HodgeFormer: Transformers for Learnable Operators on Triangular Meshes through Data-Driven Hodge Matrices

Akis Nousias[1] Stavros Nousias[2]

[1]K3Y Labs [2]Technical University of Munich

WACV 2026
TUCSON, AZ
3/6 - 3/10

Mesh learning tasks

Downstream Applications:

- Rendering & real-time graphics
- FEM simulation & physics
- Geometry processing
- ...

Learning Tasks on Meshes:

- **Segmentation**: Body parts, object components
- **Classification**: Shape categories
- **Reconstruction**: 3D from images
- **Pose estimation**: Human/object pose

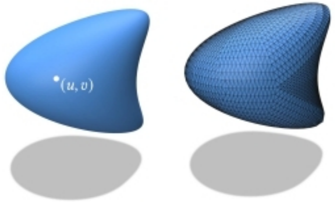


Figure: Theologou, P., Pratikakis, I. and Theoharis, T., 2015. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. Computer Vision and Image Understanding, 135, pp.49-82.

Mesh Learning

Learn the operator, not its spectrum

Traditional approaches for mesh learning:

- Computation of spectral components through eigenvalue decomposition of the Laplacian matrix
- Learning features on a fixed spectral basis

Traditional

Mesh geometry $\xrightarrow{\text{cotangent}}$ $L = \star_0^{-1} d_0^\top \star_1 d_0$ fixed Laplacian $\xrightarrow{O(n^3)\text{ CPU}}$ ~~EVD: $L = U\Lambda U^\top$ eigenvectors~~ $\xrightarrow{\text{fixed basis}}$ Network learns on spectrum

HodgeFormer

Mesh topology d_k $\xrightarrow{\text{fixed}}$ Attention learns $\star_{k,\theta}$ $\xrightarrow{\text{differentiable}}$ $L_\theta = \star_{0,\theta}^{-1} d_0^\top \star_{1,\theta} d_0$ learned Laplacian $\xrightarrow{\text{no EVD}}$ End-to-end task-adaptive

Traditional: geometry $\xrightarrow{\text{fixed}}$ operator $\xrightarrow{\text{EVD}}$ basis $\xrightarrow{\text{learn}}$ features

HodgeFormer: topology $\xrightarrow{\text{fixed}}$ operator $\xleftarrow{\text{learn}}$ data (skip EVD entirely)

Spectral Mesh Processing: Foundations

Core idea: use the mesh Laplacian's eigenstructure as a Fourier-like basis on the surface.

Mesh Laplacian

- **Combinatorial:** $L = D - A$
- **Cotan (geometric):** $w_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij})$
- Symmetric, positive semi-definite

Eigenproblem: $L\phi_\ell = \lambda_\ell \phi_\ell$

- $\{\phi_\ell\}$ form an orthonormal basis
- Low λ = smooth; high λ = detail

Why spectral?

- **Global structure:** eigenvectors capture large-scale geometry
- **Isometry invariance:** spectrum preserved under bending
- **Smooth basis:** natural for filtering and descriptors (HKS, WKS)

Represent vertex signals in the Laplacian eigenbasis to exploit global, smooth structure.

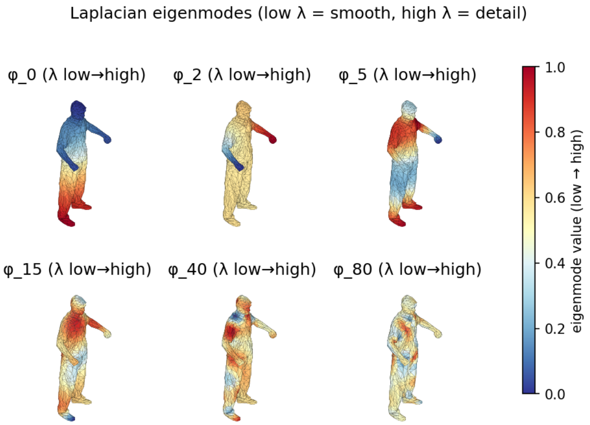
What Eigenvectors Capture

Geometric meaning of each mode:

- ϕ_1: global position (head $\leftrightarrow$ feet)
- ϕ_2: bilateral symmetry (left $\leftrightarrow$ right)
- ϕ_3: depth (front $\leftrightarrow$ back)
- Higher modes: fine details

Practical benefit:

A small set of eigenvectors gives multi-scale context—know *where* you are on the shape and at what level of detail.



