Deep Learning for Computer Vision

Fall 2024

https://cool.ntu.edu.tw/courses/41702(NTU COOL)
http://vllab.ee.ntu.edu.tw/dlcv.html (Public website)

Sheng-Yu Huang 黃聖喻 (Ph.D. Student)

Yu-Chiang Frank Wang 王鈺強, Professor

Dept. Electrical Engineering, National Taiwan University

2024/11/19

What to Cover Today?

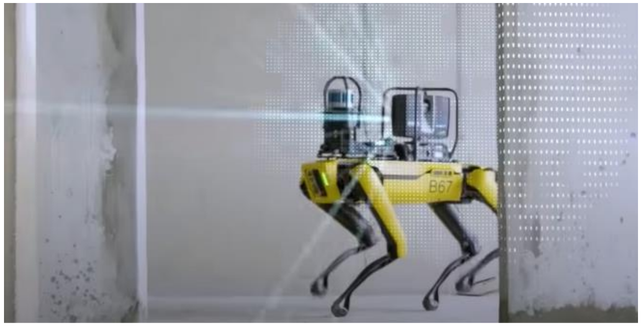
- Introduction to 3D Vision
- Part I: Traditional 3D Representation
    - Perception
    - 3D Reconstruction
- Part II: Recent 3D Representation
    - Neural Radiance Fields
    - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
    - Text-to-3D without 3D supervision
    - 4D Gaussian Splatting

What is 3D Vision?

- Enable machine to perceive and reconstruct the 3D world which we live in.

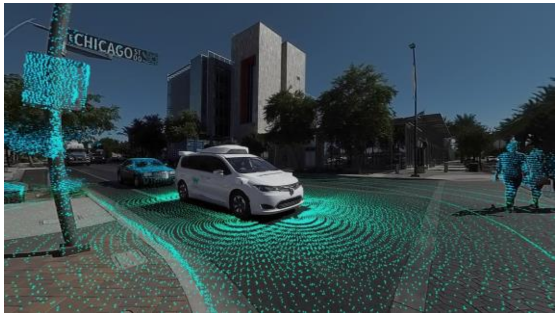
Applications of 3D Vision

- Robotics



- Augmented Reality

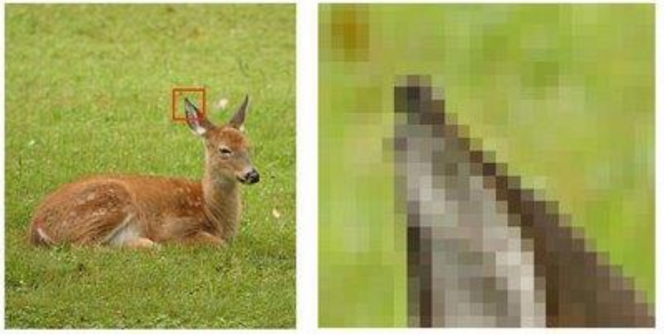
- Autonomous driving



References:
Boston Dynamics: https://www.youtube.com/watch?v=gvzljfK-PiU&ab_channel=BostonDynamics
Ikea: https://www.youtube.com/watch?v=UudV1VdFtuQ&ab_channel=IKEA
Waymo: https://www.youtube.com/watch?v=IzZcqCfA8_k&ab_channel=Waymo

How to Represent the 3D World?

- Recap: 2D representations
    - RGB pixels
    - Images/videos
    - Why 2D vision not good enough?
        - Lack of depth, scene geometry, etc. information



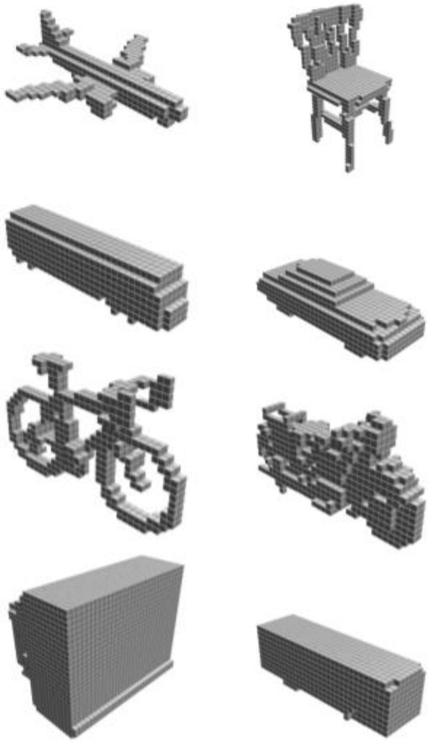
- What about 3D representations?

How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images

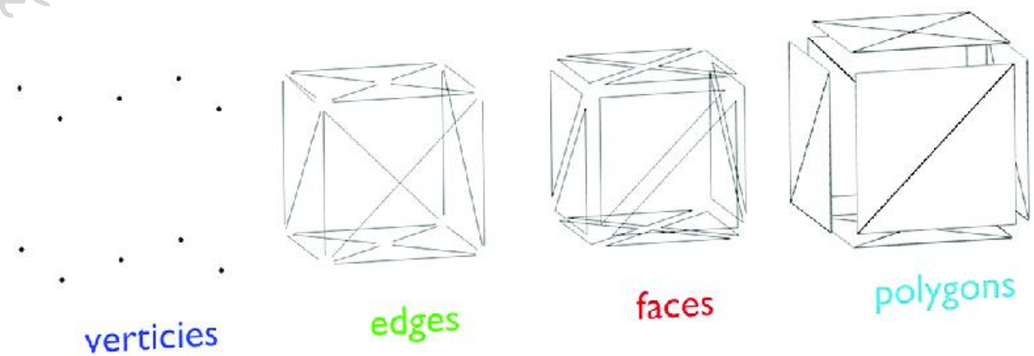
How to Represent the 3D World? (cont'd)

