IMR 2024 Short Course

Intelligent Mesh Generation

Na Lei
Dalian University of Technology

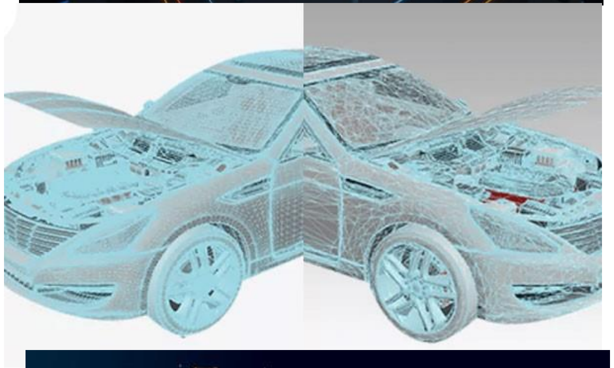
March 5, 2024

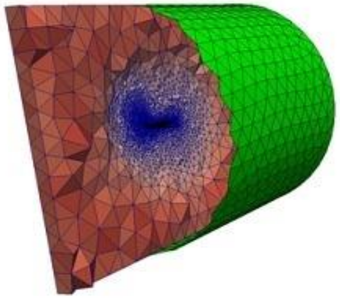
Outline

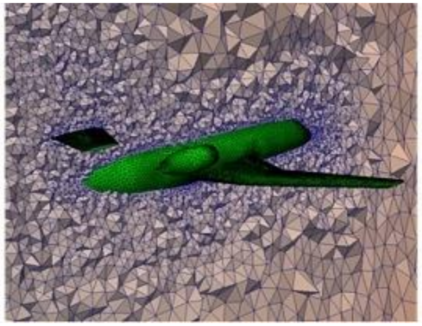
- Research Background
- Intelligent Mesh Representation(IMR)
- Intelligent Mesh Generation(IMG)
- Intelligent Mesh Evaluation(IME)
- Summary and Outlook

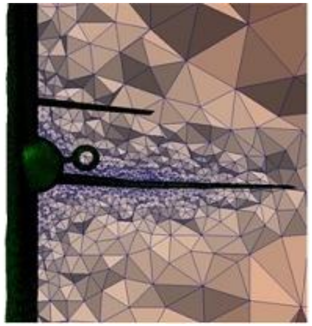
Research Background

- **The mesh is a foundational representation for 3D models, supporting applications like the metaverse, digital twins, numerical simulations, and more.**
- **Intelligent mesh generation technology significantly complements traditional methods, improving their practicality and generality, and unlocking new possibilities for mesh generation applications.**









Research Background

Definition of intelligent mesh generation

Narrow definition:

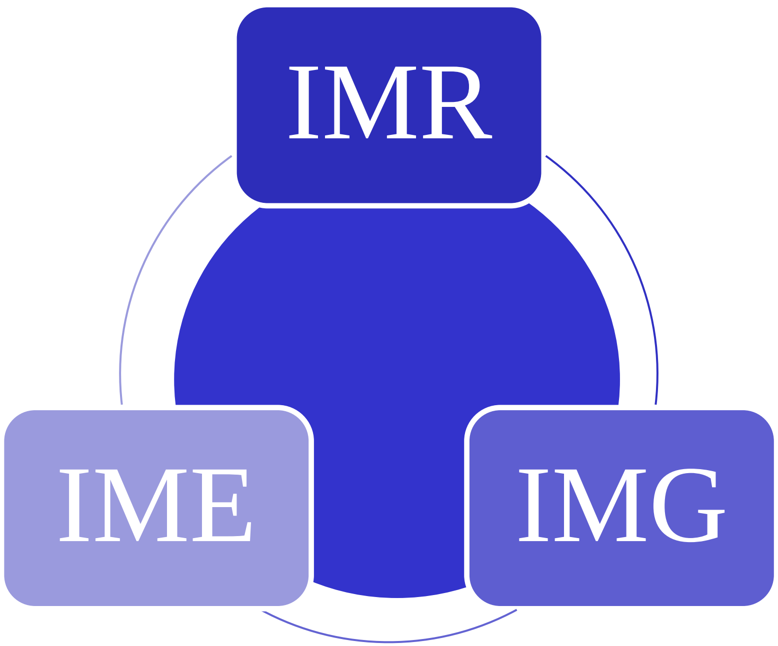
Mesh generatioin techniques in which machine learning is involved in part or all of the process.

Broad definition:

A technique involving machine learning with the mesh as the final representation.

Research Background

❑ Key components in intelligent mesh generation



Intelligent Mesh Representation

(1) How to put mesh data into a neural network?
(2) How to handle input information?
(3) How to use neural networks to extract deep features?

Intelligent Mesh Generation

(1) What kind of framework is suitable for mesh generation?
(2) What role do intelligent frameworks play in mesh generation?

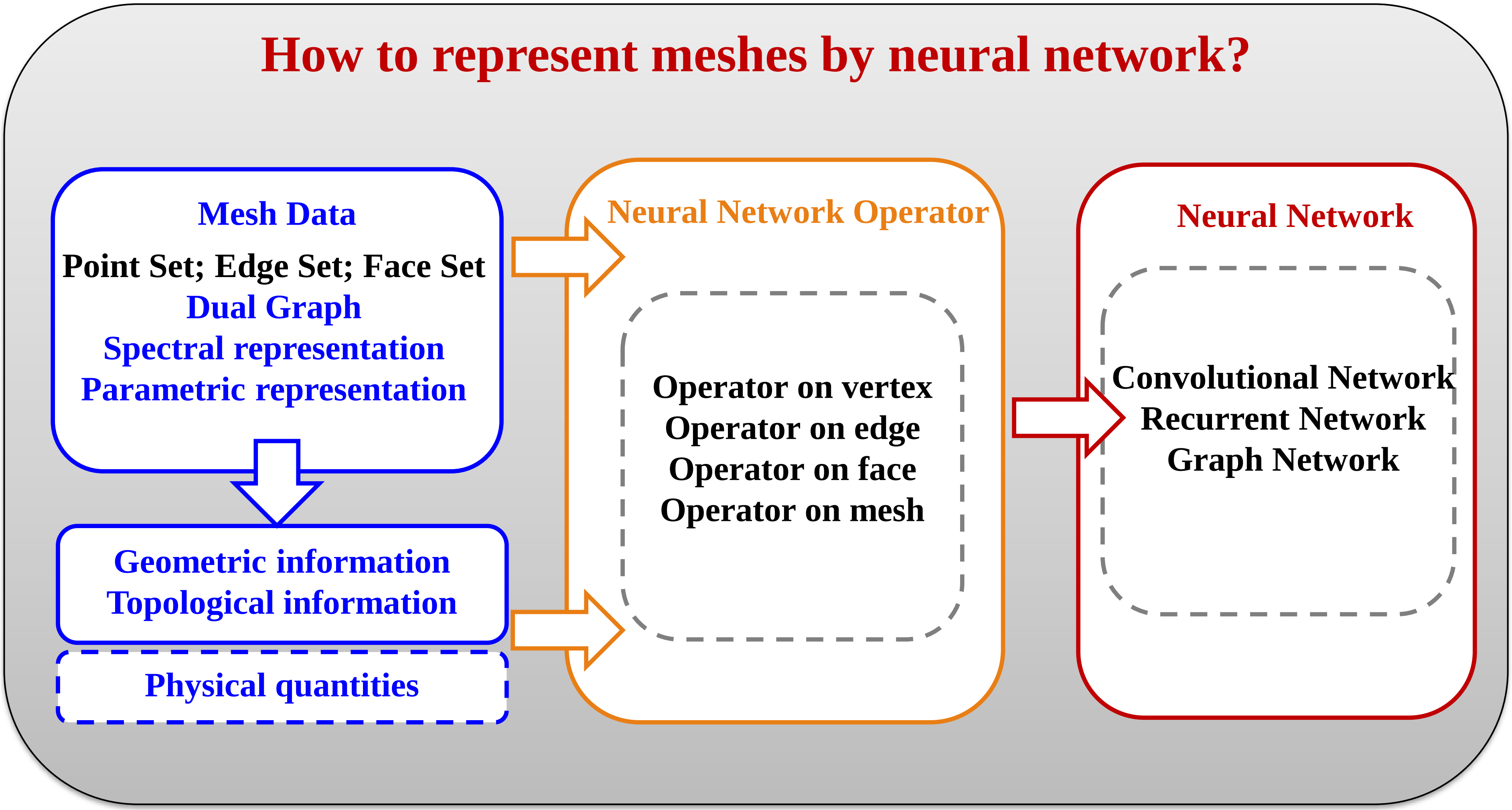
Intelligent Mesh Evaluation

(1) Conversion of metrics in traditional meshing to losses or reward functions in intelligent meshing
(2) Network gives integrated evaluation metrics

Outline

- Research Background
- Intelligent Mesh Representation(IMR)
- Intelligent Mesh Generation(IMG)
- Intelligent Mesh Evaluation(IME)
- Summary and Outlook

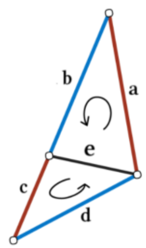
Intelligence Mesh Representation

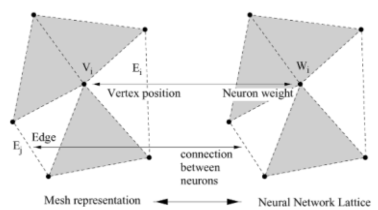


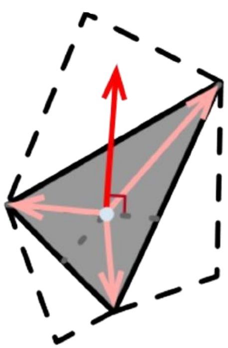
Intelligent Mesh Representation

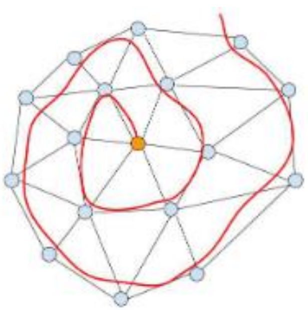
How to put mesh data into a neural network?

Primitive mesh based



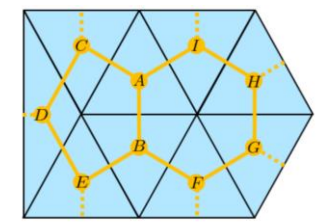


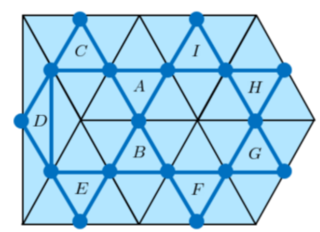




Methods defines operator on primitive mesh.

Dual graph-based





Consider meshes as dual graph-structured data

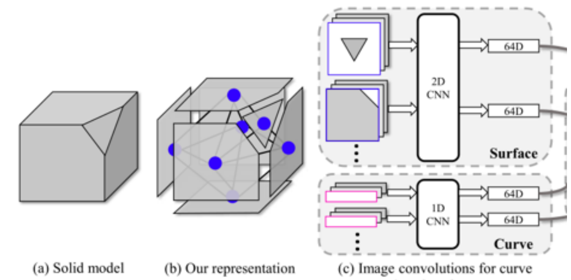
Intelligent Mesh Representation

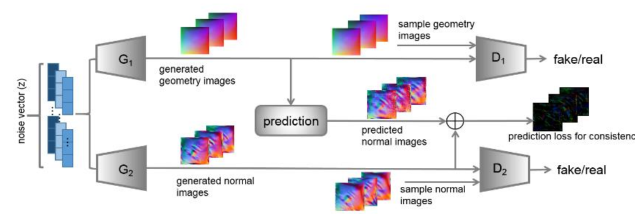
How to put mesh data into a neural network?

Spectral method-based


Learn relevant differential operators according to a geometric analysis task

Parameterization-based





Convert a mesh to an image problem using parameterization

How to put mesh data into a neural network?

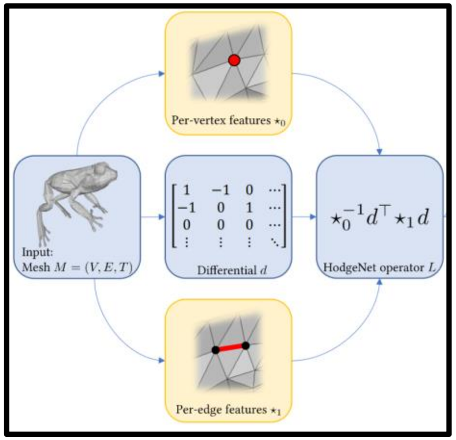
- Method 1: Employing **vertices** as the core mesh data, enriched with supplementary geometric characteristics.

✓ Utilizes the x, y, and z coordinates of vertices as input features.

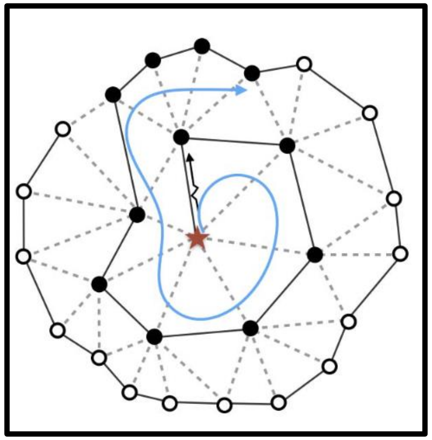
✓ [Batchsize, VertexNum, x, y, z]

✓ Some networks process these coordinates

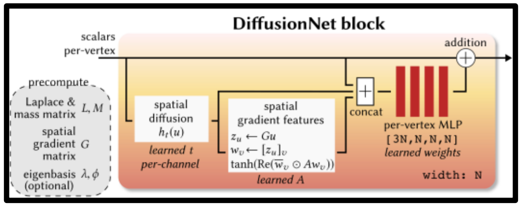
  ✓ MeshWalker employs coordinate offsets$(\Delta X, \Delta Y, \Delta Z)$ in the vertex sequence



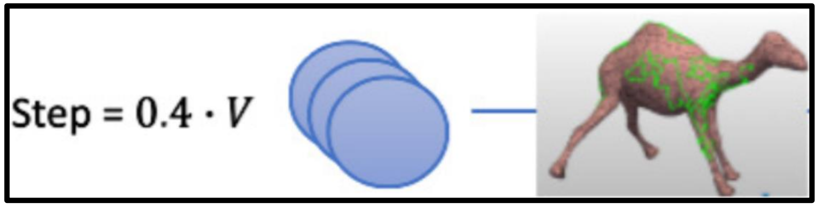
HodgeNet



SpiraNet



DiffusionNet



MeshWalker

How to put mesh data into a neural network?

- Method 2: Representing the mesh with **edges** as the primary data supplemented by other geometric properties.

✓ Utilizes the geometric properties of edges as primary data:

✓ [Batchsize, EdgeNum, N-geometric attributes]

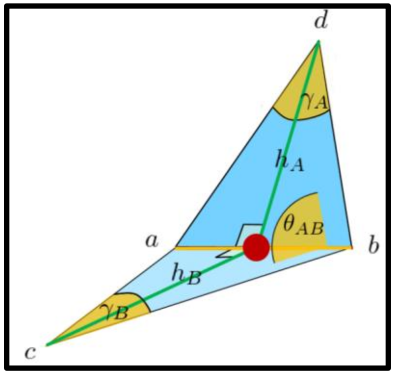
  ✓ dihedral angle

  ✓ internal angles

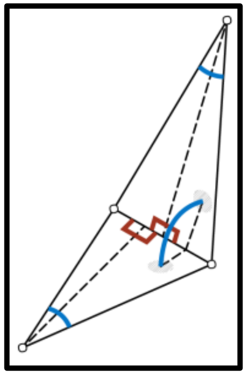
  ✓ two edge-to-height ratios

✓ Some networks add extral information

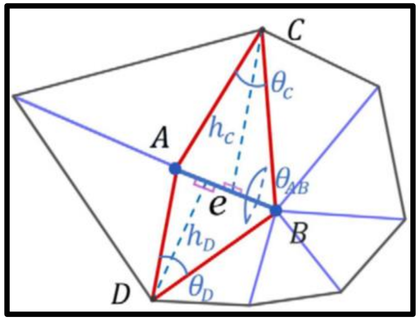
  ✓ TPnet add two vertex degrees



PD-Net



MeshCnn



TPnet

How to put mesh data into a neural network?

- Method 3: Utilizing **faces** as the fundamental data for mesh representation, augmented by additional geometric attributes.

✓ Utilizes the feature of faces as input features:

✓ [Batchsize, FaceNum, N-geometric attributes]

  ✓ Center

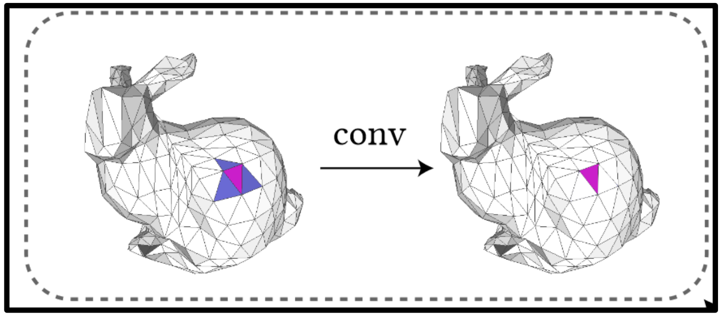
  ✓ Corner

  ✓ Normal

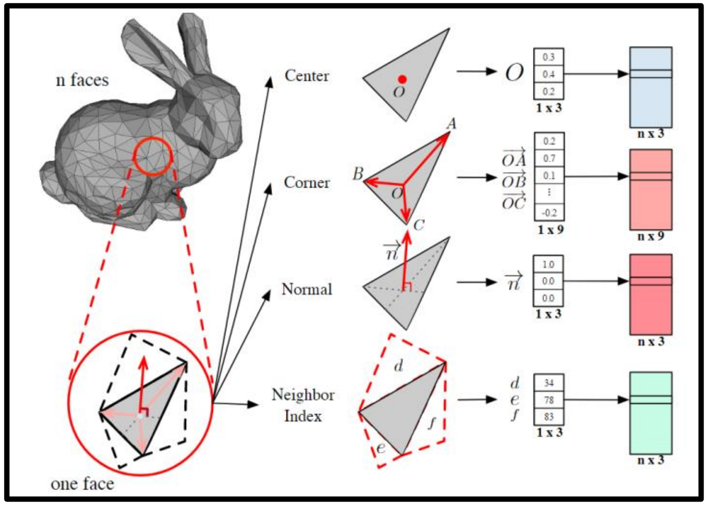
  ✓ Area

✓ Some networks process additional information

  ✓ SubdivNet employs inner products of face normal with vertex normals



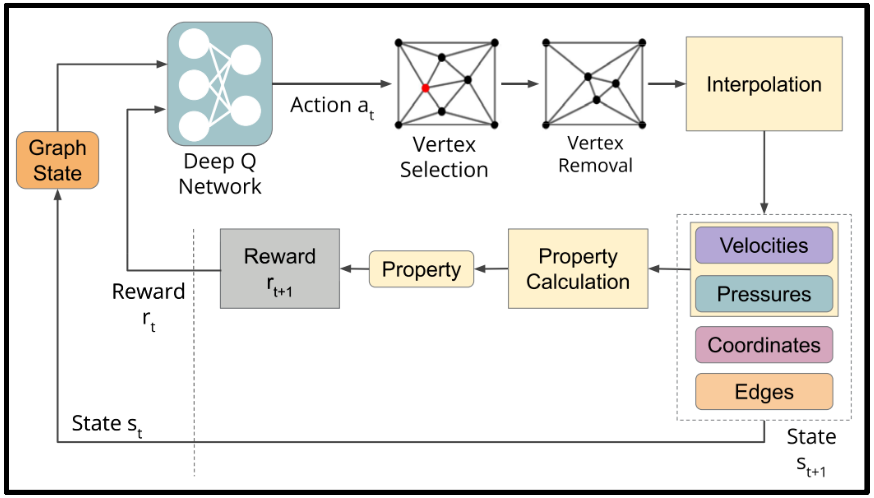
SubdivNet

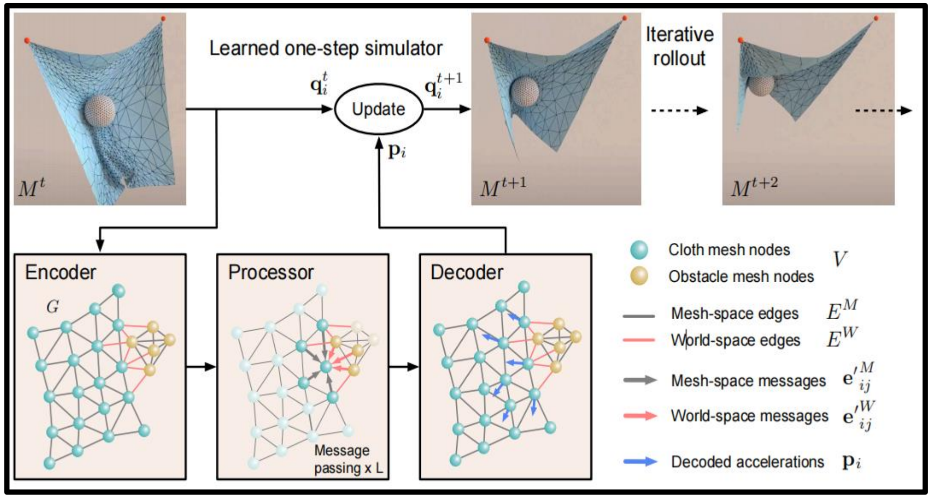


MeshNet

Mesh Representation for Numerical Simulation

- Store physical information on vertices or edges
- Fusion of mesh geometry information and physical quantities using networks





MeshDQN

The physical quantities are stored on vertices.

MeshGraphNets

The physical quantities are stored on edges.

How to handle input information?

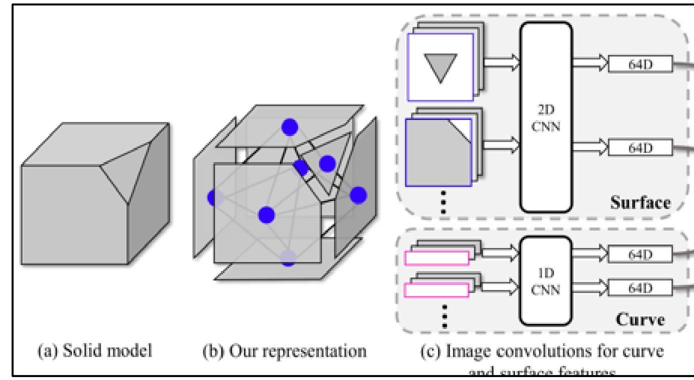
Mesh data

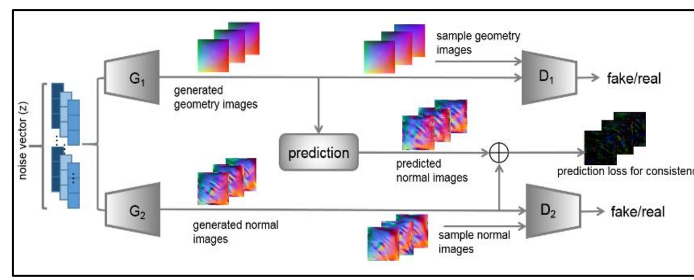
Find Data Structure

Parameter/ projection mapping

Mesh(Graph) data → **Designed operator**

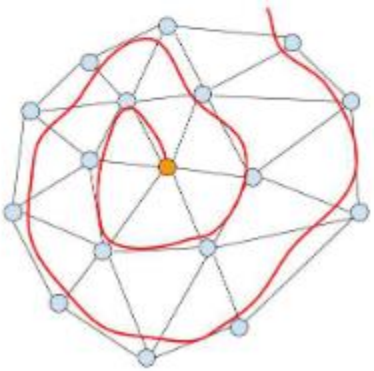
Image data → **Image operator**



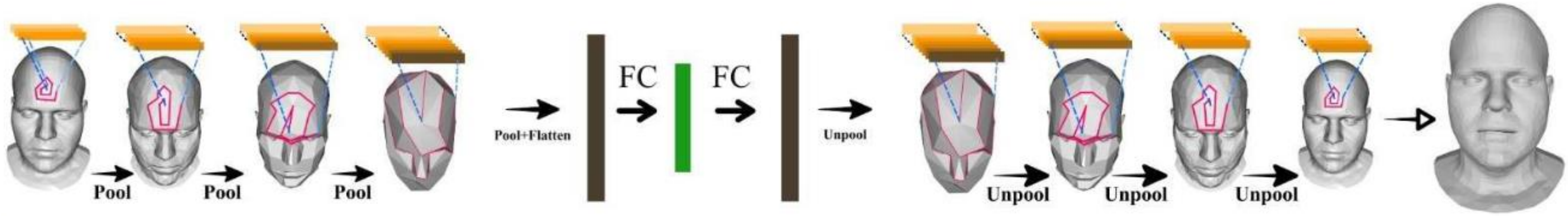


How to handle input information?

