±6.81
TNO + integral	3.12°±0.03	2.72°±0.04	31.99±4.83	81.96±11.13
TNO + integral + mass-weighted spectrum	**2.02**°±0.03	**1.81**°±0.08	**5.32**±0.58	**14.66**±0.68
C^2, zero-shot (trained at n=1,000)				
MPNN*	13.54±0.17	10.70±0.19	18.23±0.90	30.09±2.15
TNO	93.24±11.10	**1.85**±0.09	10.36±1.09	20.80±1.45
TNO + mass-weighted spectrum	5.22±1.02	2.47±0.05	7.40±0.27	16.36±1.40
C^2, two-resolution (trained at n=500 and n=1,000)				
MPNN*	8.32°±0.05	11.32°±0.07	18.65±0.16	29.45±0.99
TNO	**2.05**°±0.07	1.94°±0.04	5.37±0.45	15.52±0.80
TNO + integral	**2.05**°±0.07	1.95°±0.03	**5.21**±0.55	15.07±2.67
TNO + integral + mass-weighted spectrum	2.36°±0.08	2.41°±0.10	5.31±1.10	**15.00**±1.28

G.5 SPECTRAL FEATURES AS MPNN INPUTS

Setup. We ask whether giving the MPNN the same spectral information used by TNO can reproduce the gains from harmonic projection. We add 16 low-frequency vertex Laplacian eigenvectors (LapPE), then supply 16 edge eigenvectors either as vertex magnitudes or signed edge features. All other training settings are unchanged.

DARCY

Results. Spectral inputs give the MPNN little benefit on Darcy (tab. 14). Vertex modes improve it slightly, while edge modes worsen it. Neither approaches the gain from adding harmonic projection to TNO with copresheaf transport. Supplying spectral information alone therefore does not reproduce the effect of the TNO's use of that information.

Table 14: Spectral features as MPNN inputs on Darcy: test relative L^1 error (%), mean $\pm$std over 3 seeds. **Bold**: best; underline: second; *parameter-matched.

Variant	L^1
MPNN*	8.80±1.30
MPNN* + LapPE	8.94±0.70
MPNN* + LapPE + vertex modes	8.45±0.11
MPNN* + LapPE + edge modes	10.31±0.23
TNO + copresheaf	7.47±0.07
TNO + copresheaf + harm.	**1.89**±0.02

CURL–CURL

Results. Vertex and edge modes modestly reduce the MPNN's edge and harmonic errors (tab. 15). Its harmonic error remains well above that of TNO with harmonic projection. The same trend across both PDEs suggests that how the basis enters the operator matters more than merely providing it as an input feature. An input encoding ties encoder channel k to mode k on every mesh, although the k-th mode of one mesh need not be the k-th mode of the next. TNO instead uses the eigenbasis to project rather than as a feature, so no parameter is bound to a mode index.

G.6 CROSS-RANK COUPLING

Setup. We ask whether cross-rank coupling is needed when a target field depends on sources at another rank. We disable all paths between ranks, including in the encoder, while keeping the same-rank operations, harmonic projections, and within-rank copresheaf operations.

Results. Without cross-rank coupling, the curl–curl vertex error approaches the 100% training-mean error, and edge and face errors also rise sharply (tab. 16). The vertex head cannot access the edge source, while the other heads lose access to the vertex source. Cross-rank transport is essential for this multi-rank PDE.

G.7 COMPUTATIONAL COST

Setup. We measure the one-time construction of the eigenbases for each mesh and the recurring cost of a training step. Preprocessing runs on one CPU core per mesh. Training is profiled on one GH200 on Darcy with $\approx$1K-vertex meshes, $4{,}000$ training samples and batch size 16. Widths are matched to the MPNN's 5.36M parameters.

Results. TNO is more expensive than the MPNN. The eigensolve dominates preprocessing and grows super-linearly with mesh size (tab. 17, left). This cost is paid once per mesh and is parallelized across samples, but it becomes more substantial when each sample has its own mesh. Per step, TNO without harmonic projection or sheaf transport costs about as much as the MPNN (tab. 17, right). Harmonic projection mainly adds compute, and sheaf transport adds both compute and memory. The full TNO is $2.7\times$ slower per step and uses $1.9\times$ the peak memory. HTNO avoids the eigendecomposition by obtaining global coupling from its hierarchy.

H ADDITIONAL RELATED WORK

Neural operators. Operator learning (OL) approximates maps between infinite-dimensional function spaces, with the aim of producing models that generalize across initial conditions, coefficients, forcings, and discretizations (Kovachki et al., 2023; Boullé & Townsend, 2024; Kovachki et al.,

Table 15: Spectral features as MPNN inputs on curl–curl. We report vertex and edge prediction errors, harmonic and non-harmonic edge errors, system residual, and face consistency defect, all in %. Metrics follow tab. 11: median over test samples, mean ±std over 3 seeds. **Bold**: best; underline: second; *parameter-matched.