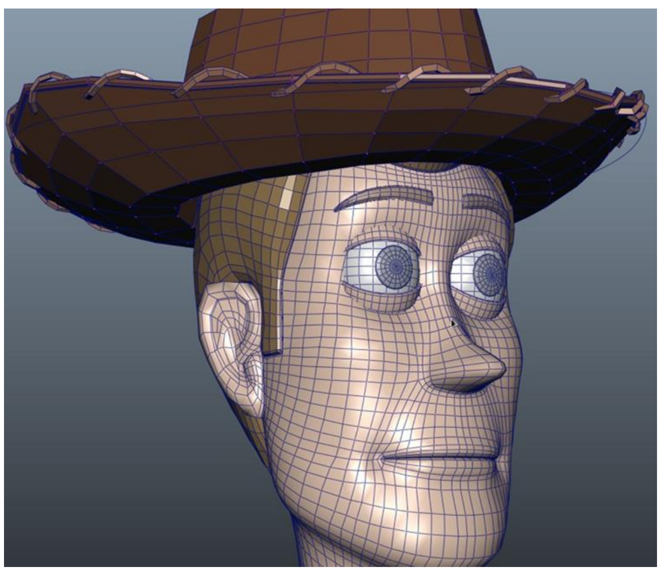
- Multi-view RGB-D images
- Voxels

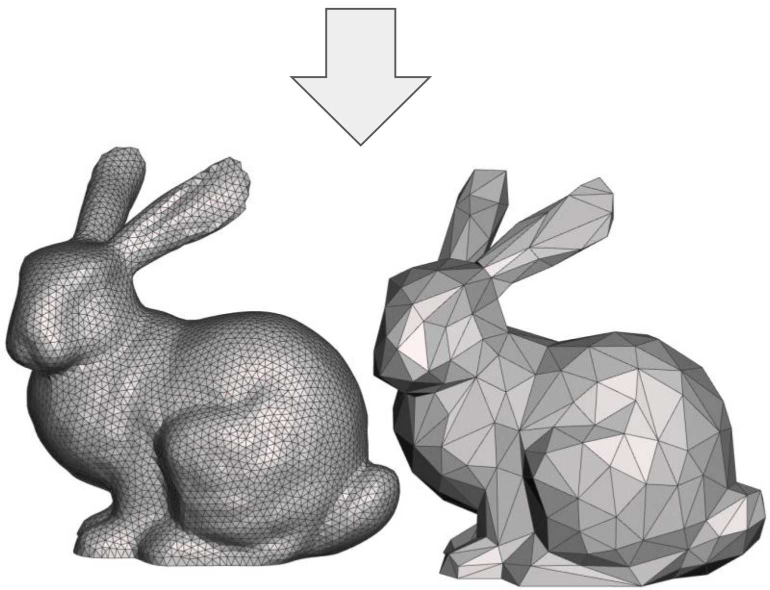


How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images
- Voxels
- Polygon Mesh

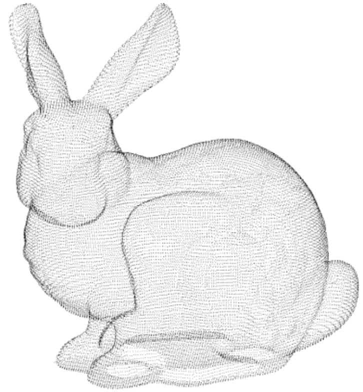


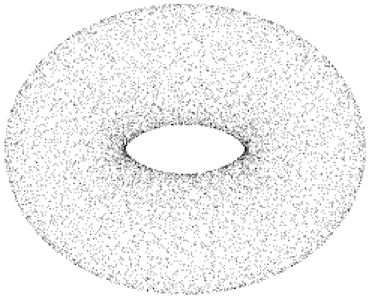




How to Represent the 3D World? (cont'd)

- Multi-view RGB-D images
- Voxels
- Polygon Mesh
- Point Cloud
- …

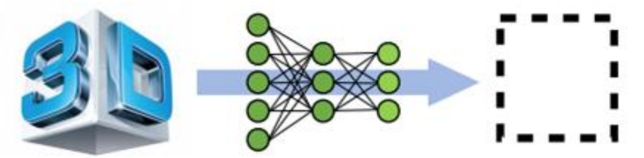




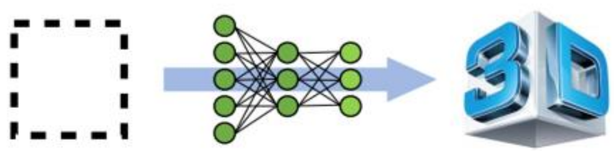
Deep Learning for 3D Vision

Traditional 3D Representation (Part 1)

- **Perception**: extract information from 3D shapes



- **Reconstruction**: synthesis 3D shapes

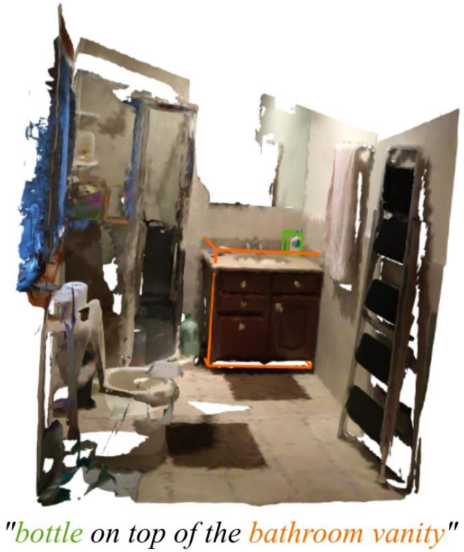


What to Cover Today?