What Eigenvectors Capture — and What They Cost

Computational cost:

- Full eigendecomposition: $O(n^3)$
- Partial (k vectors): $O(kn^2)$
- Must precompute per mesh

Common spectral features:

- Heat / Wave Kernel Signatures
- Spectral convolution (filter in eigenbasis)

Methods Comparison

Method	Year	EVD?	Type	Key Idea
Spatial Methods				
MeshCNN	2019	No	CNN	Edge conv + collapse pooling
SubdivNet	2021	No	CNN	Loop subdivision regularity
Mesh walker	2020	No	RNN	Loop subdivision regularity
Spectral Methods				
HodgeNet	2021	Yes	MLP	Learn diagonal Hodge stars
DiffusionNet	2022	Yes	CNN	Heat diffusion + HKS
Laplacian2Mesh	2022	Yes	Transformer	Transformer on eigenvectors
MeT	2023	Yes	Transformer	Laplacian PE

Spatial methods are fast but lack global context.

Spectral methods capture global geometry but require expensive eigendecomposition.

Can we get spectral method accuracy with spatial method efficiency?

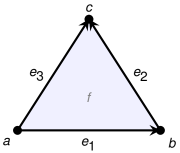
Discrete k-Forms and the Exterior Derivative

Exterior Derivative d_k

encodes **topology**

Maps k-forms to $(k+1)$-forms via signed incidence matrices:

- $d_0 \in \{-1, 0, 1\}^{n_e \times n_v}$ (discrete gradient)
- $d_1 \in \{-1, 0, 1\}^{n_f \times n_e}$ (discrete curl)
- $d_1\, d_0 = 0$ (curl of gradient $= 0$)



$$d_0 = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$d_1 = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix}$$

Topological Invariance

Deform, stretch, or bend the mesh — d_k does not change.

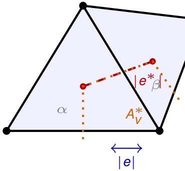
Hodge Star $\star_k$

encodes **geometry**

Maps primal k-forms to dual $(2-k)$-forms via ratio of measures:

$$(\star_k)_{\sigma\sigma} = \frac{|\star\,\sigma^k|}{|\sigma^k|}$$

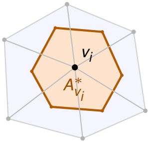
	Primal $\to$ Dual	**Diagonal entry**
$\star_0$	Vertex $\to$ Voronoi cell	A_v^*
$\star_1$	Edge $\to$ dual edge	$\frac{\cot\alpha + \cot\beta}{2}$
$\star_2$	Face $\to$ dual vertex	$1/A_f$



Red: dual mesh Orange: Voronoi cell Black: primal

Hodge Stars & the Cotangent Laplacian

$\star_0$: Vertex $\to$ Voronoi area

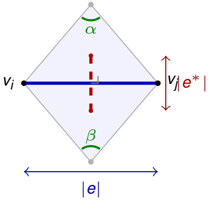


$$(\star_0)_{v_i,v_i} = A^*_{v_i}$$

Voronoi area of vertex v_i.

Normalizes the Laplacian via $\star_0^{-1}$.

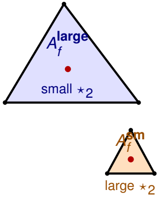
$\star_1$: Edge $\to$ dual edge ratio



$$(\star_1)_{e,e} = \frac{|e^*|}{|e|} = \frac{\cot\alpha + \cot\beta}{2}$$

Cotangent weights make the Laplacian shape-aware.

