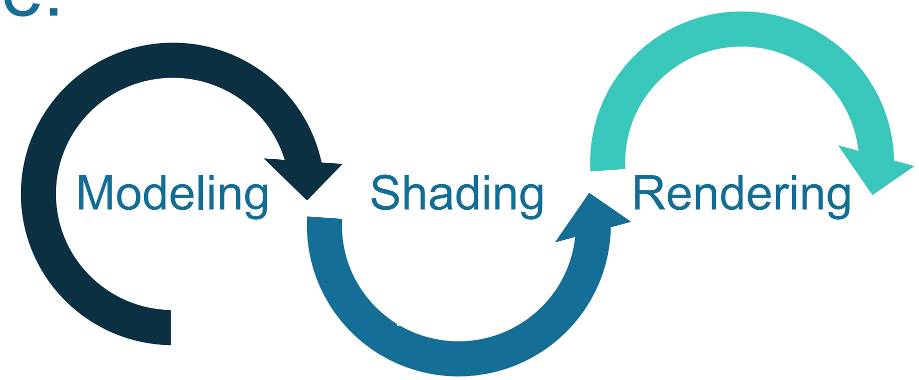
Computer Vision Applications in 3D Graphics and Visualization

Speaker: Claudia Melis Tonti

ALCOR Lab

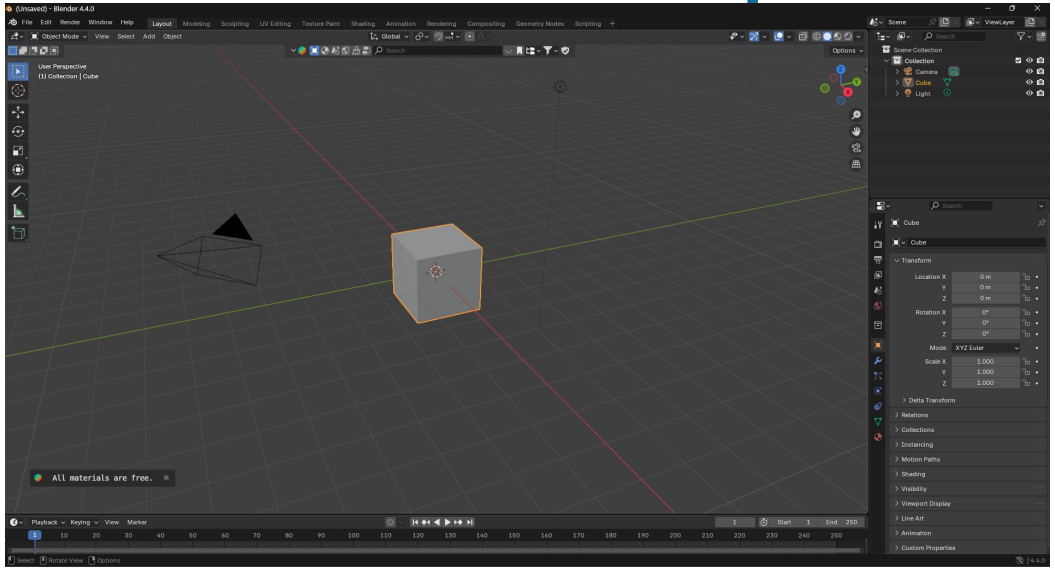
Let's start from classics

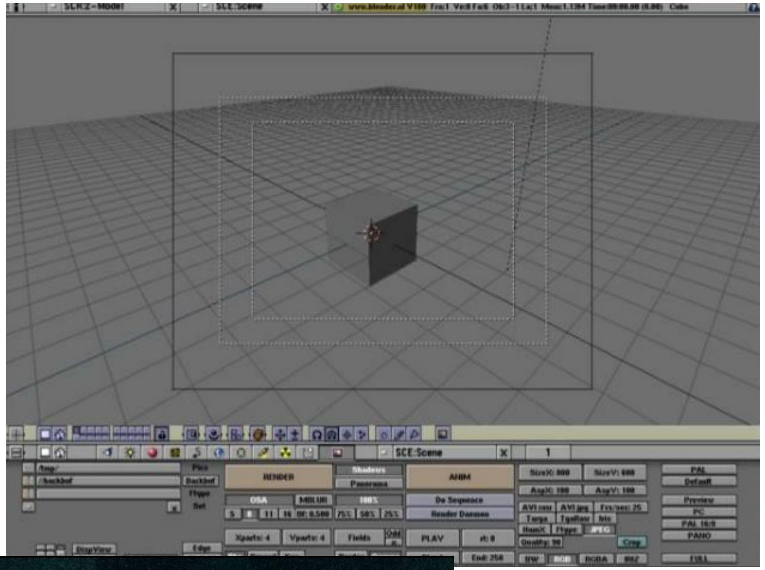
- Manual modeling (CAD, Maya, Unreal Engine, Blender, Cinema4D, Unity…)
- Physics-based rendering (ray tracing)
- Traditional pipeline:

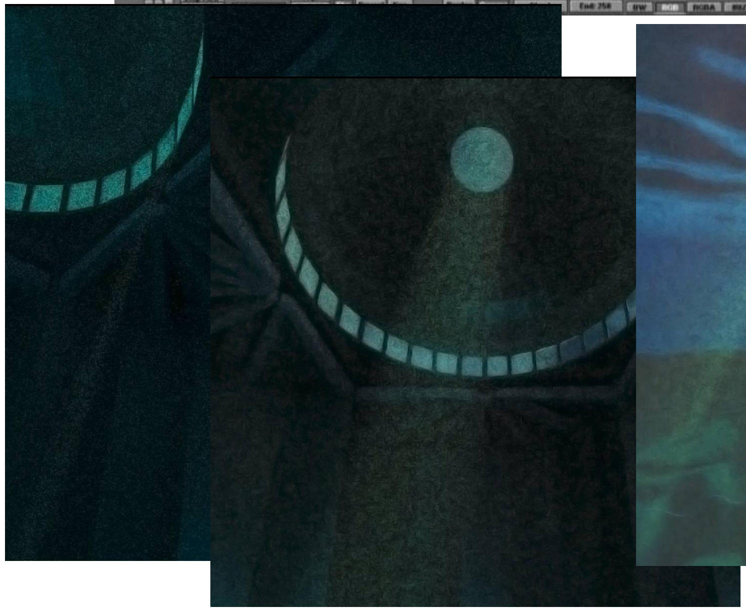


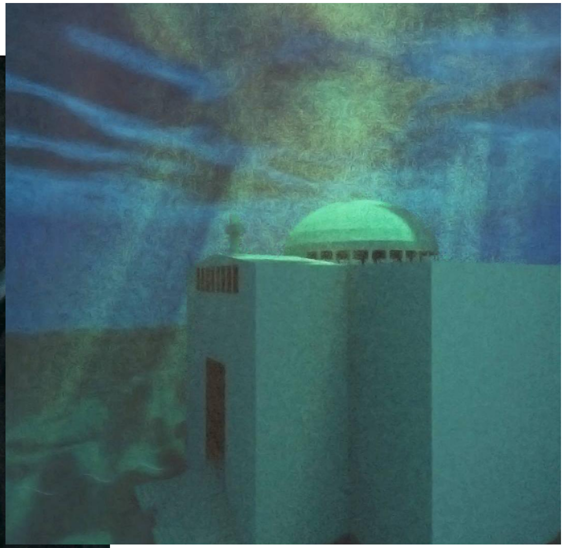
- Non-democratics (need of high-skilled artists and best performing hardware)