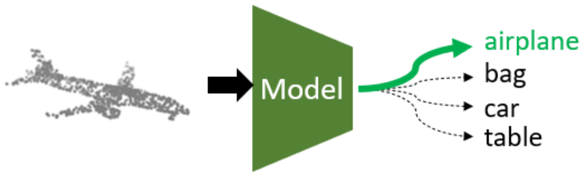
- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian Splatting

3D Perception

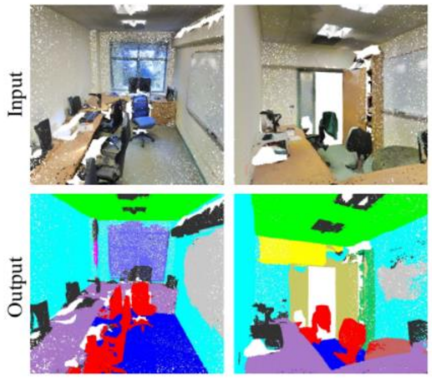
- Extract information from 3D shapes for downstream tasks
  - Classification
  - Object/scene segmentation
  - Pose estimation
  - Object detection
  - Grounding



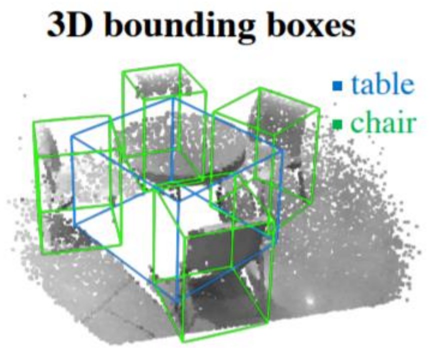
Grounding



Classification



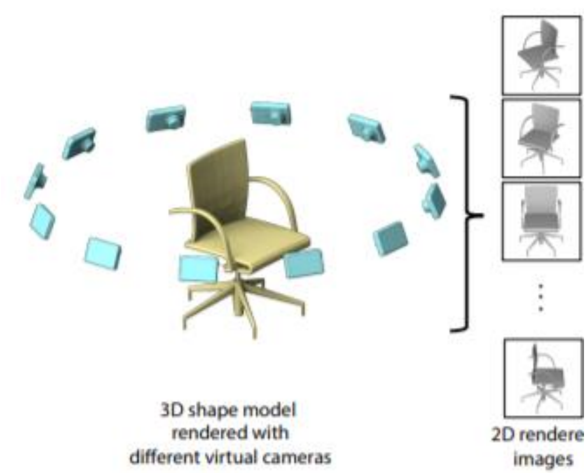
Segmentation

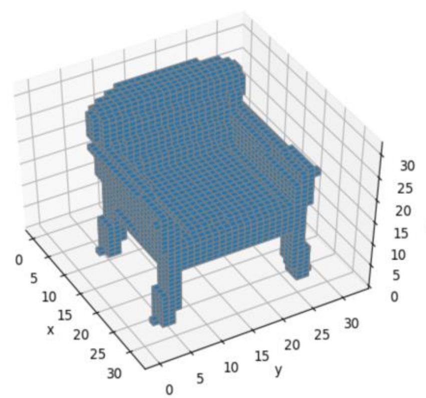


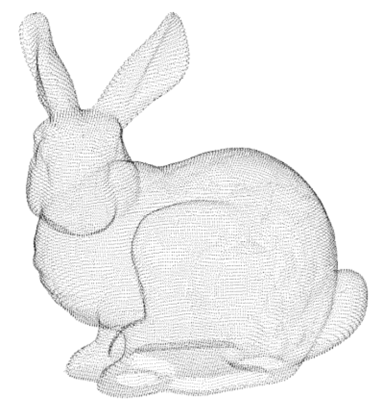
Detection

3D Perception

- In this part, we will talk about feature extraction from:
  - ~~Multi-view images~~
  - ~~Voxel~~
  - Point cloud

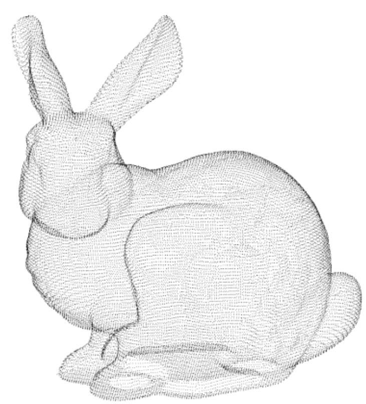


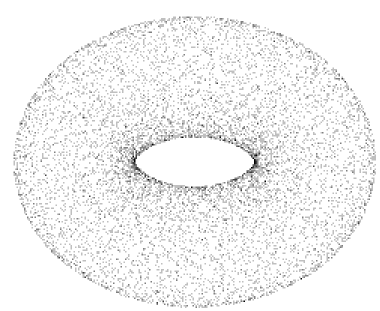


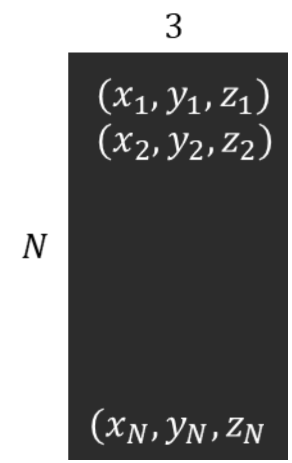


Point Cloud

- Point cloud is a point set, representing 3D shapes
- Each point is represented by coordinates (x, y, z)
- Point cloud is stored as a N×3 matrix (N: point number, 3: coordinates)

