PolyGen: An Autoregressive Generative Model of 3D Meshes

CSC2457 3D & Geometric Deep Learning

February 9th, 2021

Presenter: Alex(ander) Tessier

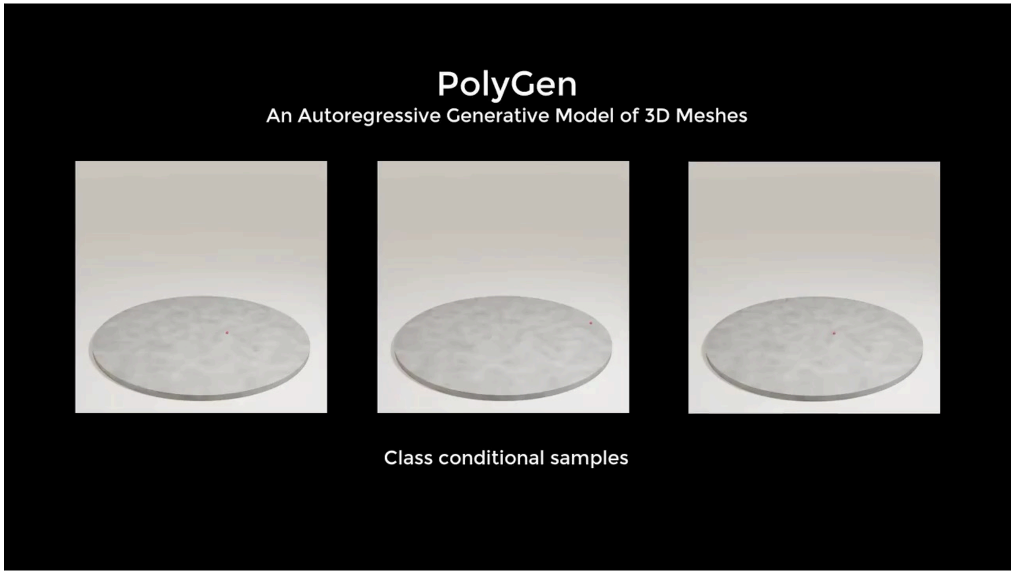
Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Motivation and Main Problem

Generate Meshes with polygonal faces!

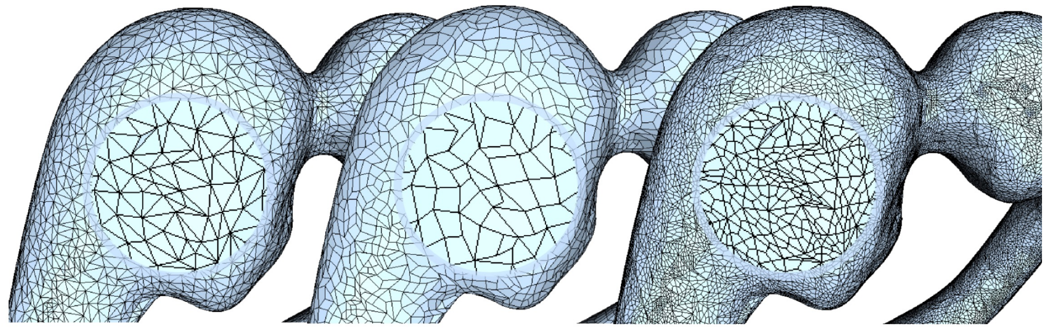
- Have an Image?
- Have a class?
- A Context?
- Some vertices?
- You can generate a mesh!
- Think of the holodeck in Star Trek…

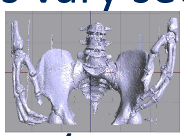


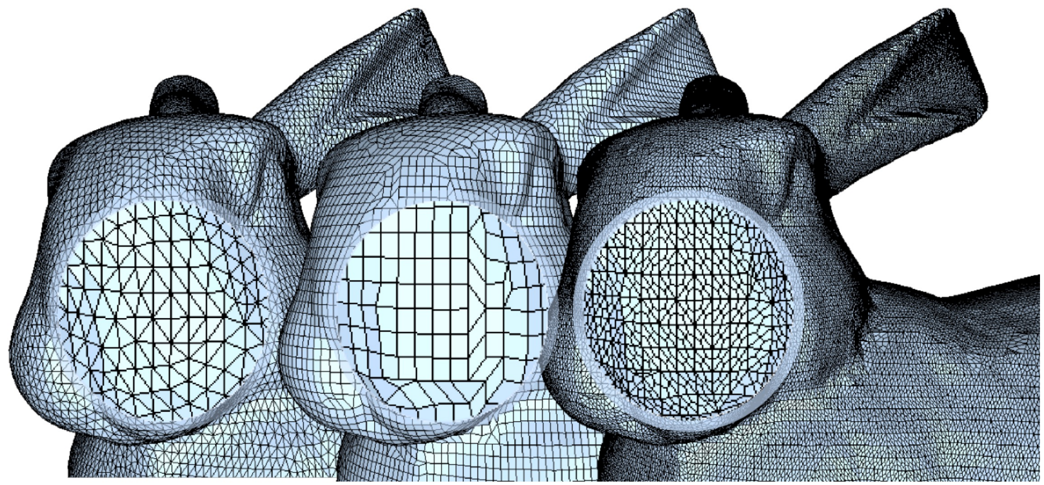
Motivation and Main Problem

- Why generative?
  - Synthesis – 'novelty'
  - Repair
  - Vision & Reasoning
  - Environments for RL and other training
  - 3D Modeling
  - 3D Printing – sample the real world

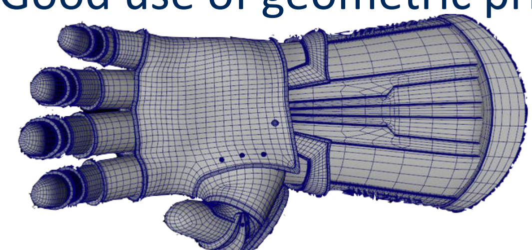
Why are generative meshes hard?



- Unordered elements
- Discrete face structure
  - N-gon means vary sequence size
- Triangle soup 

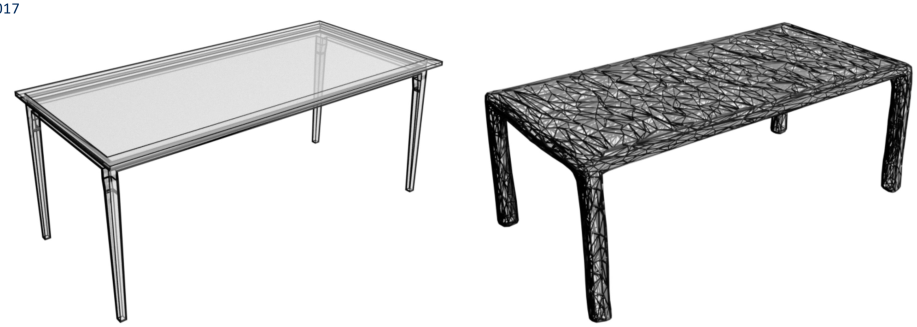


- Human meshes (games, graphics)
  - Compact
  - Good use of geometric primitives