- Operator 1: Spiral **vertex operator**, derived from SpiralNet *
  - Spiral patch operator: $S(x)=\{x, R_1^1(x), R_2^1(x), \ldots, R_{|R^h|}^h\},$
  - Spiral convolution: $(f*g)_x=\sum_{\ell=1}^{L} g_\ell\, f\big(S_\ell(x)\big).$
  - Initial point: $R_1^1(x)=\arg\min_{y\in R^1(x)} d_{\mathcal{M}}(x_0,y),$



mesh convolution

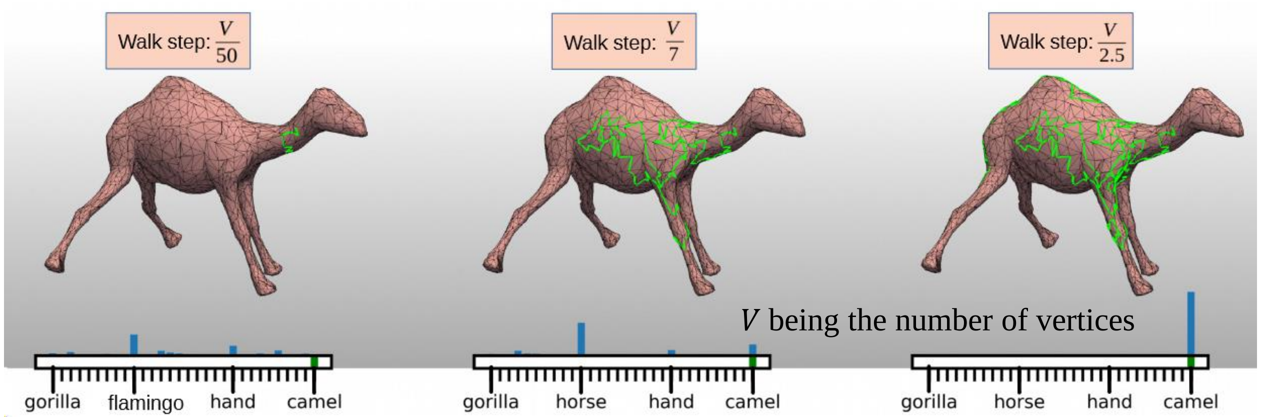


mesh pooling and unpooling

* Lim I, Dielen A, Campen M, et al. A simple approach to intrinsic correspondence learning on unstructured 3d meshes[C]//Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018: 0-0.

How to handle input information?

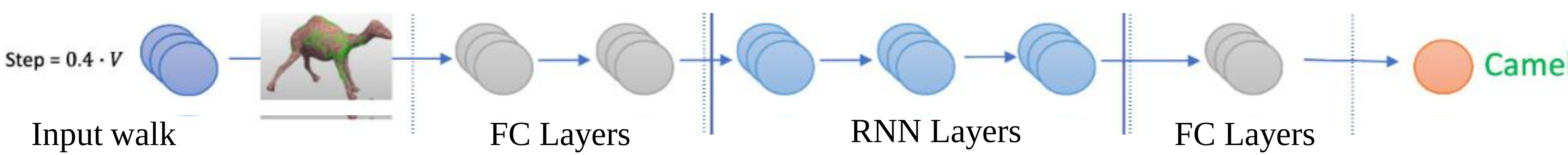
- Operator 2: Random Walk **Vertex Operator**, derived from MeshWalker *



walk (in green) proceeds along the surface

Walk: A sequence of vertices

- Generation: Randomly select the starting vertex and select the next vertex according to the adjacency until the present walk length is reached
- Representation: Each vertex is represented as the 3D translation from the previous vertex in the walk ($\Delta X, \Delta Y, \Delta Z$)

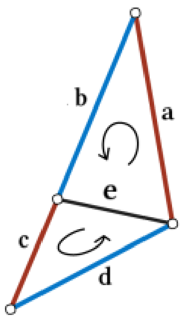


* Lahav A, Tal A. Meshwalker: Deep mesh understanding by random walks[J]. ACM Transactions on Graphics (TOG), 2020, 39(6): 1-13.

How to handle input information?

- Operator 3: Classic **edge operator**, derived from MeshCNN *

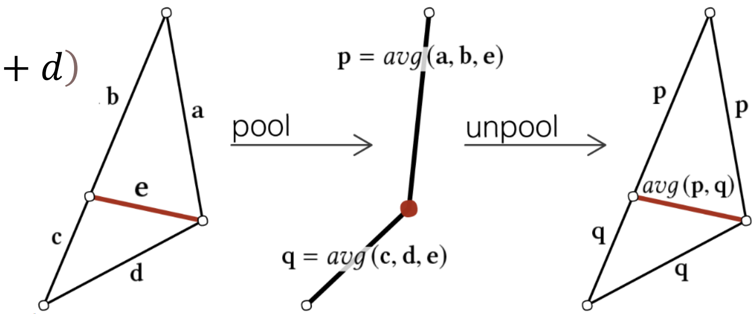
Invariant convolutions

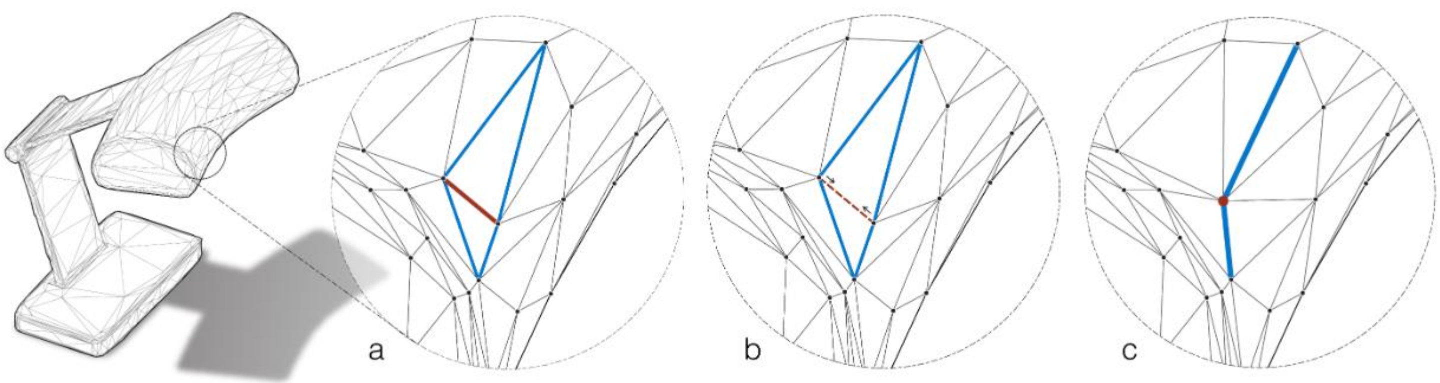


$$(e^1, e^2, e^3, e^4) = (|a - c|, a + c, |b - d|, b + d)$$

$$e \cdot k_0 + \sum_{j=1}^{4} k_j \cdot e^j,$$

mesh pooling and unpooling



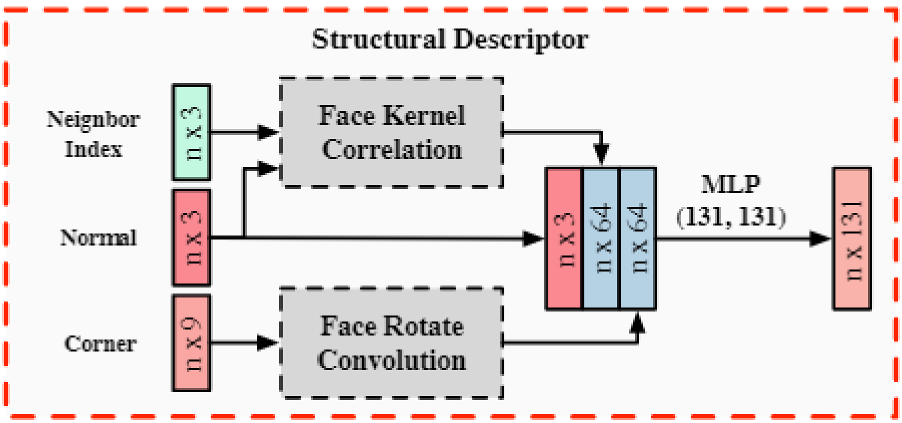


MeshCNN pooling example

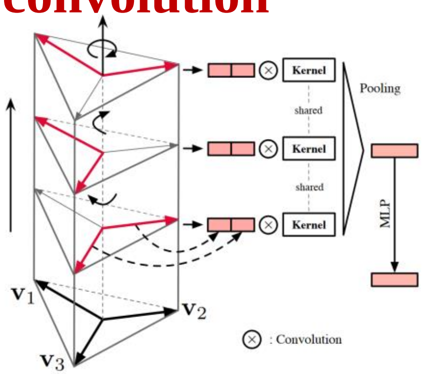
* Hanocka R, Hertz A, Fish N, et al. Meshcnn: a network with an edge[J]. ACM Transactions on Graphics (ToG), 2019, 38(4): 1-12.

How to handle input information?

- Operator 4: Rotational **face operator**, derived from MeshNet *



Face rotate convolution



$$g\left(\frac{1}{3}\left(f(\mathbf{v}_1,\mathbf{v}_2)+f(\mathbf{v}_2,\mathbf{v}_3)+f(\mathbf{v}_3,\mathbf{v}_1)\right)\right)$$

Face kernel correlation

Define the face kernel as M learnable normals and correlation refers to the similarity between the face normals and the kernel normal.

$$KC(i,k)=\frac{1}{|N_i||M_k|}\sum_{\boldsymbol{n}\in Ni}\sum_{\boldsymbol{m}\in Mk}K_\sigma(\boldsymbol{n},\boldsymbol{m})$$

$$K_\sigma(\boldsymbol{n},\boldsymbol{m})=\exp\left(-\frac{\|\boldsymbol{n}-\boldsymbol{m}\|^2}{2\sigma^2}\right)$$

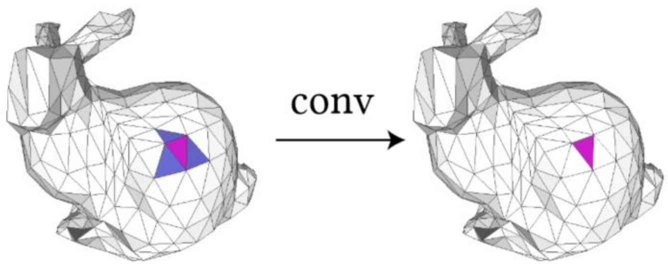
N_i:the set of normal of the *i-th* face and its neighbor faces

M_k:the set of normals in the *k-th* kernel

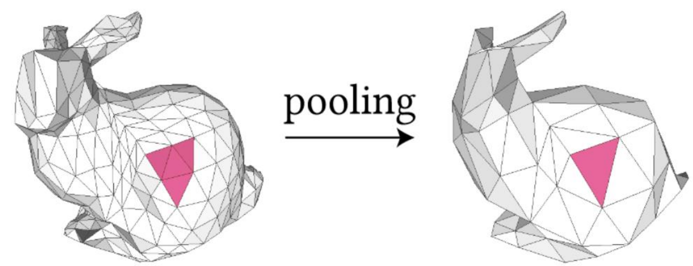
* Feng Y, Feng Y, You H, et al. Meshnet: Mesh neural network for 3d shape representation[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 8279-8286.

How to handle input information?

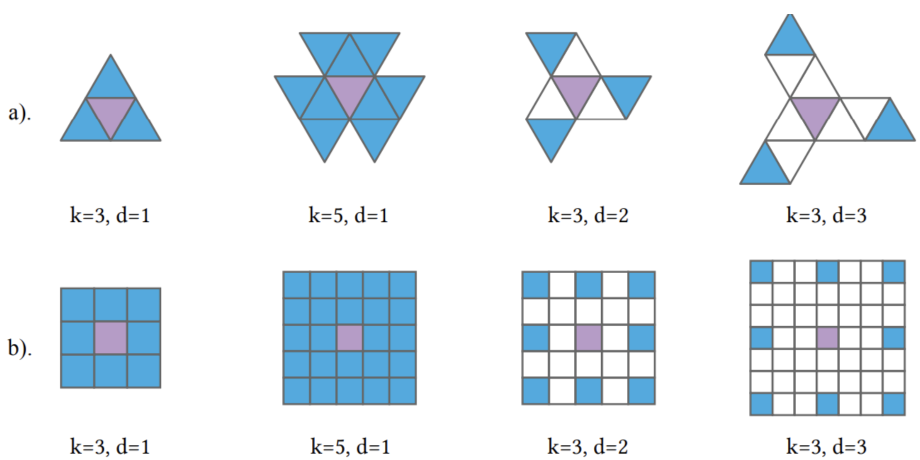
- Operator 5: Analogous to 2-D convolution, **face operator** derived from SubdivNet *



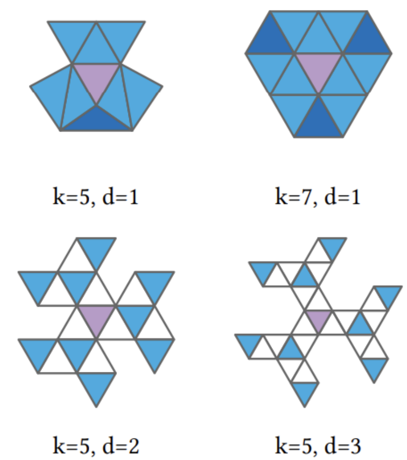
mesh convolution



mesh pooling



Face convolution kernel analogous to 2D convolution

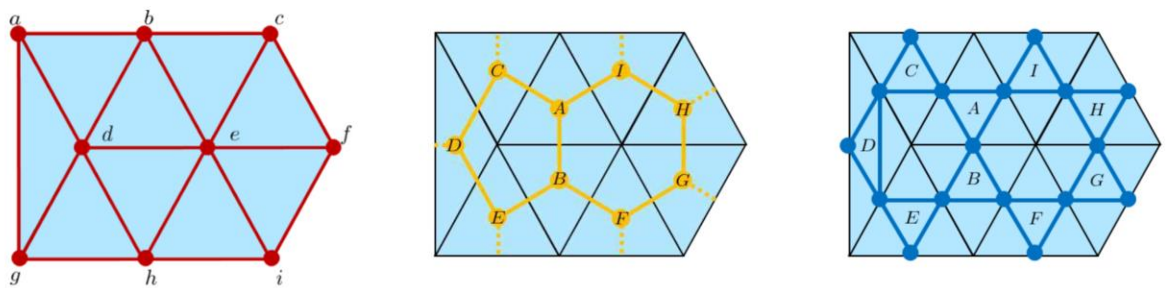


Zig-zag dilated convolution

* Hu S M, Liu Z N, Guo M H, et al. Subdivision-based mesh convolution networks[J]. ACM Transactions on Graphics (TOG), 2022, 41(3): 1-16.

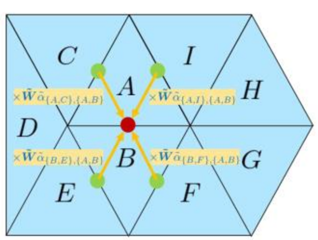
How to handle input information?

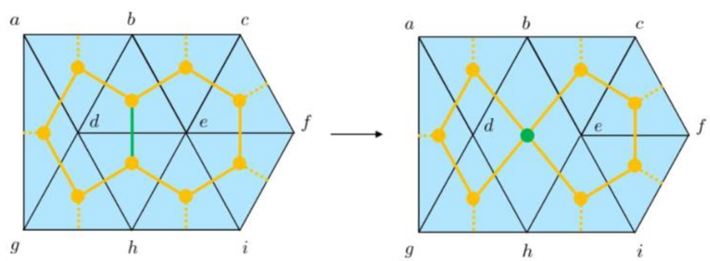
- Operator 6: **Mesh operator** based on the dual graph, derived from PD-NET *

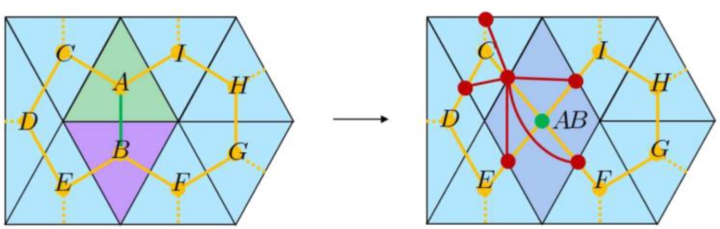


(a) Input mesh $\mathcal{M}$ (b) Primal graph $\mathcal{P}(\mathcal{M})$ (c) Dual graph $\mathcal{D}(\mathcal{M})$

Primal-dual graphs associated to an input mesh in PD-MeshNet.





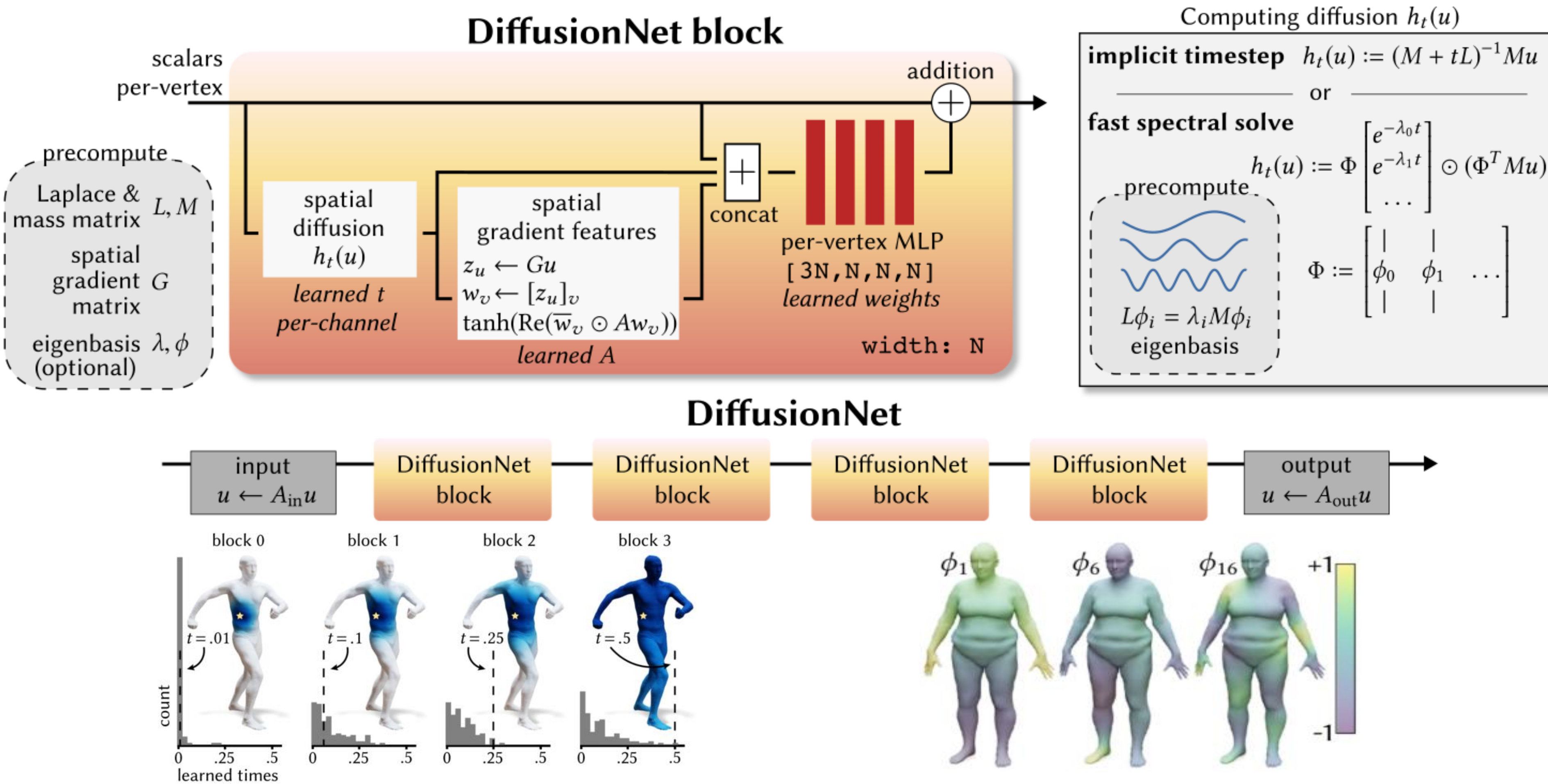


mesh pooling in P(M) **Invariant convolutions in D(M)**

* Milano F, Loquercio A, Rosinol A, et al. Primal-dual mesh convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2020, 33: 952-963.

How to handle input information?

- Operator 7: Diffusion **Mesh Operator**, derived from DiffusionNet *



* Sharp N, Attaiki S, Crane K, et al. Diffusionnet: Discretization agnostic learning on surfaces[J]. ACM Transactions on Graphics (TOG), 2022, 41(3): 1-16.

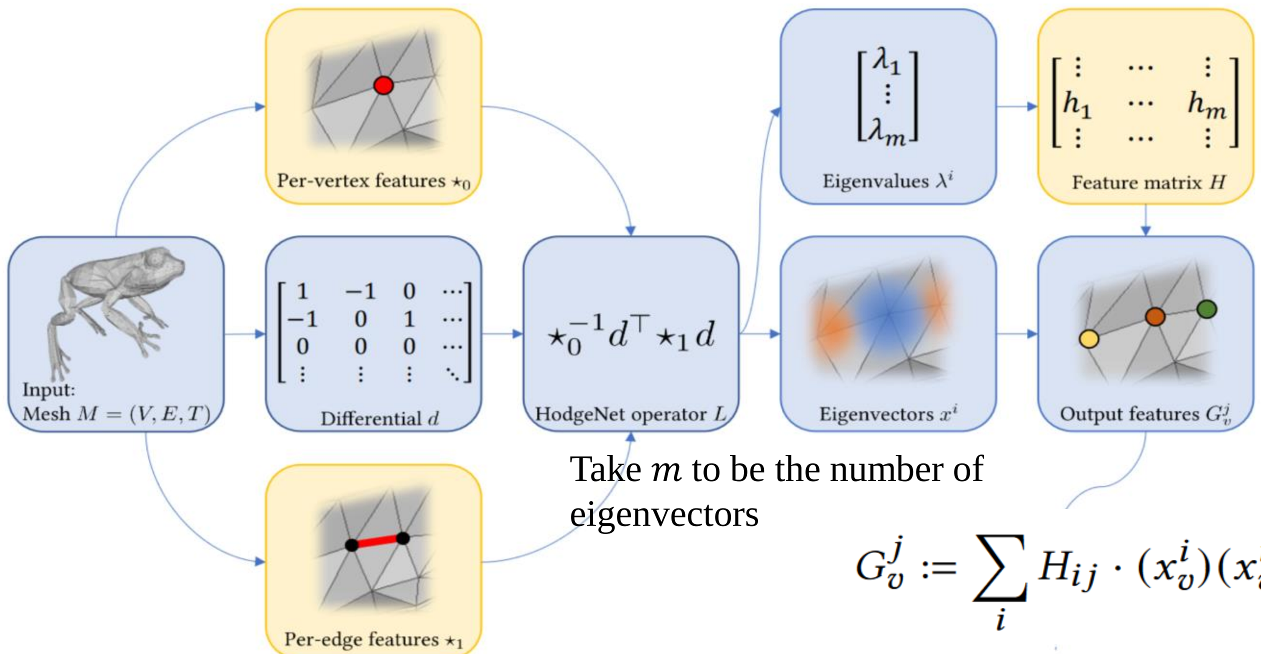
How to handle input information?