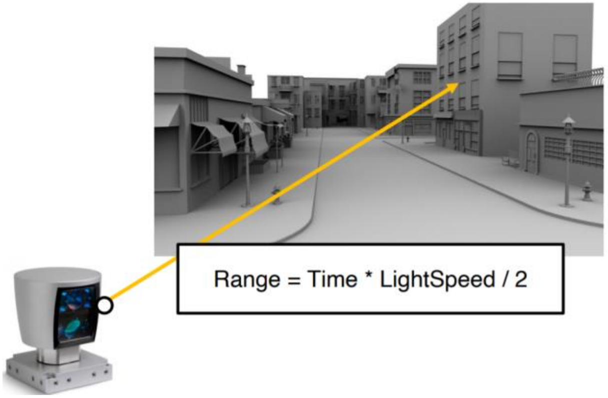






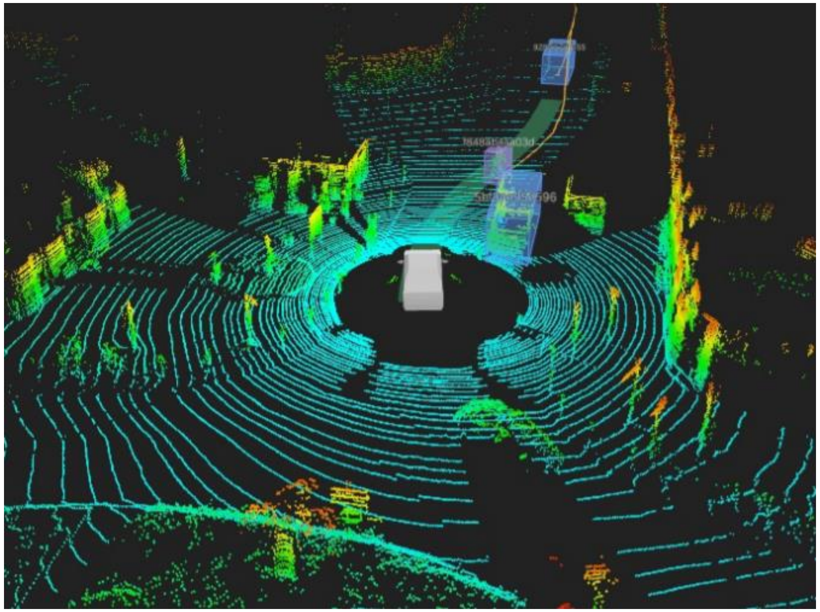
Point Cloud (cont'd)

- Point cloud can be obtained from LiDAR sensors
- Can capture scene geometry



Autonomous driving

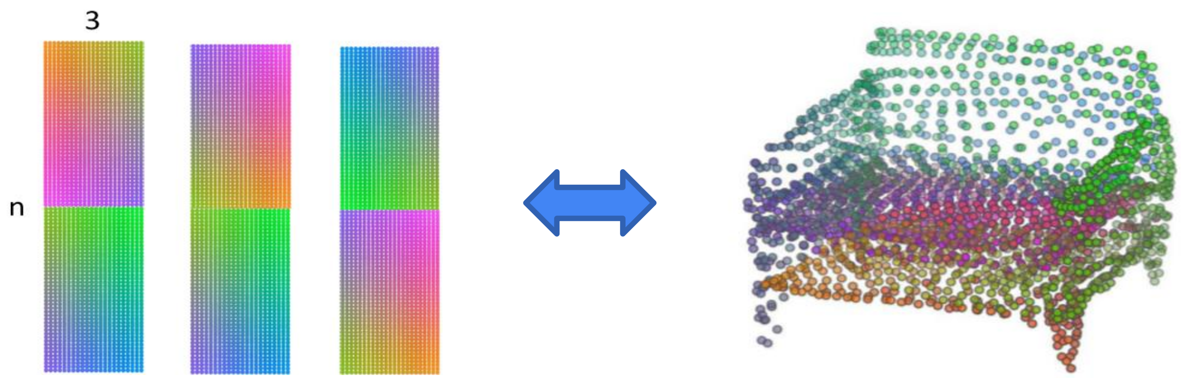
Augmented Reality (AR)



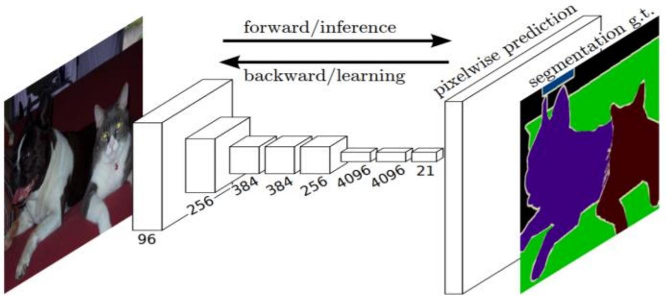
Reference: Robot Perception, taught by Prof. Shenlong Wang, UIUC
https://shenlong.web.illinois.edu/teaching/cs598fall21/assets/slides/lecture3_sensors.pdf

Challenges in Point Cloud

- Can we directly apply CNN on point cloud?
    - No, because point cloud is not grid-structured.
- The shape object can be represented in different orders
- Unknown shape transformation (e.g., translation, rotation…)

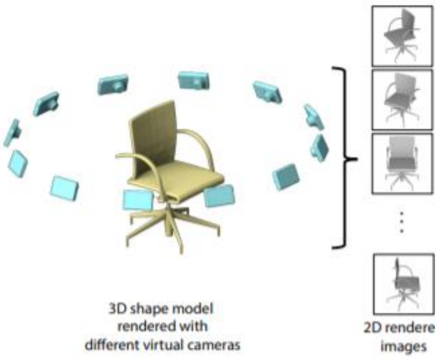


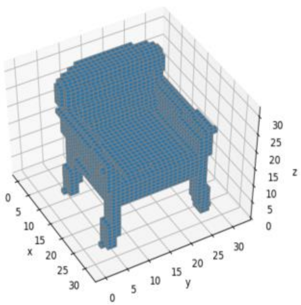
Limitation of CNN

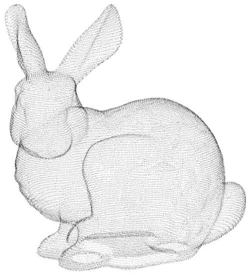


- Can we directly apply CNN on 3D data?
  - Well, it depends...