Variant	C^0	C^1	Harmonic	Non-harm.	Residual	C^2 defect
MPNN*	22.56±0.20	22.03±0.08	13.81±0.58	24.72±0.33	2781±37	448±2
MPNN* + LapPE + vertex modes	23.59±0.19	21.77±0.22	12.81±0.48	24.14±0.40	2493±81	396±11
MPNN* + LapPE + edge modes	22.94±0.19	21.72±0.35	13.14±0.46	24.01±0.39	2432±45	393±11
TNO	19.16±0.38	18.47±0.39	11.06±0.67	20.54±0.61	3111±397	549±73
TNO + harm. basis	**2.22**±0.36	**2.87**±0.08	**2.38**±0.26	**2.99**±0.01	**617**±33	**74**±5

Table 16: Cross-rank coupling on curl–curl. We report prediction errors at each rank and errors in the harmonic and non-harmonic edge components, all in %, as mean ±std over 3 seeds. The variant without cross-rank coupling uses diagonal copresheaf maps; the full TNO uses Cayley maps. **Bold**: best; underline: second; *parameter-matched.

Variant	C^0	C^1	C^2	Harmonic	Non-harm.
MPNN*	22.57±0.20	22.03±0.08	9.80±0.04	13.81±0.58	24.72±0.33
TNO	19.18±0.37	18.47±0.38	9.98±0.16	11.06±0.67	20.54±0.61
TNO + copresheaf + harm.	**1.60**±0.33	**2.37**±0.11	**1.87**±0.07	**2.22**±0.13	**2.36**±0.12
TNO + copresheaf + harm., no cross-rank	99.18±1.37	32.42±2.10	45.40±0.70	13.13±1.02	36.24±1.97

2024; Azizzadenesheli et al., 2024). *DeepONet* (Lu et al., 2021) and its physics-informed extensions (Goswami et al., 2023; Jiao et al., 2025) learn a branch–trunk factorization of the operator with a universal approximation guarantee in function spaces. The *neural operator* family of Li et al. (2020) instead realizes the operator through learned integral kernels: the Graph Neural Operator (GNO) approximates these kernels by message passing on sampled point clouds, while the Fourier Neural Operator (FNO) (Li et al., 2021) replaces the kernel by a parameterized spectral multiplier on a regular grid, with subsequent factorized (Tran et al., 2023), wavelet (Tripura & Chakraborty, 2023), and convolutional (Raonic et al., 2023) variants. Physics-informed neural networks complement these by deriving losses from the governing PDEs (Li et al., 2024; Viswanath et al., 2026; Zhong & Meidani, 2025), and a separate line studies OL via model reduction (Bhattacharya et al., 2021).

Neural operators on irregular geometries. PDEs on irregular domains have motivated a body of operator-learning extensions beyond regular Cartesian grids. Geo-FNO (Li et al., 2023b) learns a deformation from the physical mesh to a uniform latent grid on which the FNO is applied; DAFNO (Liu et al., 2023) embeds geometry through smoothed indicator masks; and Beyond-Regular-Grids (Lingsch et al., 2024) evaluates truncated spectral transforms directly on arbitrary point sets. GINO (Li et al., 2023c) couples a graph encoder/decoder with an FNO on a uniform latent grid, scaling operator learning to large 3D meshes. Mesh-based simulators include MeshGraphNet (Pfaff et al., 2021), which propagates information on the simulation mesh together with a separate world-edge graph, and RIGNO (Mousavi et al., 2025), which uses message passing on multiscale graphs; GAOT (Wen et al., 2025) pairs a multiscale attentional graph encoder with a transformer processor on a latent grid. Transformer-based operators (Li et al., 2023a; Hao et al., 2023; Wu et al., 2024; Bryutkin et al., 2024; Calvello et al., 2025) treat unstructured nodes as tokens and attend either across all points or across learned cluster representatives, with universal physics transformers (Alkin et al., 2024; 2025) and PDE foundation models (Herde et al., 2024) extending this idea to large-scale 3D aerodynamics (Paischer et al., 2025).

A complementary line treats geometry intrinsically: DIMON (Yin et al., 2024) learns the operator across a diffeomorphic family of domains, geometric NOs on Riemannian manifolds (Quackenbush & Atzberger, 2024; Chen et al., 2024) and gauge-equivariant intrinsic operators (Cheng, 2026; Cheng & Peng, 2023) build manifold structure into the architecture, and Neural Green's Functions (Yoo et al., 2025) parameterize geometry-dependent fundamental solutions. What separates these methods from TNO is that physical quantities are still represented as functions on *points* (graph nodes, deformed grid samples, or tokens); cross-dimensional differential structure is recovered indirectly through

Table 17: Computational cost. *Left:* one-time rank-1 eigenbasis construction per mesh. *Right:* per-step training cost of parameter-matched variants on Darcy, with time and peak-memory ratios to the MPNN.