- Operator 8: **Mesh Operator** based on the Hodge star, derived from HodgeNet *

$$(\star_0(F))_{vv} = \varepsilon + \sum_{t\sim v} f_\Phi(F_{v_1}, F_{v_2}, F_{v_3})^2$$

n to be the number of output features

$$h_\Phi : \mathbb{R} \to \mathbb{R}^n$$



Take m to be the number of eigenvectors

$$G_v^j := \sum_i H_{ij} \cdot (x_v^i)(x_v^i)^\top,$$

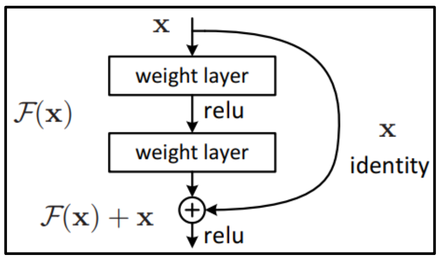
$$(\star_1(F))_{ee} = \varepsilon + g_\Phi(F_{v_1}, F_{v_2}, F_{v_3}, F_{v_4})^2,$$

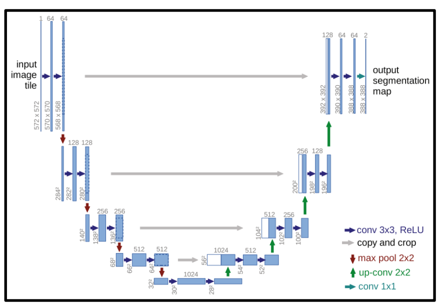
* Smirnov D, Solomon J. HodgeNet: Learning spectral geometry on triangle meshes[J]. ACM Transactions on Graphics (TOG), 2021, 40(4): 1-11.

Intelligent Mesh Representation

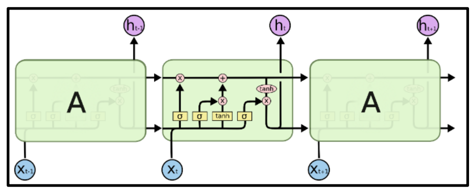
How to use neural networks to extract deep features?

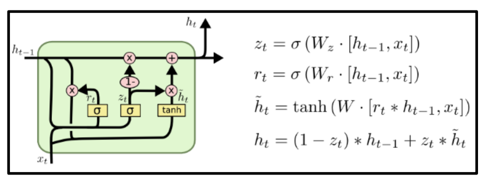
Convolutional Network



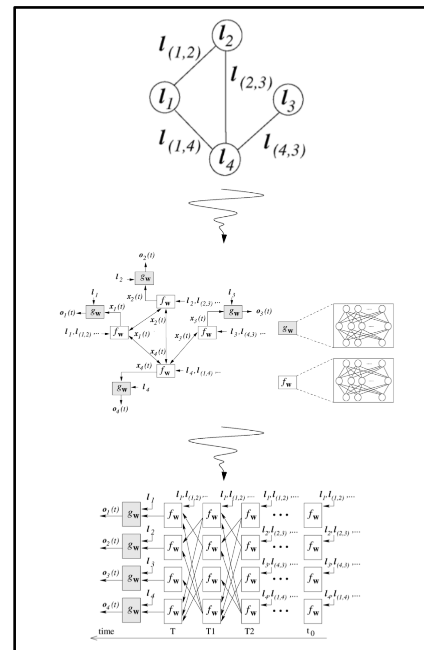


Recurrent Network



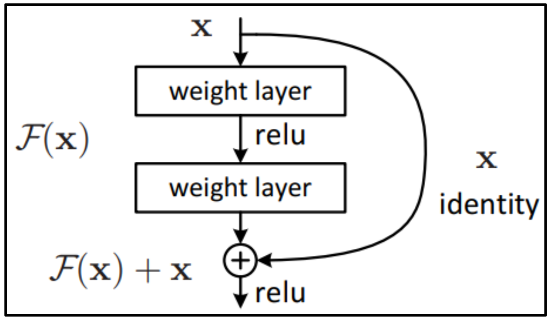


Graph Network

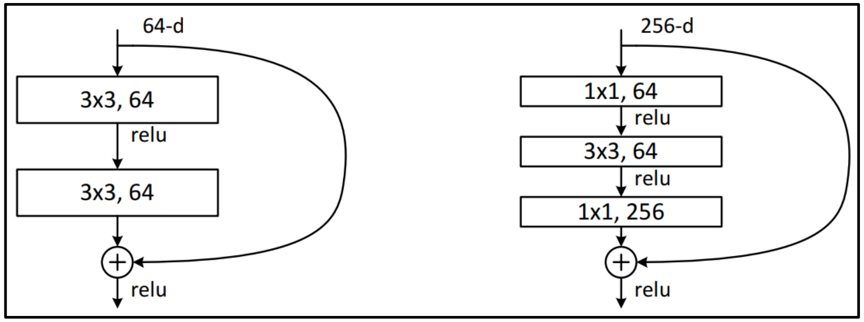


Convolutional Network

ResNet*



Residual Block



Residual Block Instance

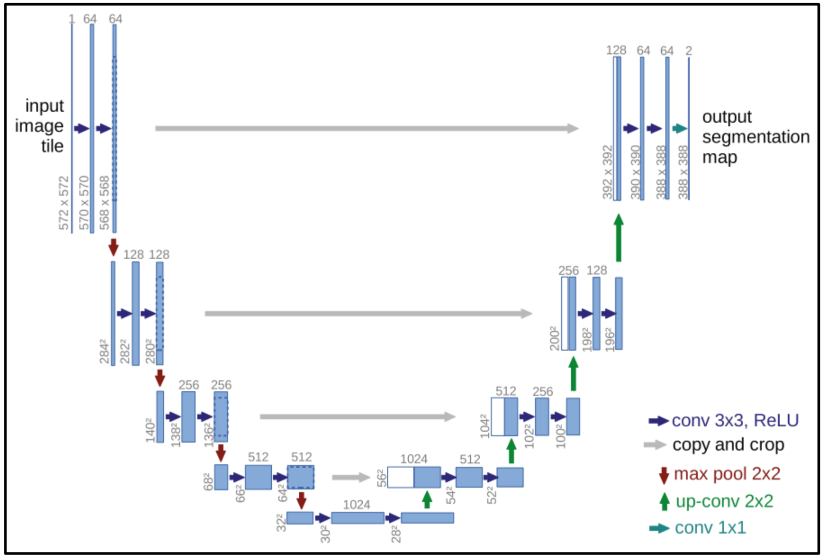
- Utilize shortcut connection to construct a residual learning framework
- Solve the problem of deep network degradation and difficulty in training
- Suitable for various Computer Vision(CV) tasks(e.g. image classification, object detection)

* He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

Convolutional Network

■ U-Net*

U-Net Architecture

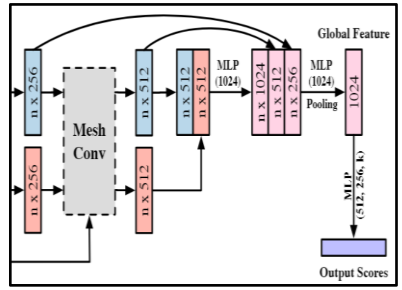


- Comprise contracting path and symmetric expansive path, with skip-connections between corresponding levels
- Improve performance under limited training samples through data augmentation
- Applicable to various biomedical image segmentation tasks

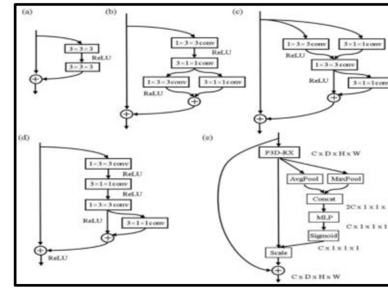
* Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015: 234-241.

Convolutional Network

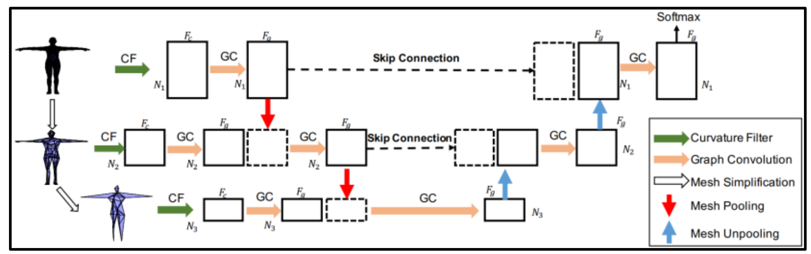
- For ResNet like networks, residual connections deepen it to obtain semantic features of mesh information
- For U-Net like networks, hierarchical structure can obtain multi-granularity deep features
- Combine the above two neural structures gradually becomes a broader idea



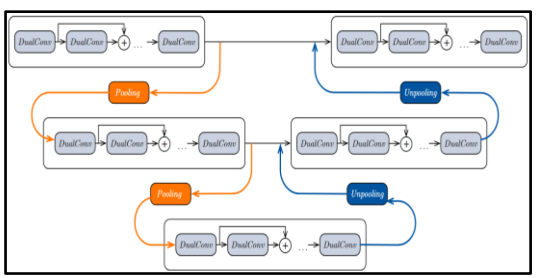
MeshNet



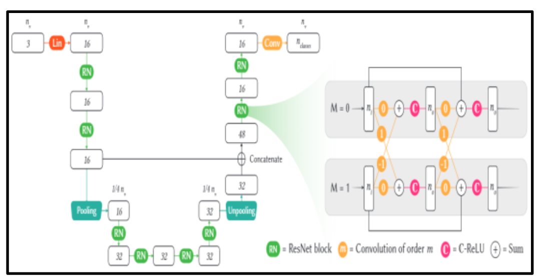
PointerNet



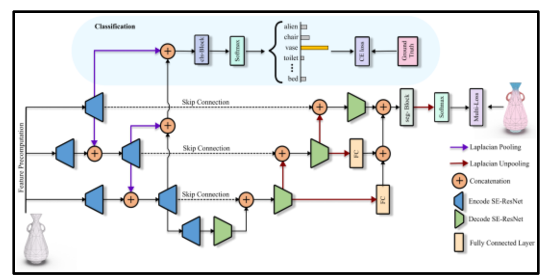
CurvaNet



DualConvMesh-Net



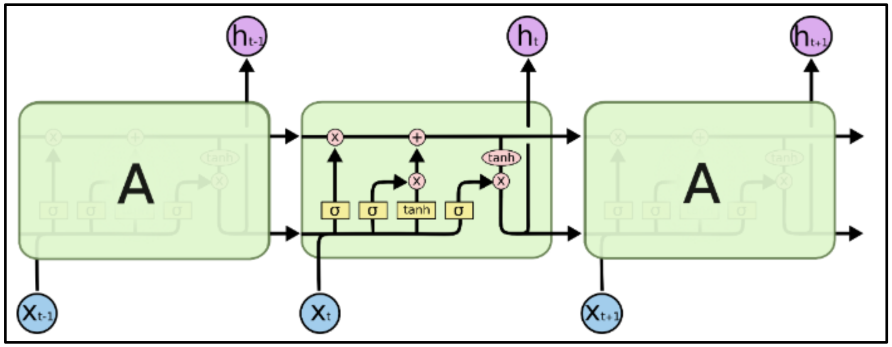
HSN



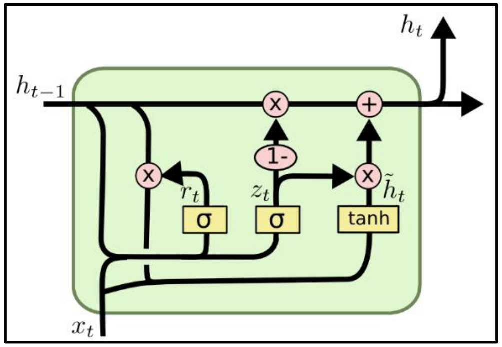
Laplacian2Mesh

Recurrent Network

LSTM[1]&GRU[2]



Long Short-Term Memory

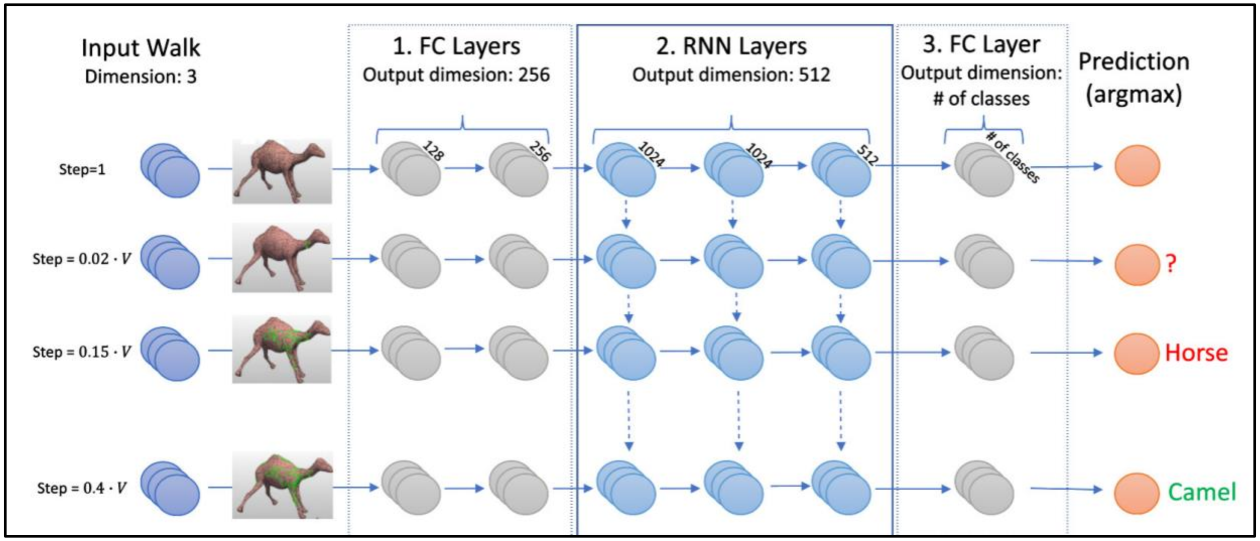


Gated Recurrent Unit

- LSTM: Utilize a series of interconnected units to construct a recurrent learning framework and avoid gradient anomaly problem
- GRU: Simplify the structure of LSTM and process simpler sequence data more efficiently
- Suitable for processing various sequence data-related tasks(e.g. Natural Language Processing(NLP))

[1] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[2] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arxiv preprint:1406.1078, 2014.

Recurrent Network



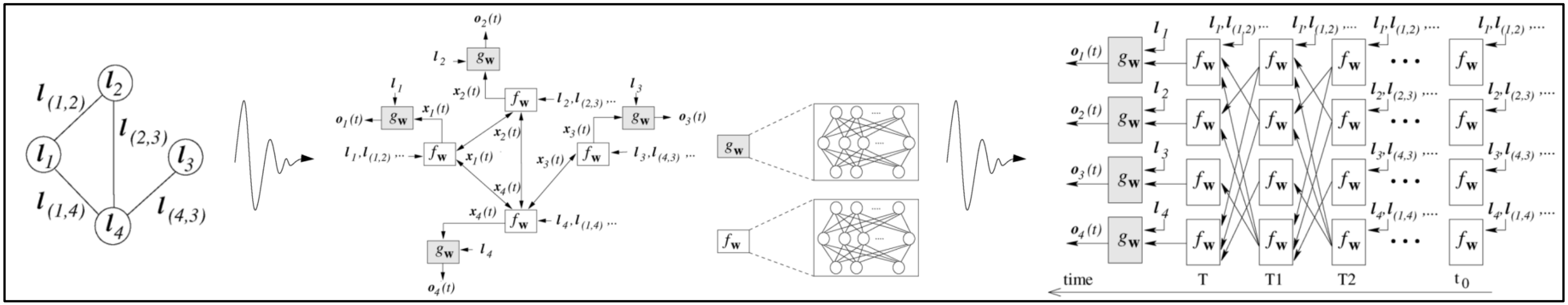
MeshWalker*

- Represent the mesh by random walks along the surface, which explore the mesh's geometry and topology
- Treat walk as sequence data and feed it into a GRU-based RNN that remembers the history of the walk
- Achieves top results for mesh-based classification and segmentation

* Lahav A, Tal A. Meshwalker: Deep mesh understanding by random walks[J]. ACM Transactions on Graphics (TOG), 2020, 39(6): 1-13.

Graph Network

GNN*

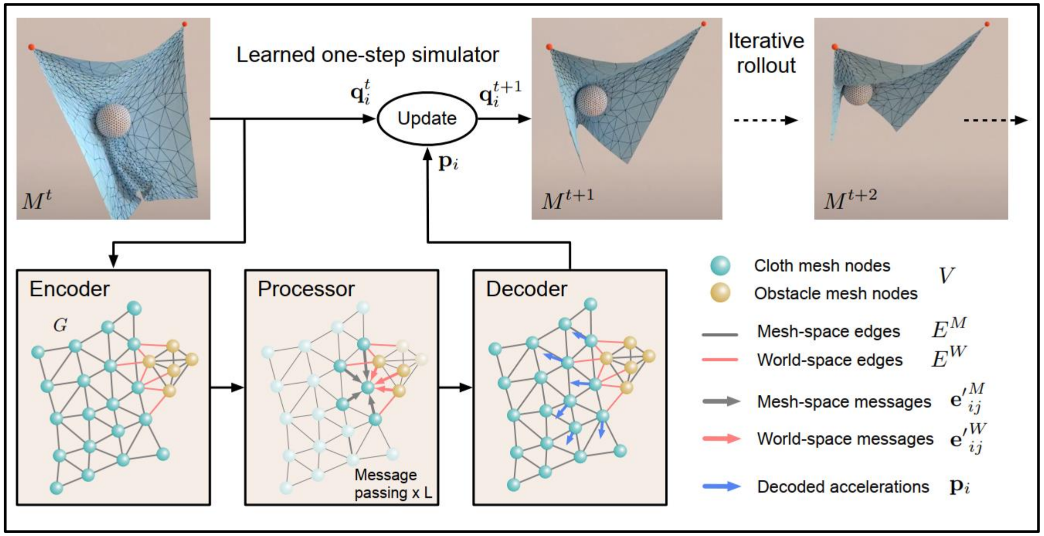


Graph Neural Network

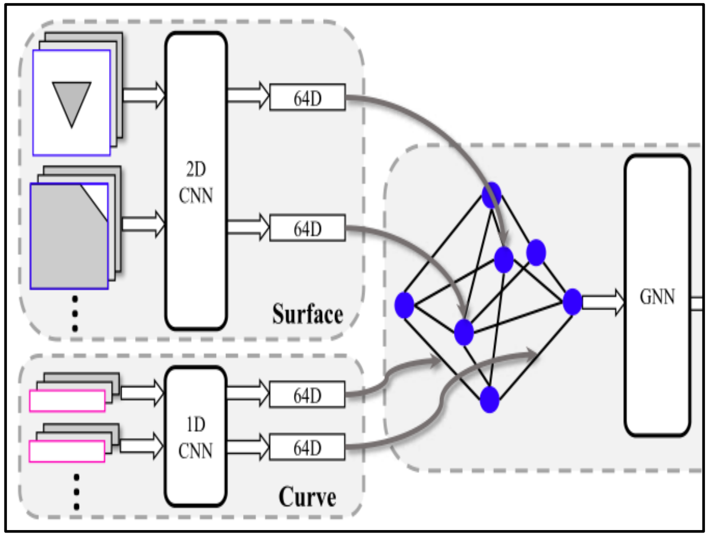
- Update nodes’ state and exchange information
- Solve the problem of losing topology information when preprocessing graph type data in traditional machine learning
- Suitable for various relationship-related tasks (e.g. knowledge graph, proteomics, social analysis)

* Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE transactions on neural networks, 2008, 20(1): 61-80.

Graph Network



MeshGraphNets[1]



UV-Net[2]

- Transform the mesh information into a graph
- Characterize the transfer of information (eg. geometry, physics) as updates of GNNs

[1] Pfaff T, Fortunato M, Sanchez-Gonzalez A, et al. Learning mesh-based simulation with graph networks[J]. arXiv preprint arXiv:2010.03409, 2020.
[2] Jayaraman, Pradeep Kumar and Sanghi, Aditya and Lambourne, Joseph and Willis, Karl and Davies, Thomas and Shayani, Hooman and Morris, Nigel. "Uv-net: Learning from boundary representations." CVPR. 2021.

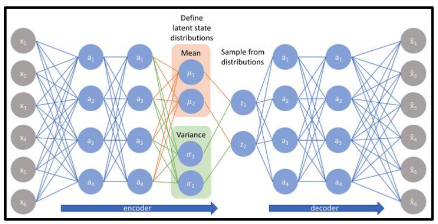
Outline

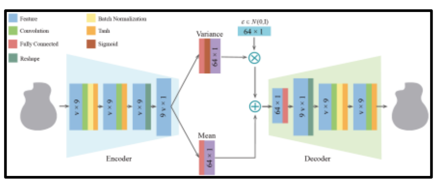
- **Research Background**
- **Intelligent Mesh Representation(IMR)**
- **Intelligence Mesh Generation (IMG)**
- **Intelligent Mesh Evaluation(IME)**
- **Summary and Outlook**

Intelligence Mesh Generation

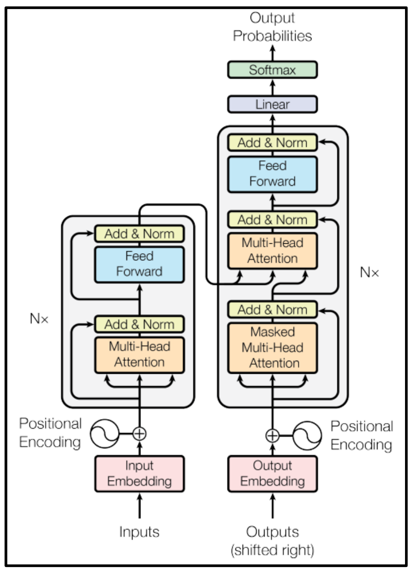
What kind of framework is suitable for mesh generation?