3D Representation	CNN applicable?
Multi-view images	O
Voxel	O
Point Cloud	x

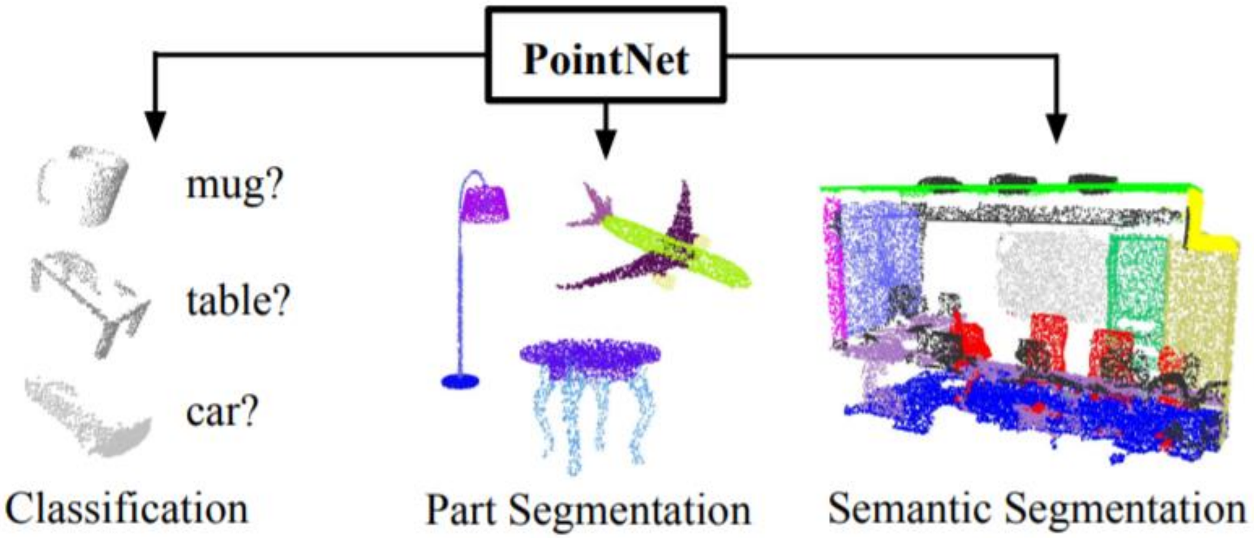






PointNet

- Goal: Point cloud classification & segmentation

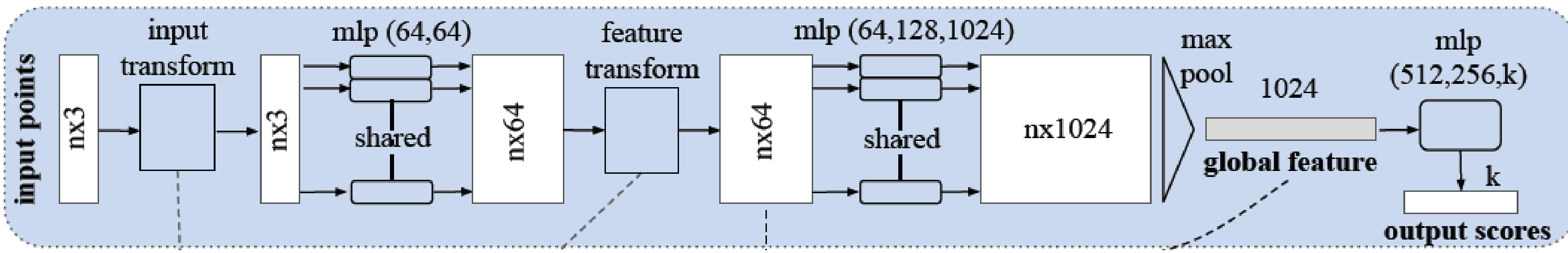


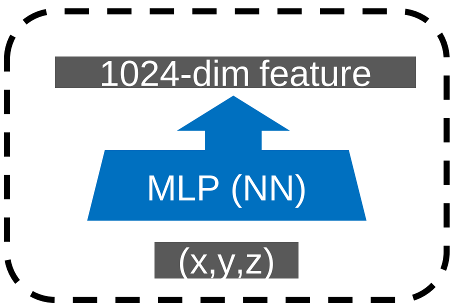
PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

PointNet

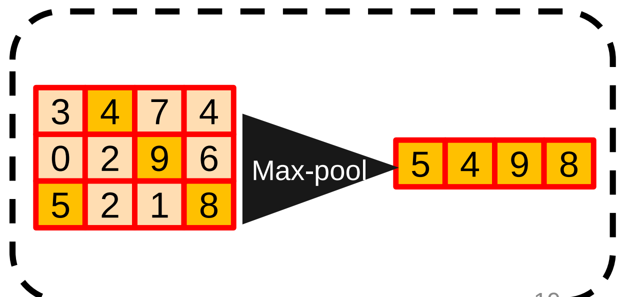
- Goal: Point cloud classification & segmentation
- Classification