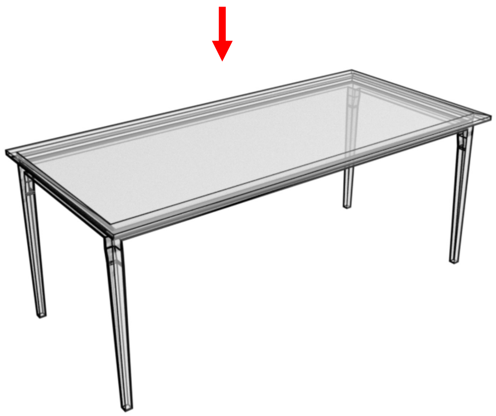


Previous Work

- "…there were no existing methods that directly model mesh vertices and faces"
- Compression – yes – Draco (google)
- Ordered and unordered point clouds
  - The shape variational autoencoder: A deep generative model of part-segmented 3d objects - Nash & Williams 2017
  - Point cloud GAN – Li et. al 2019
  - Pointflow: 3d point cloud generation with continuous normalizing flows. Yang et. al 2019
  - PCT: Point Cloud Transformer. Guo 2020
- Voxels
  - A unified approach for single and multi- view 3d object reconstruction. Choy et.al 2016
  - Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Wu et. al 2016
  - Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. Tarchenko et. al 2017
  - Unsupervised learning of 3d structure from images. 2016
- Functional representations: SDF, Implicits in general
  - Learning continuous signed distance functions for shape representation. Parks et.al 2019
  - Occupancy networks: Learning 3d reconstruction in function space. **Mescheder et. al. 2019**
- Parameterized deformable meshes
  - A papier-maˆche´ approach to learning 3d sur- face generation **Groueix et.al. 2018**



PolyGen



Occupancy Networks (Mescheder 2019)

AtlasNet / Papier-mâché (Groueix 2018)

Implicit Representations – Gradient Fields (Cai 2020)

Differences in approach

- Here, we directly model and generate meshes that are similar to those created by *people*
- Probabilistic model – yields diverse (creative?) output – robust to ambiguous input – principled
- Vertex model like PointGrow (Sun et al, 2020)
  - autoregressive decomposition to model point 3D clouds
    - Fixed length point clouds vs variable vertex sequences
    - Hand-crafted self-attention mechanism vs SOA deep architectures -> to model verts and faces – generation of high quality meshes

Contributions

- First: Unconditional mesh vertex & face models create n-gon meshes – directly, no post-processing! Autoregressive!
- Demonstrated conditional generation given any and/or only one of:
  - Object Class
  - Image
  - Voxels
- Application of Transformers to Meshes: vertex & face distributions, robust to bad input
- Output is diverse, realistic and directly usable in graphics applications – unlike previous post-processed output

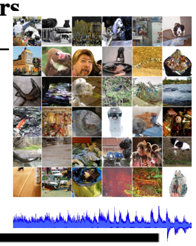
General Background

PolyGen: An Autoregressive Generative Model of 3D Meshes

=

Charlie Nash [1] Yaroslav Ganin [1] S. M. Ali Eslami [1] Peter W. Battaglia [1]

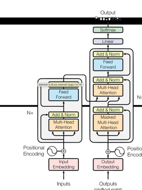
Generating Long Sequences with Sparse Transformers



Rewon Child [1] Scott Gray [1] Alec Radford [1] Ilya Sutskever [1]

+

Attention Is All You Need



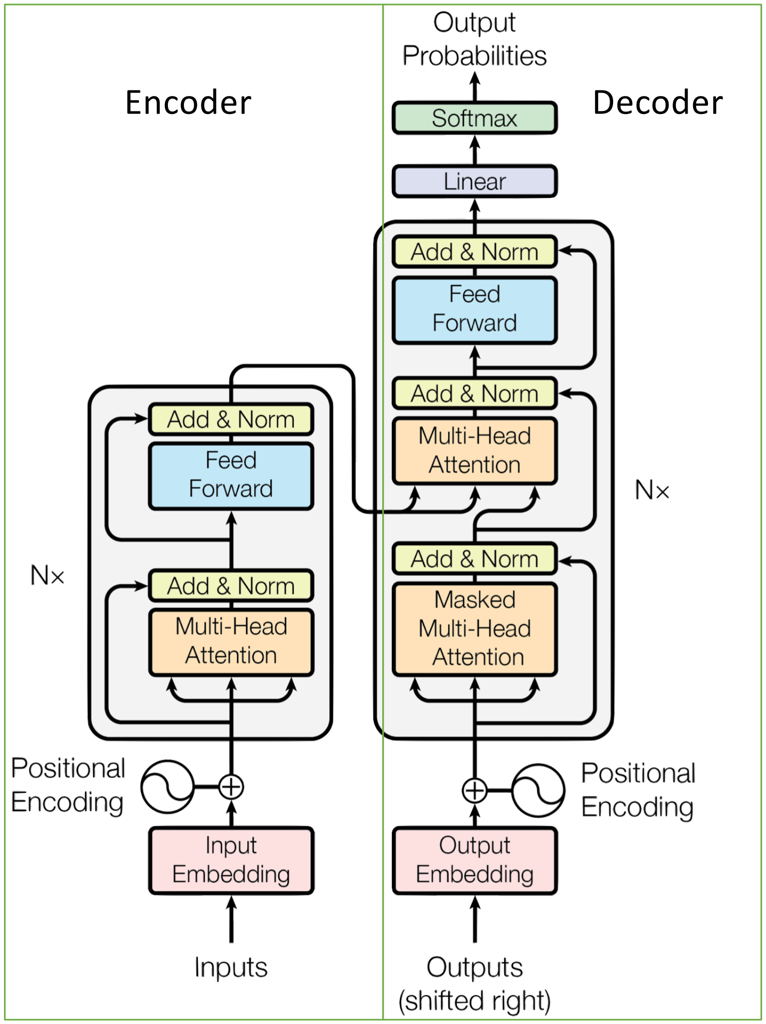
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

+

Pointer Networks

Oriol Vinyals* Google Brain
Meire Fortunato* Department of Mathematics, UC Berkeley
Navdeep Jaitly Google Brain

General Background – **Transformers &** Attention



Why Transformers for Meshes?

- Short & Long range dependencies (context)
- Relationships between faces and vertices
  - Symmetries
  - Non-local dependencies
  - Arbitrary distributions
- Distribution over sequences
- Analogous to words / sentences / paragraphs
  - Autoregressive language models

Problem Setting

A mesh, $\mathcal{M}$ has 3d vertices $\mathcal{V}$, which are indexed to form faces, $\mathcal{F}$

$$p(\mathcal{M}) = p(\mathcal{V}, \mathcal{F}) = p(\mathcal{F}|\mathcal{V})p(\mathcal{V})$$

$$\mathcal{F}_{\text{tri}} = \left\{ \left(f_1^{(i)}, f_2^{(i)}, f_3^{(i)} \right) \right\}_i$$

$$\mathcal{F}_{n\text{-gon}} = \left\{ \left(f_1^{(i)}, f_2^{(i)}, \ldots, f_{N_i}^{(i)} \right) \right\}_i$$

Problem Setting

A mesh, $\mathcal{M}$ has 3d vertices $\mathcal{V}$, which are indexed to form faces, $\mathcal{F}$