Variational Auto-Encoders

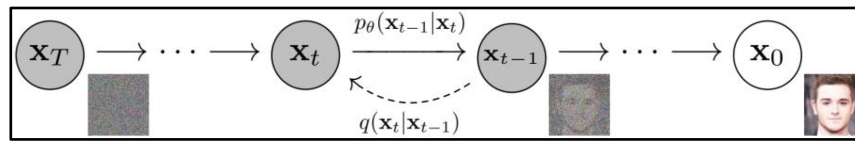




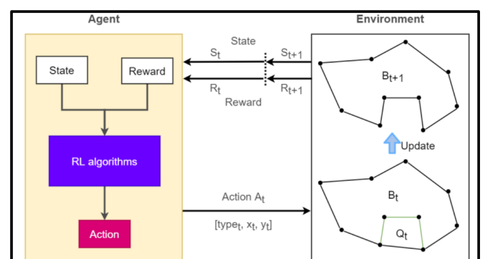
Transformer Architecture

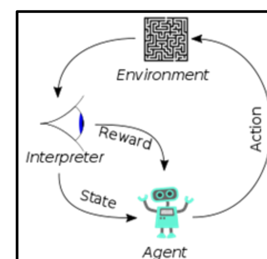


Diffusion Model



Reinforcement Learning





Framework for Mesh generation

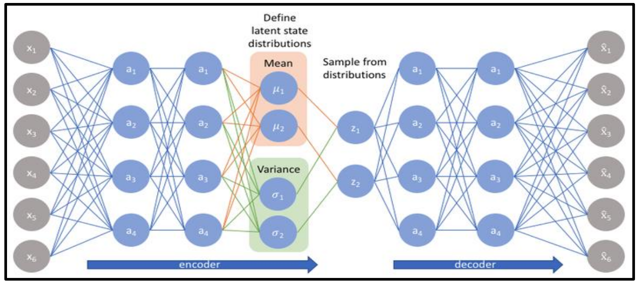
Variational Auto-Encoders

- Minimize the re-parameterized variational lower bound, to learn distribution of data
- Suitable for image generation task

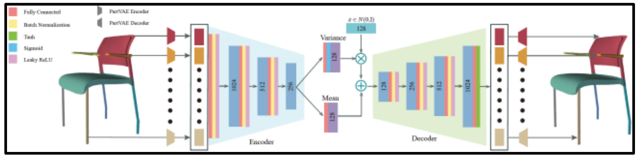
For mesh

- Generate mesh models with the same topology but different geometries (geometrically similar)
- The generation process requires only the input of the latent vectors and decoding them with the previously trained decoder

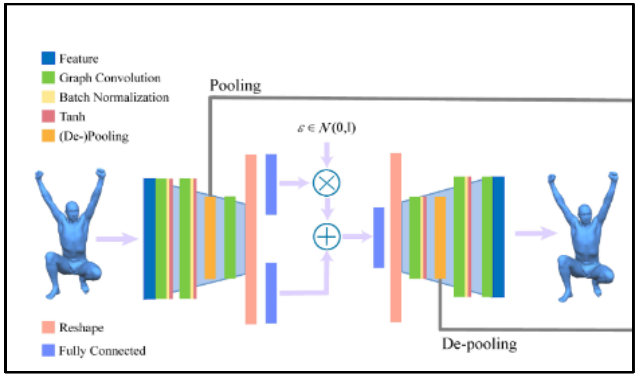
VAE



SP-VAE



MeshVAE



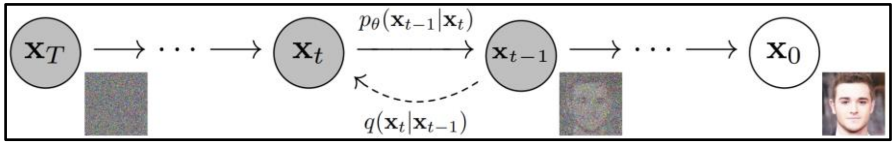
Framework for Mesh generation

Diffusion Model

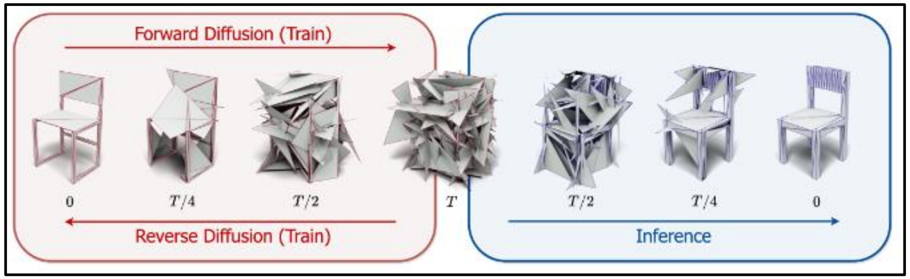
- Diffusion process involves gradually adding Gaussian noise to the original image.
- Denoising process gradually generates images through the learned Markov chain
- Suitable for image generation task

For mesh

- Learns the categorical distribution of the mesh data



Diffusion and Denoising Process



PolyDiff

Framework for Mesh generation

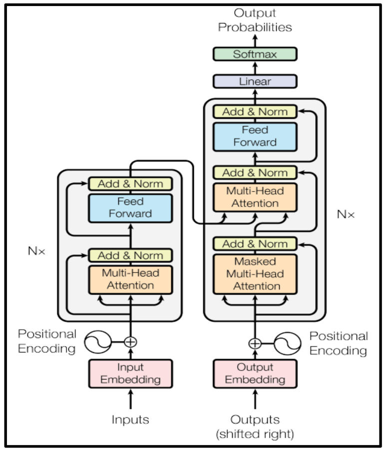
Transformer Architecture

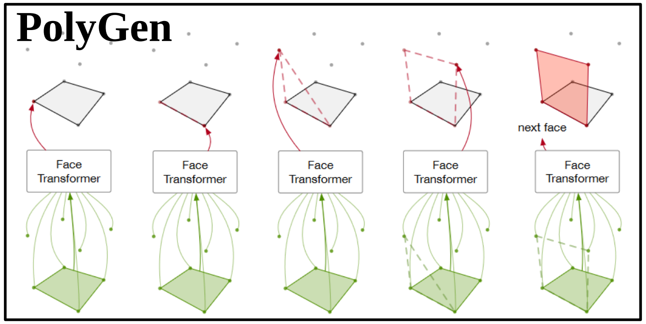
- Use self-attention mechanism to focus on information at different positions of sequence data
- Suitable for language generation task

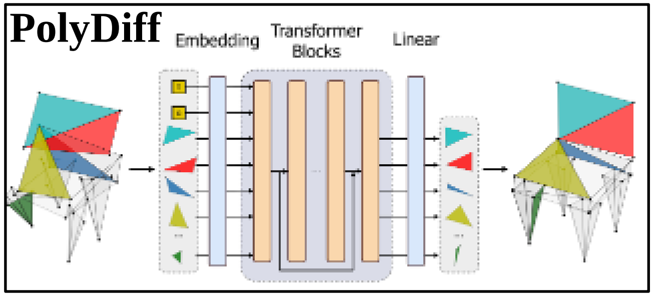
For mesh

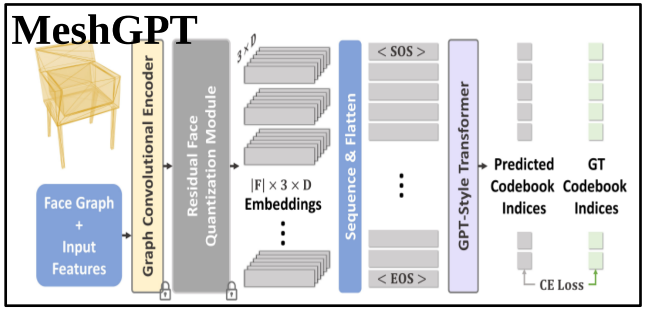
- During training, the mesh objects are viewed as ordered lists of points and faces
- The whole process can be viewed as conditional generation of sequence faces

Transformer





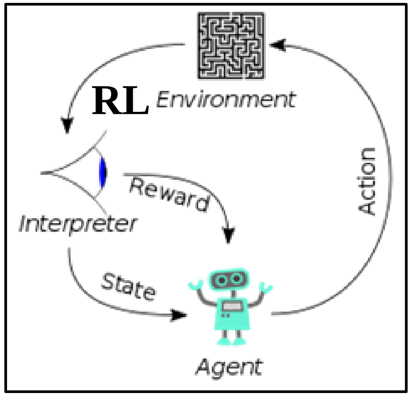




Framework for Mesh generation

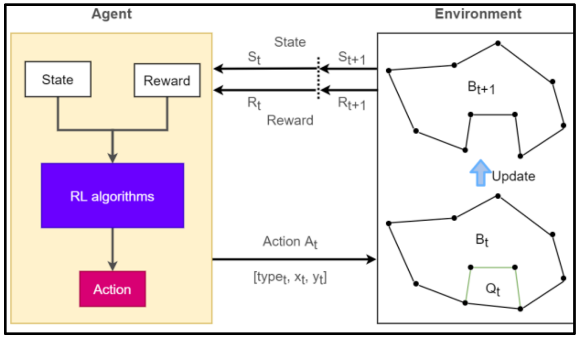
Reinforcement Learning

- Reinforcement Learning focuses on how software agents should take actions in an environment to maximize some notion of cumulative reward.



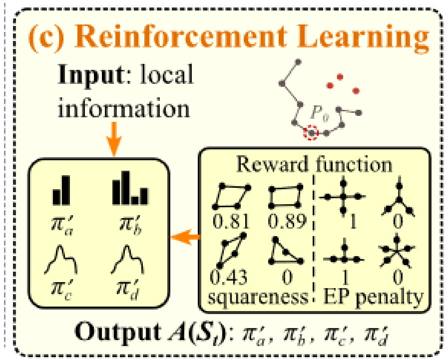
For mesh

RLQMG



- By formulating the mesh generation as a Markov decision process (MDP) problem, we are able to use a reinforcement learning algorithm

SRL-assisted AFM



Intelligence Mesh Generation

What role do intelligent frameworks play in mesh generation?

Mesh Generation Algorithm

- Advancing front Method
- Cross field Method
- Parameterization Method
- etc.

Full process intelligent
Partial process intelligent
Extended Input Data Types

Provide algorithmic framework
Provide loss and valuation criteria

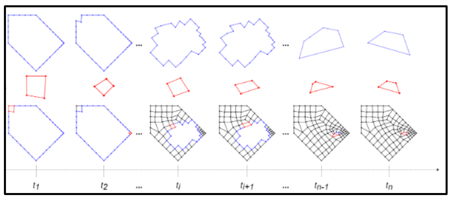
Intelligent Framework

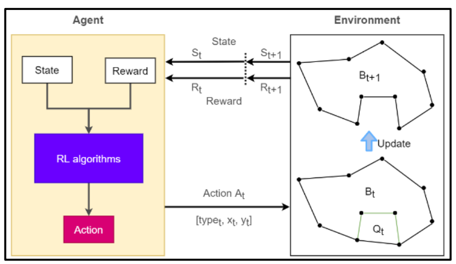
- Reinforcement Learning
- Variational Auto-Encoders
- Transformer Architecture
- etc.

The Role of Intelligent Methods in Mesh Generation

Full process intelligent mesh generation

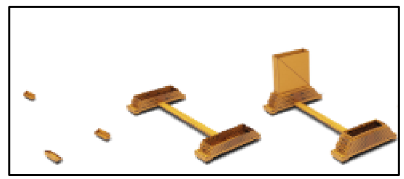
AFM Based IMG





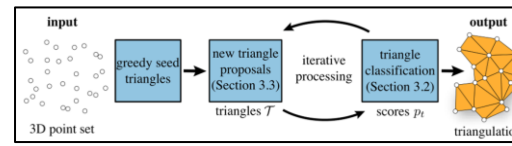
AFM using **Reinforcement Learning** Architecture

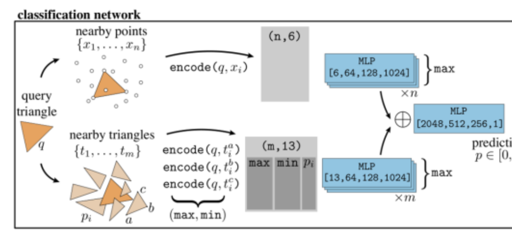
Sequencing based IMG



Representing mesh data as **sequenced data** to be processed using the **Transformer**

Classification based IMG



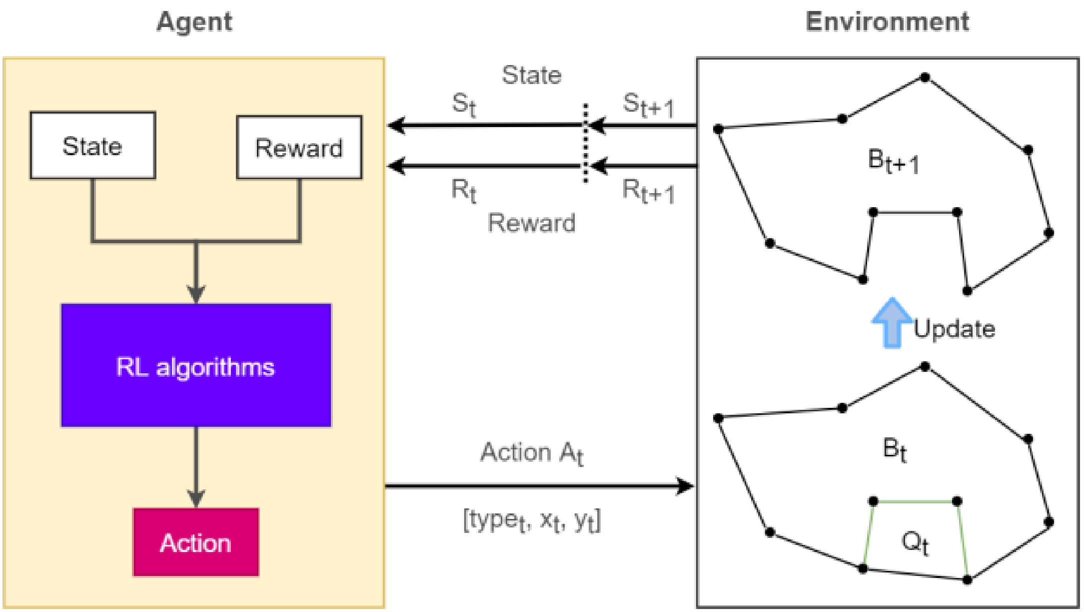


Estimate the existence or non-existence of points or surfaces

AFM-based IMG: RLQMG

➤ **Reinforcement learning for automatic quadrilateral mesh generation: a soft actor-critic approach**

- First, view mesh generation as a Markov decision process, given the initial boundary state S, the possible action set A, the reward R, and the initial state transition probability $P(S_{t+1}, R_{t+1} | S_t, R_t |)$;
- Based on the current state S_t, take action A_t, obtain the next state S_{t+1}, and calculate the corresponding reward R_t.
- Repeat the second step, using existing reinforcement learning techniques like soft actor-critic (SAC) to solve and implement mesh generation.



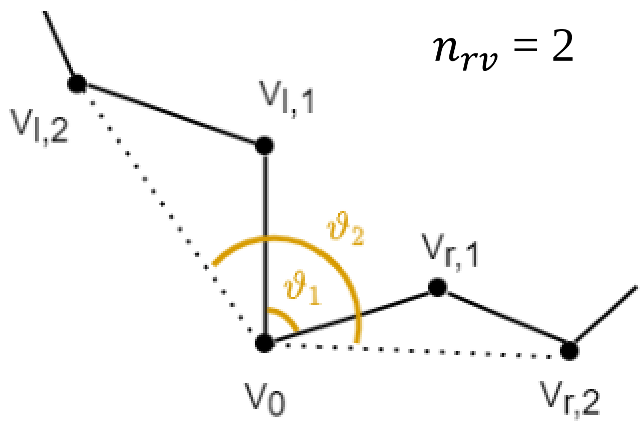
Pan, Jie and Huang, Jingwei and Cheng, Gengdong and Zeng, Yong. Reinforcement learning for automatic quadrilateral mesh generation: A soft actor–critic approach[J]. Neural Networks, 2023

AFM-based IMG: RLQMG

Selection of initial reference points.

$$V_i^* = \arg\min_{V_i} \frac{1}{n_{rv}} \sum_{j=1}^{n_{rv}} \angle V_{l,j} V_i V_{r,j}, i \in N_B,$$

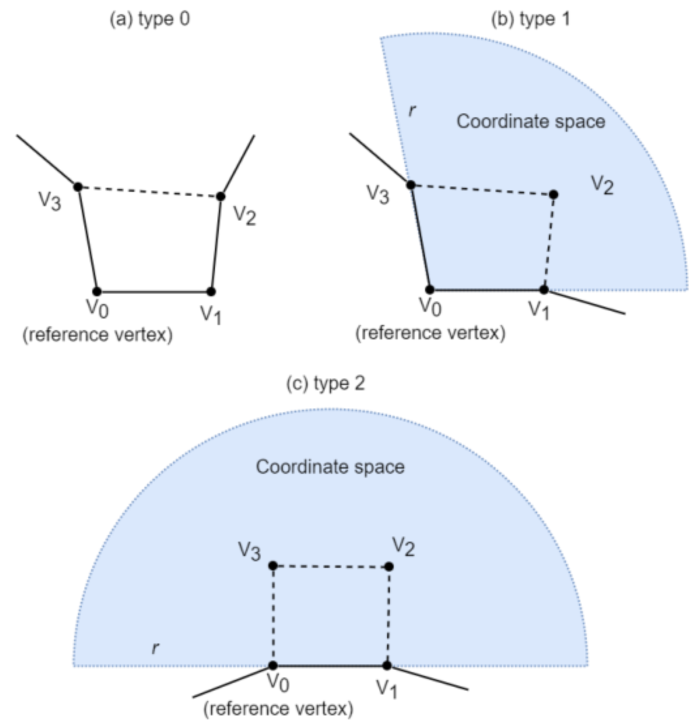
$n_{rv} = 2$



$$r = \alpha * L,$$

$$L = \frac{1}{2n} \sum_{j=0}^{n} |V_{l,j} V_{l,j+1}| + |V_{r,j} V_{r,j+1}|, 0 < n < N/2$$

Action Set A [type, V_1 , V_2]



The light blue area represents the sampling area for V_1 , V_2 with a length of r

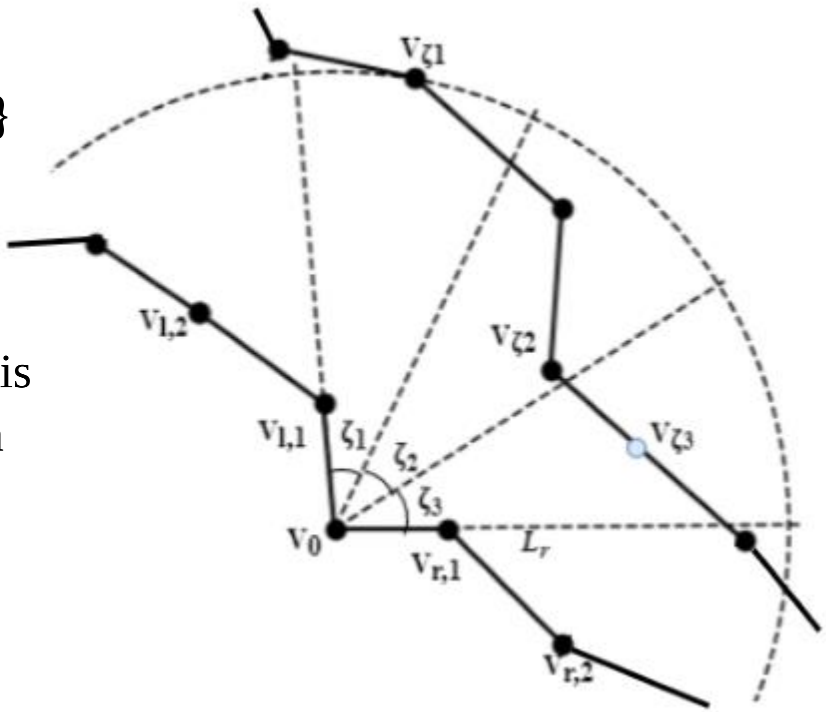
AFM-based IMG: RLQMG

State Set S

$$S_t = \{V_{ln}, \dots, V_{l1}, V_0, V_{r1}, \dots, V_{rn}, V_{\zeta 1}, \dots, V_{\zeta g}, \rho_t\}$$

$$L_r = \beta * L, \quad \beta = 4, n = 2, g = 3$$

V_0 is the reference point, V_l and V_r are the left and right boundary points, respectively, and V_ζ is the inner point closest to V_0 in each sector, with a sector radius of L_r



Reward Function R

$$r_t(s_t, a_t) = \begin{cases} -0.1, & \text{invalid element;} \\ 10, & \text{the element is the last element;} \\ m_t, & \text{otherwise.} \end{cases}$$

where $\quad m_t = \eta_t^e + \eta_t^b + \mu_t.$

AFM-based IMG: RLQMG

Element quality:

$$\eta_t^e = \sqrt{q^{edge} q^{angle}},$$

$$q^{edge} = \frac{\sqrt{2} min_{j\in\{0,1,2,3\}}\{l_j\}}{D_{max}},$$

$$q^{angle} = \frac{min_{j\in\{0,1,2,3\}}\{angle_j\}}{max_{j\in\{0,1,2,3\}}\{angle_j\}},$$

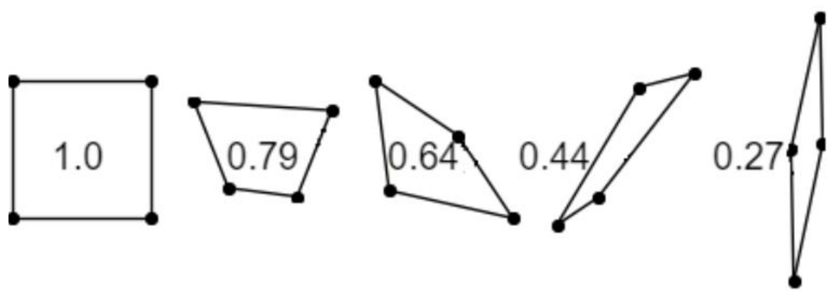
$$\mu_t = \begin{cases} -1, & \text{if } \mathcal{A}_t < \mathcal{A}_{min}; \\ \frac{\mathcal{A}_t - \mathcal{A}_{min}}{\mathcal{A}_{max} - \mathcal{A}_{min}}, & \text{if } \mathcal{A}_{min} \le \mathcal{A}_t < \mathcal{A}_{max}; \\ 0, & \text{otherwise.} \end{cases}$$

Boundary quality:

$$\eta_t^b = \sqrt{\frac{min_{k\in\{1,2\}}\{min(\varsigma_k, M_{angle})\}}{M_{angle}} q^{dist}} - 1,$$

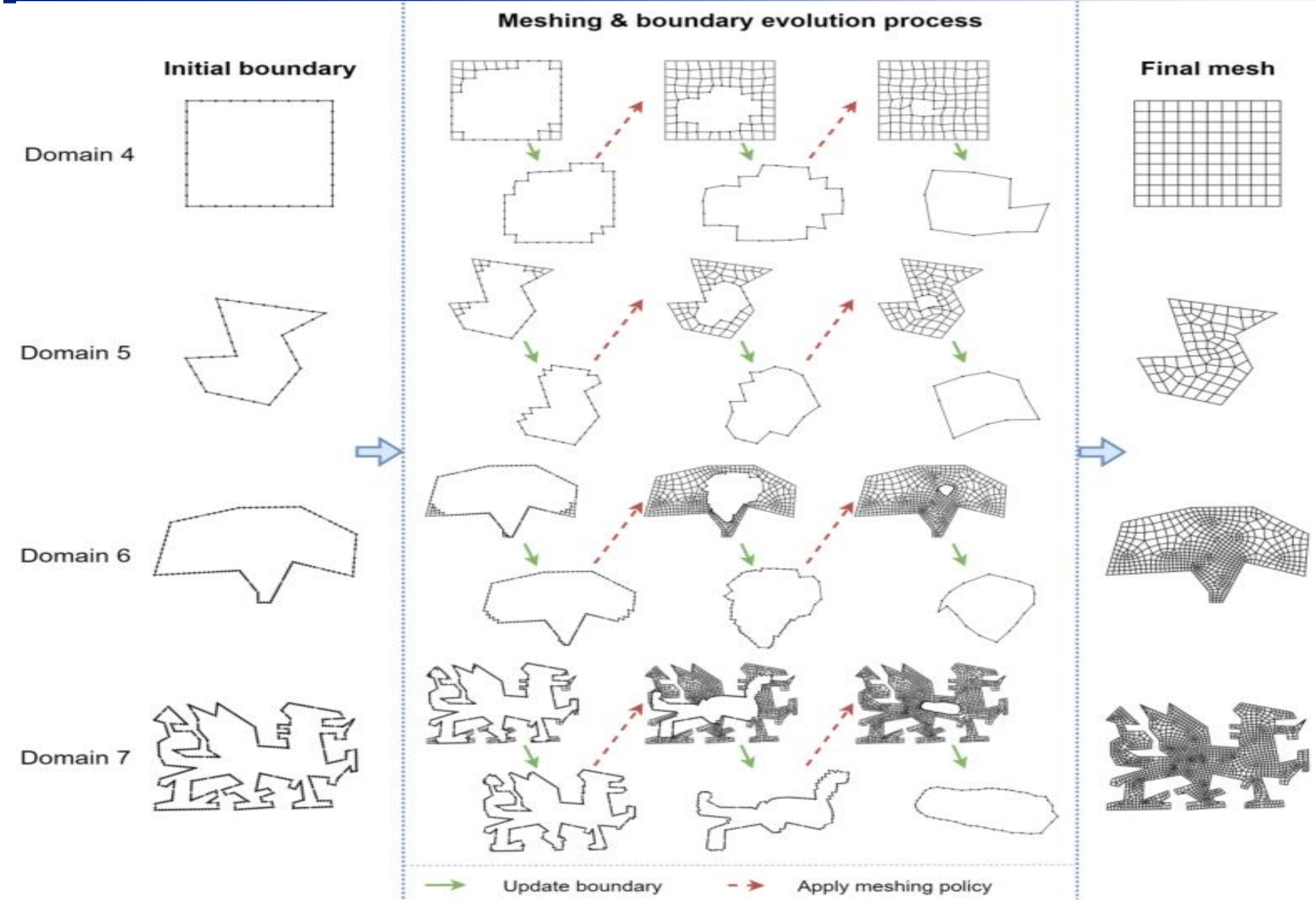
$$q^{dist} = \begin{cases} \frac{d_{min}}{(d_1+d_2)/2}, & \text{if } d_{min} < (d_1+d_2)/2; \\ 1, & \text{otherwise.} \end{cases}$$

D_{max} represents the maximum diagonal length of the quadrilateral, A_t represents the area of the quadrilateral, $M_{angle} = 60°$, and ς_k represents the angle of the newly generated facet.