Blender examples







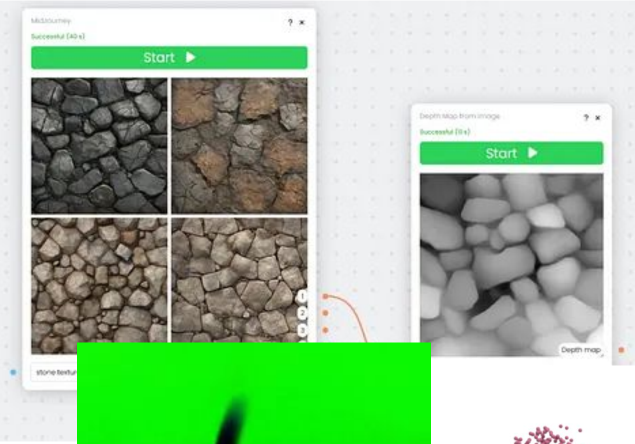


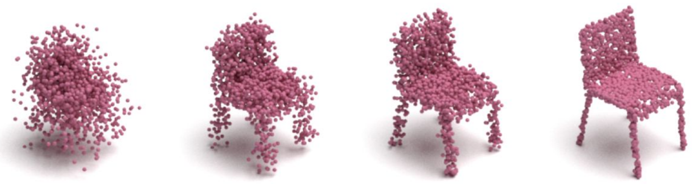
The coming of AI

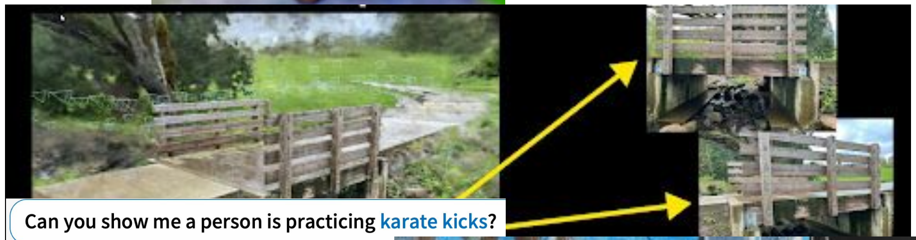
- Generative AI:
  - textures (SDXL, MidJourney, DALL-E, NeRF …),

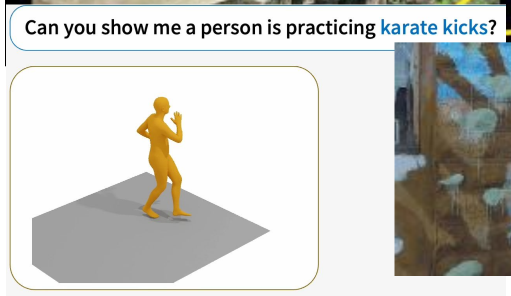
  - meshes (Diffusion Point Cloud, NeRF),
  - environments/scenarios (NeRF),

  - animations (MotionGPT, NeRF)



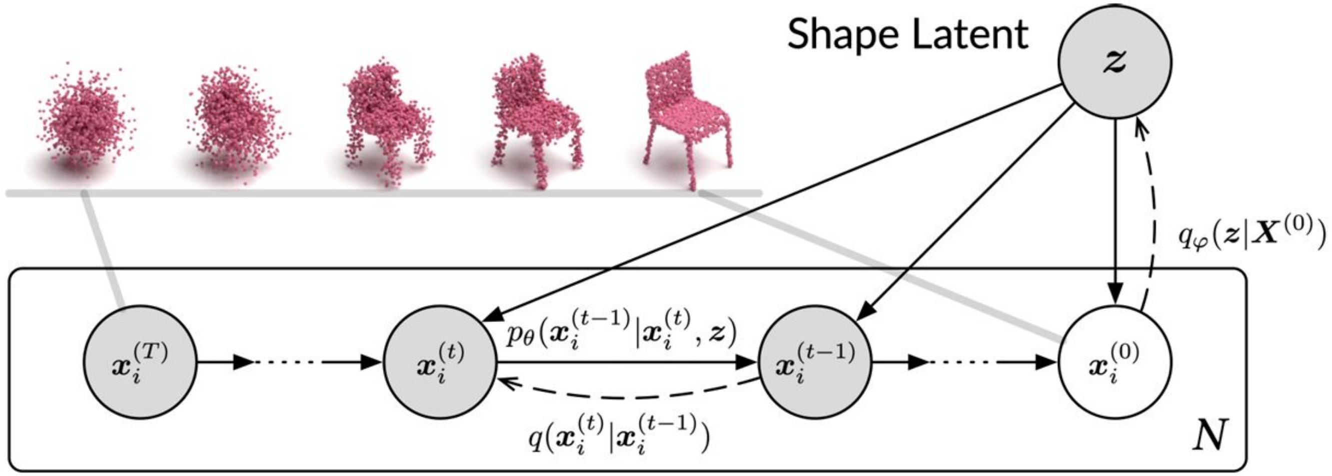




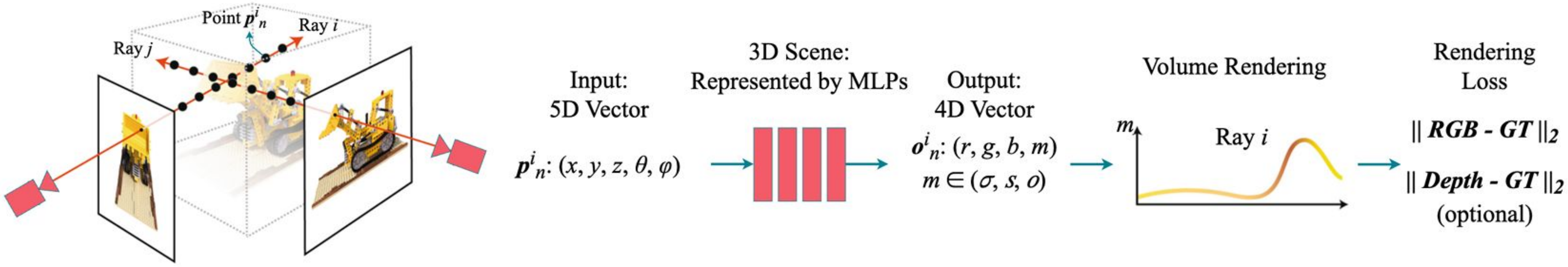


Most used AI models for Computer Graphics

- Diffusion Models



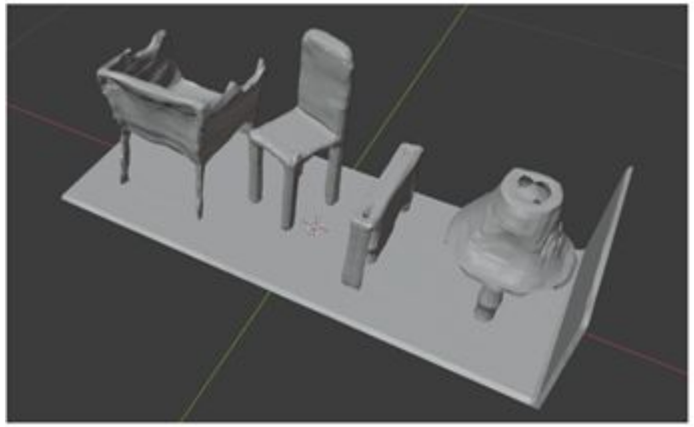
- NeRF



Let's compare

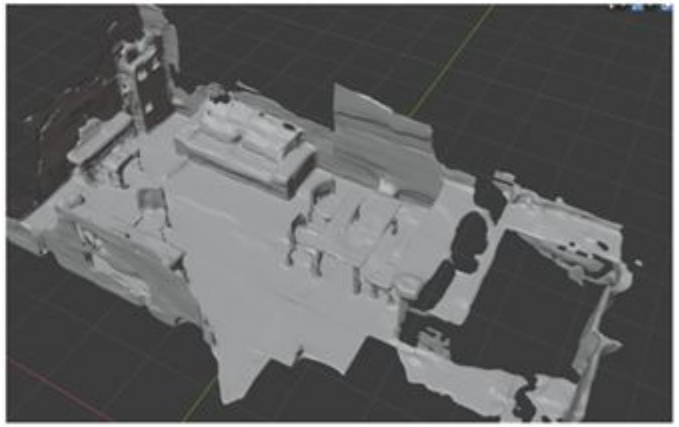
- Scene generation with AI

Dynamic Planes ConvONet



3 dynamic planes : 5 min

Neuralblox [1]



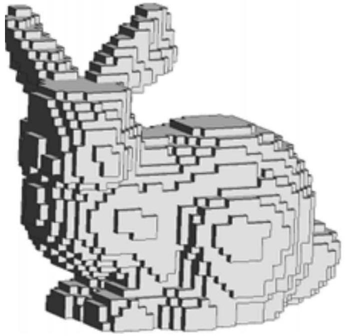
3-4 min

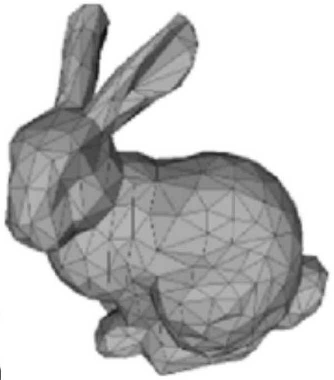
- Scene generation by hand: more or less 2-3 working days

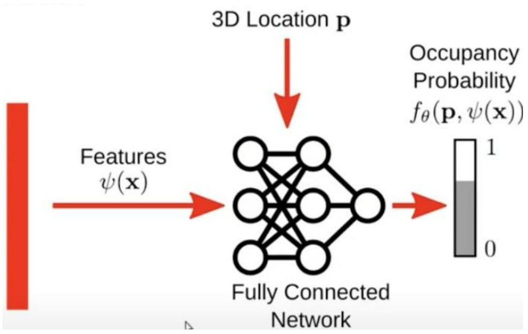
Lionar, S., Schmid, L., Cadena, C., Siegwart, R., and Cra mariuc, A. (2021b). Neuralblox: Real-time neural representation fusion for robust volumetric mapping. In 2021 International Conference on 3D Vision (3DV), pages 1279–1289. IEEE.