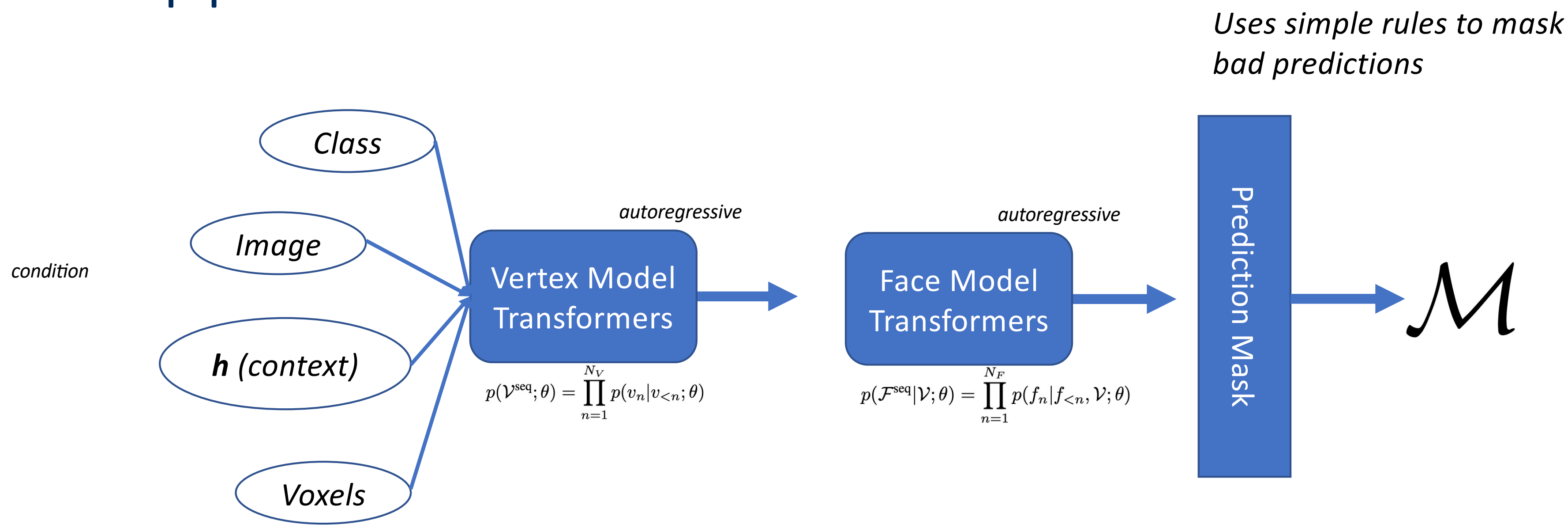
$$\underbrace{\mathcal{V}^{\text{seq}} = v_{\text{n}}\,,\, n = 1, \ldots, N_V}_{\text{8 bit-quantization } (0\text{-}255)}$$

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

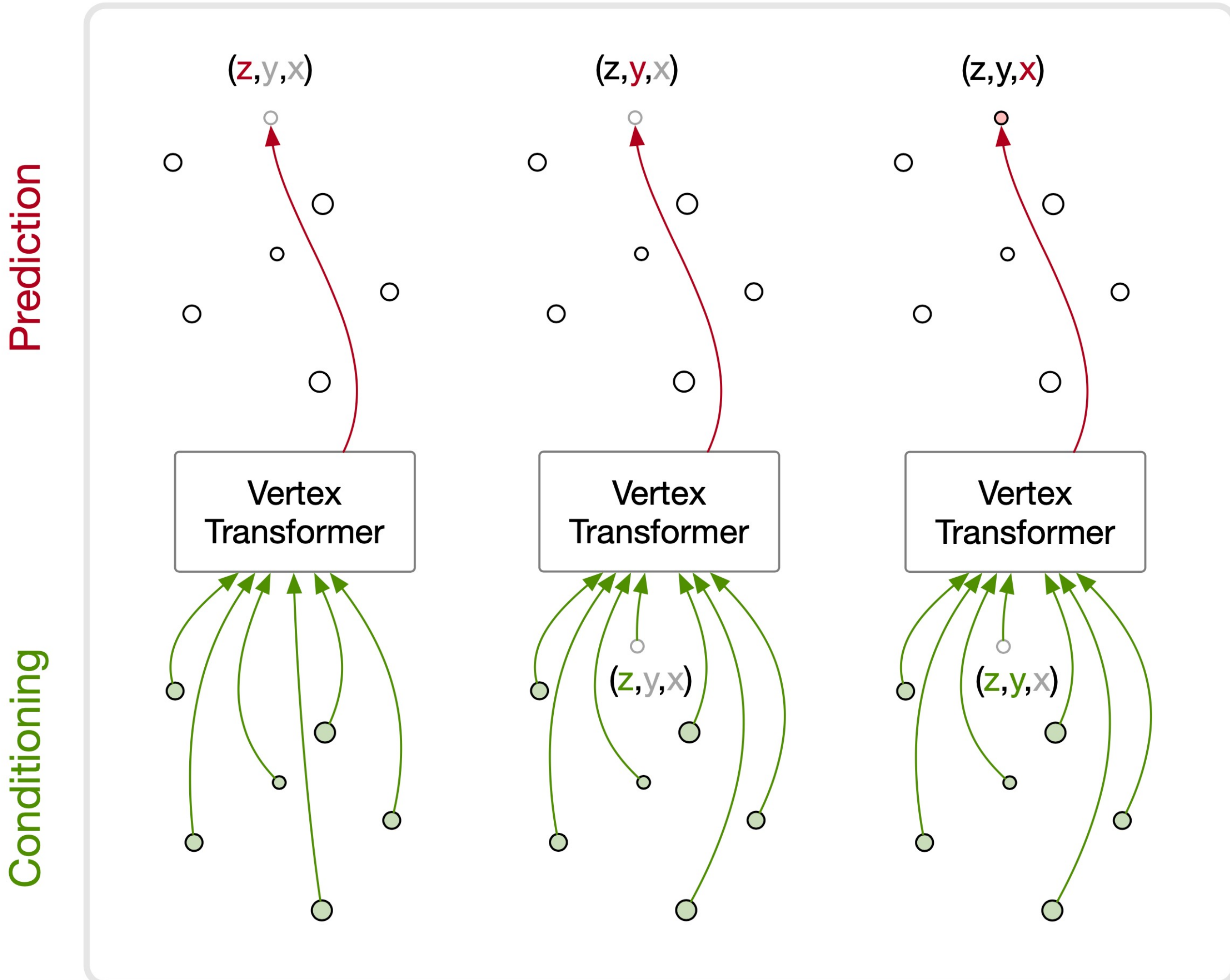
$$\max(\log p(\mathcal{V}^{\text{seq}} | \theta))$$

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

Approach



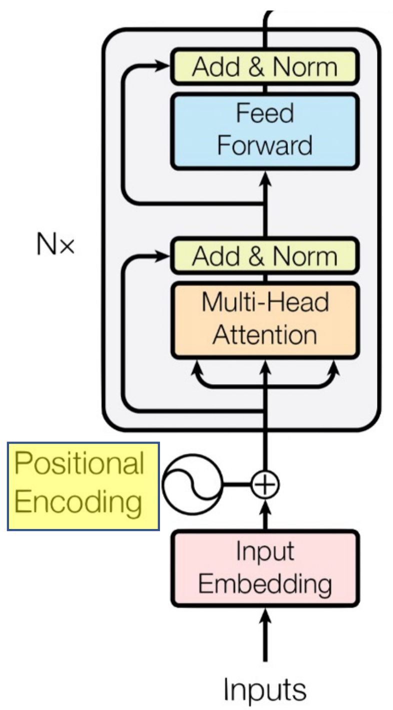
$$\mathcal{V}^{\text{seq}} = v_{\text{n}}\,,\, n = 1,\, \ldots,\, N_V$$