$\star_2$: Face $\to$ inverse area



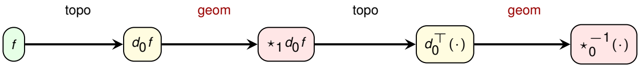
$$(\star_2)_{f_k,f_k} = \frac{1}{A_{f_k}}$$

Total on face $\to$ density at the dual vertex.

Cotangent Laplacian:

$$L_0 = \star_0^{-1}\, d_0^\top\, \star_1\, d_0$$

$$(L_0 f)_i = \frac{1}{A_i^*}\sum_{j\sim i}\frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2}(f_j - f_i)$$



From Diagonal Hodge Stars to Learned Attention

Classical $\star_k$: diagonal, one weight per element, no coupling.

$$\star_k = \mathrm{diag}(w_1, \ldots, w_{n_k}) \qquad \text{fails on irregular meshes}$$

Galerkin $\star_k$: non-diagonal mass matrix, captures neighbor coupling.

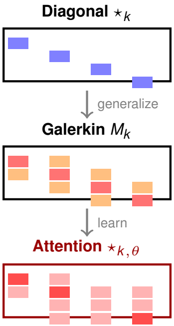
$$M_{ij} = \int_\Omega \phi_i \cdot \phi_j \, dx \qquad \text{sparse, SPD, but fixed}$$

Attention: same structure as Galerkin, but learned from data.

$$A_{ij} = \sigma\left(Q_i K_j^\top / \sqrt{d}\right) \qquad \text{sparse, non-diagonal, learned}$$

$M \approx \star_k^{\mathrm{Gal}}$ and $A \approx M$, therefore **attention can learn Hodge stars**:

$$\boxed{\star_{k,\theta} = \sigma\left(Q_k K_k^\top / \sqrt{d_h}\right)}$$



HodgeFormer: Overview

Methodology:

1. **Fix** incidence matrices d_0, d_1 from mesh topology
2. **Learn** non-diagonal Hodge stars $\star_0, \star_1, \star_2$ via sparse attention
3. **Assemble** Hodge Laplacians: $L_v = \star_0^{-1} d_0^\top \star_1 d_0$ (and L_e, L_f)
4. **Apply** operators end-to-end

Benefits:

1. **Non-diagonal** Hodge stars capture neighbor coupling
2. **Implicit** Laplacians without costly eigenvalue decomposition
3. Fully **end-to-end** differentiable on GPU

Transformer Self-Attention

Standard Transformer Layer

$T_l(x) = f_l(A_l(x) + x)$

Self-Attention Mechanism:

$$Q = xW_Q, \quad K = xW_K, \quad V = xW_V$$

$$\text{Attention}(Q, K, V) = \underbrace{\text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right)}_{\text{Attention Matrix}} V$$

Learning Hodge Stars via Sparse Attention

Parametrize Hodge Star via Attention:

$$\star_k(x_k) = \sigma\left(\frac{Q_k K_k^T}{\sqrt{d_h}}\right)$$

for $k \in \{v, e, f\}$ (vertices, edges, faces)

Properties of Learned $\star_k$:

- **Non-diagonal**: Captures basis overlaps
- **Non-PSD**
- **Sparse**: $\sim \sqrt{n}$ neighbors

Learned Hodge Laplacians for each k-form:

$$L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0$$

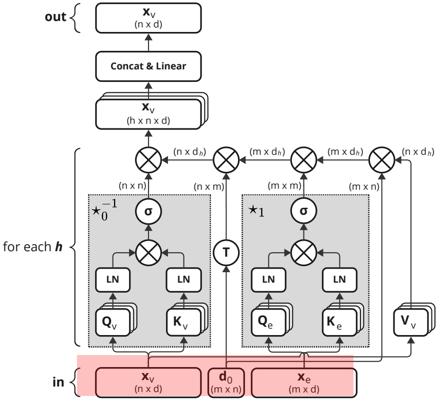
$$L_e := d_0 \cdot \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) + \star_1^{-1}(x_e) \cdot d_1^T \cdot \star_2(x_f) \cdot d_1$$

$$L_f := d_1 \cdot \star_1^{-1}(x_e) \cdot d_1^T \cdot \star_2(x_f)$$

Operator	Acts on	Learns	Updates
L_v	vertices	$\star_0^{-1}, \star_1$	x_v
L_e	edges	$\star_0^{-1}, \star_1^{-1}, \star_1, \star_2$	x_e
L_f	faces	$\star_1^{-1}, \star_2$	x_f