Classification Network





Multi-Layer Perceptron

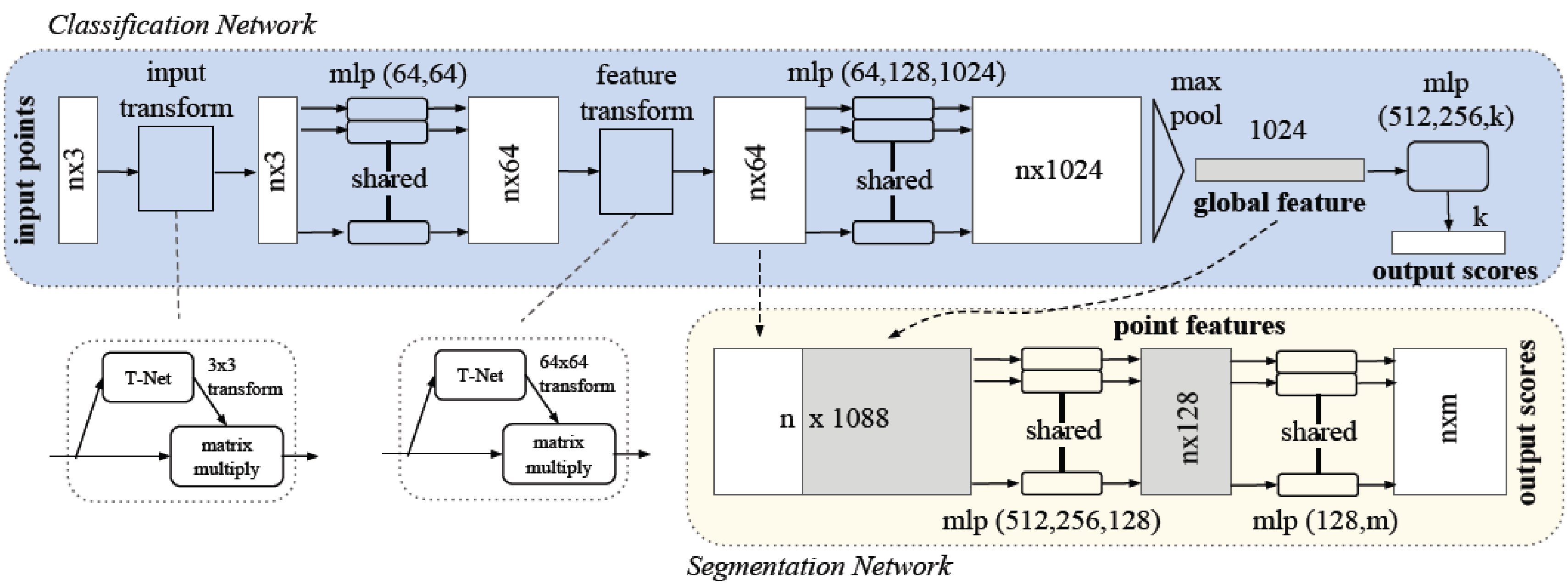


Channel-wise max-pooling

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

PointNet

- Goal: Point cloud classification & segmentation
- Classification
- Segmentation

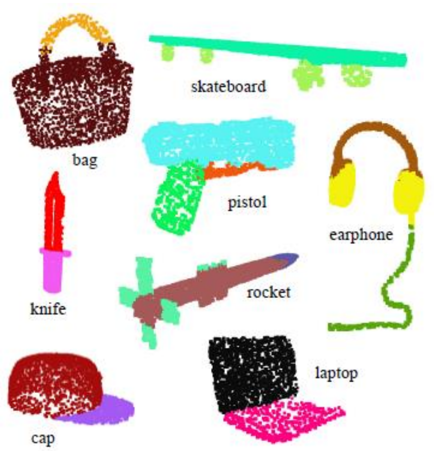


PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

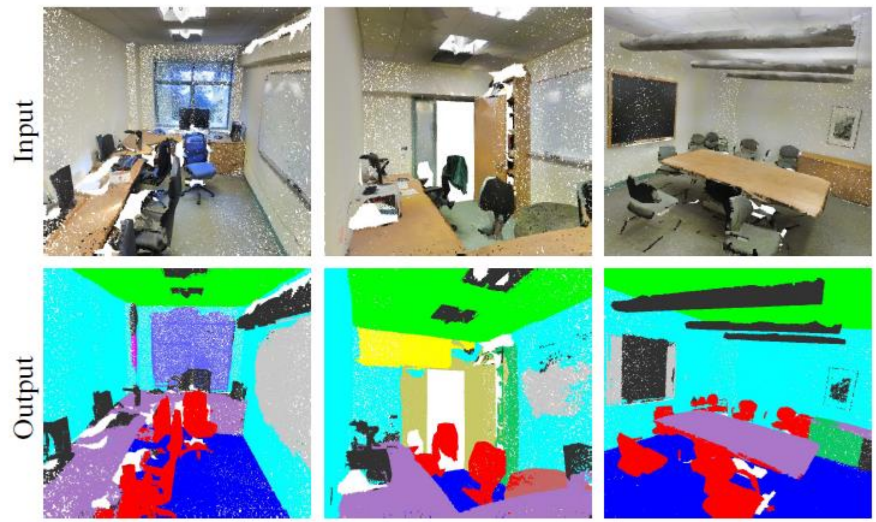
PointNet

- Goal: Point cloud classification & segmentation
- Classification & segmentation
- Qualitative results

Point: (xyz, rgb)



Part segmentation



Scene segmentation

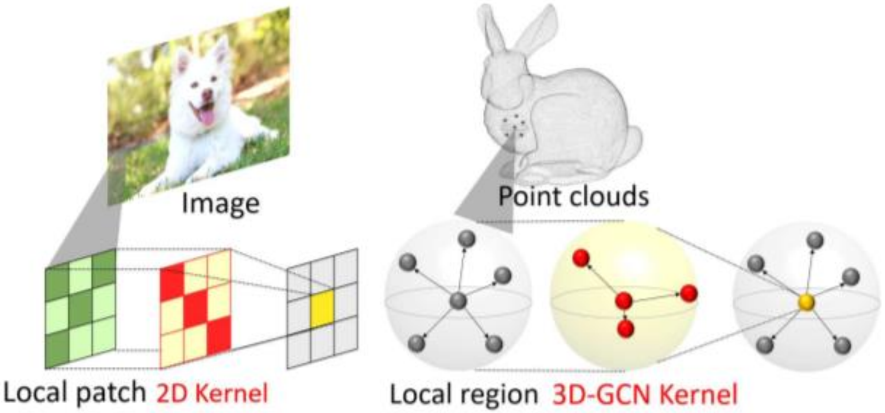
PointNet

- Goal: Point cloud classification & segmentation
- Classification & segmentation
- Qualitative results
- Remarks
  - Pros: extract features from unordered points
  - Cons:
    - Outlier/noisy point cloud data
    - Cannot capture **detailed geometry** (global pooling)
    - Might not robust to transformation like **translation, scaling, rotation**