But first, an overview of 3D Computer Graphics:
Modeling

- 3D Object representations:
  - Voxels
  - Point Clouds
  - Meshes
  - Implicit Representations

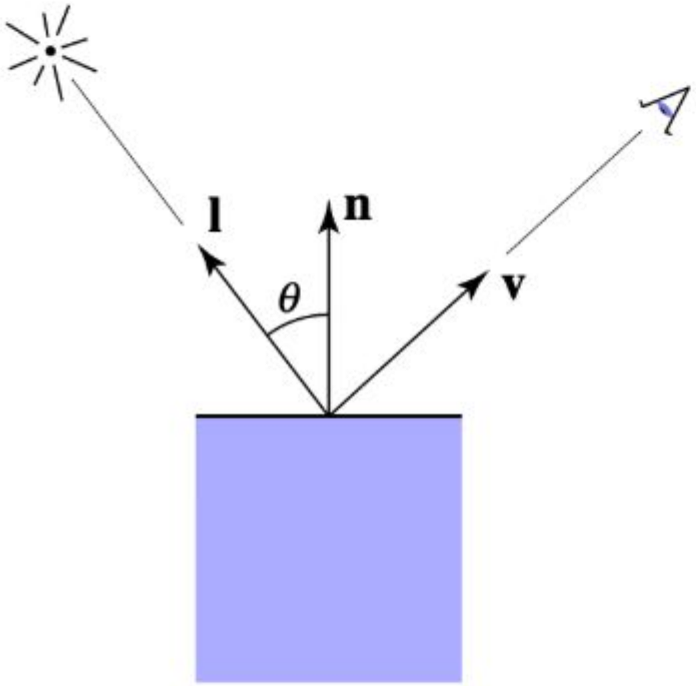


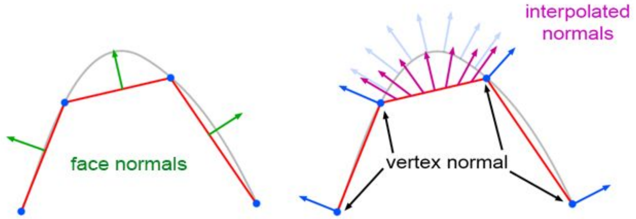


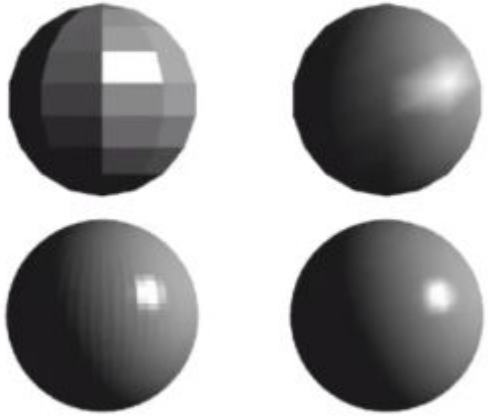


But first, an overview of 3D representations: Shading

- Represents the light bouncing on objects and reaching our eyes
- Gives representation to materials and textures
- Powerful tool to give illusion of high definition to meshes with low level of details

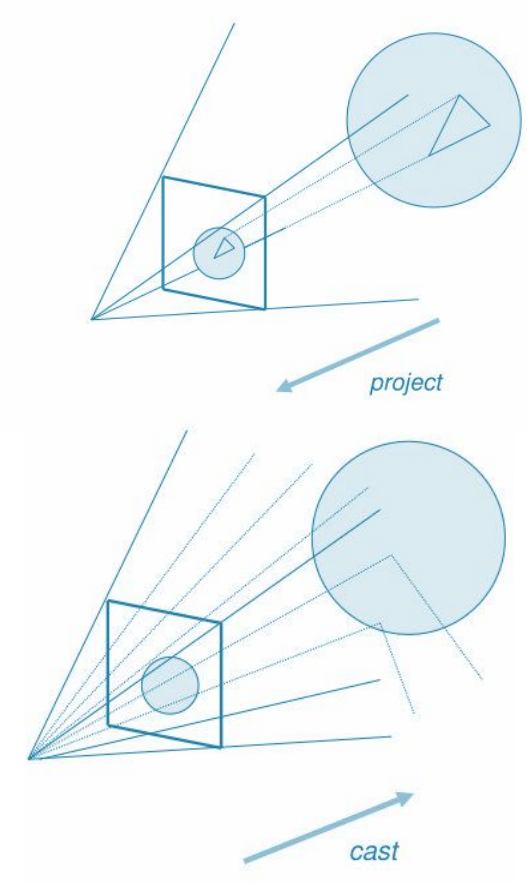




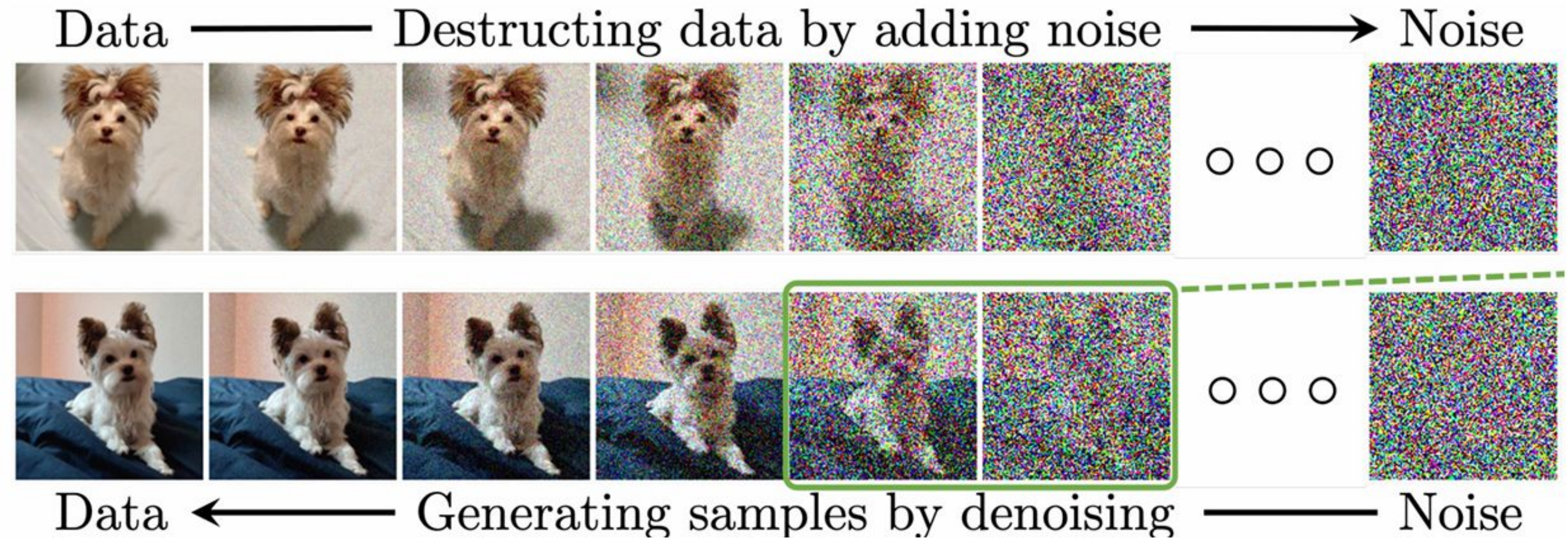


But first, an overview of 3D representations: Rendering

- Process of generation of 2D image from a 3D scene
  - Rasterization
    - Projects polygons onto picture plane
    - efficient
  - Ray-tracing
    - Cast light rays into scene through picture plane
    - accurate