Multi-Head Hodge Attention

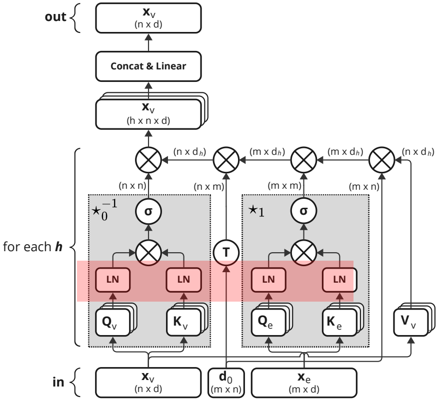
Vertex Features



$$L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0$$

Multi-Head Hodge Attention

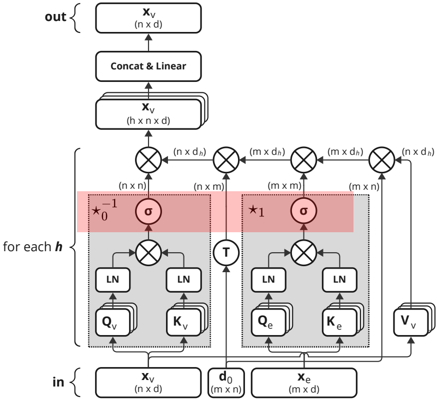
Vertex Features



$$\boxed{L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0}$$

Multi-Head Hodge Attention

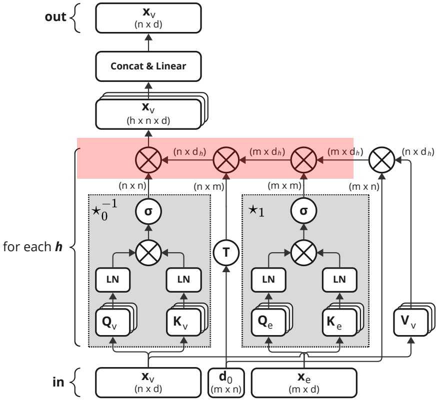
Vertex Features



$$\boxed{L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0}$$

Multi-Head Hodge Attention

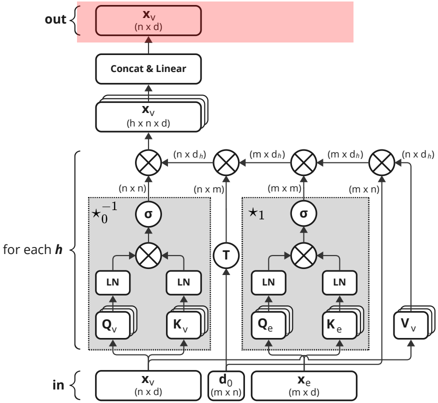
Vertex Features



$$\boxed{L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0}$$

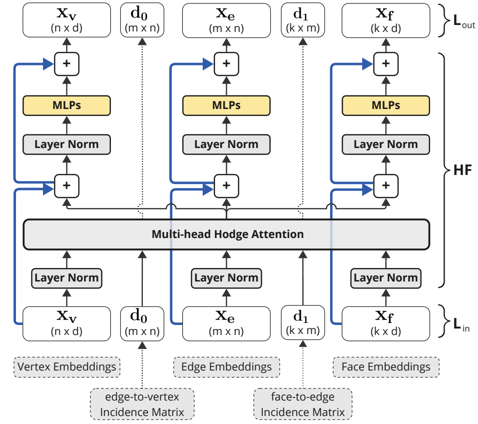
Multi-Head Hodge Attention

Vertex Features



$$L_v := \star_0^{-1}(x_v) \cdot d_0^T \cdot \star_1(x_e) \cdot d_0$$

HodgeFormer Layer

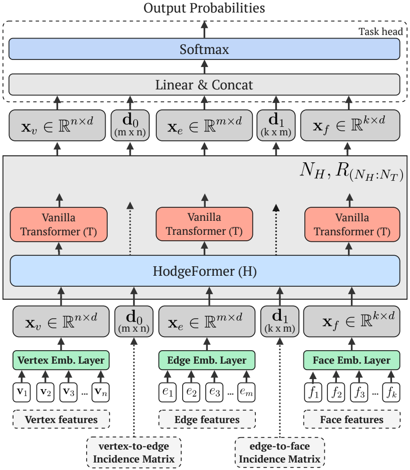


HodgeFormer Layer:

$H_l(x_v, x_e, x_f) :=$

$G_l(A_{H_l}(x_v, x_e, x_f) + (x_v, x_e, x_f))$

End-to-End Architecture



Architecture Components:

1. **Embedding Layers**: Map vertex, edge, face features to latent dimension
2. **HodgeFormer + Transformer Layers**: Local operators + global mixers
3. **Task Head**: Classification or segmentation output

From Mesh Features to Learned Hodge Stars

Vertex ($x_{v_{\text{in}}}$)
- 3D coordinates
- Normal: weighted average of face normals
- Area: weighted average of face areas

Edge ($x_{e_{\text{in}}}$)
- Endpoint 3D coords
- Opposite vertex coords
- Normal: average of vertex normals
- Incident-face edge lengths

Face ($x_{f_{\text{in}}}$)