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017

Extensions of PointNet

- **PointNet++**: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, NIPS 2017
- **Dynamic Graph CNN** for Learning on Point Clouds, TOG 2019
- **KPconv**: Flexible and deformable convolution for point clouds, ICCV 2019
- **Convolution in the cloud**: Learning deformable kernels in 3D graph convolution networks for point cloud analysis, CVPR 2020 (VLLab @ NTU)
- **3D-SelfCutMix:** Self-Supervised Learning for 3D Point Cloud Analysis, ICIP 2022 (VLLab @ NTU)

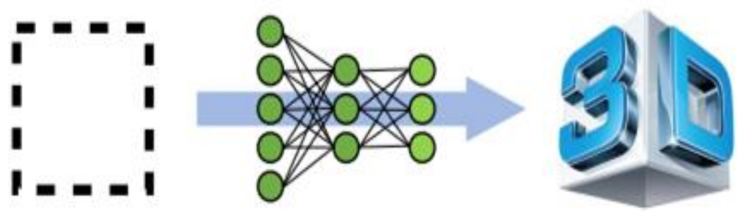


What to Cover Today?

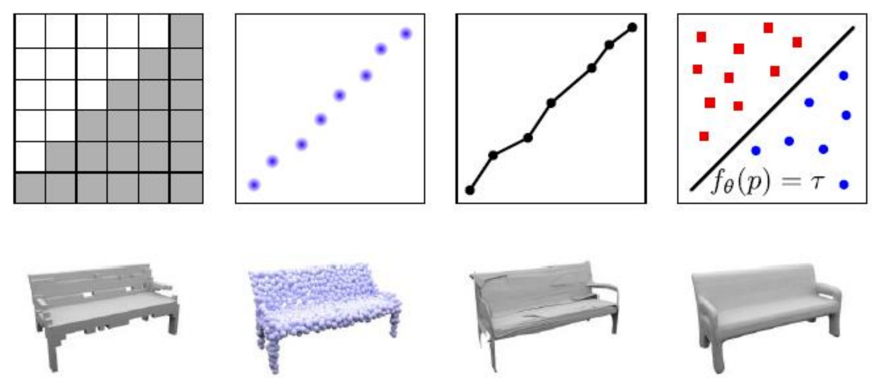
- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian Splatting

3D Reconstruction

- Reconstruct 3D shapes/scenes from partial observations
  - Single/multi-view images
  - Videos
  - Incomplete point cloud
  - text

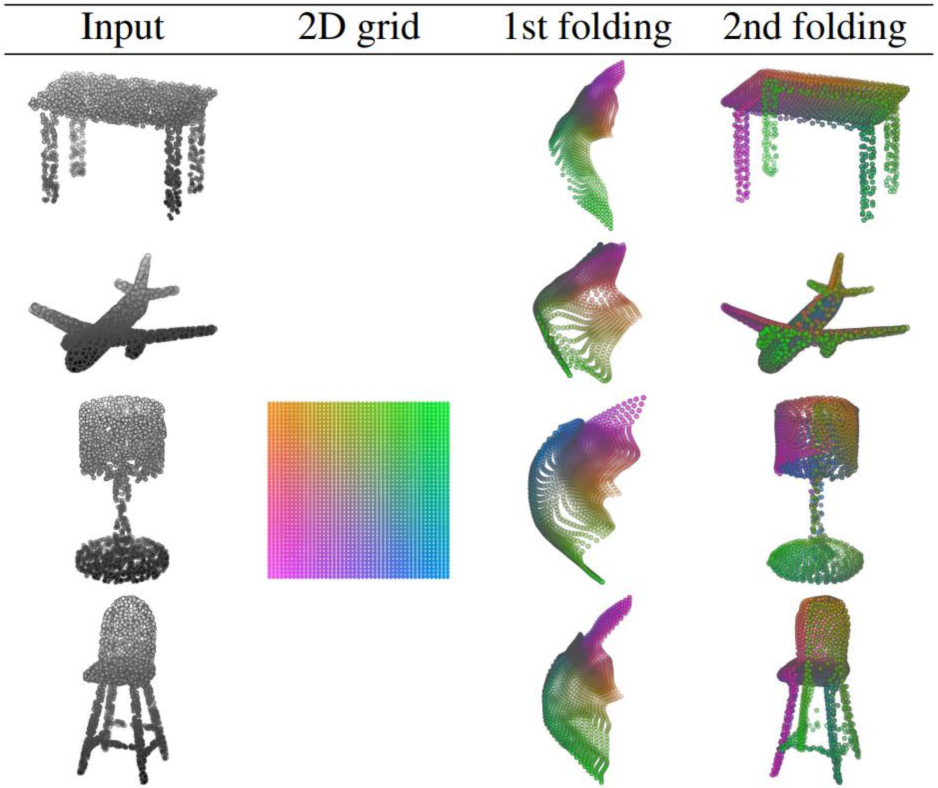


- In this part, we will talk about how to reconstruct
  - ~~Depth~~
  - ~~Voxels~~
  - Point cloud
  - ~~Mesh~~
  - Implicit Representation (Function) (??)