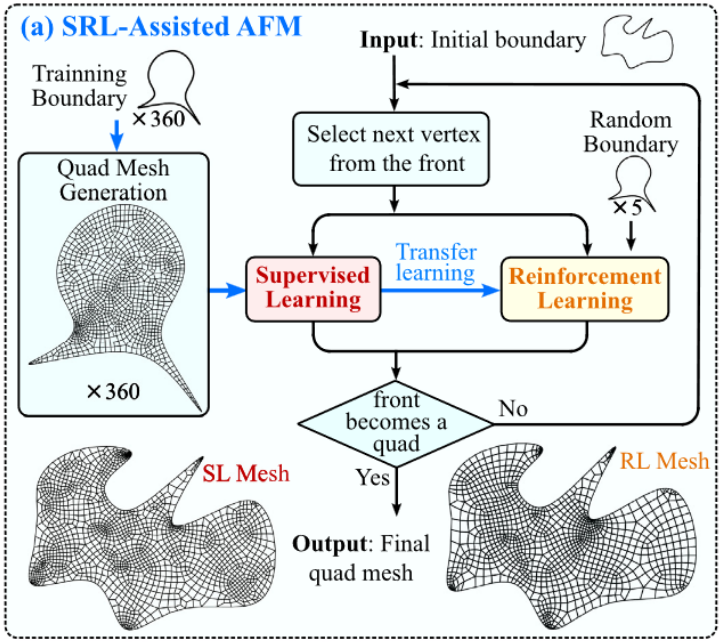
The corresponding reward value for each quadrilateral facet.

AFM-based IMG: RLQMG



AFM-based IMG: SRL-assisted AFM

- **SRL-assisted AFM: Generating planar unstructured quadrilateral meshes with supervised and reinforcement learning-assisted advancing front method**



Training: (Blue arrow)

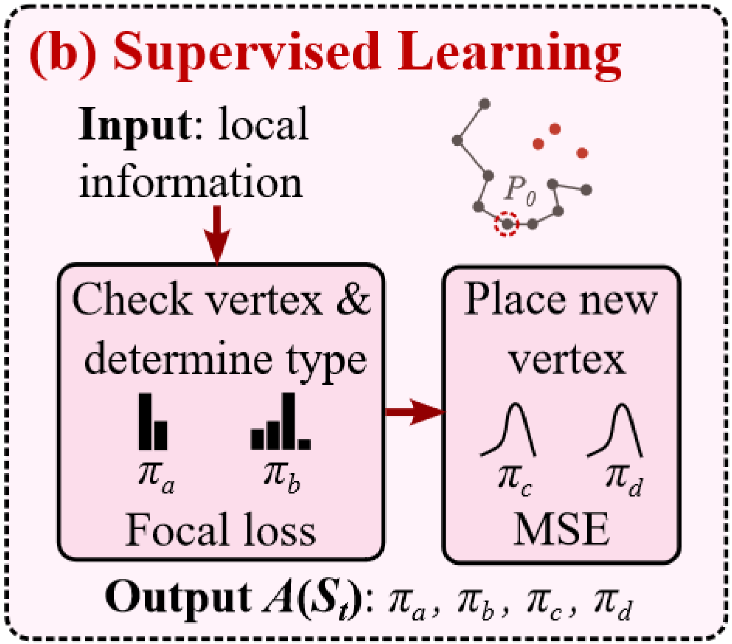
1. Input planar closed manifold boundaries
2. Use ANSYS to generate quad meshes for SL training
3. Transferring Supervised learning (SL) parameters to Reinforcement learning (RL)
4. Train Iteratively to improve mesh quality

Inference (Black arrow)

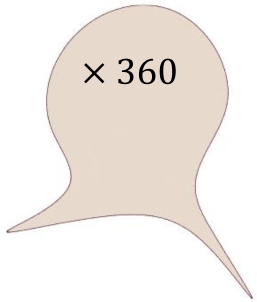
- Replace rule-based algorithms with policy neural networks (SL and RL)
- Generate high quality mesh with primitive feature

Hua Tong, Kuanren Qian, Eni Halilaj, Yongjie Jessica Zhang, SRL-assisted AFM: Generating planar unstructured quadrilateral meshes with supervised and reinforcement learning-assisted advancing front method, Journal of Computational Science,Volume 72,2023,102109.

AFM-based IMG: SRL-assisted AFM



Training data:



(1) Initial boundary (360 spline models):

(2) Set seed interval function $s(i)$:

$$s(i) = k(i)\rho(i), \qquad where\ \rho(i) \neq 0.$$

$\rho(i)$ represents curvature, $k(i) = \frac{l_{tol}}{s_a \sum_{i=1}^{N} \frac{l(i)}{\rho(i)}}$.

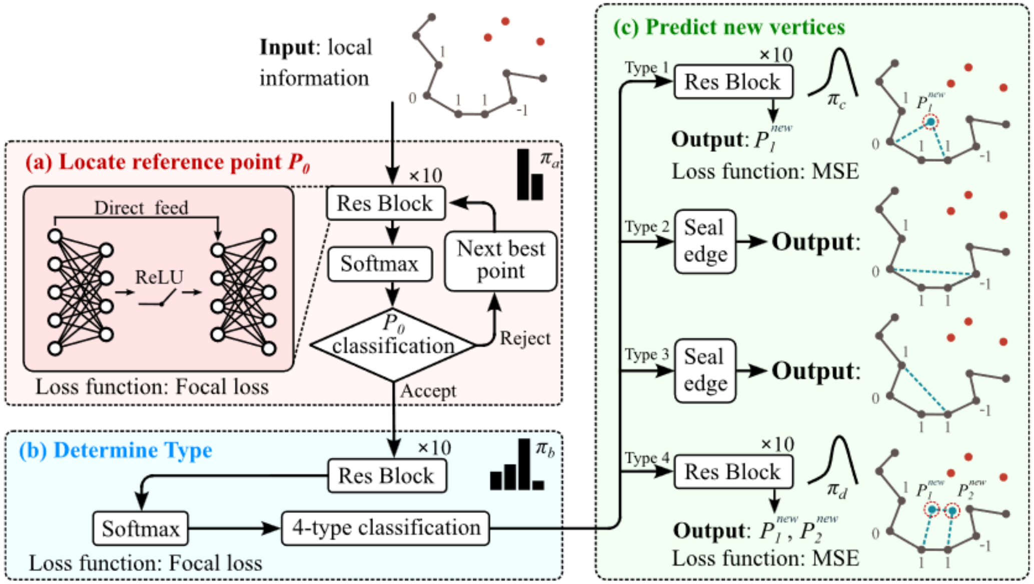
For sampling points by curvature

(3) Get local information for input:

Similar to above

(4) Calculation of supervisory signals

AFM-based IMG: SRL-assisted AFM



The role of neural networks:

π_a: The classification network determines the reference point P_0.

π_b: The classification network determines the type of action.

π_c: Network for predicting the location of a single point .

π_d:Network for predicting the position of two points.

Loss function:

For π_a and π_b

$$FocalLoss_{\pi_{a/b}} = -\frac{1}{N_{a/b}} \alpha_{a/b} \sum_{i=1}^{N_{a/b}} \left[(1-p^i)^{\gamma} \log p^i \right],$$

$N_{a/b}$ is training rows for π_a or π_b, $\alpha_{a/b}$ is a scaling factor of two/four classes.

For π_c and π_d

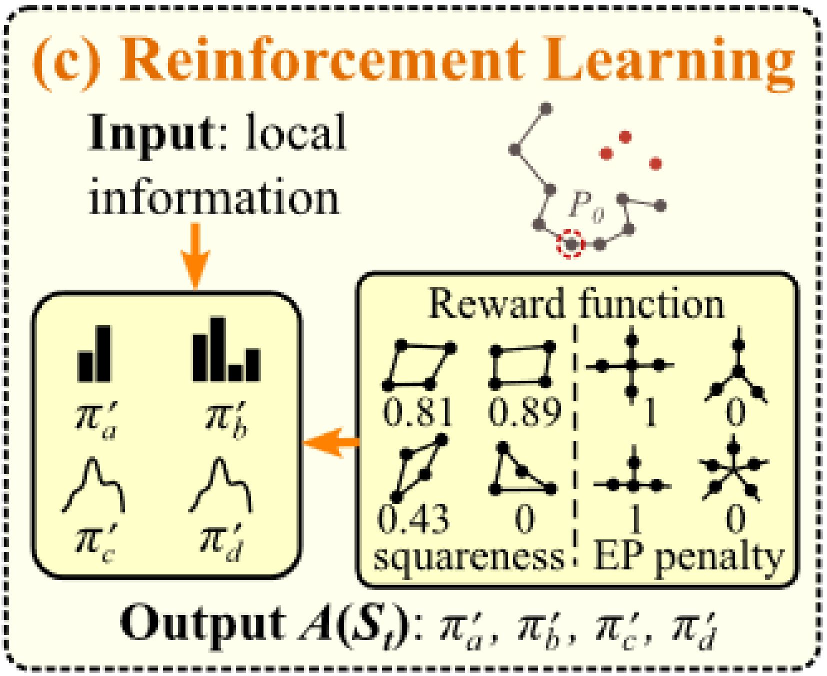
$$MSE_{\pi_c} = \frac{1}{N_c} \sum_{i=1}^{N_c} \left[(\theta_1^i - \hat{\theta}_1^i)^2 + (\rho_1^i - \hat{\rho}_1^i)^2 \right],$$

$$MSE_{\pi_d} = \frac{1}{N_d} \sum_{i=1}^{N_d} \left[(\theta_1^i - \hat{\theta}_1^i)^2 + (\rho_1^i - \hat{\rho}_1^i)^2 + (\theta_2^i - \hat{\theta}_2^i)^2 + (\rho_2^i - \hat{\rho}_2^i)^2 \right]$$

$N_{c/d}$ is training rows for π_c or π_d;
θ, ρ are normalized polar angle and polar radius;
$\hat{\theta}, \hat{\rho}$ are ground truth (given by ansys result).

AFM-based IMG: SRL-assisted AFM

Reward:



Squareness reward function

$$R^s = \sqrt[3]{\frac{\min\{\theta_1,\theta_2,\theta_3,\theta_4\}}{90^\circ}\left(2-\frac{\max\{\theta_1,\theta_2,\theta_3,\theta_4\}}{90^\circ}\right)\frac{\min\{l_1,l_2,l_3,l_4\}}{\max\{l_1,l_2,l_3,l_4\}}}.$$

EP penalty reward function

$$R^{ep} = \left(1-\frac{N_{ep}}{N_{tot}}\right)\left(1-\frac{N_{cep}}{N_{tot}}\right).$$

$$R^{fin} = \frac{1}{M}\sum_{i=1}^{M} R_i^s R_i^{ep} + \min\{R_1^s R_1^{ep}, R_2^s R_2^{ep}, \ldots, R_M^s R_M^{ep}\},$$

where M is the number of quads produced.

Noise:

$$A_t^{a/b} \sim \eta\pi'_{a/b}(S_t) + (1-\eta)Dir(\alpha^{a/b}),$$

$$A_t^{c/d} \sim \mathcal{N}\left(\pi'_{c/d}(S_t), \Sigma^{c/d}\right).$$

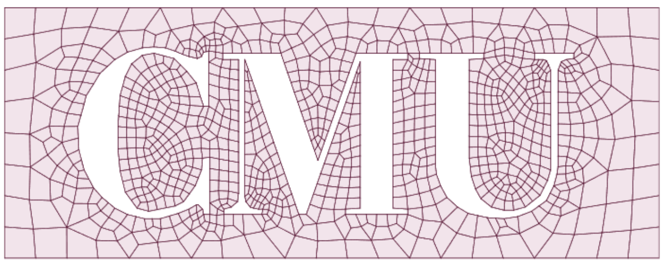
Introduce additional exploration to the neural network-guided action by adding noise to RL neural networks.

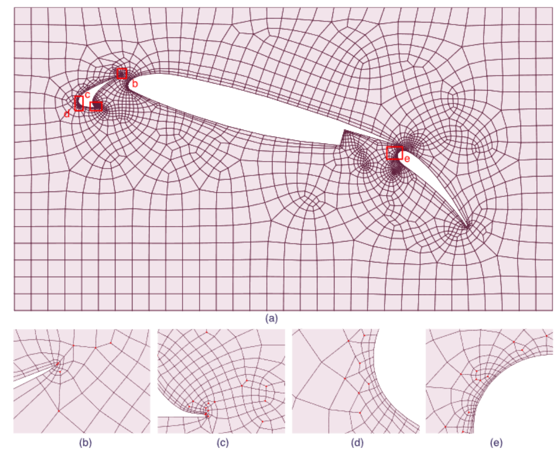
Structured metrics added to the reward function

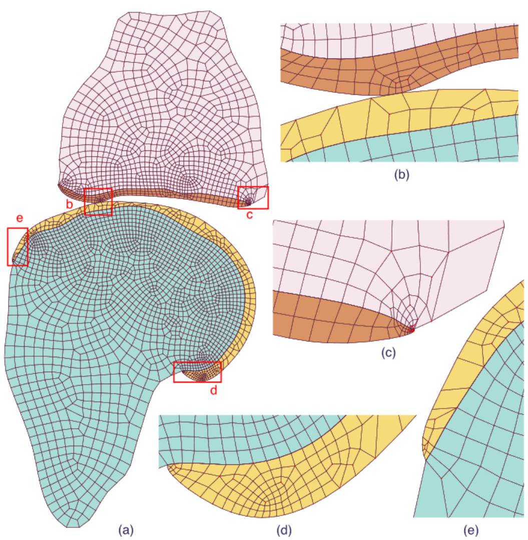
AFM-based IMG: SRL-assisted AFM

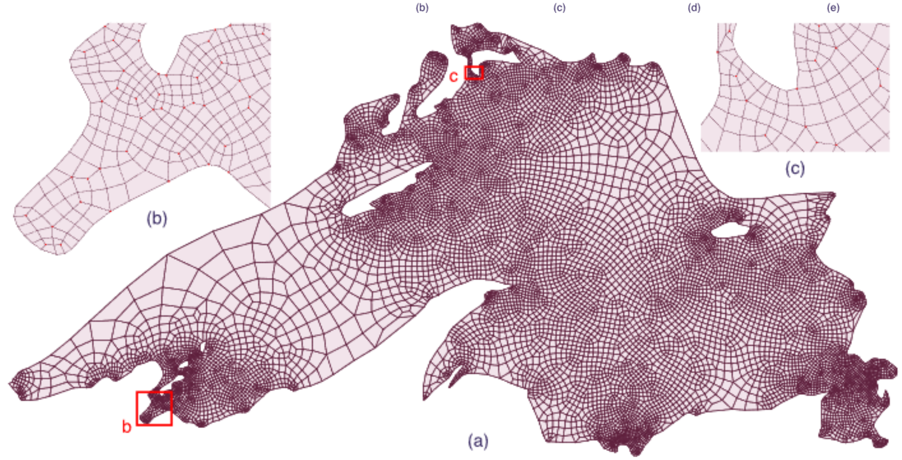
Domain	Mesh size [Vert#, Elem#]	Aspect ratio [Best, Worst]	Valence [EP, cEP[a]]	Angle [Min, Max]	Jacobian [Worst, Best]	Time (s)
Curve	[1361, 1420]	[1.0, 3.8]	[128, 88]	[35°, 148°]	[0.75, 1.0]	0.8
CMU Logo	[924, 1115]	[1.0, 3.5]	[111, 92]	[24°, 140°]	[0.68, 1.0]	0.4
Knee Joint	[2694, 2931]	[1.0, 2.4]	[210, 153]	[27°, 147°]	[0.72, 1.0]	8.1
Air Foil	[2782, 2953]	[1.0, 4.8][b]	[250, 167]	[29°, 150°]	[0.68, 1.0]	8.1
Lake Superior	[12, 150, 11, 618]	[1.0, 7.2]	[389, 223]	[15°, 156°]	[0.60, 1.0]	118.1

[a]cEP is the number of pairs of EPs that are adjacent to each other.
[b]After adding boundary layers, the worst aspect ratio becomes 16.0.

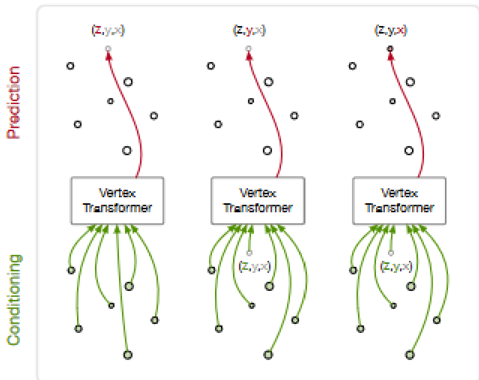




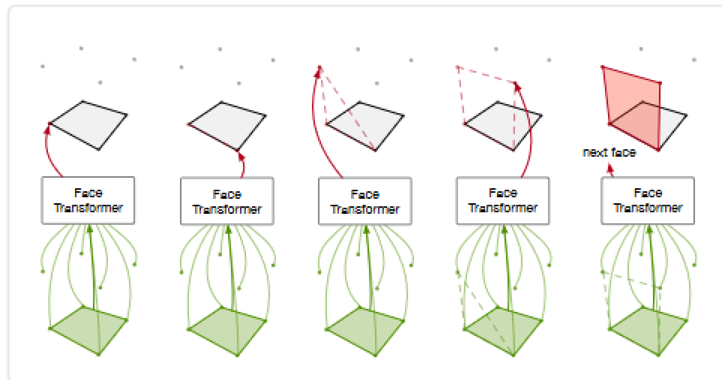




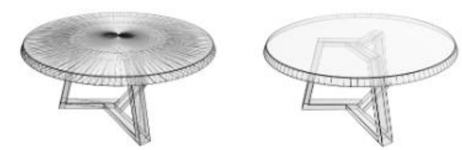
Sequencing based IMG: PolyGen



Vertex Model



Face Model



(a) Triangle mesh (b) n-gon mesh

$$p(\mathcal{V}^{\text{seq}};\theta) = \prod_{n=1}^{N_V} p(v_n | v_{<n};\theta)$$

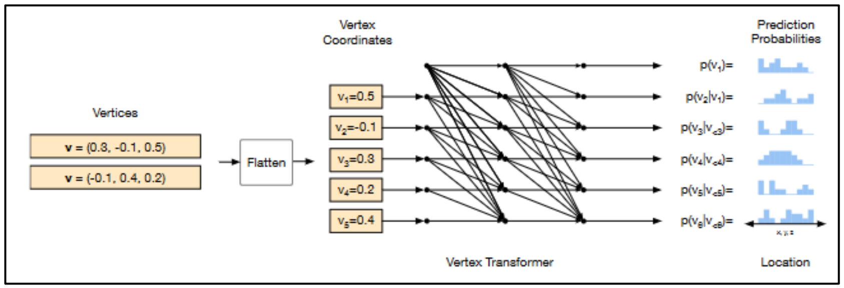
$$p(\mathcal{F}^{\text{seq}}|\mathcal{V};\theta) = \prod_{n=1}^{N_F} p(f_n | f_{<n}, \mathcal{V};\theta)$$

$$\begin{aligned} p(\mathcal{M}) &= p(\mathcal{V}, \mathcal{F}) \\ &= p(\mathcal{F}|\mathcal{V})p(\mathcal{V}) \end{aligned}$$

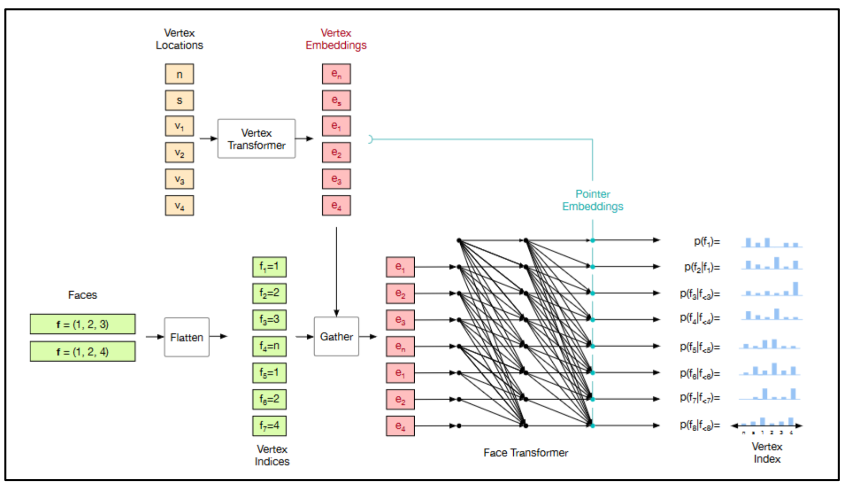
- Focus on polygon mesh generation.
- Both of vertex and face models are Transformer based Autoregressive.
- To generate a mesh, first sample the vertex model, and then pass the resulting vertices as input to the face model, from sample faces.
- In addition, conditional generation is also possible, such as mesh class identity, an input image, or a voxelized shape.

Nash C, Ganin Y, Eslami S M A, et al. Polygen: An autoregressive generative model of 3d meshes[C]//International conference on machine learning. PMLR, 2020: 7220-7229.

Sequencing based IMG : PolyGen



The Vertex Transformer outputs discrete distributions over the individual coordinate locations, as well as the stopping token *s*.

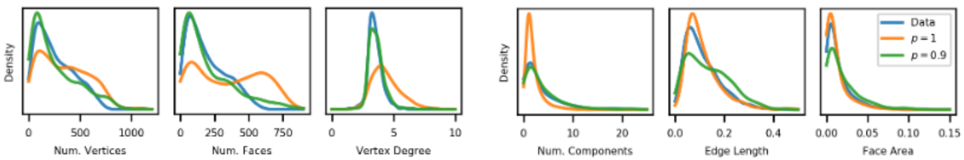


The Face Transformer outputs pointer embeddings which are compared to the vertex embeddings using a dot-product to produce the desired distributions.

- PolyGen first generates mesh vertices, and then generates mesh faces conditioned on those vertices.
- Vertices are generated sequentially from lowest to highest on the vertical axis.
- To generate the next vertex the current sequence of vertex coordinates is passed as context to a vertex Transformer, which outputs a predictive distribution for the next coordinate.
- The face model takes as input a collection of vertices, and the current sequence of face indices, and outputs a distribution over vertex indices.

Sequencing based IMG: PolyGen

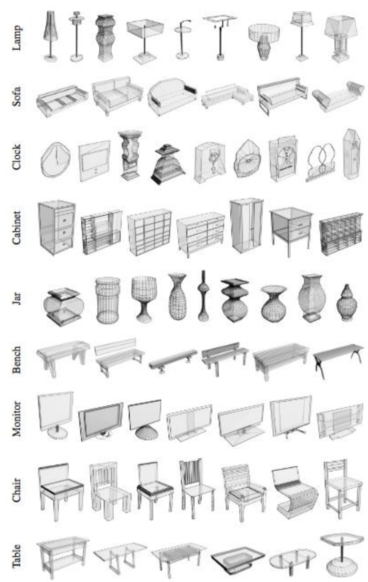
Model	Bits per vertex: Vertices	Bits per vertex: Faces	Accuracy: Vertices	Accuracy: Faces
Uniform	24.08	39.73	0.004	0.002
Valid predictions	21.41	25.79	0.009	0.038
Draco* (Google)	Total: 27.68		-	-
PolyGen	2.46	1.79	0.851	0.900
- valid predictions	2.47	1.82	0.851	0.900
- discr. embed. (V)	2.56	-	0.844	-
- data augmentation	3.39	2.52	0.803	0.868
+ cross attention (F)	-	1.87	-	0.899



Distribution of mesh statistics for unconditional samples from our model and the ShapeNet test set. We compare samples generated using with nucleus sampling and top-p = 0.9, to true model samples (p = 1).