Diffusion Probabilistic Models (DM)



- Idea: using Markov chain to convert noise distribution to data distribution to generate **unconditional** data and images.
- Some of the most recent works use DM to solve **conditional** generation problem

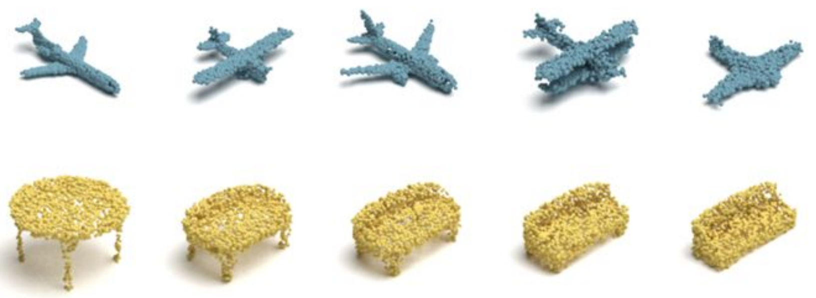
Diffusion Probabilistic Models (DM) [2]

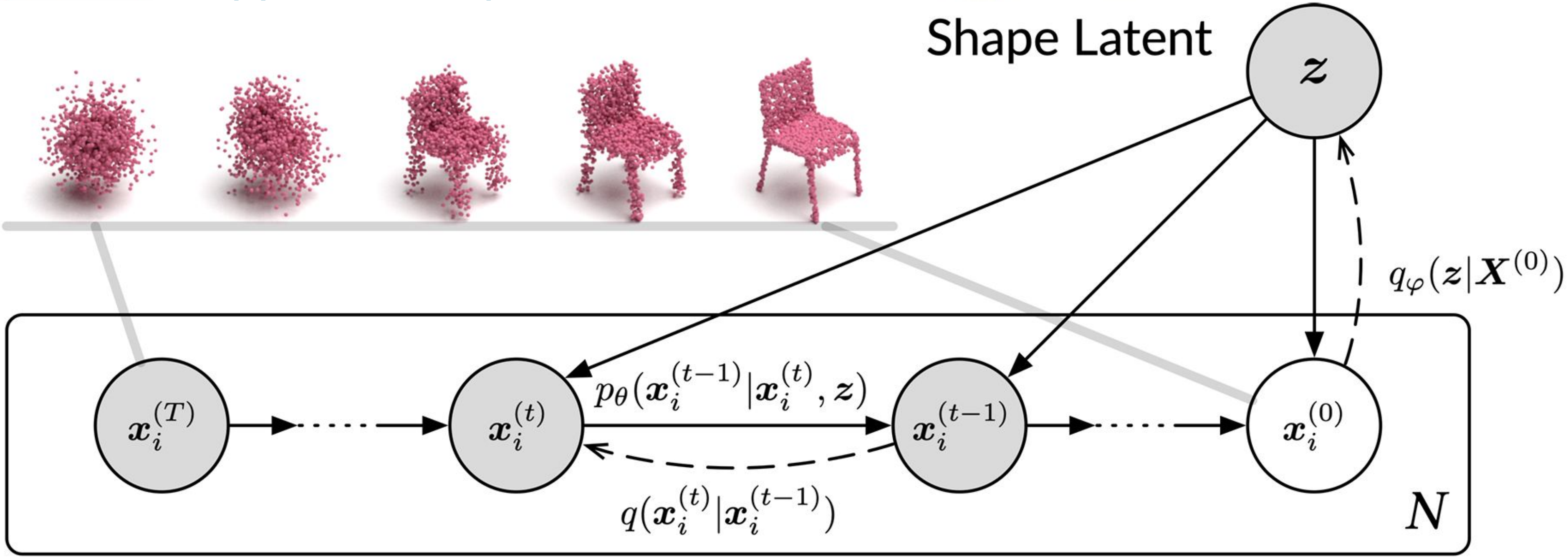
$q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)})$:forward diffusion process

$p_\theta(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)},\boldsymbol{z})$:backward diffusion process

N:number of point cloud points

$q_\varphi(\boldsymbol{z}|\boldsymbol{X}^{(0)})$:approximate posterior distribution



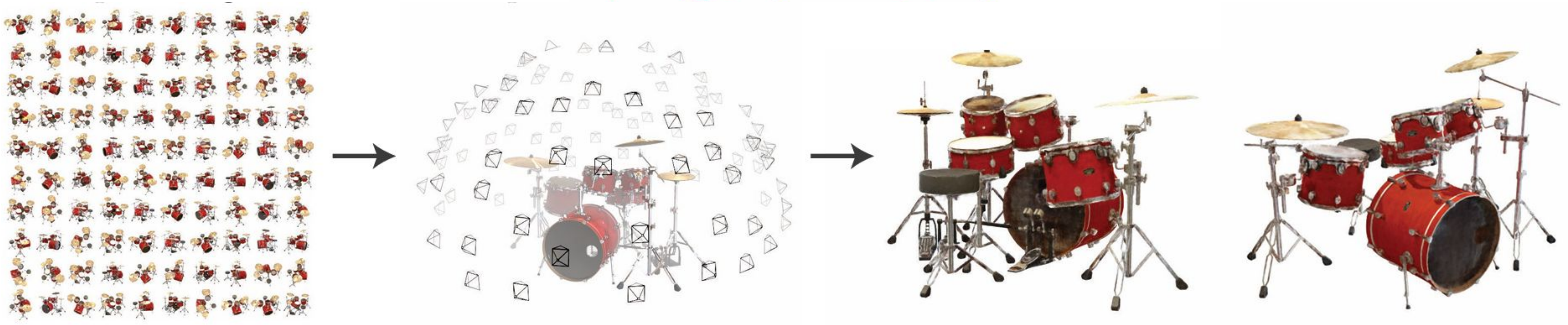


S. Luo and W. Hu, ‘Diffusion probabilistic models for 3D point cloud generation’, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.

NeRF: Neural Radiance Field [3]

- Defines a 3D scene as a continuous volumetric function.
- Given a 3D positior $\mathbf{p} \in \mathbb{R}^3$ in the world system and a view direction $\mathbf{d} \in \mathbb{R}^2$, NeRF outputs the volume dens $\sigma \in \mathbb{R}$ and the emitted r $\mathbf{c} \in \mathbb{R}^3$ e

$$\mathcal{F}_\theta : (\mathbf{p}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$



Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65.1 (2021): 99-106.

What are the common issues?

High computational costly

Require high performing hardware

Not suitable for small and wireless devices (ex. Oculus)

Slow generation of results

Professional artists hate you

What´s left?

- Field of Computer Graphics is highly covered by AI research
- The evolution of Generative models is so fast
- Every new paper talks about a model which is already obsolete
- New models are developed by huge industries (Google, NVIDIA, Amazon…) with an incredible speed

Mesh compression

Inspiration:

Autodesk Maya Dependency Graph

Goal:

avoid overusing RAM for rendering the entire scene when using viewport

Hardware used for scene rendering and visualization:

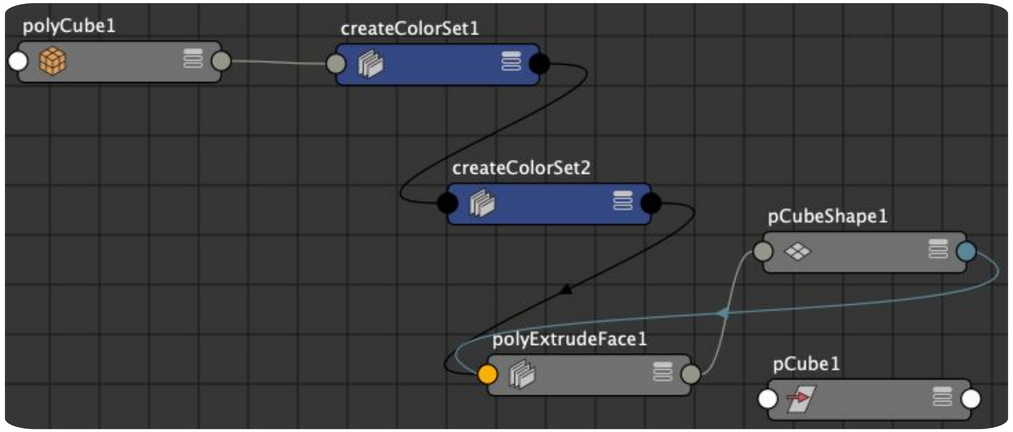
CPU for reading graph

RAM for generating meshes, textures, materials, keyframes

GPU for shading, rendering, deformations and skinning

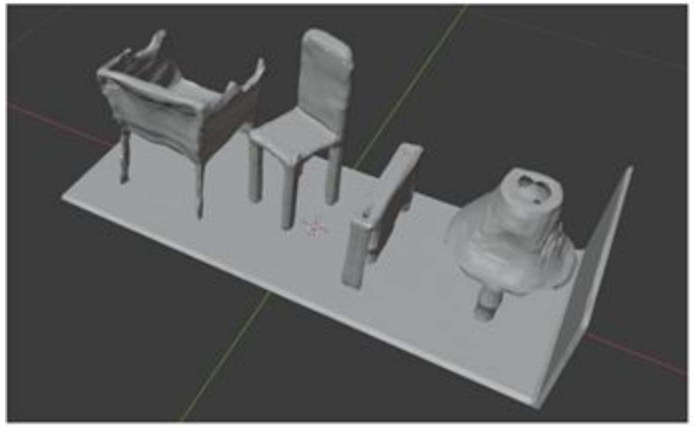
Consequence:

Real-time rendering is overconsuming



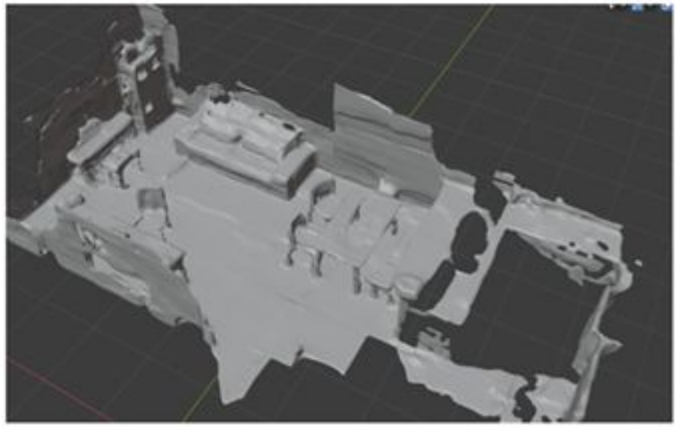
Mesh extraction

Dynamic Planes ConvONet



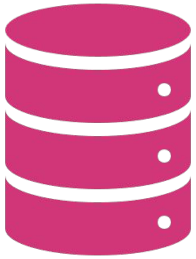
3 dynamic planes : 5 min

Neuralblox



3-4 min

AI 3D mesh compression

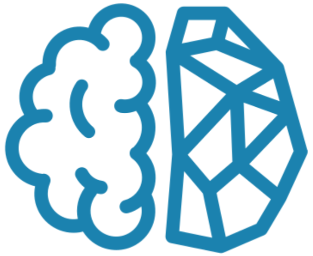


Point Cloud compression:

Group of coordinates

Unordered points

Need information about normals and occupancy



Low-poly meshes compression:

Group of points, edges and faces

Need groups of points to know the order of faces

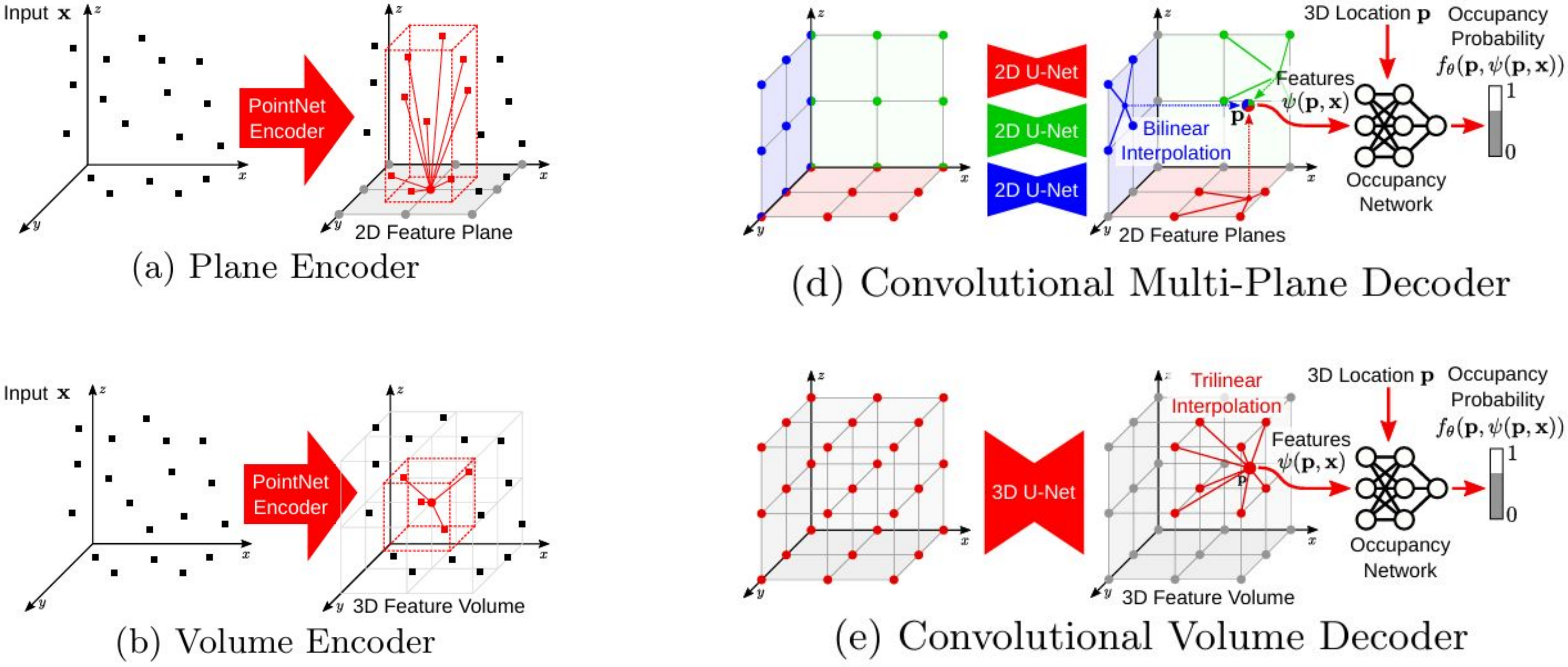
Ordered points and faces

The details can be enhanced with normal, bump and parallax mappings

Point Cloud compression

- Idea: compress a 3D mesh as a sparse point cloud and reconstruct the objects only when needed
- Objectives:
  - Remove as much as possible points from point cloud, keeping only the most informative ones
  - Decrease at minimum the generation time of the mesh

Convolutional Occupancy Networks (ConvONet) [4]



Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In Computer Vision–ECCV 2020: 16th Euro
pean Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 523–540. Springer.