- Vertex 3D coords (oriented)
- Face normal
- Face area

↓ ↓ ↓

for each $k \in \{v, e, f\}$

Embedding Layer $x_{k_{\text{in}}} \to x_k$

$$x_k = \mathrm{MLP}(\, x_{k_{\text{in}}} + A_k \cdot x_{k_{\text{in}}})$$

1. One-hop aggregation via adjacency A_k
2. MLP projection to latent dim d

Neighbors' info incorporated *before* projection.

Learned Hodge Star via Attention

$$Q_k = x_k W_{Q_k}, \quad K_k = x_k W_{K_k} \qquad W \in \mathbb{R}^{d \times d_h}$$

$$\boxed{\star_k(x_k) = \mathrm{softmax}\left(\frac{Q_k K_k^\top}{\sqrt{d_h}}\right)}$$

Row-stochastic · Non-diagonal · Data-dependent

xyz coordinates serve as positional encoding — features encode both **primal** and **dual** geometry, exactly what $\star_k$ depends on.

Sparse Attention for Efficiency

Challenge: Full attention has $O(n^2)$ complexity

Solution: Define sparsity patterns based on local neighborhoods

$$x_i = \sum_{j \in S_i} A_{ij} V_j, \quad A_{ij} = \operatorname*{softmax}_{j \in S_i} \left(\frac{Q_i \cdot K_j^T}{\sqrt{d}} \right)$$

Sparsity Pattern S_i:

- Local neighbors via BFS on mesh adjacency
- Random connections ($\sqrt{n}$ total neighbors)

Complexity

Overall computational complexity: $O(n^{1.5} d)$

Experimental Setup

Implementation:

- PyTorch with standard backpropagation
- BFS operations via GraphBLAS framework
- Single NVIDIA RTX 4090 GPU (24GB VRAM)

Architecture Configuration:

- Latent dimension: $d = 256$
- MLP hidden dimension: $d_h = 512$
- Attention heads: $h = 4$
- Layers: 6 HodgeFormer + 2 Transformer