Image condition

Voxel condition



Class condition

Random unconditional samples

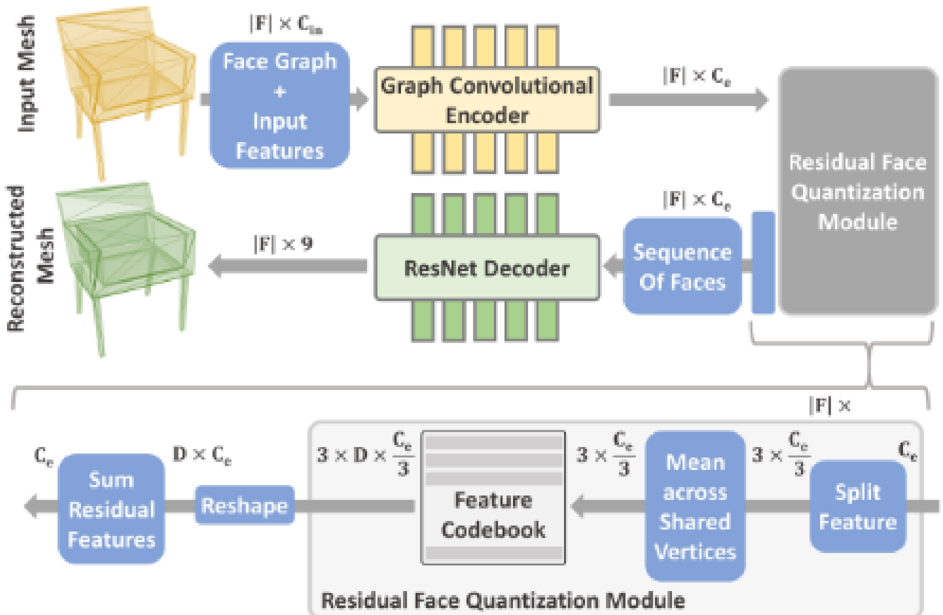
Sequencing based IMG: MeshGPT

➢ **MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers**

- Utilizing graph convolutional neural network and to learn a vocabulary of geometric embedding from a large dataset.
- Training a decoder-only transformer (GPT) for mesh generation.

Siddiqui Y, Alliegro A, Artemov A, et al. MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers[J]. CVPR 2024.

Sequencing based IMG: MeshGPT



All of the above are trained with reconstruction loss and commit loss.

$$\mathcal{L}_{\text{recon}} = \sum_{n=1}^{N} \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{128} \mathrm{w}_{nijk} \log \mathcal{P}_{nijk}$$

$$\mathcal{L}_{\text{commit}}(\mathbf{z}, \hat{\mathbf{z}}) = \sum_{d=1}^{D} \|\mathbf{z} - \mathrm{sg}[\hat{\mathbf{z}}^{(d)}]\|_2^2$$

(1)Utilizing GNN on the dual graphs of the mesh to encode faces .

$$\mathbf{Z} = (z_1, z_2, \ldots, z_N) = E(\mathcal{M}),$$

(2)These features are then quantized into codebook embeddings using residual quantization.

$$\mathbf{T} = (t_1, t_2, \ldots, t_N) = \mathrm{RQ}(\mathbf{Z}; \mathcal{C}, D),$$

$$t_i = (t_i^1, t_i^2, \ldots, t_i^D),$$

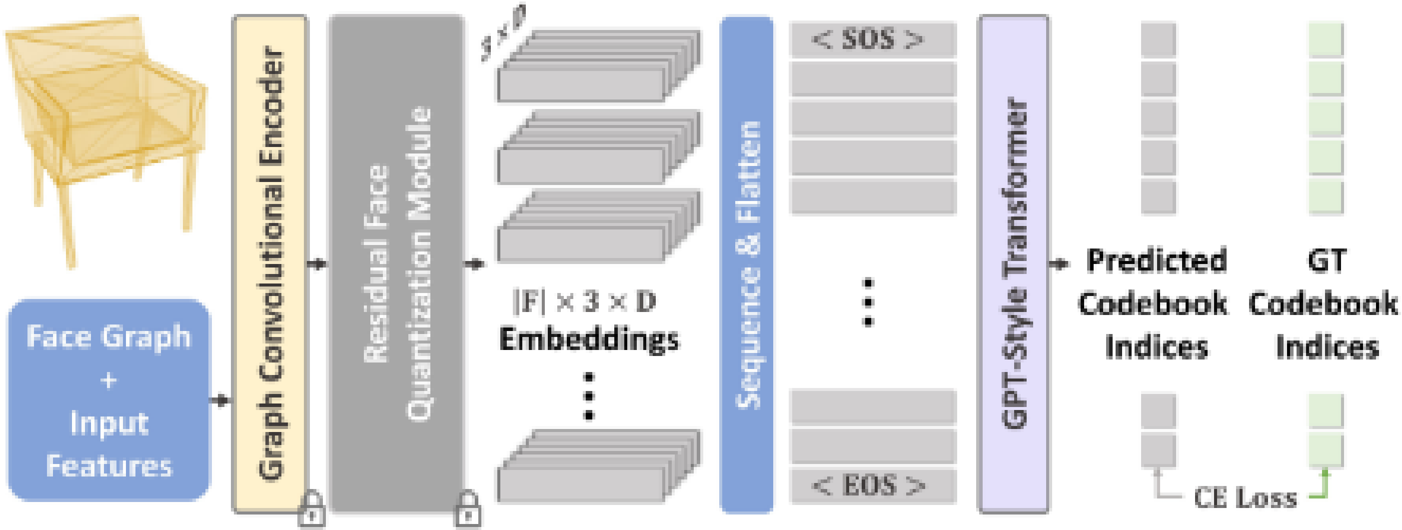
Vertex feature to face feature

$$\hat{\mathbf{Z}} = (\hat{z}_1, \ldots, \hat{z}_N), \text{ with } \hat{z}_i = \oplus_{v=0}^{2} \sum_{d=1}^{\frac{D}{3}} \mathrm{e}(t_i^{3.v+d}).$$

(3)Decoding the quantized embeddings through a 1D Resnet.

$$\hat{\mathcal{M}} = G(\hat{\mathbf{Z}})$$

Sequencing based IMG: MeshGPT



The same token decoder is used to generate the mesh

$$\hat{\mathcal{M}} = G(\hat{\mathbf{Z}})$$

Needs post-processing to remove duplicate points.

This transformer decoder predicts the subsequent codebook index for each embedding, optimized via cross-entropy loss.

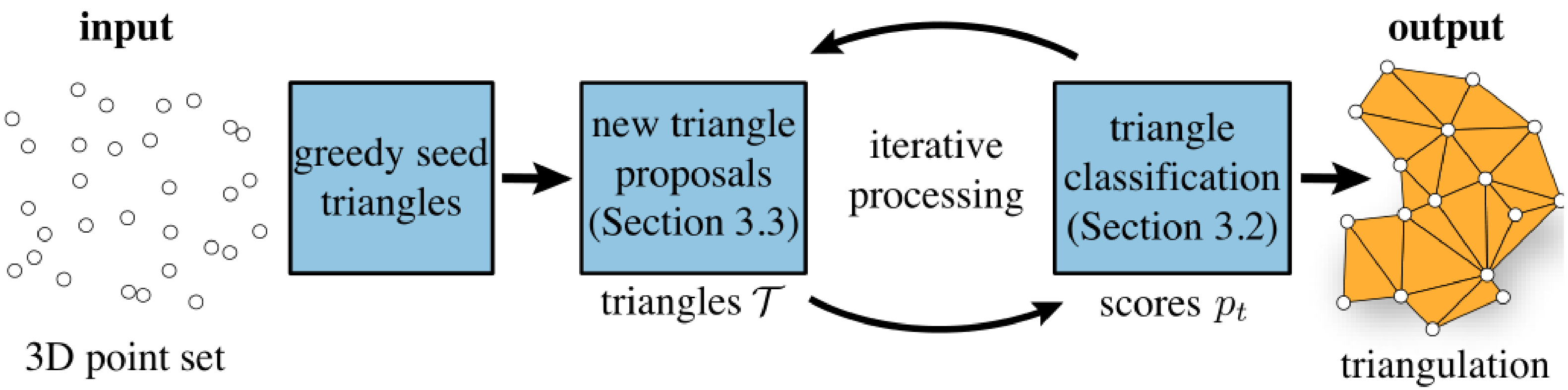
$$\mathcal{L}_{recon} = -\sum_{i=1}^{N}\sum_{j=1}^{D}\sum_{k=1}^{|C|} logp\left(s_i^k = t_i^j\right)$$

Sequencing based IMG: MeshGPT

Class	Method	COV↑	MMD↓	1-NNA	FID↓	KID↓	|V|	|F|
Chair	AtlasNet [18]	9.03	4.05	95.13	170.71	0.169	2500	4050
	BSPNet [7]	16.48	3.62	91.75	46.73	0.030	673	1165
	Polygen [43]	31.22	4.41	93.56	61.10	0.043	248	603
	GET3D [14]	40.85	3.56	83.04	81.45	0.054	13725	27457
	GET3D*	38.75	3.57	84.07	78.29	0.065	199	399
	MeshGPT	**43.28**	**3.29**	**75.51**	**18.46**	**0.010**	125	228
Table	AtlasNet [18]	7.16	3.85	96.30	161.38	0.150	2500	4050
	BSPNet [7]	16.83	3.14	93.58	30.78	0.017	420	699
	Polygen [43]	32.99	3.00	88.65	38.53	0.029	147	454
	GET3D [14]	41.70	2.78	85.54	93.93	0.076	13767	27537
	GET3D*	37.95	2.85	81.93	50.46	0.037	199	399
	MeshGPT	**45.68**	**2.36**	**72.88**	**6.24**	**0.002**	99	187
Bench	AtlasNet [18]	20.53	2.47	90.58	189.39	0.163	2500	4050
	BSPNet [7]	28.74	2.05	88.44	59.11	0.030	457	756
	Polygen [43]	51.92	1.97	76.98	49.34	0.031	172	430
	MeshGPT	**55.23**	**1.44**	**68.24**	**8.72**	**0.001**	159	291
Lamp	AtlasNet [18]	19.97	4.68	91.85	177.91	0.139	2500	4050
	BSPNet [7]	18.38	5.32	93.13	112.65	0.077	587	1011
	Polygen [43]	47.86	4.18	81.42	52.48	0.025	185	558
	MeshGPT	**53.88**	**3.94**	**65.73**	**19.91**	**0.004**	150	288

Classification-based IMG: PointTriNet

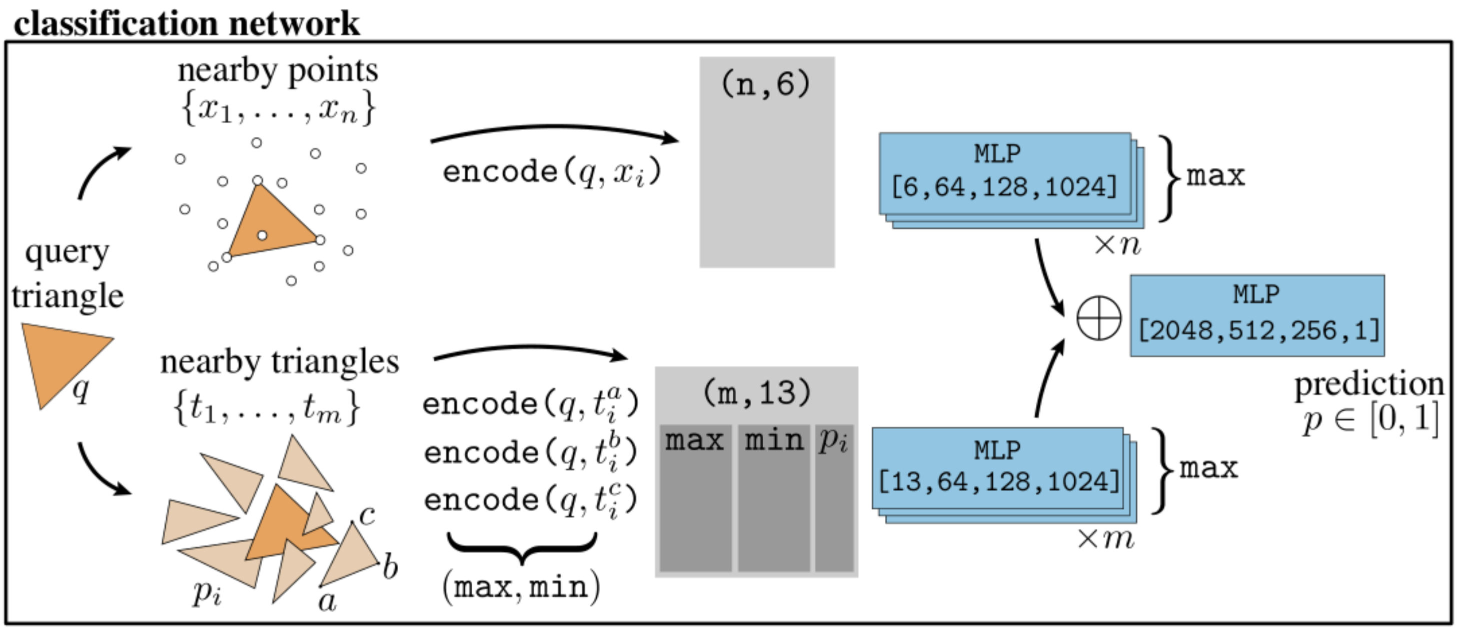
➢ **PointTriNet: Learned Triangulation of 3D Point Sets**



- The initial candidate triangular facets are first constructed by selecting the two nearest neighbors centered on each vertex;
- The probability of candidate triangles appearing is predicted using a PointNet-based classification network, while another PointNet-based proposal network is used to give the candidate triangul for the next step;
- Repeat the second step five times, leaving at the end the triangular with probability greater than a given threshold.

Sharp, Nicholas and Ovsjanikov, Maks. Pointtrinet: Learned triangulation of 3d point sets[C]. ECCV 2020

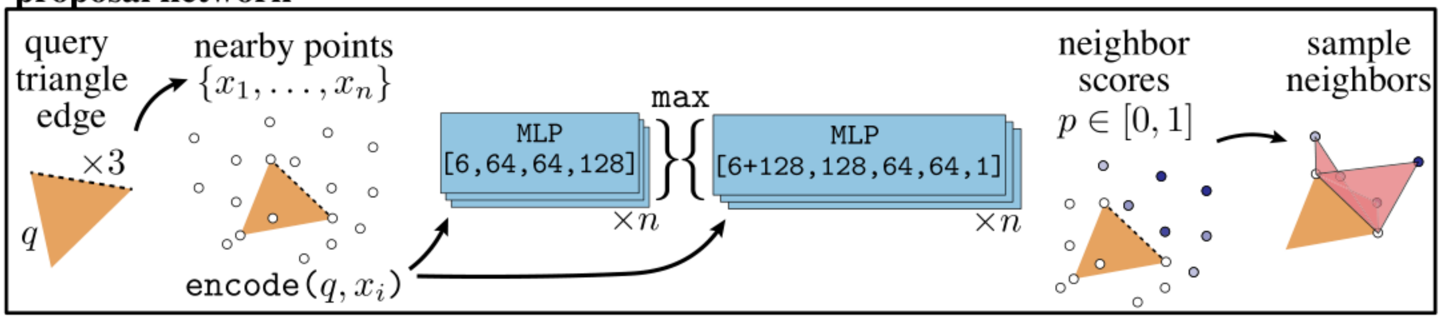
Classification-based IMG: PointTriNet



- encode(t, p) → $[x', y', z', u, v, w]$. Rule-based encoding of information about the triangles to be queried and neighboring triangles into a 6-dimensional vector
- t^a, t^b, t^c refers to the three vertices of the neighboring triangle
- **max** and **min** are obtained on all neighboring triangles *t*

Classification-based IMG: PointTriNet

proposal network

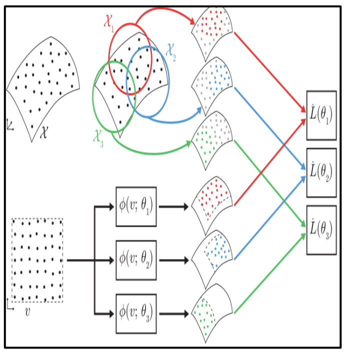


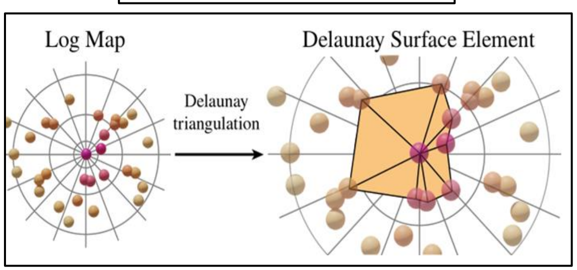
input point set　learned triangulation　+ learned vertex offset　+ fill small holes

The Role of Intelligent Methods in Mesh Generation

Partial Process Intelligent Mesh Generation method

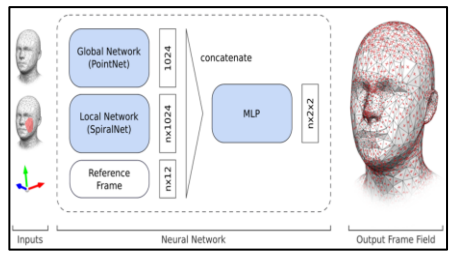
Parameterization Based IMG





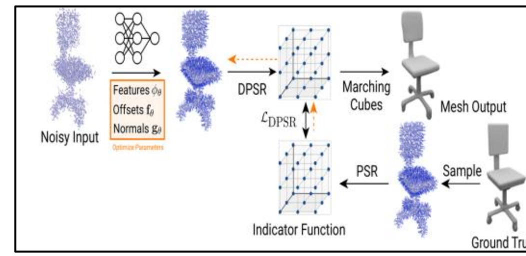
Network learning parameter mapping

Cross field based IMG



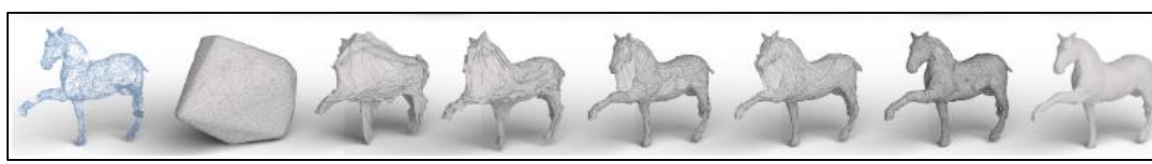
Learn cross field to generate meshes

Iso-surfaces based IMG



Network learning iso-surfaces and combining Marching cube to build meshes.

Deform Based IMG

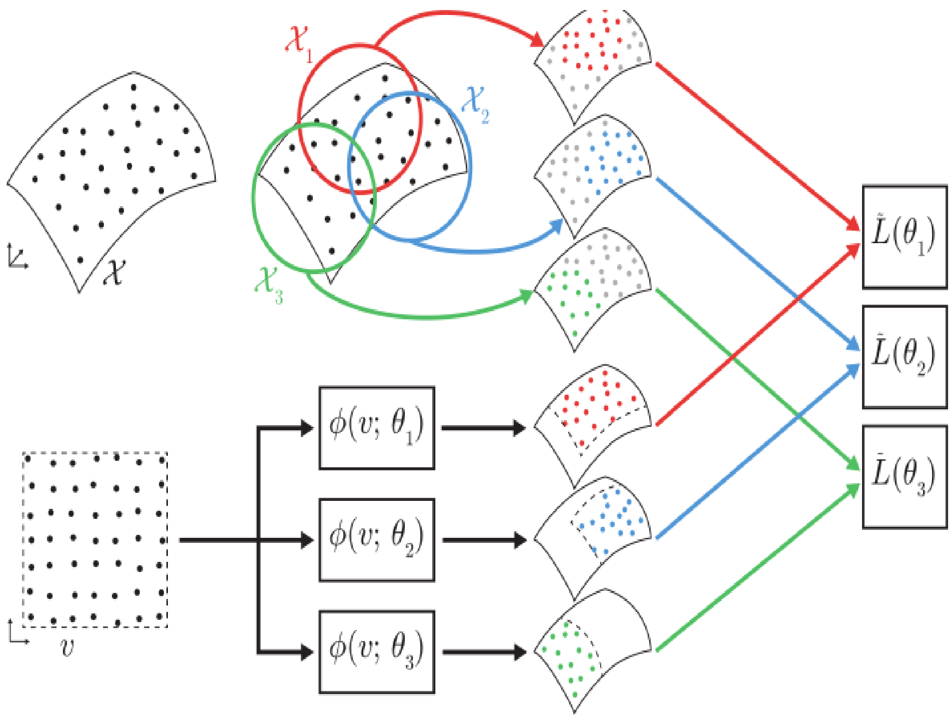


This class method is an intelligent extension to 3D Alpha Wrapping

Parameterization-based DGP

➢ Deep Geometric Prior for Surface Reconstruction

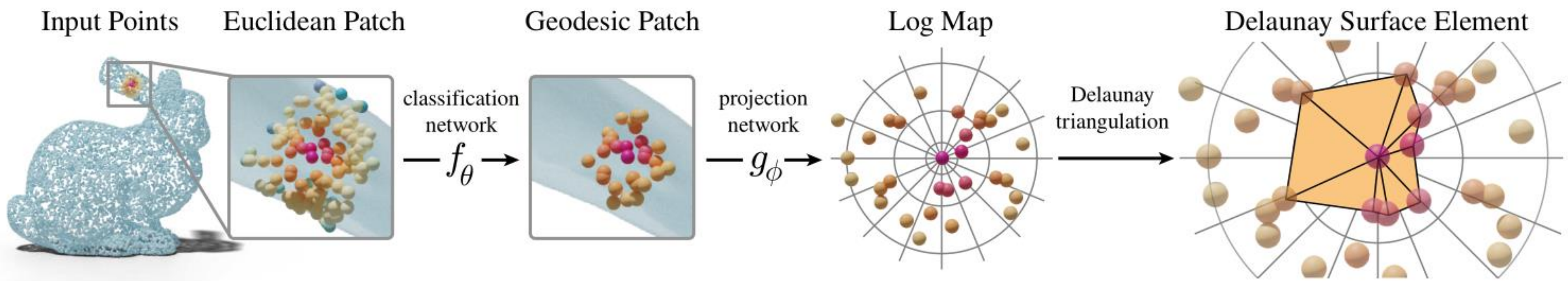
- First, the input point cloud is divided into overlapping local blocks χ;
- The neural network $\boldsymbol{\phi}$ learns the 2D to 3D inverse parametrization mapping guided by the minimization of the Wasserstein distance.
- By minimizing the error between different local blocks' overlapping areas, these parametrizations are made consistent with each other in their overlapping portions.
- Once the local mappings are established, the mesh on the 2D plane is correspondingly mapped onto the target surface.



Williams F, Schneider T, Silva C, et al. Deep geometric prior for surface reconstruction[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 10130-10139.

Parameterization based IMG: DSE

➢ **Learning Delaunay Surface Elements for Mesh Reconstruction**



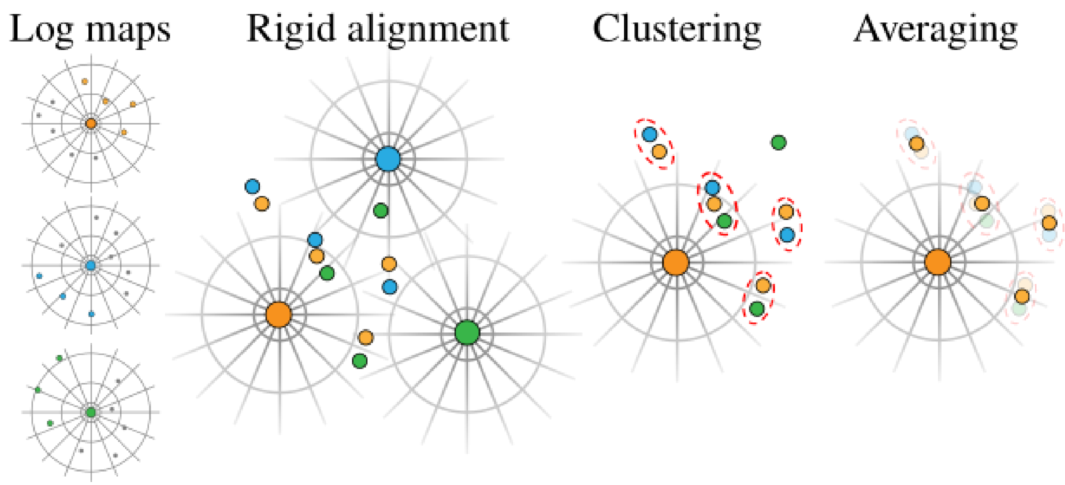
- Initially, a neighborhood of 120 points is constructed around each point p_{i},after which a classification network f_θ is trained to select the closest 30 points in terms of geodesic distance, resulting in local blocks P_{i} for each point.
- Subsequently, another network g_ϕ computes the logarithmic map U_{i} for each P_{i} ,which represents the Euclidean coordinates with the central point p_{i}as the origin.
- This is followed by local alignment of U_{i} to enhance consistency between neighboring U_{i};
- A triangulation is performed on all U_{i} resulting in corresponding Delaunay Surface Elements (DSEs), from which triangles belonging to three DSEs are selected to compose the final mesh.