Convolutional Occupancy Networks (ConvONet)

- Metrics

$$MAD_{accuracy} = \frac{\sum_i^n ||p_i - g_i||}{n}$$

$$MAD_{completeness} = \frac{\sum_i^n ||g_i - p_i||}{n}$$

$$IoU = \frac{Volume\ of\ Overlap}{Volume\ of\ Union}$$

$$Chamfer - L1 = \frac{MAD_{accuracy} + MAD_{completeness}}{2}$$

$$F - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- Loss
  - Binary Cross-Entropy between predicted occupancy and ground truth occupancy

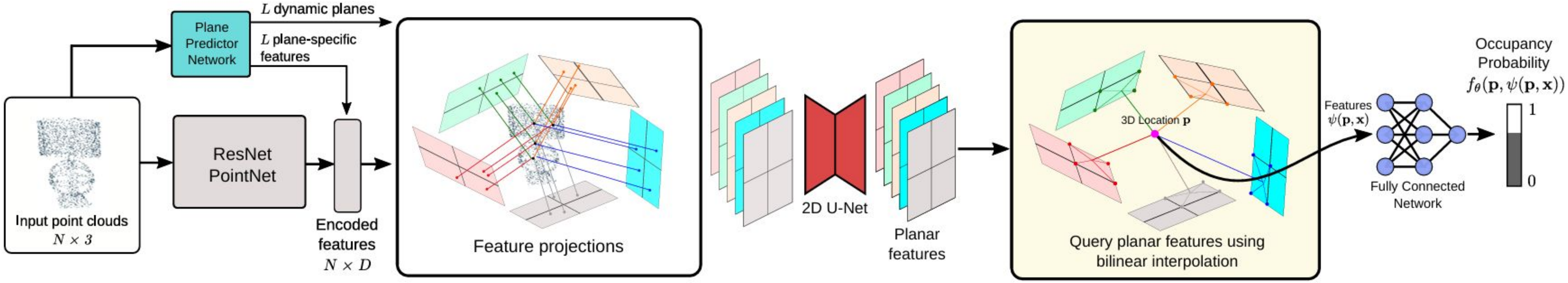
$$\mathcal{L}(\hat{o}_{\mathbf{p}}, o_{\mathbf{p}}) = -[o_{\mathbf{p}} \cdot \log(\hat{o}_{\mathbf{p}}) + (1 - o_{\mathbf{p}}) \cdot \log(1 - \hat{o}_{\mathbf{p}})]$$

- Mesh generation:
  - Multiresolution IsoSurface Extraction (MISE)

Convolutional Occupancy Networks (ConvONet)

- What we need:
  - Point Cloud
  - Normals associated to points
  - Occupancy: points that are inside the mesh
- Pros:
  - It works perfectly without an order of the points
  - Discrete level of details
- Cons:
  - Not enough high level of details
  - High computational demanding
  - Mesh Generation not fast enough

Dynamic Plane ConvONet [5]



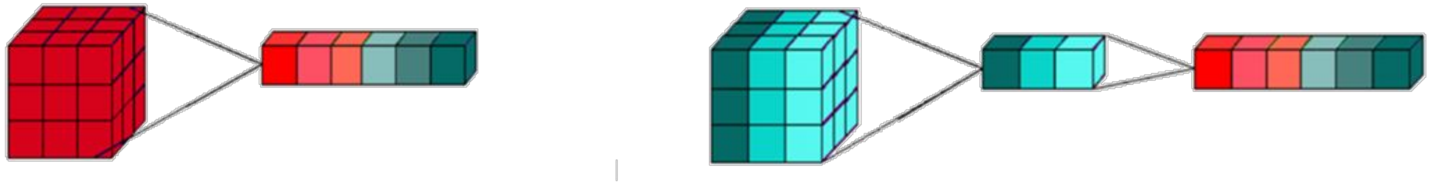
- Projection planes chosen by the encoder
- Pros :
  - Higher level of details (the highest number of planes, the better)
  - Transfer problem from 3D to 2D decreasing the computational cost of the network
- Cons:
  - Not real-time
  - Can only generate small scenes with a low number of meshes

Lionar, S., Emtsev, D., Svilarkovic, D., and Peng, S. (2021a). Dynamic plane convolutional occupancy net works. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1829–1838.

Fine Tuning is the answer: Light ConvONet [6]

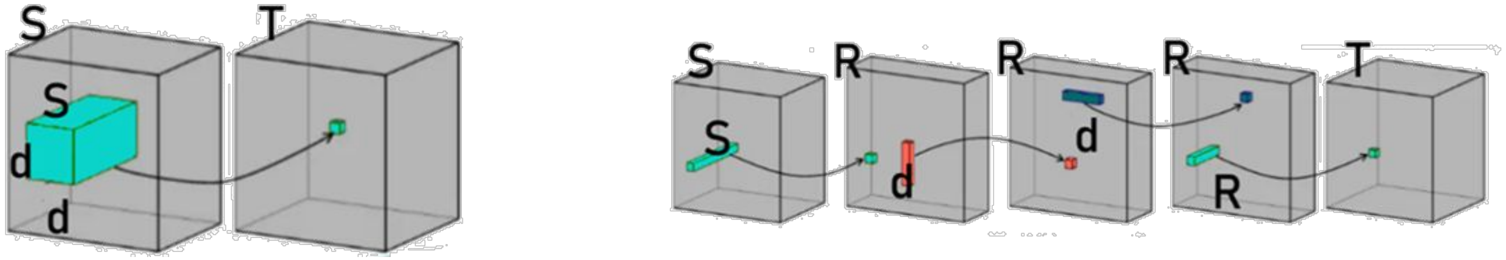
- Depth-wise

separable convolutions [7]



+

- CP-Decomposition [8]



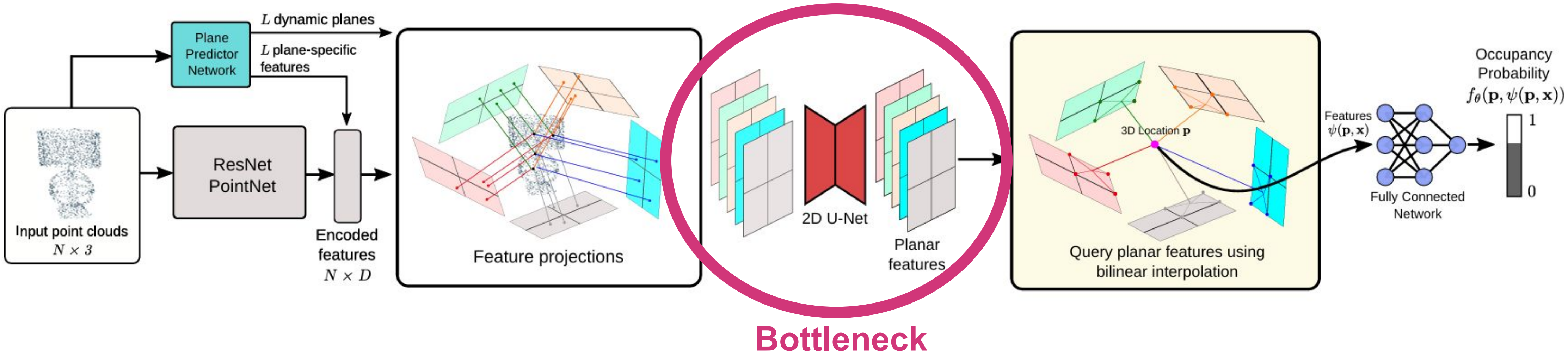
+

- Pruning [9] =

Model	Completeness ↓	Accuracy ↓	Chamfer-L_1 ↓	F-score ↑	IoU ↑	inference time ↓
DPConvONet [15]	**0.0047**	**0.0039**	**0.0043**	**0.9433**	**0.8879**	0.86 s/mesh
Ours	0.0058	0.0048	0.0053	0.9037	0.8499	0.48 s/mesh

Tonti, C. M., Papa, L., and Amerini, I. (2024). Lightweight 3-D Convolutional Occupancy Networks for Virtual Object Reconstruction . IEEE Computer Graphics and Applications, 44(02):23–36.