Results

Method	Type	Acts On	EVD	SHREC11 (split-10)	Cube Engrav.	Human Simp.	COSEG Vases	COSEG Chairs	COSEG Aliens
HodgeNet	mlp	v	Yes	94.7%	n/a	85.0%	90.3%	95.7%	96.0%
DiffusionNet	mlp	v	Yes	99.5%	n/a	90.8%	n/a	n/a	n/a
LaplacianNet	mlp	v	Yes	n/a	n/a	n/a	92.2%	94.2%	93.9%
Laplacian2Mesh	cnn	v	Yes	**100.0%**	**91.5%**	**88.6%**	94.6%	96.6%	95.0%
MeT	trns	f	Yes	n/a	n/a	n/a	**99.8%**	**98.9%**	**99.3%**
MeshCNN	cnn	e	No	91.0%	92.2%	85.4%	92.4%	93.0%	96.3%
PD-MeshNet	cnn	ef	No	99.1%	94.4%	85.6%	95.4%	97.2%	98.2%
MeshWalker	rnn	v	No	97.1%	98.6%	n/a	**99.6%**	98.7%	**99.1%**
SubDivNet	cnn	f	No	99.5%	**98.9%**	**91.7%**	96.7%	96.7%	97.3%
EMNN (MC+H)	gnn	ef	No	**100%**†	n/a	88.7%†	n/a	n/a	n/a
EGNN (MC+H)	gnn	ef	No	99.6%†	n/a	87.2%†	n/a	n/a	n/a
HodgeFormer	trns	vef	No	98.7%	95.3%	90.3%	94.3%	**98.8%**	98.3%

HodgeFormer achieves results comparable to the state-of-the-art without spectral features, eigenvalue decomposition operations or complex complementary structures.

Mesh Size (n_v)	2^8	2^{10}	2^{12}	2^{14}
		Compute Time (ms)		
HF Encoder (Train)	5.78	8.98	29.72	197.9
HF Encoder (Infer)	2.50	3.42	10.96	66.13
HF Layer (Infer)	1.08	1.91	9.42	55.25
		Peak Memory Usage (GBs)		
HF Encoder (Train)	0.12	0.40	2.54	19.23
HF Encoder (Infer)	0.09	0.31	2.08	15.89
HF Layer (Infer)	0.09	0.31	2.08	15.89

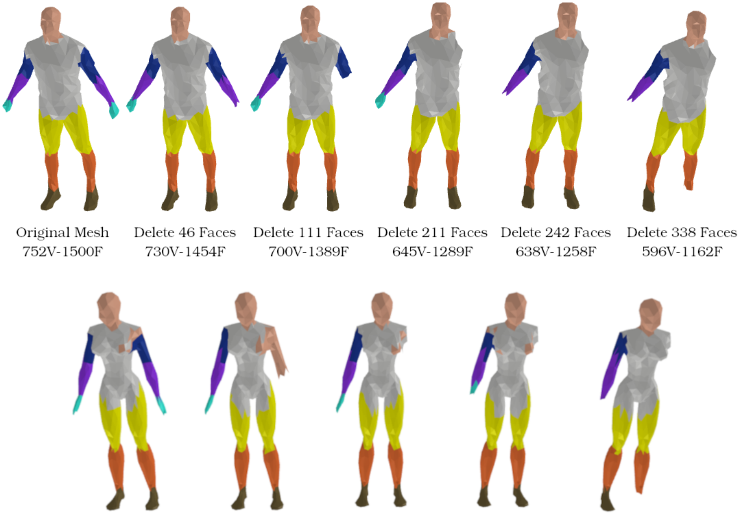
	Exec Time (s)	GPU Time (s)	GPU Batch (ms)	Peak Mem (GB)
		Training (batch size $=$ 12)		
MeshCNN	26.84	25.59	806.29	**4.41**
Laplacian2Mesh	641.34	605.17	423.57	4.80
HodgeFormer	**17.58**	**8.36**	**263.57**	14.21
		Testing (batch size $=$ 1)		
MeshCNN	1.36	1.09	60.40	0.19
Laplacian2Mesh	2.96	0.13	7.98	0.95
HodgeFormer	1.97	0.40	22.41	0.41

Qualitative Results

Mesh segmentation results on the Human Body test set.

Qualitative Results

Robustness to incomplete meshes.