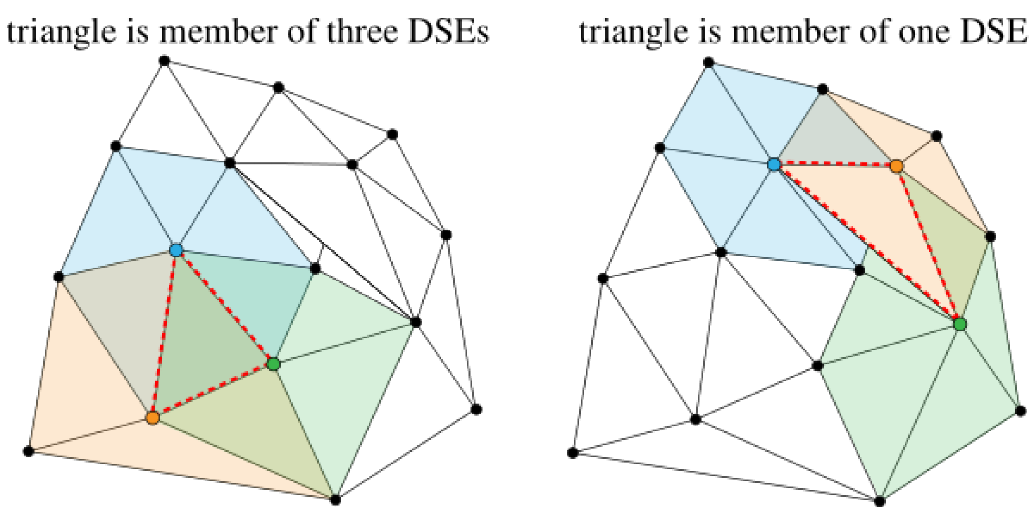
Rakotosaona, Marie-Julie and Guerrero, Paul and Aigerman, Noam and Mitra, Niloy J and Ovsjanikov, Maks. Learning Delaunay surface elements for mesh reconstruction[C]//CVPR. 2021

Parameterization based IMG : DSE

Log map Alignment

Corresponding points in adjacent U_i are aligned using a rigid transformation. Then, the resulting set of corresponding points is clustered to remove outliers, and the average is taken to obtain the final two-dimensional coordinates.



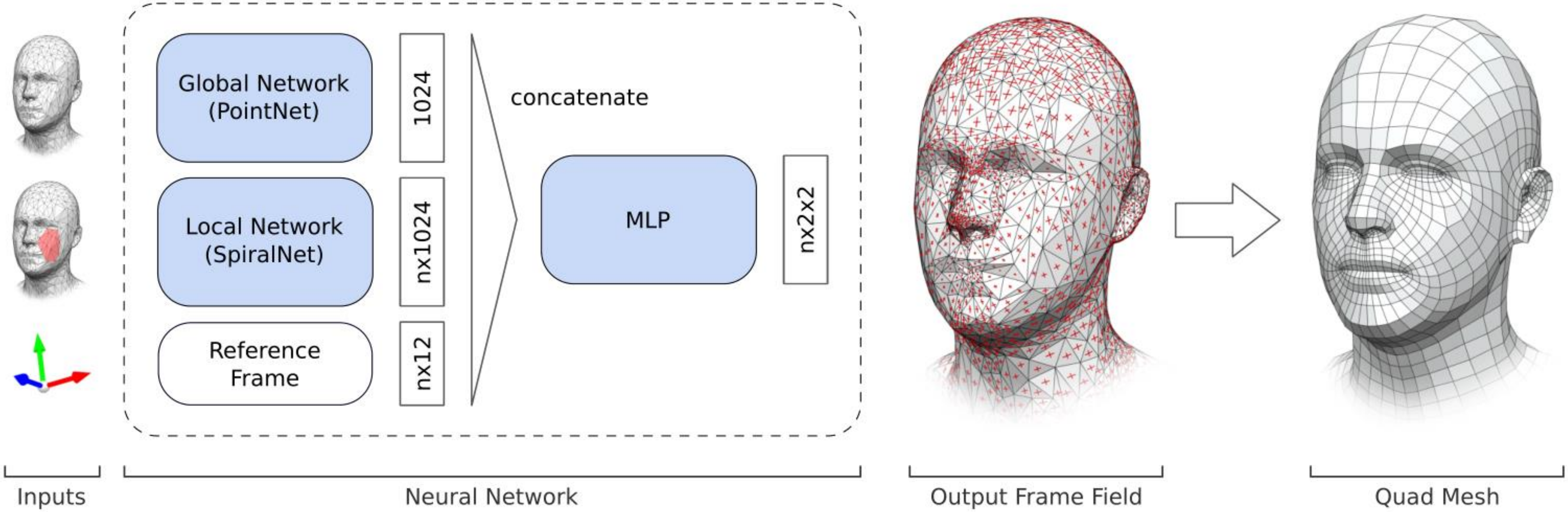


Triangle Selection Criteria

To obtain a mesh as close to the manifold as possible, the criteria established is that triangles appearing in three distinct DSEs are considered most likely to be present in the manifold mesh. Triangles that appear only once are deemed least likely to be present.

Cross field based IMG: LDFQ

➢ **Learning Direction Fields for Quad Mesh Generation**

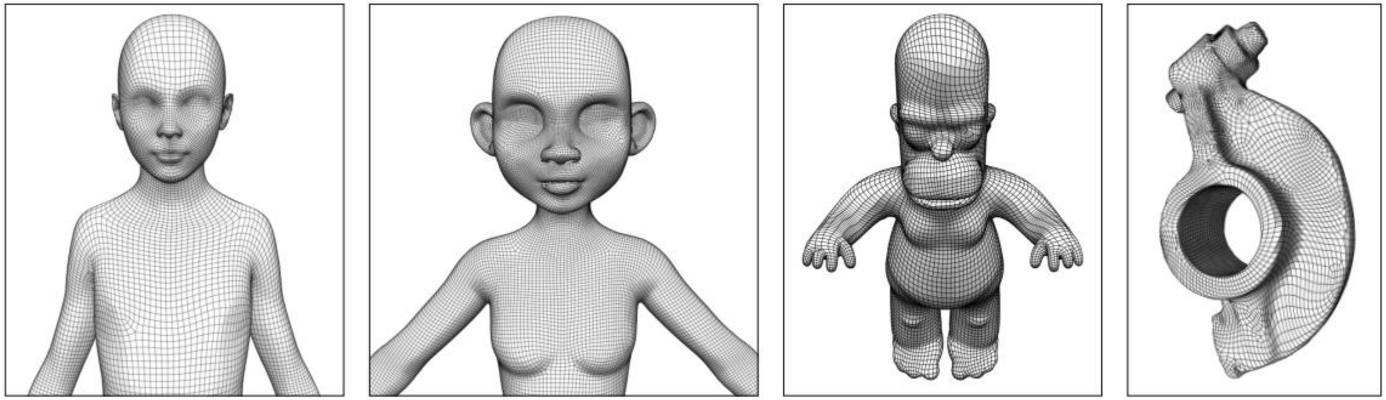


The problem of quadrilateral mesh generation is transformed into the learning of a frame field.

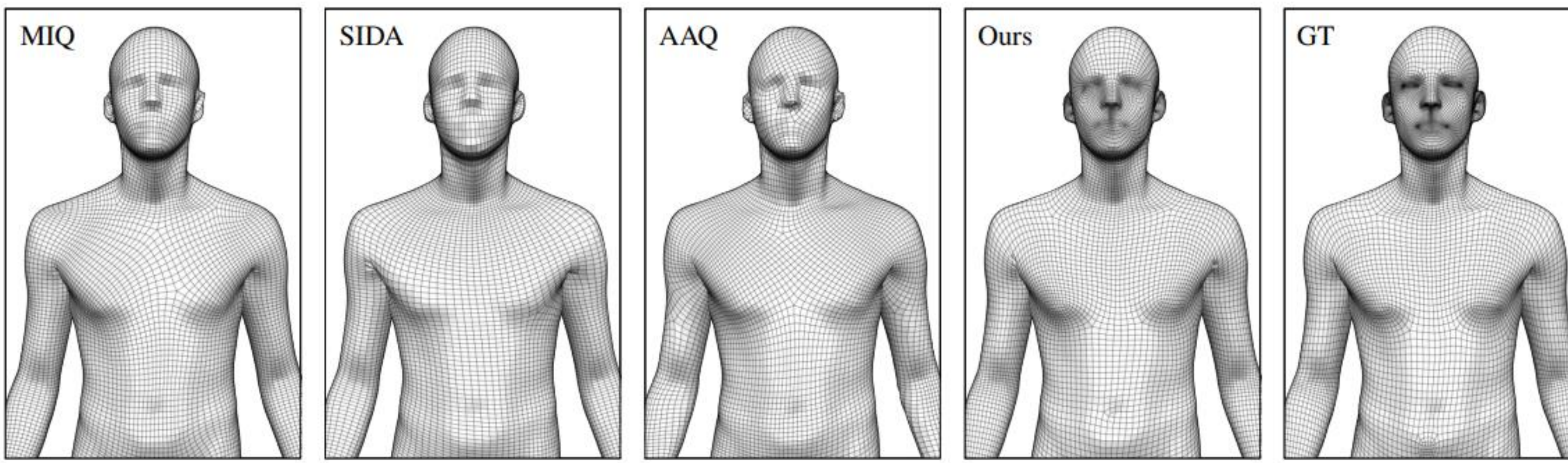
- The geometric information of the mesh is extracted through global and local networks. The reference frame provides the current tangent plane.
- The output is a frame field, which is then used to generate the quadrilateral mesh using traditional methods.

Dielen, Alexander and Lim, Isaak and Lyon, Max and Kobbelt, Leif. Computer Graphics Forum. 2021

Cross field based IMG: LDFQ



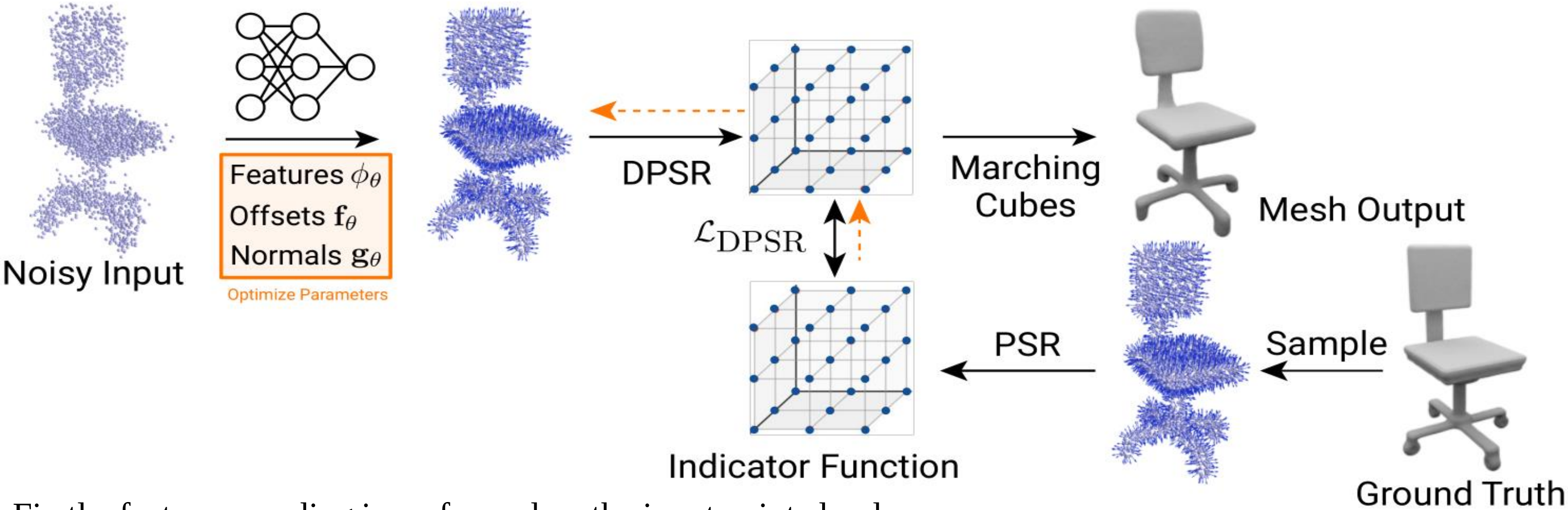
Based on the learned frame field, the final quadrilateral mesh generation result is obtained using traditional methods.



Comparison with other quadrilateral mesh generation algorithms and ground truth data.

Isosurface-based :SAP

- **Shape As Points: A Differentiable Poisson Solver**

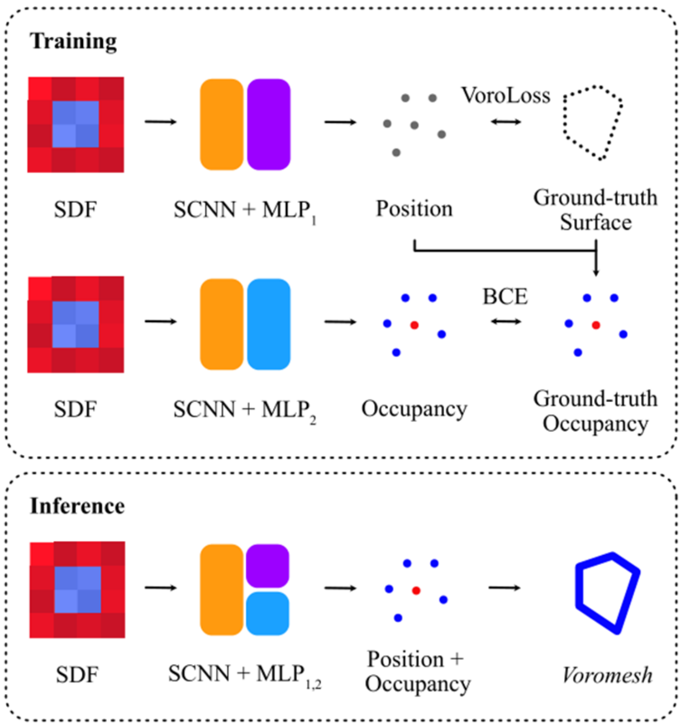


- Firstly, feature encoding is performed on the input point cloud.
- Then, an MLP network f_θ predicts displacement values for each point, resulting in an offset and k-times upsampled point cloud.
- Next, another MLP network g_θ is applied to the resulting point cloud from the previous step to obtain corresponding normal vectors.
- Based on the normal vector field obtained earlier, the Poisson equation is solved to derive the implicit function field.
- Finally, the Marching Cubes algorithm is employed to generate the mesh.

Peng, Songyou and Jiang, Chiyu and Liao, Yiyi and Niemeyer, Michael and Pollefeys, Marc and Geiger, Andreas, A. Shape as points: A differentiable poisson solver[J]. NeurIPS, 2021

Iso-surface based IMG: VoroMesh

➤ **VoroMesh: Learning Watertight Surface Meshes with Voronoi Diagrams**



$$q_i = v_i + MLP_1(SCNN(SDF(v_i)))$$
$$o_i = MLP_2(SCNN(SDF(v_i)))$$

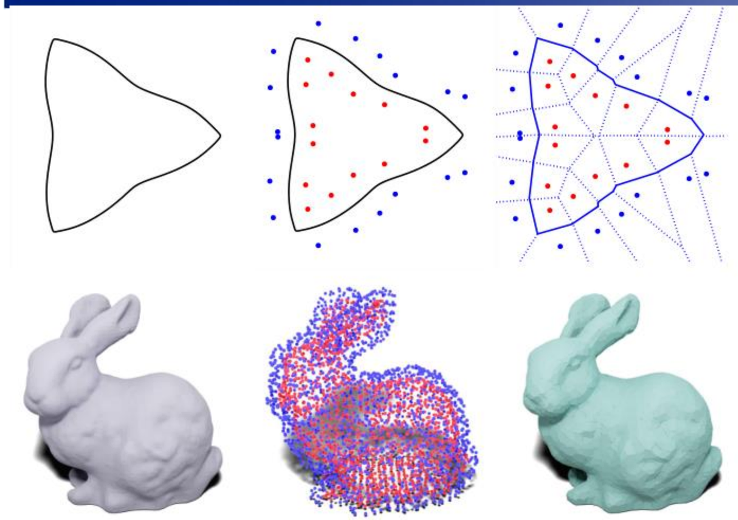
- Find a concise, learnable discrete representation of 3D surfaces
- Reconstruct watertight and non-self-intersecting meshes

To generate a VoroMesh from a grid $I \in R^{N\times N\times N}$ of signed distances field (SDF).

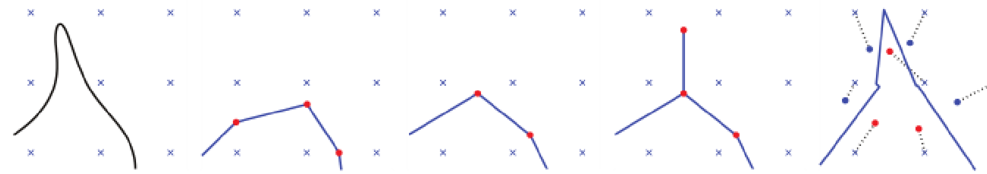
1. Densely sample a set of points $X \in \mathbb{R}^{M\times 3}$ from a ground truth surface;
2. Selected grid points close to the surface serve as initialized $Q \in \mathbb{R}^{N\times 3}$
3. Compute the Voronoi diagram of Q;
4. Minimize $VoroLoss(X, Q)$;
5. Determine the ground truth occupancy O of the barycenter of each Voronoi cell;
6. Compute the final polygonal mesh $VoroMesh(Q, O)$.

Maruani, Nissim and Roman Klokov and Maks Ovsjanikov. VoroMesh: Learning Watertight Surface Meshes with Voronoi Diagrams[C]//ICCV. 2023

Iso-surface based IMG: VoroMesh

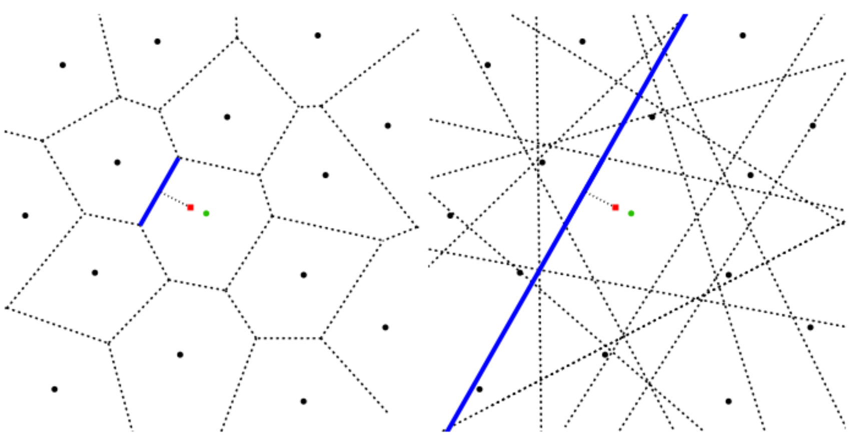


(a) Target surface (b) Optimization (c) *VoroMesh*



(a) Target (b) MC [26] (c) DC [18] (d) UDC [6] (e) Ours

Figure 2: Marching Cubes (b) and Dual Contouring (c) cannot capture details of a target shape (a) smaller than the grid size; UDC (d), based on edge-crossings, can but at the price of a non-manifold elements. *VoroMesh* (e) both captures the details and returns a closed and manifold mesh



(a) Closest face (b) Closest bisector

Theorem 1 *The distance from x to its closest face in a Voronoi diagram equals the distance from x to the closest bisector $H_{i_x,j}$ formed between q_{i_x} whose Voronoi cell contains x and another Voronoi site q_j:*

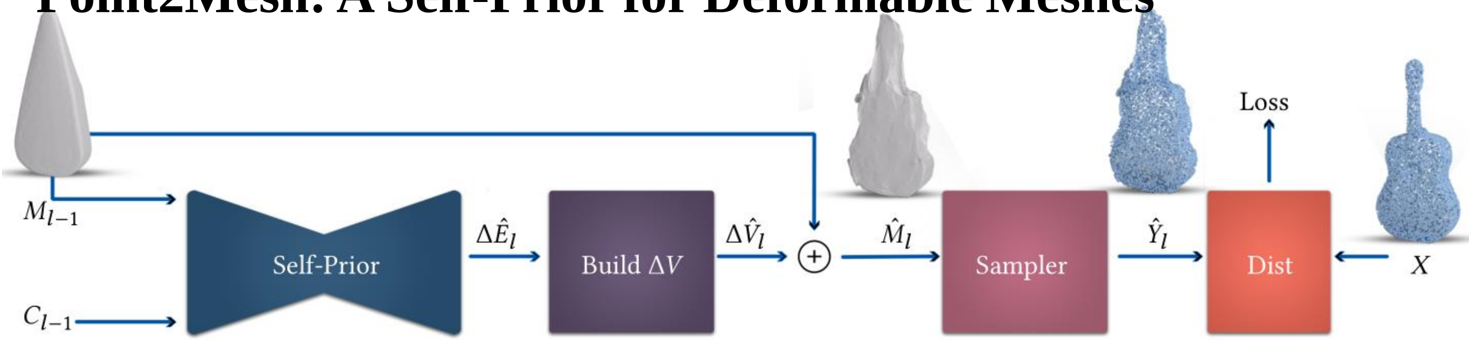
$$\|x - \partial V_i\| = \min_{j \neq i_x} \|x - H_{i_x,j}\|.$$

We thus introduce a loss function, dubbed *VoroLoss*:

$$\mathit{VoroLoss}(X, \mathbf{Q}) := \sum_{x \in X} \min_{j \neq i_x} \|x - H_{i_x,j}\|^2$$

Deformation-based IMG: Point2Mesh

- **Point2Mesh: A Self-Prior for Deformable Meshes**



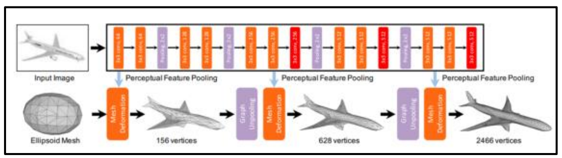
- Firstly, construct an template mesh that is topologically equivalent to the target object. If the genus of the target object is zero, a convex hull is constructed. If the genus is not zero, the alpha shape algorithm is used to construct a concave hull, or Poisson reconstruction is performed based on the point cloud.
- Input the current mesh along with the initial displacement of each edge into MeshCNN, and sequentially obtain the displacement of faces, edges, and vertices.
- Update the coordinates of the vertices based on the predicted displacement, thereby progressively approaching the target surface.
- Repeat the second and third steps.