Mesh	V	Cells	Time (s)	RSS (MB)
Darcy	1,031	4,002	0.10	9
Elasticity	972	5,383	0.11	13
Poisson-Gauss	9,216	50,426	2.55	110
PCS	16,431	86,164	5.23	188

Variant	w	ms/step	Peak (GiB)	Ratio
MPNN	272	37.6	7.41	—
TNO	220	42.8	7.28	1.14 / 0.98
TNO + harm.	210	68.8	8.54	1.83 / 1.15
TNO + copresheaf	172	78.4	12.88	2.09 / 1.74
TNO + copresheaf + harm.	168	103.2	14.29	2.74 / 1.93

learned kernels. Our DEC realization addresses this gap by keeping fields at their physical cochain degrees and routing them through boundary, coboundary, and Hodge operators. The underlying differential operators satisfy $d \circ d = 0$; conservation of the network's predictions requires additional constraints on how these channels are used.

Discrete exterior calculus and structure-preserving discretizations. The mathematical foundation behind TNOs is the discretization of differential forms. Discrete exterior calculus (Hirani, 2003; Desbrun et al., 2005; Crane et al., 2013; Crane, 2018; Ptáčková & Velho, 2021; Bell & Hirani, 2012) models physical quantities as cochains on oriented cells and replaces gradient, curl, and divergence by signed boundary operators that satisfy a discrete Stokes' theorem. The same viewpoint underlies Whitney forms (Bossavit, 1988; 1998), mixed and Nédélec finite elements (Nédélec, 1980; Hiptmair, 2002), mimetic finite differences (Hyman et al., 1997; Lipnikov et al., 2014), and the finite element exterior calculus of Arnold et al. (2006; 2010); Arnold (2018), all of which preserve the de Rham complex and the algebraic identity $B_k B_{k+1} = 0$ at the discrete level. The fixed DEC realization uses this structure: the cellular operators $d^k = B_{k+1}^\top$, δ^k, and the Hodge Laplacian Δ_k used in these layers are exactly the DEC operators on the underlying complex.

Topological deep learning (TDL) & architectures for PDE learning. TDL (Papamarkou et al., 2024; Hajij et al., 2023b) extends graph neural networks by allowing message passing across cells of multiple ranks (vertices, edges, faces, and higher). Simplicial complex networks generalize the Weisfeiler–Leman test to higher-order incidences (Bodnar et al., 2021b), CW networks operate on regular cell complexes (Bodnar et al., 2021a). Combinatorial complexes (Hajij et al., 2023a) unify these under a single neighborhood scheme. Sheaf neural networks (Hansen & Gebhart, 2020; Bodnar et al., 2022) and copresheaf topological networks (Hajij et al., 2025) attach learnable fiber maps to incidence relations, while Hodge Laplacian random walks (Schaub et al., 2020; Viganò et al., 2026) and topology-aware GNNs (Leventhal et al., 2023), learning the Hodge star (Smirnov & Solomon, 2021), provide foundations. Despite the appeal of the cellular viewpoint for physics, topological architectures remain almost absent from the OL literature, and the closest prior works are graph-based and message-passing in flavor: SNN-PDE (Choi et al., 2024) uses simplicial convolutions for **time-dependent** PDEs, SCCNNs develop Hodge-aware convolutions (Yang et al., 2025). Trask et al. (2022) learn Whitney-form-style metric data on a fixed graph to obtain a data-driven exterior calculus that enforces conservation. Finally, DEC-HOGNN (Liao et al., 2025) builds a higher-order GNN on DEC and FEEC primitives for boundary-value electromagnetic problems. While these methods exploit cell-level features, **they remain GNN-style architectures**.

Concurrently, cellular sheaf neural operators (Shikhman & Gilbertie, 2026) place PDE states on the cells of a fixed cubical grid and couple adjacent ranks through learned restriction maps; the method is neither formulated for nor evaluated on irregular geometries, and it exposes no Hodge decomposition. HSD (Zheng et al., 2026) splits the solution operator along the Hodge decomposition, representing topology-dominated components with discrete differential forms and the remainder in an orthogonal ambient space. Apart from these concurrent works, none define true function-space operators, expose the Hodge decomposition as an architectural component or address multi-degree mPDE systems

where unknowns at different ranks couple simultaneously through d and δ. TNOs address these by fixing information flow to the DEC operators of an arbitrary cell complex and by exposing the Hodge decomposition in the layer (eq. (9)), so that exact, coexact, and harmonic channels carry physically meaningful components, and achieve strong performance on irregular meshes that vary per sample.