Qualitative Results

Robustness to Gaussian Noise and remeshing.

Original Meshes

Gaussian Noise (λ=0.020)

QEM Remesh (2000 Faces)

HUMAN Test Set Variant	Accuracy	Acc.Drop
Original	90.3%	n/a
Gaussian Noise ($\lambda = 0.005$)	88.3%	2.0%
Gaussian Noise ($\lambda = 0.010$)	87.3%	3.0%
Gaussian Noise ($\lambda = 0.020$)	81.1%	9.2%
QEM Remesh ($1000F$)	87.2%	3.1%
QEM Remesh ($2000F$)	86.2%	4.1%
Face Removal ($p = 0.04$)	87.8%	2.5%
Face Removal ($p = 0.10$)	85.9%	4.4%
Face Removal ($p = 0.20$)	81.6%	8.7%
Patch removal	86.8%	3.5%

Face Removal (p = 0.20)

Patch Removal

Ablation Studies

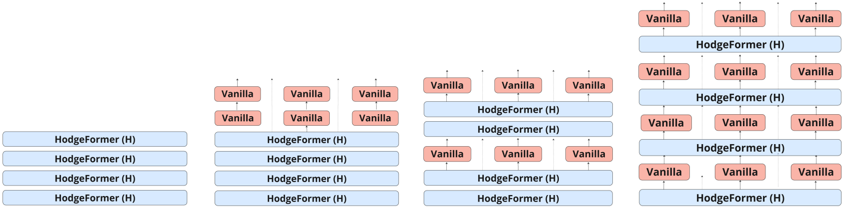
Transformer	Embedding	Vases
Vanilla	FNN	84.82%
Vanilla	Neighbors + FNN	87.28%
HodgeFormer	FNN	91.67%
HodgeFormer	Neighbors + FNN	92.02%

No. Neighbors (# V, # E)	Vases
1v, 1e	88.37%
8v, 12e	90.72%
16v, 24e	90.84%
32v, 48e	92.02%
64v, 96e	92.06%
128v, 196e	92.13%

Input Features	Vases
coords	86.57%
normals	56.41%
coords-normals	91.66%
coords-areas	88.22%
all (coords-normals-areas)	92.02%

Ablation Studies

Mixing HodgeFormer and Transformer layers.

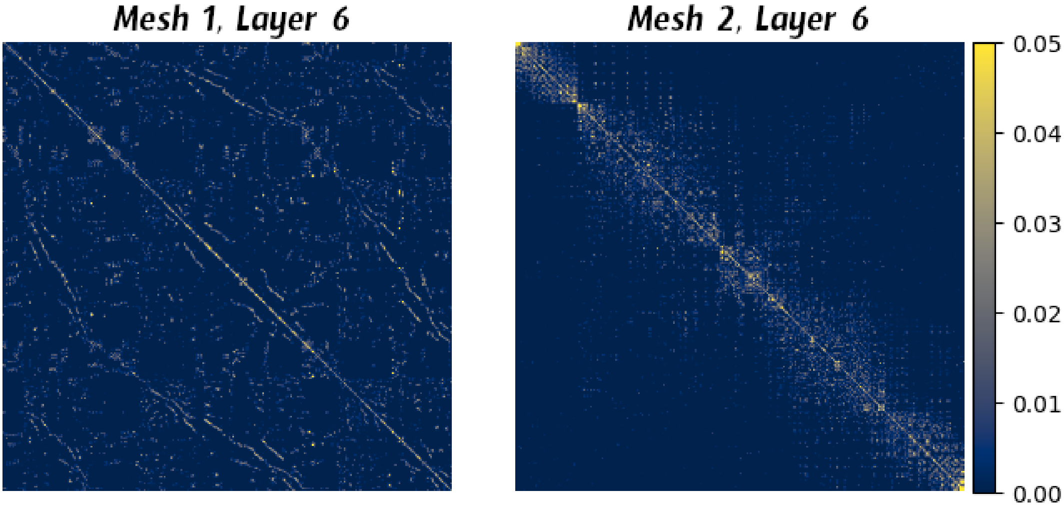


(a) (b) (c) (d)