Fine Tuning is the answer



- Can we have only one plane to process?
- Transformers have enough capacity of maintaining both local and global information
- Issue: Transformers are computational demanding

Fine Tuning is the answer: TransONet [10]

Solution: use a lightweight Vision Transformer

Dynamic Vision Transformer [11]:

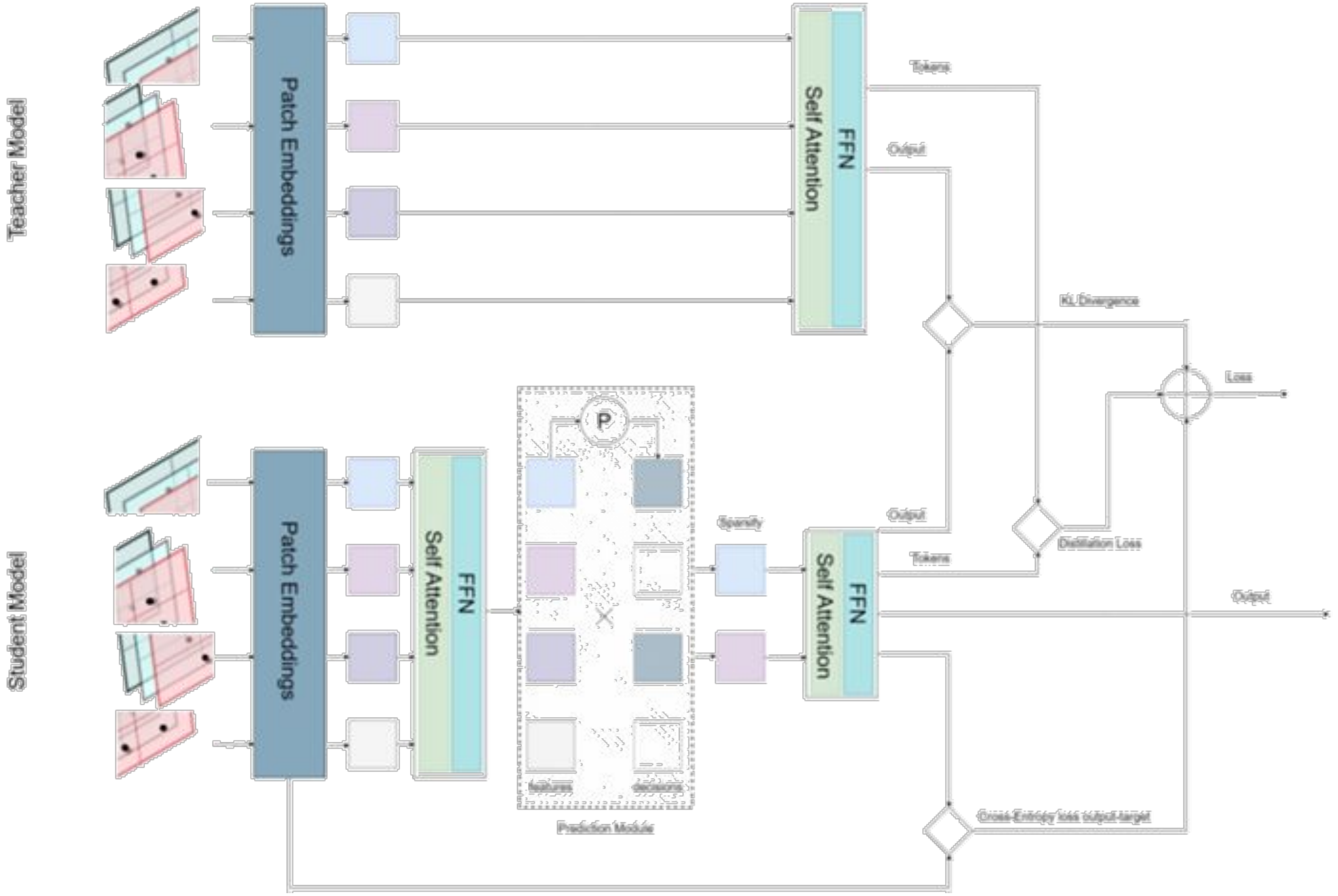
- Uses knowledge distillation to prune least necessary weights

- Losses:

$$\mathcal{L}_{ce} = CrossEntropy(\mathbf{y}, \bar{\mathbf{y}})$$

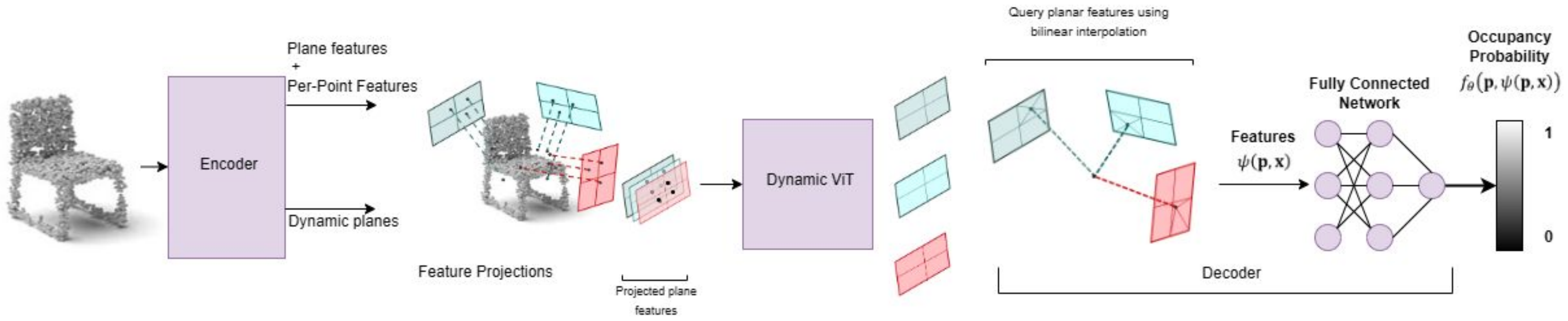
$$\mathcal{L}_{KL} = KL(\mathbf{y}||\mathbf{y}')$$

$$\mathcal{L}_{distill} = \frac{1}{\sum_{b=1}^{B} \sum_{i=1}^{N} \hat{\mathbf{D}}_i^{b,S}} \sum_{b=1} B \sum_{i=1} N \hat{\mathbf{D}}_i^{b,S}(\mathbf{t}_i - \mathbf{t}'_i)$$



C. Tonti and I. Amerini, ‘Lightweight transformer occupancy networks for 3D virtual object reconstruction’, in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Porto, Portugal, 2025, pp. 408–414.

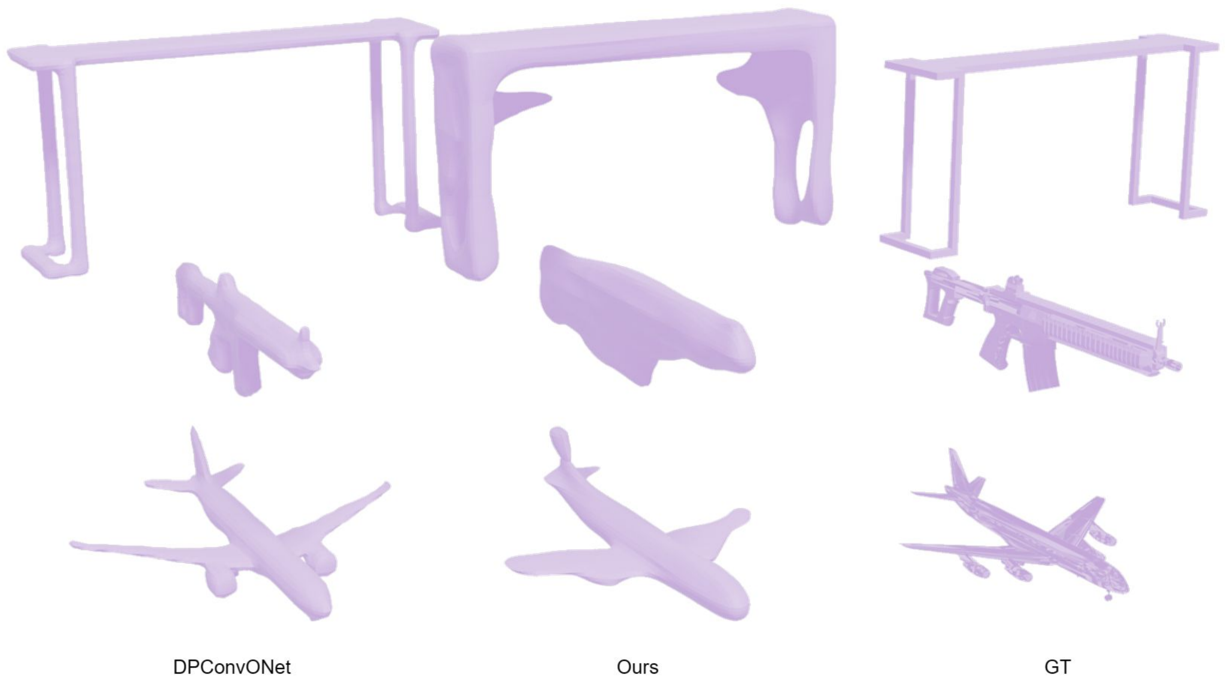
Fine Tuning is the answer: TransONet pipeline



- Results:

Model	Accuracy ↓	Completeness ↓	Chamfer-L_1 ↓	F-score ↑	IoU ↑	Inference Time ↓
DP ConvONet	**0.0047**	**0.0039**	**0.0043**	**0.9433**	**0.8879**	0.86 s/mesh
Light DP ConvONet	0.0058	0.0048	0.0053	0.9037	0.8499	0.48 s/mesh
Our	0.0342	0.0263	0.0302	0.3297	0.4924	**0.40** s/mesh

Quick survey



- Which do you think is more important: faster generation or higher visual quality of the mesh?