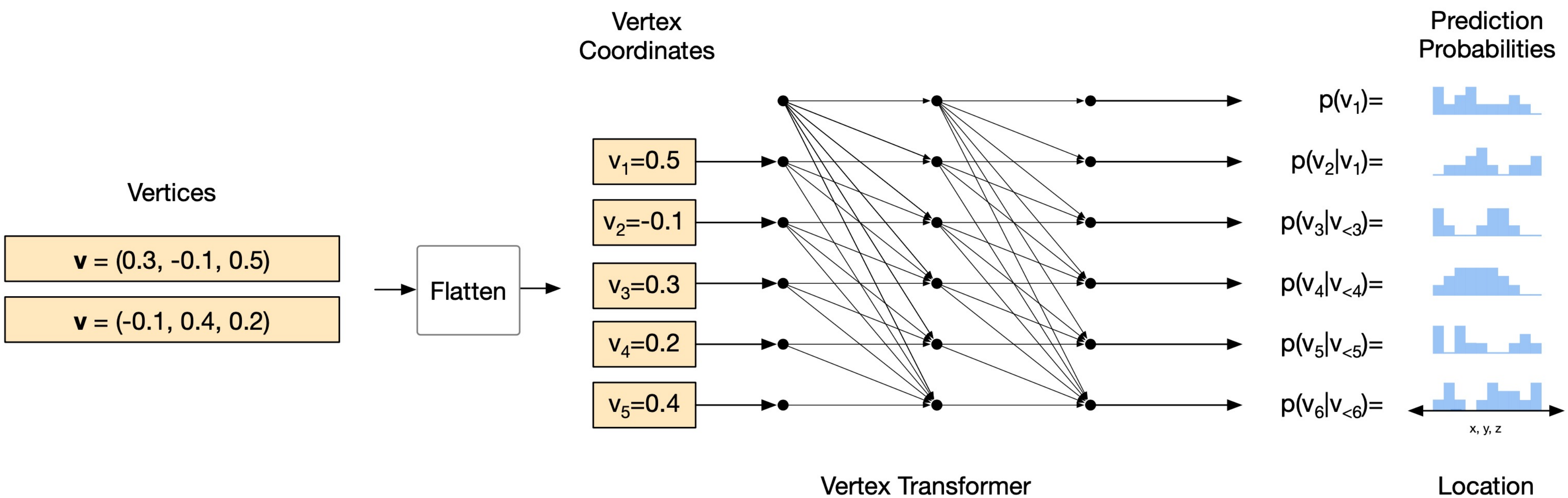


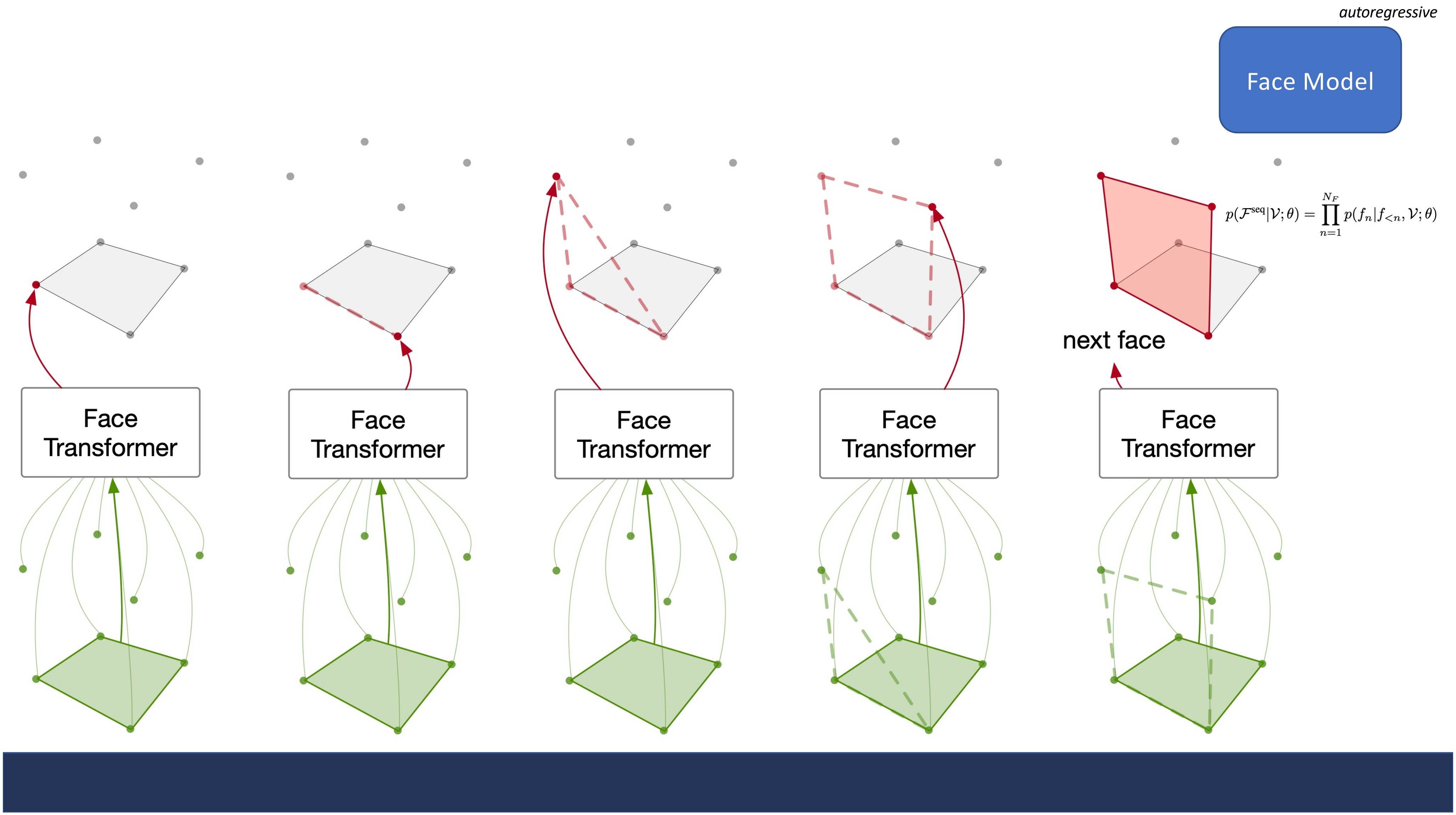
autoregressive

Vertex Model

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



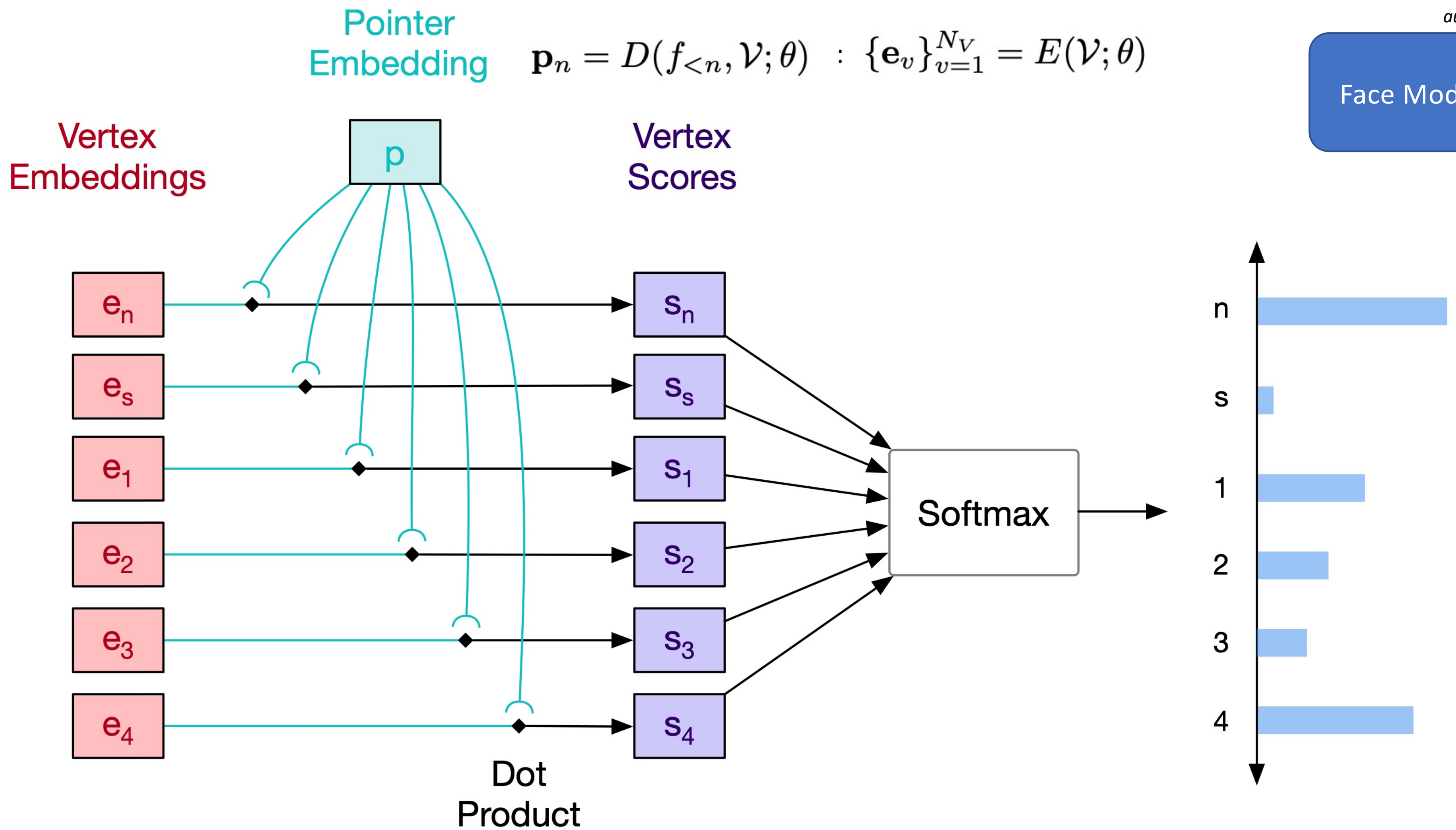

Vertices
v = (0.3, -0.1, 0.5)
v = (-0.1, 0.4, 0.2)
Flatten
Vertex Coordinates
v1=0.5
v2=-0.1
v3=0.3
v4=0.2
v5=0.4
Vertex Transformer
Prediction Probabilities
p(v1)=
p(v2|v1)=
p(v3|v<3)=
p(v4|v<4)=
p(v5|v<5)=
p(v6|v<6)=
x, y, z
Location


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

$$\mathbf{p}_n = D(f_{<n}, \mathcal{V}; \theta) \quad : \quad \{\mathbf{e}_v\}_{v=1}^{N_V} = E(\mathcal{V}; \theta)$$