Layers	Vases (COSEG)	Ref
$N_H = 4$ & $R_{(N_H,N_T)} = 4:0$	92.02%	(a)
$N_H = 4$ & $R_{(N_H,N_T)} = 4:2$	93.04%	(b)
$N_H = 4$ & $R_{(N_H,N_T)} = 2:1$	92.59%	(c)
$N_H = 4$ & $R_{(N_H,N_T)} = 1:1$	92.85%	(d)

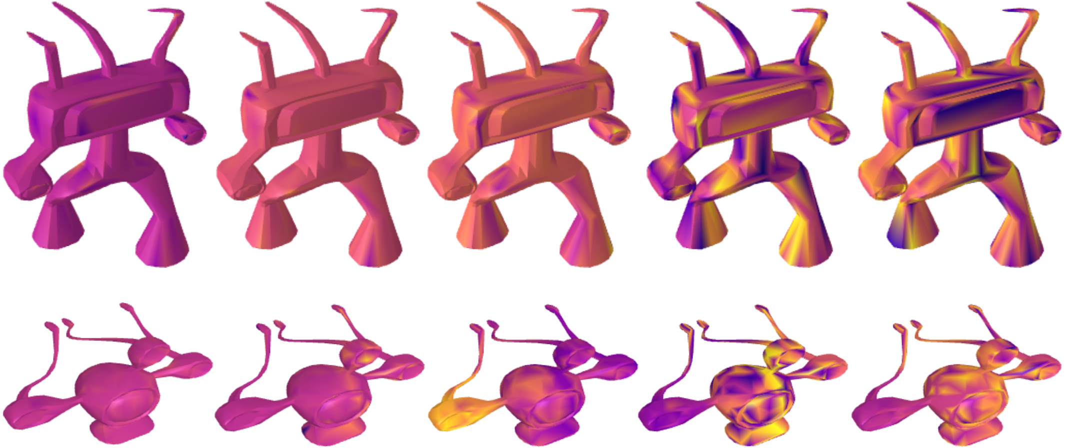
N_H: number of HodgeFormer layers; $R_{H:T} = (N_H : N_T)$: ratio to Vanilla Transformer layers.

Attention maps



Eigenfunctions of learned operators

Computed in Layer 6 Head 1



Conclusion

HodgeFormer: Key Contributions

1. **Novel connection**: Hodge stars $\approx$ Galerkin mass matrices $\approx$ Attention
2. **Learn operators**: Data-driven Hodge Laplacians via multi-head attention
3. **No EVD**: Eliminates $O(n^3)$ eigendecomposition bottleneck
4. **Efficient**: Sparse attention achieves $O(n^{1.5}d)$ complexity
5. **Competitive**: Near state-of-the-art accuracy, dramatically faster training

The Paradigm Shift

Not all geometric information needs eigendecomposition. By learning Hodge stars via attention, we get the **structural benefits of spectral methods** in a **fully differentiable, GPU-friendly framework**.

Limitations & Open Questions

Current Limitations:

- Some segmentation tasks: spectral Transformers still slightly better (e.g., COSEG vases)
- Higher memory footprint than some spatial methods (14GB vs 4GB)
- Theoretical expressiveness guarantees not fully established

Open Research Questions:

- **Scalability**: How does it perform on 100K+ vertex meshes?
- **Volume meshes**: Extension to 3D tetrahedral complexes?
- **Transfer learning**: Do learned Hodge stars generalize across domains?
- **Universal approximation**: Can attention-based Hodge stars approximate any Galerkin operator?

Future Directions

Large-scale pretraining, self-supervised learning (MeshMAE-style), PDE solving on meshes

Thank You!

HodgeFormer

Transformers for Learnable Operators on Triangular Meshes
through Data-Driven Hodge Matrices

Code & Data:

https://github.com/hodgeformer/

Contact:

Akis Nousias: anousias@k3y.bg
Stavros Nousias: stavros.nousias@tum.de