Occupancy Networks Issues

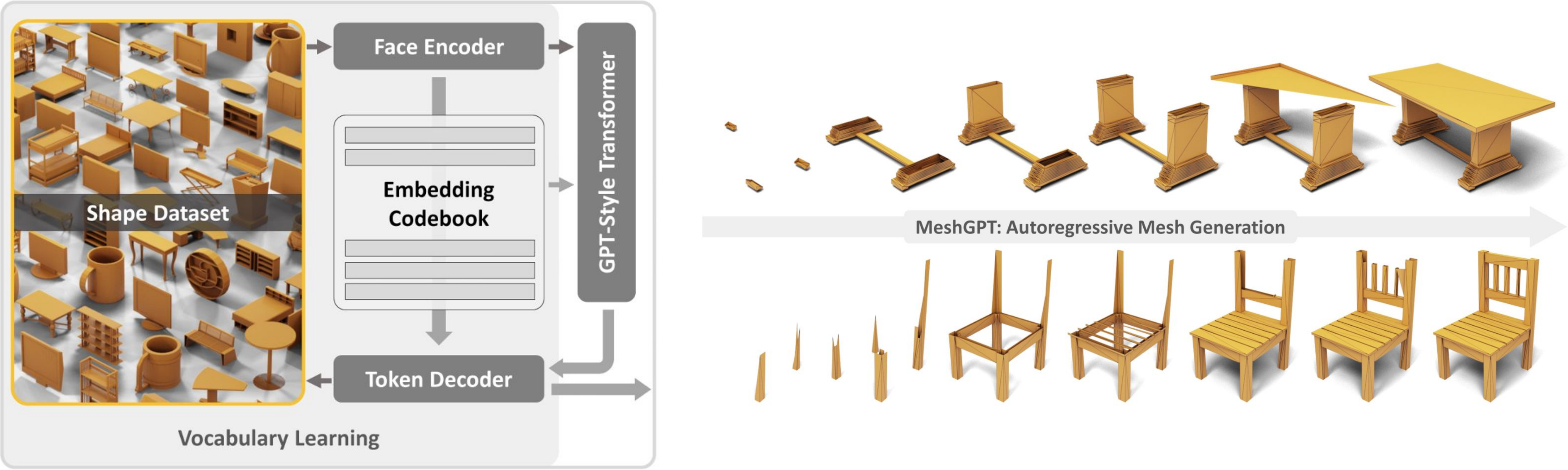
Biased by the dataset

Need much information (points, occupancy, normals)

Mesh have lots of faces and vertices, so rendering it may be computational demanding

MeshGPT [12]

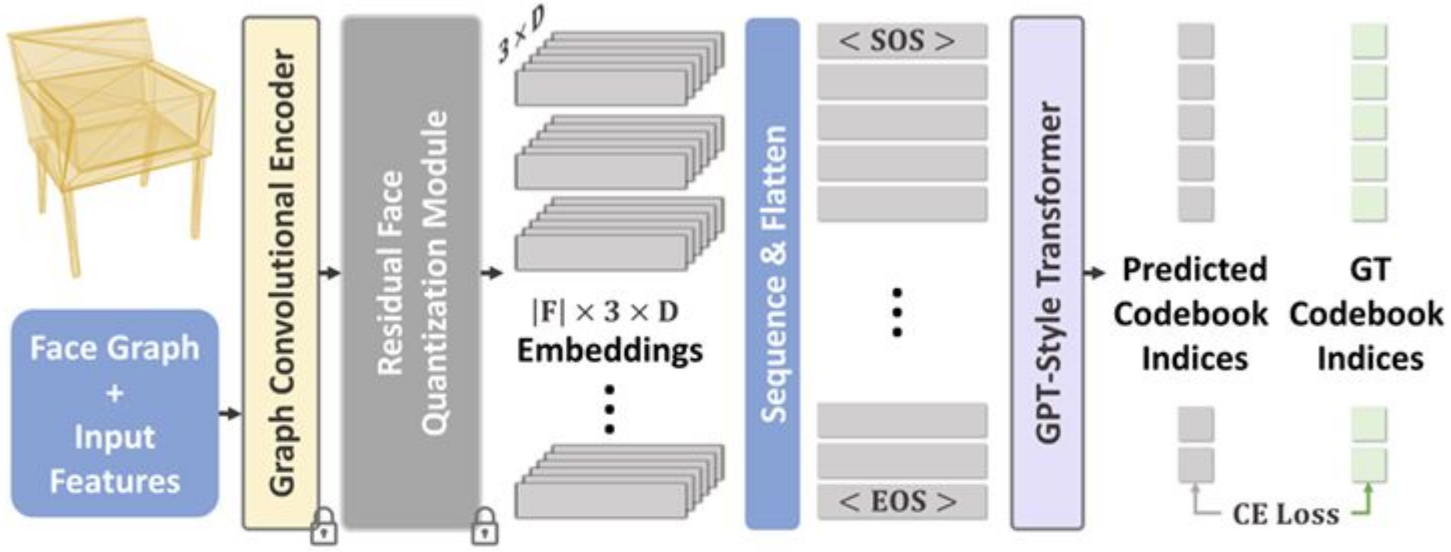
- Idea: use a transformer architecture to learn vertices and faces distribution of a class of meshes to generate new shaped 3D object of that class.



Siddiqui, Yawar, et al. "Meshgpt: Generating triangle meshes with decoder-only transformers." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024.

MeshGPT

- GPT (Generative-Pretrained Transformer): decoder-only transformer. During training, a graph encoder extracts features from mesh faces, which are quantized into a set of face embeddings. The decoder predicts the subsequent codebook index for each embedding, optimized via cross-entropy loss.



Let's talk about the project

- Fine-tuned ConvONet for object reconstruction: TransONet
- Modify it to make it work for the scene dataset (synthetic_room_dataset) or to enhance the quality on object dataset (ShapeNet). You can add layers and architectures but always keep an eye to inference time.
- Make some experiments and compare the results with
- Github repository with instruction to install everything: ALCOR-Lab-DIAG/TransONet
- Contact me for questions at the email: melistonti@diag.uniroma1.it

References

1. Lionar, S., Schmid, L., Cadena, C., Siegwart, R., and Cra mariuc, A. (2021b). Neuralblox: Real-time neural representation fusion for robust volumetric mapping. In 2021 International Conference on 3D Vision (3DV), pages 1279–1289. IEEE.
2. S. Luo and W. Hu, ‘Diffusion probabilistic models for 3D point cloud generation’, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.
3. Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65.1 (2021): 99-106.
4. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 523–540. Springer.
5. Lionar, S., Emtsev, D., Svilarkovic, D., and Peng, S. (2021a). Dynamic plane convolutional occupancy net works. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1829–1838.
6. Tonti, C. M., Papa, L., and Amerini, I. (2024). Lightweight 3-D Convolutional Occupancy Networks for Virtual Object Reconstruction . IEEE Computer Graphics and Applications, 44(02):23–36.
7. V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, “Speeding-up convolutional neural networks using fine-tuned CP-decomposition,” in Proc. 3rd Int. Conf. Learn. Representations, 2015, pp. 1–11.
8. T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” SIAM Rev., vol. 51, no. 3, pp. 455–500, 2009.
9. K. Persand, A. Anderson, and D. Gregg, “Composition of saliency metrics for pruning with a myopic oracle,” in Proc. IEEE Symp. Ser. Comput. Intell., 2020, pp. 753–759.
10. C. Tonti and I. Amerini, ‘Lightweight transformer occupancy networks for 3D virtual object reconstruction’, in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Porto, Portugal, 2025, pp. 408–414.
11. Rao, Y., Liu, Z., Zhao, W., Zhou, J., and Lu, J. (2023). Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intel ligence, 45(9):10883–10897.
12. Siddiqui, Yawar, et al. "Meshgpt: Generating triangle meshes with decoder-only transformers." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024.