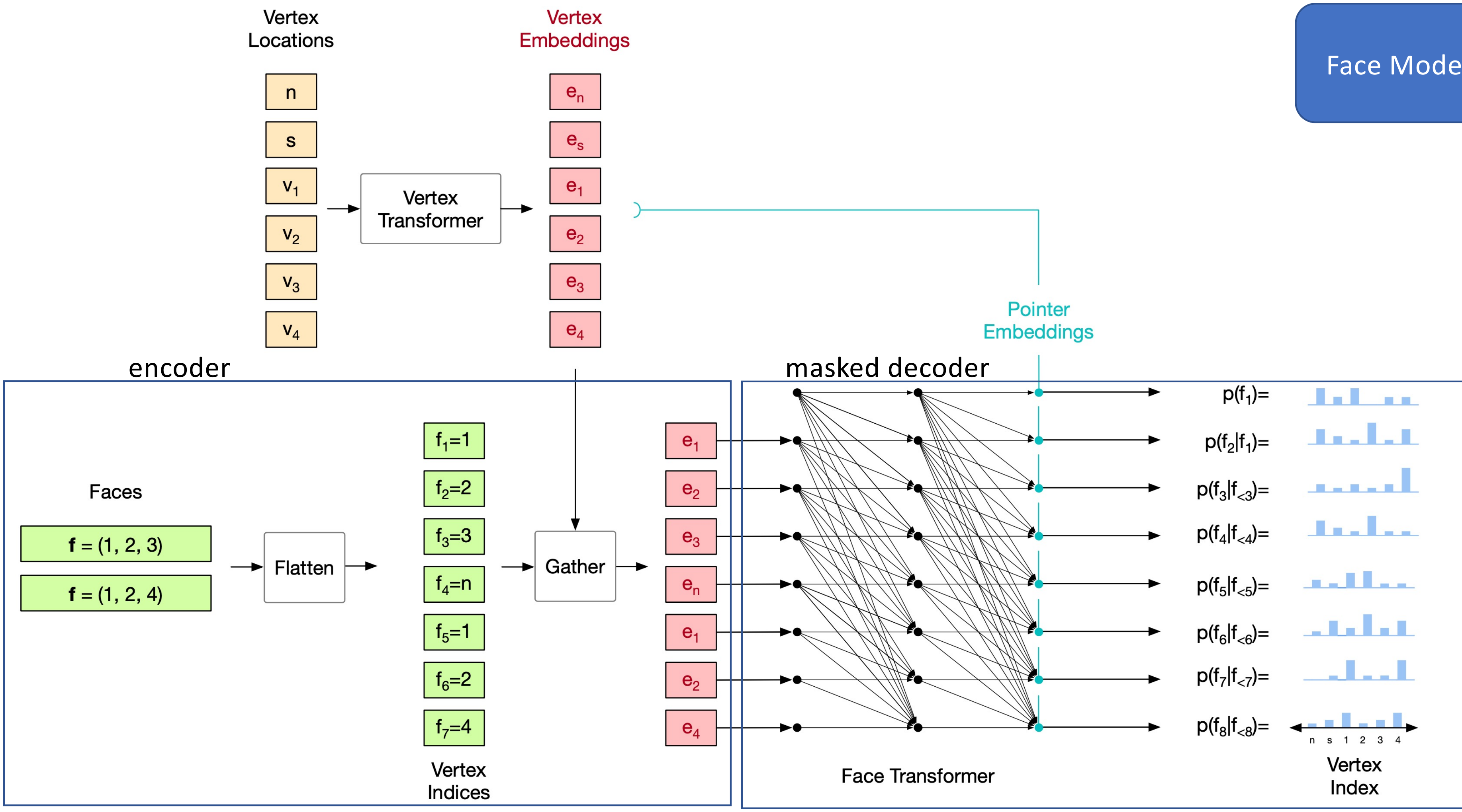
autoregressive

Face Model

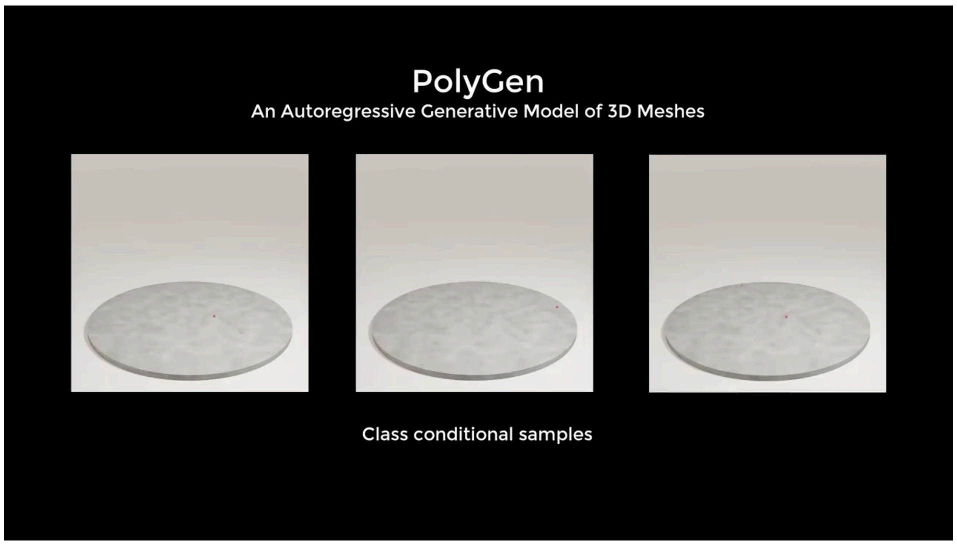


autoregressive

Face Model



Discussion of results



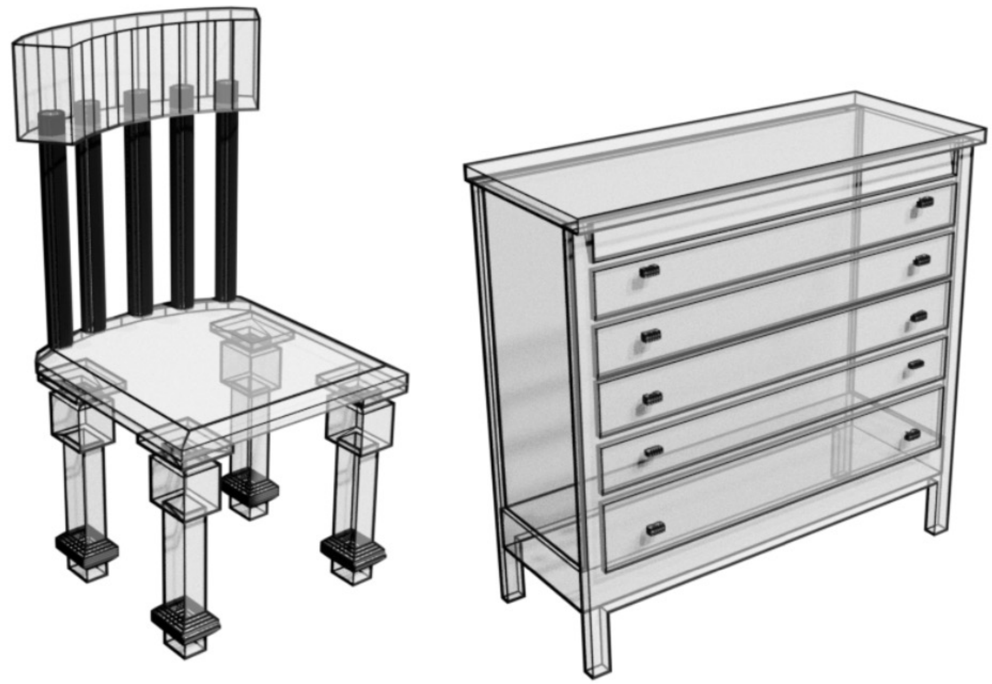
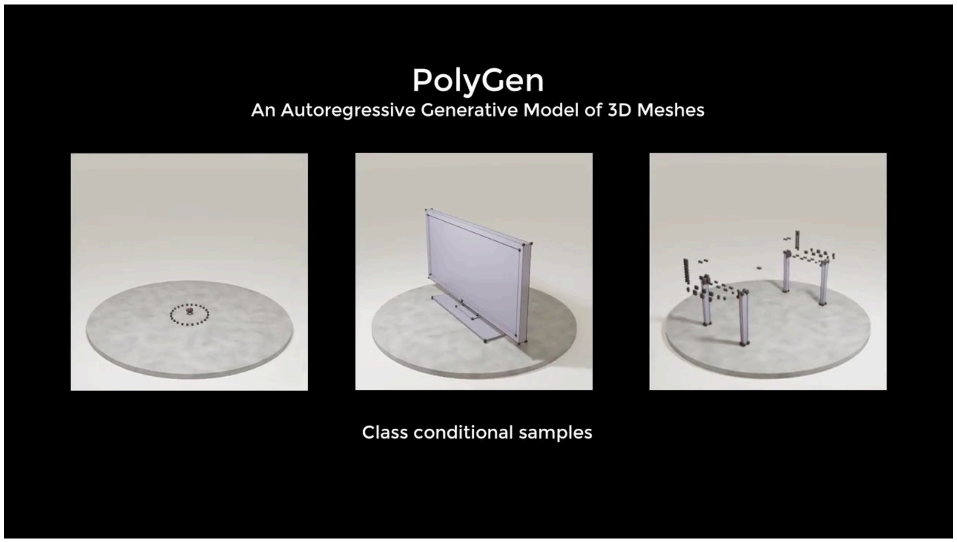


Figure 1. Class conditional n-gon meshes generated by PolyGen.