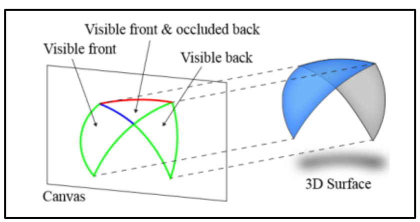
Hanocka, Rana and Metzer, Gal and Giryes, Raja and Cohen-Or, Daniel. Point2Mesh: a self-prior for deformable meshes[J]. ACM Transactions on Graphics (TOG), 2020

The Role of Intelligent Methods in Mesh Generation

Extended Input Data Types for Mesh Generation

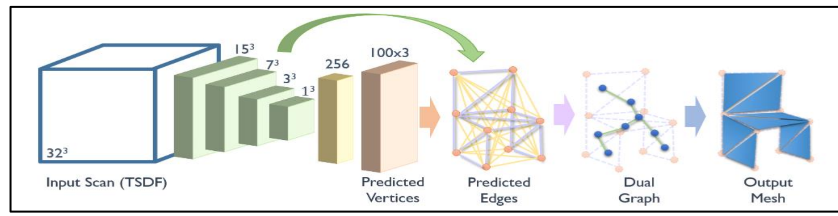
Image to mesh





Generate meshes from images or sketches

Voxel to mesh



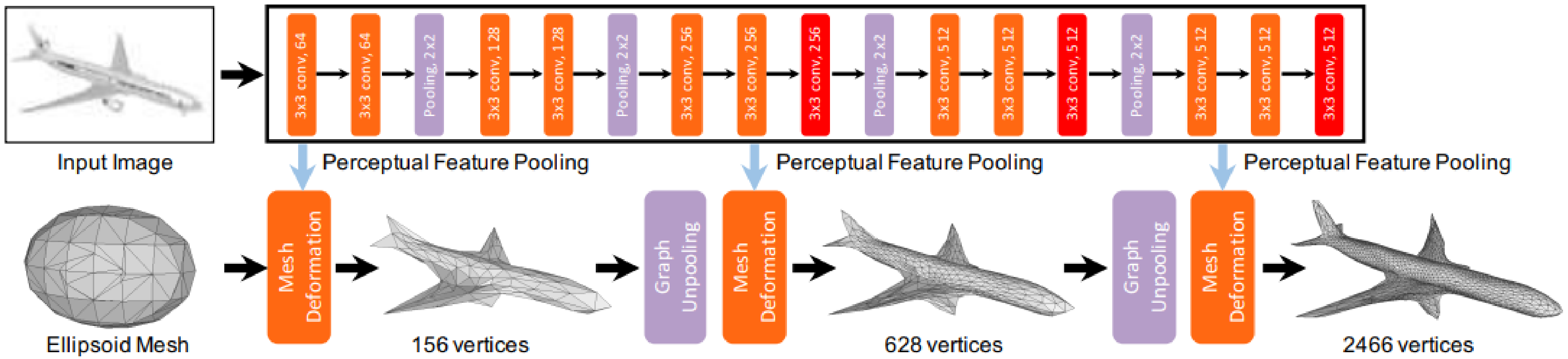
Generate meshes from Voxel or SDF

Text to mesh

Generate meshes from text

Image to mesh: Pixel2Mesh

➢ **Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images**



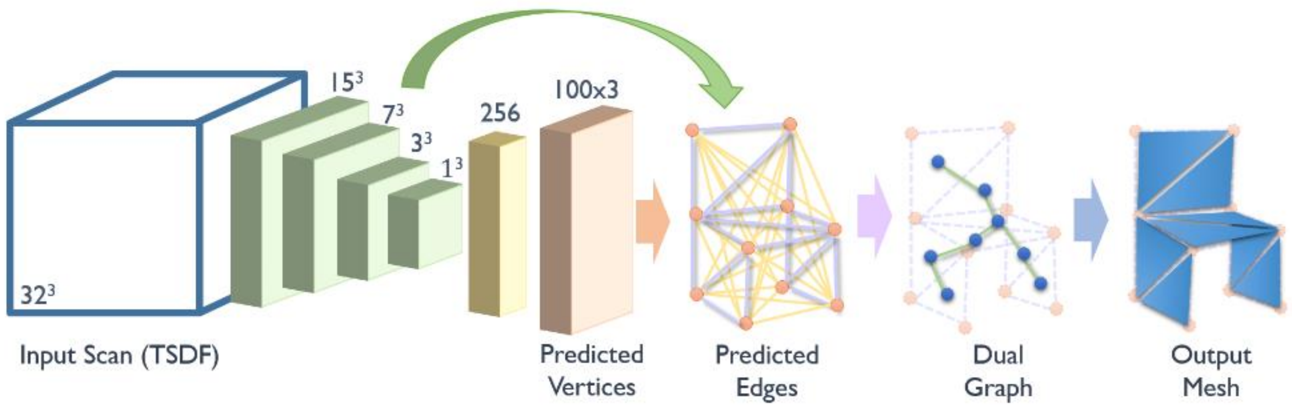
The entire network consists of an Image Feature Network and a Cascaded Mesh Deformation Network.

- The Image Feature Network comprises a 2D CNN that extracts perceptual features from the input image.
- The Mesh Deformation Network gradually deforms an ellipsoid mesh into the desired 3D model based on the perceptual features extracted from the image.

Wang, Nanyang and Zhang, Yinda and Li, Zhuwen and Fu, Yanwei and Liu, Wei and Jiang, Yu-Gang. ECCV. 2018

Voxel to mesh: Scan2Mesh

➢ **Scan2Mesh: From Unstructured Range Scans to 3D Meshes**

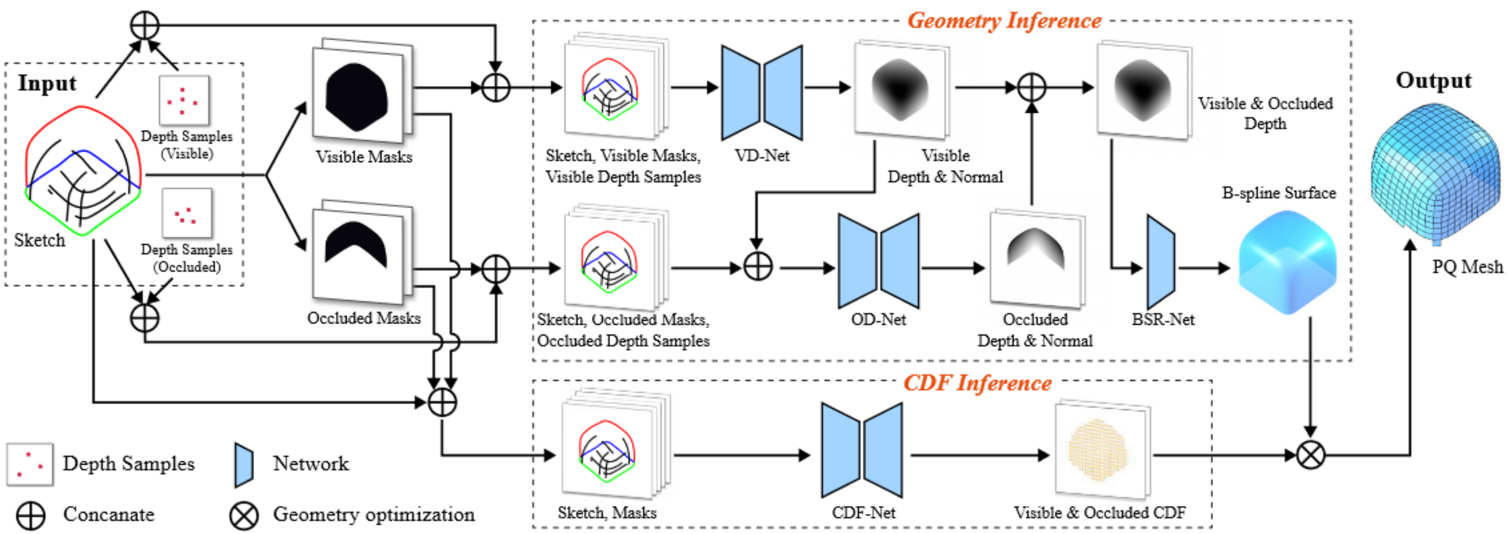


- First, based on the input truncated signed distance field (TSDF), a neural network module predicts the positions of n vertices;
- Using another neural network module, the existence of each edge is predicted based on the coordinates of the corresponding points and their associated features.
- Treating each face as a node, a neural network module predicts the existence of each face. The features of each face are represented by an 8-dimensional vector composed of barycentric coordinates, normal vectors, area, and circumscribing circle radius.

Dai, Angela and Niebner, Matthias. CVPR 2019

Sketch to mesh: Sketch2PQ

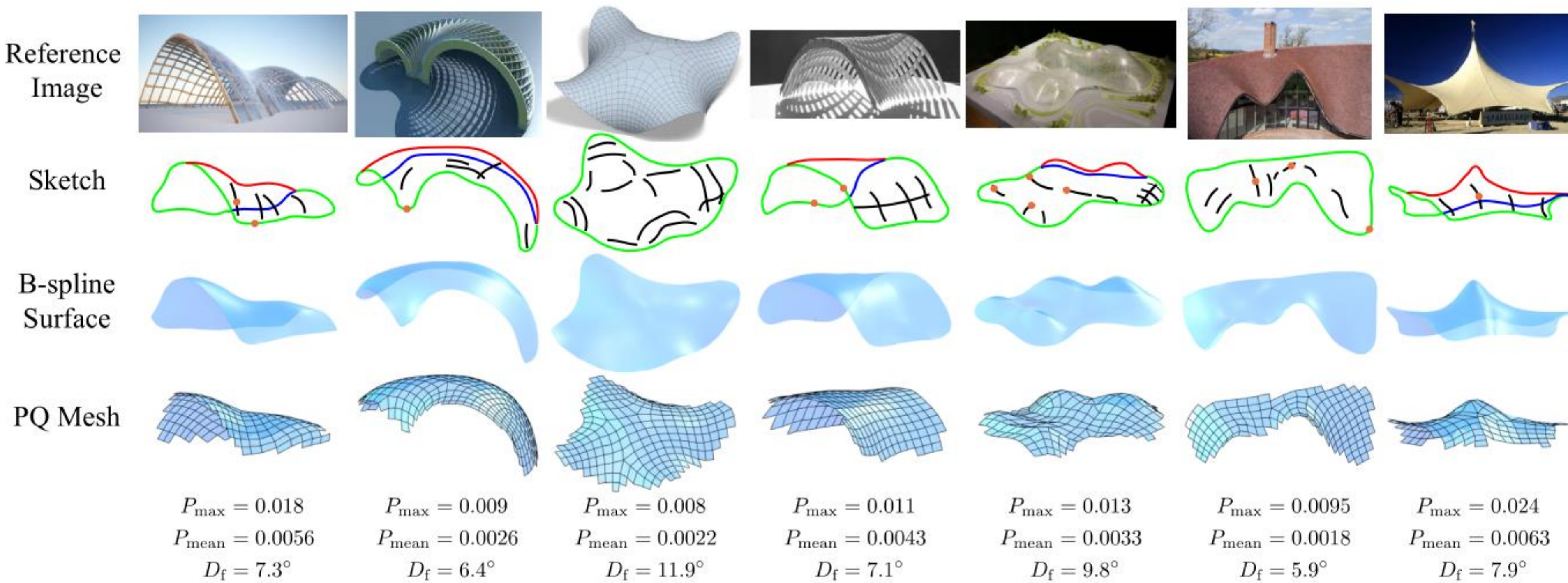
➢ **Sketch2PQ: Freeform Planar Quadrilateral Mesh Design via a Single Sketch**



- Using stroke lines, depth sampling, and visible/occluded region masks derived from sketches as inputs;
- A geometric inference module predicts depth maps and normal maps for visible and occluded regions, using them to infer B-spline surfaces;
- A conjugate direction field (CDF) inference module predicts an approximate CDF layout for PQ mesh;
- Finally, PQ mesh is extracted from B-spline surfaces and CDF using geometric optimization.

Deng, Zhi and Liu, Yang and Pan, Hao and Jabi, Wassim and Zhang, Juyong and Deng, Bailin. IEEE Transactions on Visualization and Computer Graphics (2022)

Sketch to mesh: Sketch2PQ



P_{mean} and P_{max} represent the average planarity error and maximum planarity error of the entire mesh, respectively.

D_{f} is an angle-based alignment error that measures the angle between the direction of feature lines and their corresponding projected lines on the plane.

Text to mesh : CLIP-Mesh

- Making it possible to generate meshes by zero-shot textguided with a differentiable renderer

- Using the analytical expression of the Loop subdivision limit surface as an implicit regularizer.
- Introducing a set of render augmentations and incorporating a text to image embedding prior.

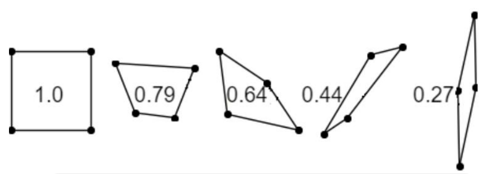
Mohammad Khalid N, Xie T, Belilovsky E, et al. Clip-mesh: Generating textured meshes from text using pretrained image-text models[C]//SIGGRAPH Asia 2022 conference papers. 2022: 1-8.

Outline

- Research Background
- Intelligent Mesh Representation(IMR)
- Intelligence Mesh Generation(IMG)
- Intelligent Mesh Evaluation(IME)
- Summary and Outlook

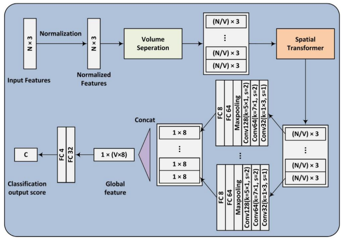
Intelligent Mesh Evaluation

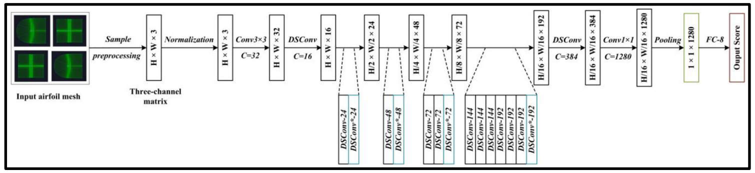
Traditional metric-based



Aspect Ratio
Vertex Valance
Min-max Angle
Jacobian

Neural network-based





Intelligent Mesh Evaluation

Traditional metric-based

RLQMG & SRL-assisted AFM

$$r_t(s_t, a_t) = \begin{cases} -0.1, & \text{invalid element;} \\ 10, & \text{the element is the last element;} \\ m_t, & \text{otherwise.} \end{cases}$$

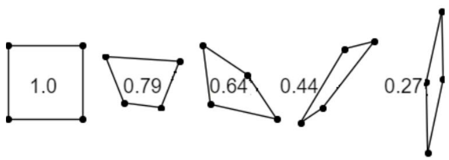
$$m_t = \eta_t^e + \eta_t^b + \mu_t.$$

$$\eta_t^e = \sqrt{q^{edge} q^{angle}},$$

$$q^{edge} = \frac{\sqrt{2} min_{j\in\{0,1,2,3\}}\{l_j\}}{D_{max}},$$

$$q^{angle} = \frac{min_{j\in\{0,1,2,3\}}\{angle_j\}}{max_{j\in\{0,1,2,3\}}\{angle_j\}},$$

$$R^s = \sqrt[3]{\frac{\min\{\theta_1, \theta_2, \theta_3, \theta_4\}}{90^\circ}\left(2 - \frac{\max\{\theta_1, \theta_2, \theta_3, \theta_4\}}{90^\circ}\right)\frac{\min\{l_1, l_2, l_3, l_4\}}{\max\{l_1, l_2, l_3, l_4\}}}.$$



$$R^{ep} = \left(1 - \frac{N_{ep}}{N_{tot}}\right)\left(1 - \frac{N_{cep}}{N_{tot}}\right)$$

$$\eta_t^b = \sqrt{\frac{min_{k\in\{1,2\}}\{min(\varsigma_k, M_{angle})\}}{M_{angle}} q^{dist}} - 1,$$

$$q^{dist} = \begin{cases} \frac{d_{min}}{(d_1+d_2)/2}, & \text{if } d_{min} < (d_1 + d_2)/2; \\ 1, & \text{otherwise.} \end{cases}$$

$$\mu_t = \begin{cases} -1, & \text{if } \mathcal{A}_t < \mathcal{A}_{min}; \\ \frac{\mathcal{A}_t - \mathcal{A}_{min}}{\mathcal{A}_{max} - \mathcal{A}_{min}}, & \text{if } \mathcal{A}_{min} \le \mathcal{A}_t < \mathcal{A}_{max}; \\ 0, & \text{otherwise.} \end{cases}$$

$$R^{fin} = \frac{1}{M}\sum_{i=1}^{M} R_i^s R_i^{ep} + \min\{R_1^s R_1^{ep}, R_2^s R_2^{ep}, \cdots, R_M^s R_M^{ep}\}$$

- Utilizing traditional metrics, including the Jacobian metric, maximum and minimum angles, aspect ratios, etc., as reward functions in reinforcement learning-based mesh generation.

1. Pan, Jie and Huang, Jingwei and Cheng, Gengdong and Zeng, Yong. Reinforcement learning for automatic quadrilateral mesh generation: A soft actor–critic approach[J]. Neural Networks, 2023

2. Hua Tong, Kuanren Qian, Eni Halilaj, Yongjie Jessica Zhang,SRL-assisted AFM: Generating planar unstructured quadrilateral meshes with supervised and reinforcement learning-assisted advancing front method, Journal of Computational Science,Volume 72,2023,102109.

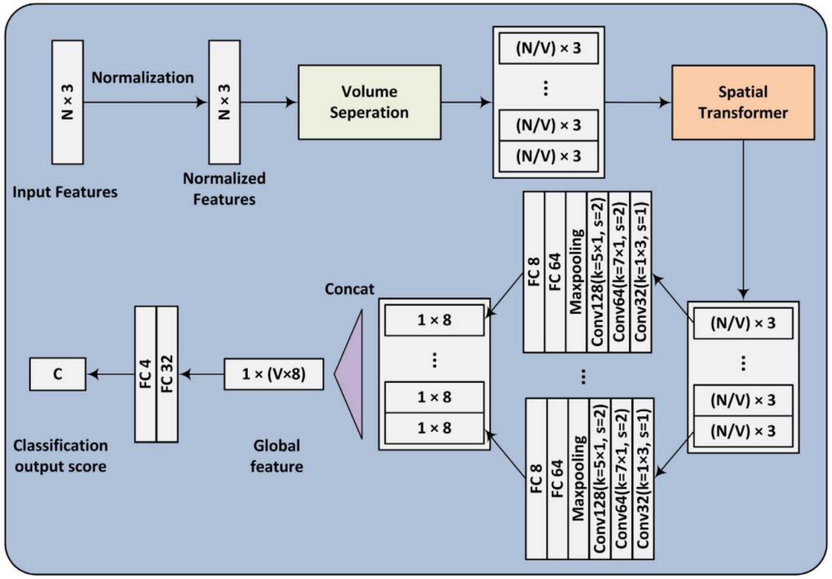
Intelligent Mesh Evaluation

Neural network-based

- **MVE-Net***

Input: Point Coordinates

Output: High-quality Mesh / Non-orthogonal Mesh / Non-smooth Mesh / Poor-quality Mesh



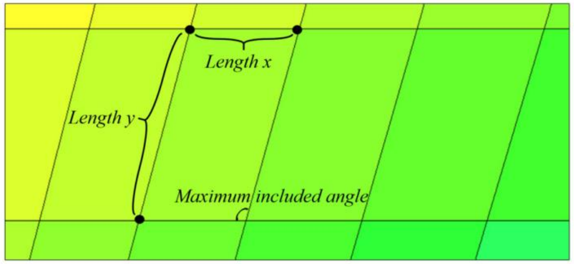
- Employs a two-step structure (volume and global features learning) to map between the input (ordered point coordinates) and mesh validity
- Studies the role of mesh point distribution on numerical accuracy, and outputs the overall validity for the simulation
- Since MVE-Net is a black box, its evaluation metrics are not interpretable.

* Chen X, Liu J, Gong C, et al. MVE-Net: An automatic 3-D structured mesh validity evaluation framework using deep neural networks[J]. Computer-Aided Design, 2021, 141: 103104.

Intelligent Mesh Evaluation

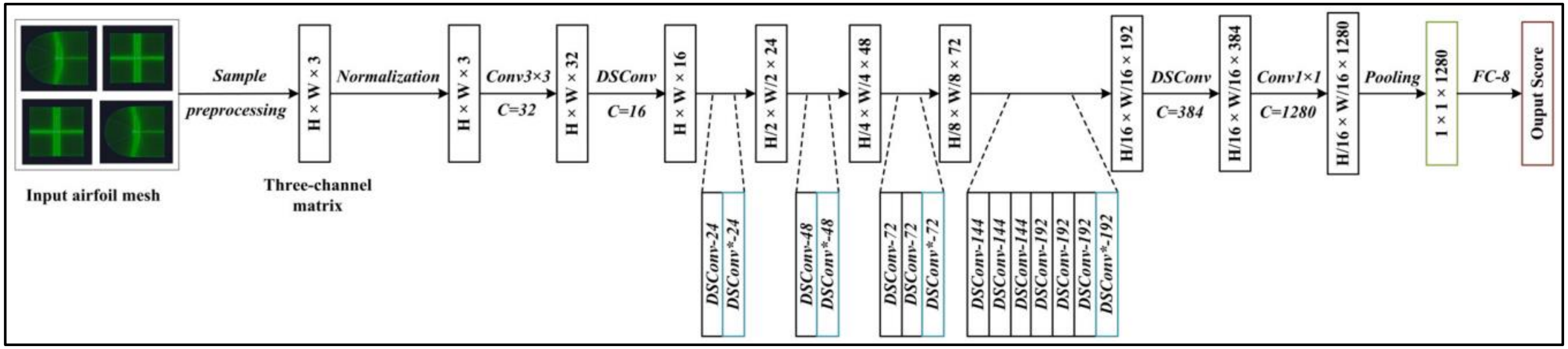
Neural network-based

➢ MQNet*



Input **Output**

Label	Name
1 (W)	Well-shaped
2 (P-O)	Poor orthogonality
3 (P-S)	Poor smoothness
4 (P-D)	Poor density
5 (P-OS)	Poor orthogonality and smoothness
6 (P-OD)	Poor orthogonality and density
7 (P-SD)	Poor smoothness and density
8 (P-OSD)	Poorly-shape



- Learns the quality-related attributes(e.g. mesh orthogonality, smoothness)
- Accounts for both individual element geometry and neighboring attributes

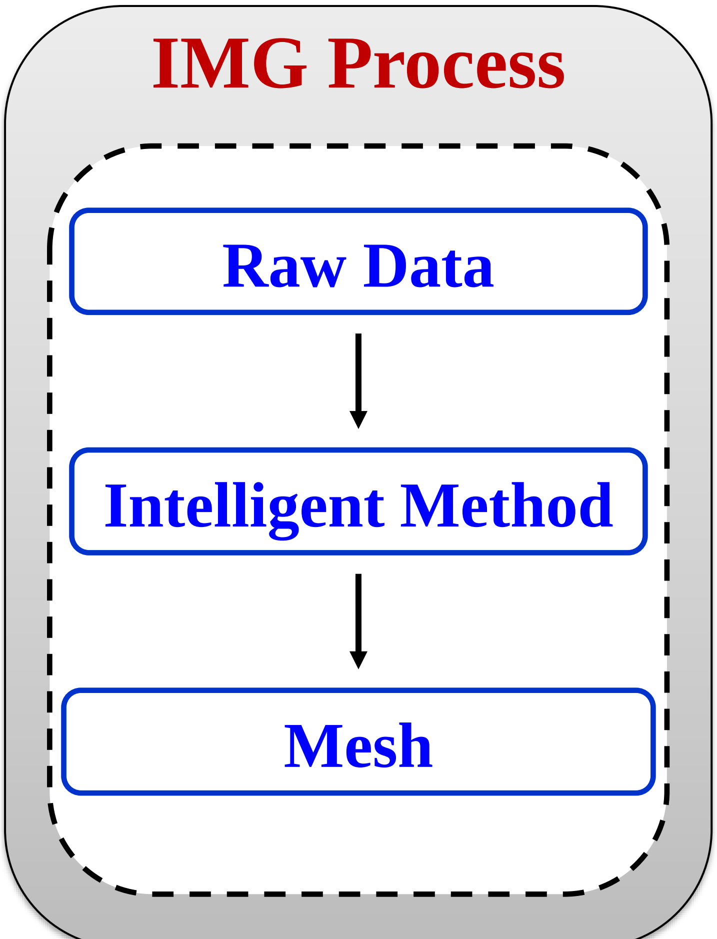
* Chen X, Gong C, Liu J, et al. A novel neural network approach for airfoil mesh quality evaluation[J]. Journal of Parallel and Distributed Computing, 2022, 164: 123-132.

Outline

- Research Background
- Intelligent Mesh Representation(IMR)
- Intelligence Mesh Generation(IMG)
- Intelligent Mesh Evaluation(IME)
- Summary and Outlook

Summary and Outlook

What are the limits of IMG?



Data Bottleneck

- **High Cost:** Quality assessment of meshes is often costly (numerical simulation accuracy)
- **Less Open Source:** High-quality reference data is often difficult to obtain

Suboptimal Mesh Quality

- **Geometric Information:** Most intelligent methods can only maintain geometric characteristics
- **Topology Information:** Only Reinforcement Learning(RL) can guarantee topological characteristics at present

Summary and Outlook

What are the advantages of IMG?

Excellent IMG Method

- **Generalization:** Applicable to various models, i.e. not limited to their type
- **Automation:** Automatically generate grid without parameter tuning
- **Efficiency:** Generate meshes in real-time
- **Robustness:** Avoid the impact of low-quality raw data

Thanks for listening