Discussion of results



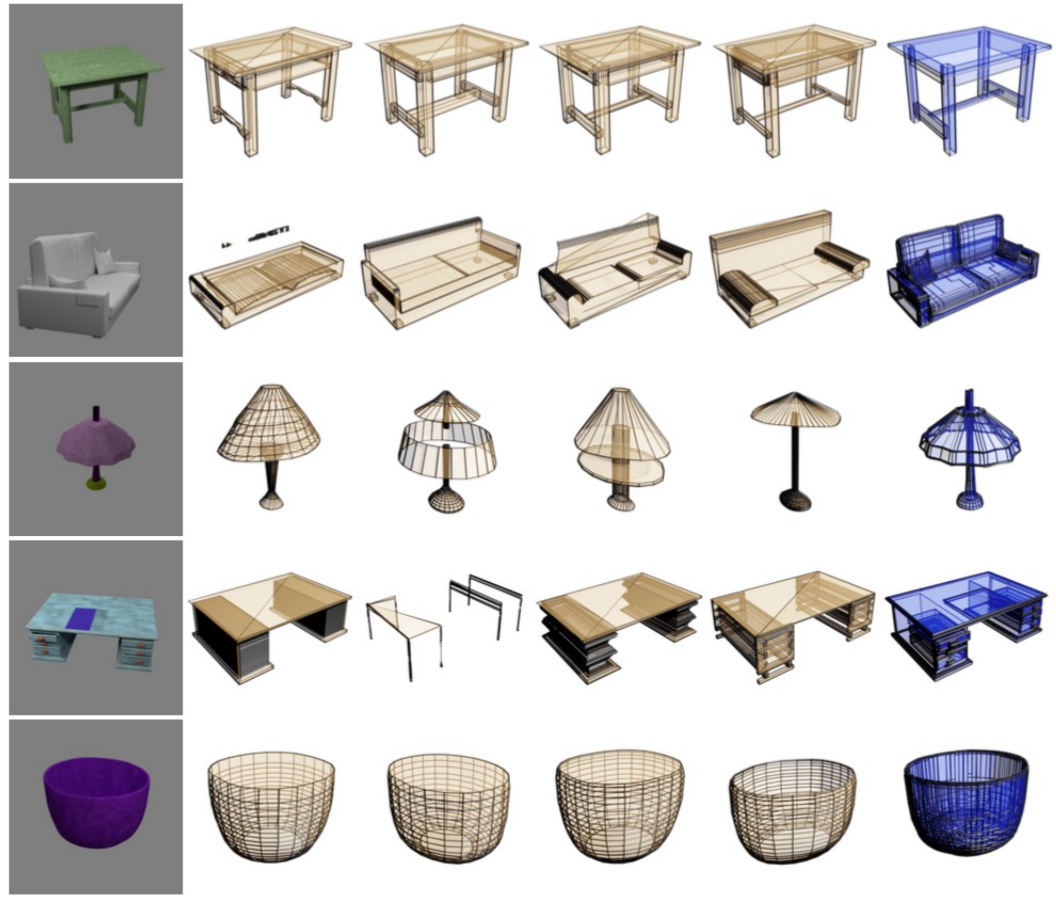


Figure 6. Image conditional samples (yellow) generated using nucleus sampling with top-p=0.9 and ground truth meshes (blue).

Discussion of results

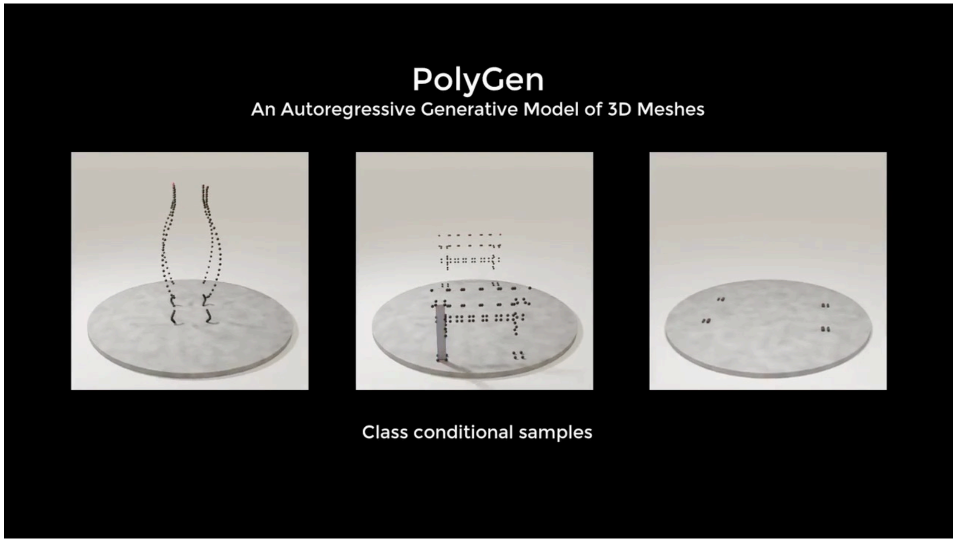


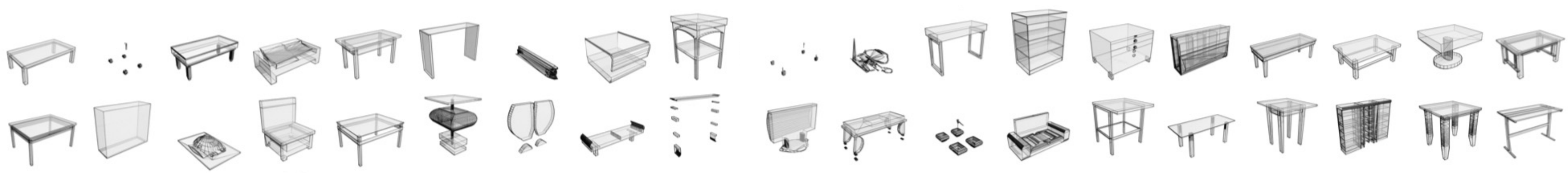
Figure 10. Voxel conditional (blue, left) samples generated using nucleus sampling with top-p=0.9 (yellow) and ground truth meshes (blue, right).

Experimental Results

- Mesh Generation
  - Best model achieves:
    - log-likelihood 4.26 bits/vertex
    - 85% prediction for Vertices
    - 90% prediction for faces
  - Accuracy of next vert coordinate & choice of vertex for next face
  - No great metrics so…

Negative log-likelihood

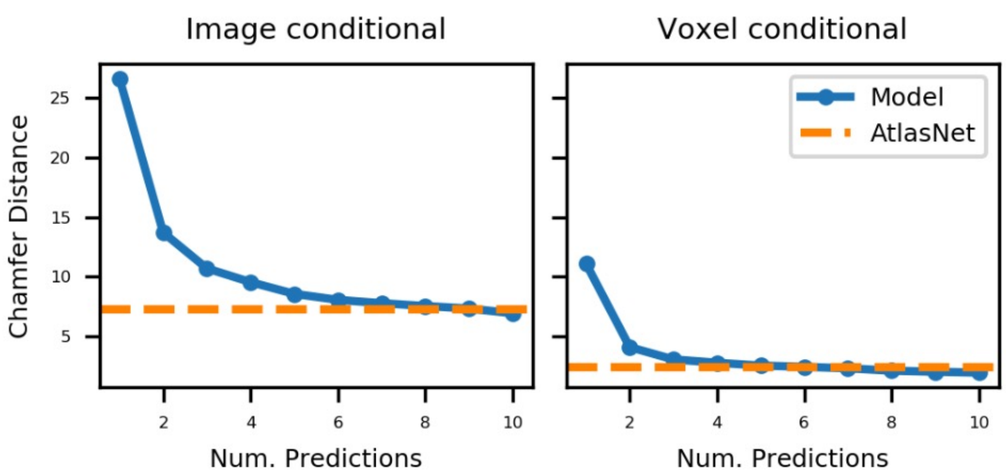
	Bits per vertex		Accuracy	
Model	Vertices	Faces	Vertices	Faces
Uniform	24.08	39.73	0.004	0.002
Valid predictions	21.41	25.79	0.009	0.038
Draco* (Google)	Total: 27.68		-	-
PolyGen	2.46	1.79	0.851	0.900
- valid predictions	2.47	1.82	0.851	0.900
- discr. embed. (V)	2.56	-	0.844	-
- data augmentation	3.39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



Experimental Results vs. Mesh Reconstruction

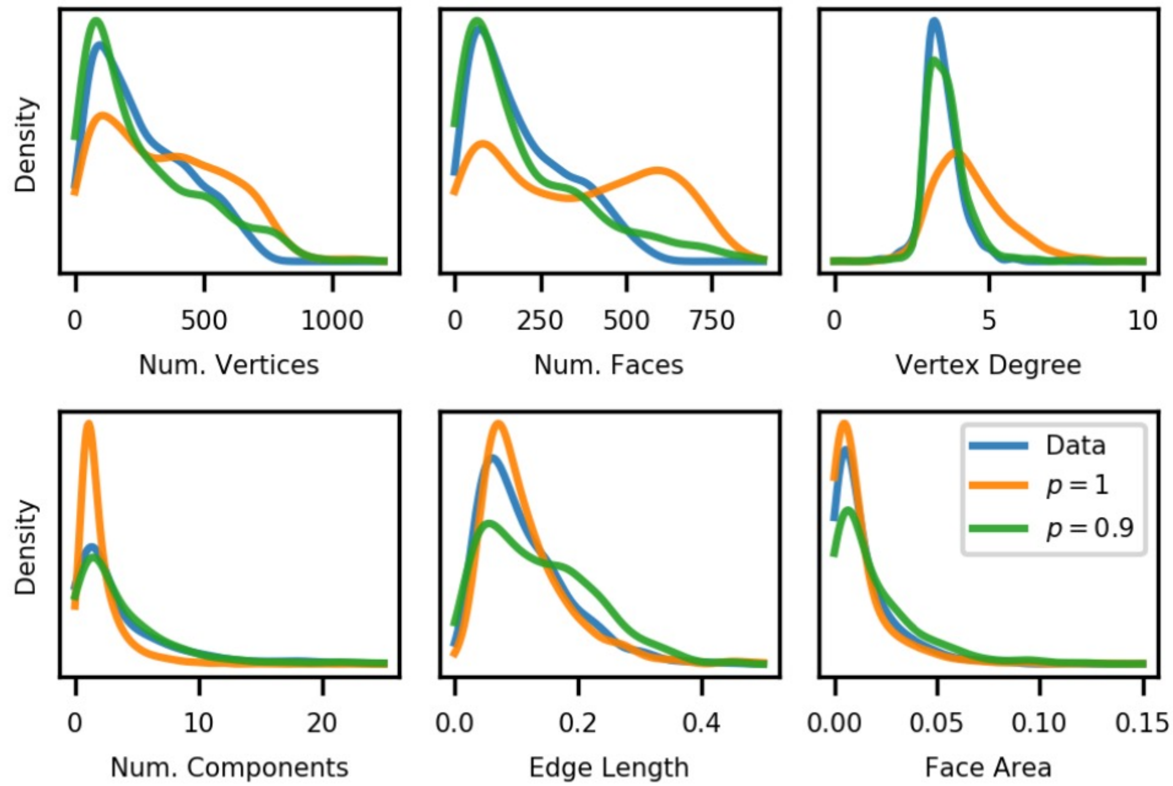
Symmetric Chamfer Distance between two point sets

$$\mathcal{L}(\boxed{\mathcal{P}}, \boxed{\mathcal{Q}}) = \sum_{\boxed{\mathbf{p}\in\mathcal{P}}} \min_{\mathbf{q}\in\mathcal{Q}} (\mathbf{p}-\mathbf{q})^2 + \sum_{\boxed{\mathbf{q}\in\mathcal{Q}}} \min_{\mathbf{p}\in\mathcal{P}} (\mathbf{p}-\mathbf{q})^2$$



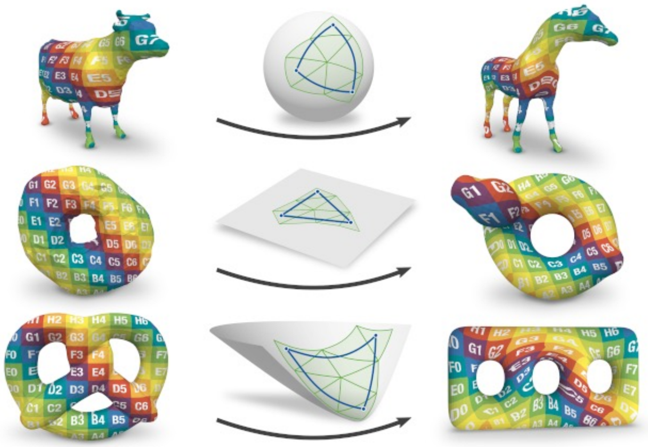
Experimental Results

- But are the meshes any good?
- Some crude statistics…
- Blue line is ground truth from ShapeNet



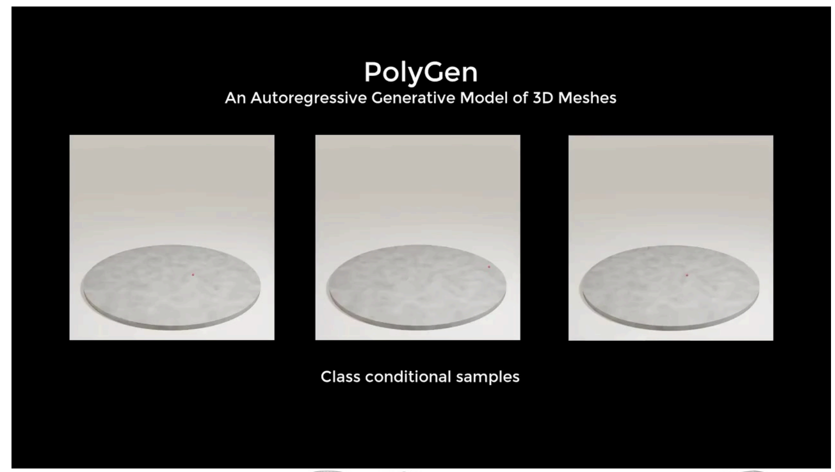
Critique / Limitations / Open Issues

- Transformers are expensive to train
  - Attention is quadratic in nature $O(L^2)$
    - New mechanism is O(L log(L)) ! <Reformer: The efficient Transformer>
  - Can limit mesh sizes (Scalability <vertex history>)
    - Don’t know what the limits are… not in paper, or code…
    - Quantization hides some of this
    - Need better quality metrics for understanding statistical variations
- Rotations
- Unclear on performance with generalized curvature
- No new benchmark metrics (yet)…
- Generation could be slow since it’s sequential
- Non-planar n-gons (probably filtered out organic/curvy things!)
- Low res 256x256x256 mesh/voxel coords.



Contributions recap

- First: Unconditional mesh vertex & face models create polygon meshes!
- Demonstrated conditional generation given
  - Object Class
  - Image
  - Voxels
- Novel application of Transformers to Meshes
  - vertex & face distributions, robust to bad input
- Output is diverse, realistic and directly usable in graphics applications – unlike previous post-processed output



Thank You!