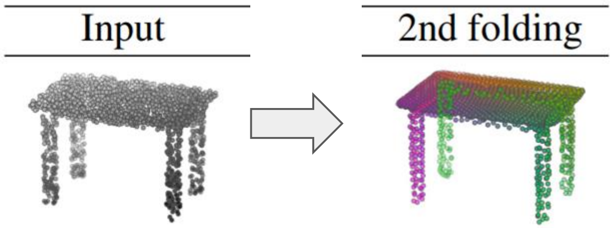
FoldingNet

- 3D reconstruction via point cloud (Auto Encoder)
- Input: point cloud object
- Output: reconstructed point cloud

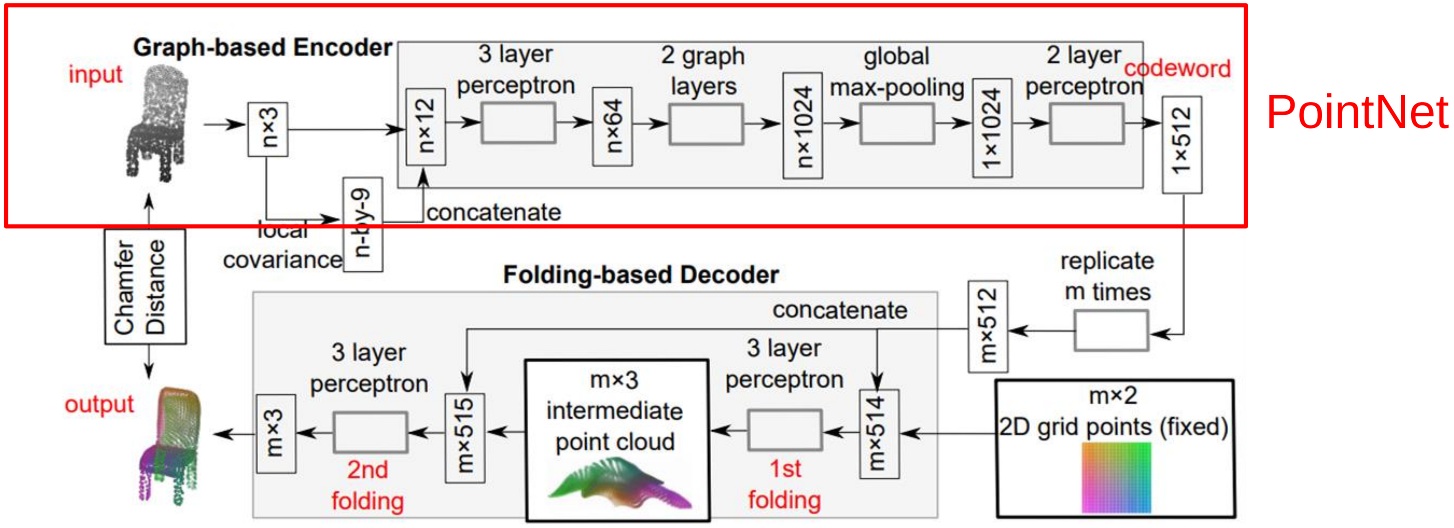


FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, CVPR 2018

FoldingNet

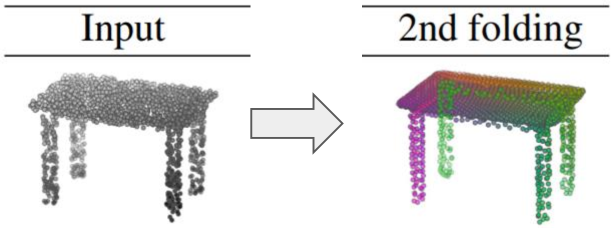


- 3D reconstruction via point cloud (Auto Encoder)
- Input: point cloud object
- Output: reconstructed point cloud
- Decoder: concat 2D coordinates with object feature and pass to MLPs



FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, CVPR 2018

FoldingNet

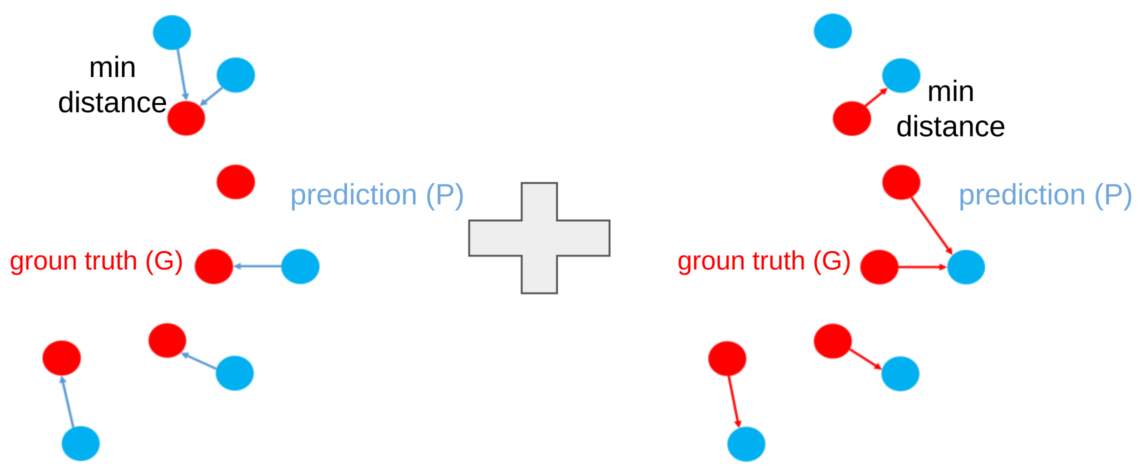


- 3D reconstruction via point cloud
- Input: point cloud object
- Output: reconstructed point cloud
- Loss function: **Chamfer distance**

What if only one-sided?

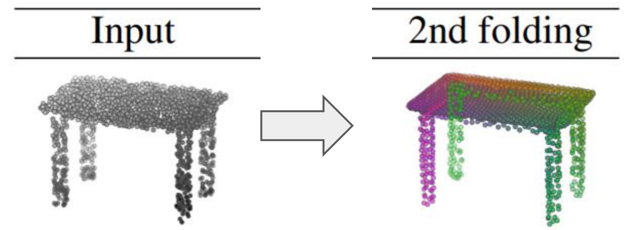
$$d_{CD}(\boxed{\mathcal{P}}, \boxed{\mathcal{G}}) = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\min_{g\in\mathcal{G}}\|p-g\| + \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\min_{p\in\mathcal{P}}\|g-p\|$$

prediction ground truth



FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, CVPR 2018

FoldingNet



- 3D reconstruction via point cloud
- Input: point cloud object
- Output: reconstructed point cloud
- Example results

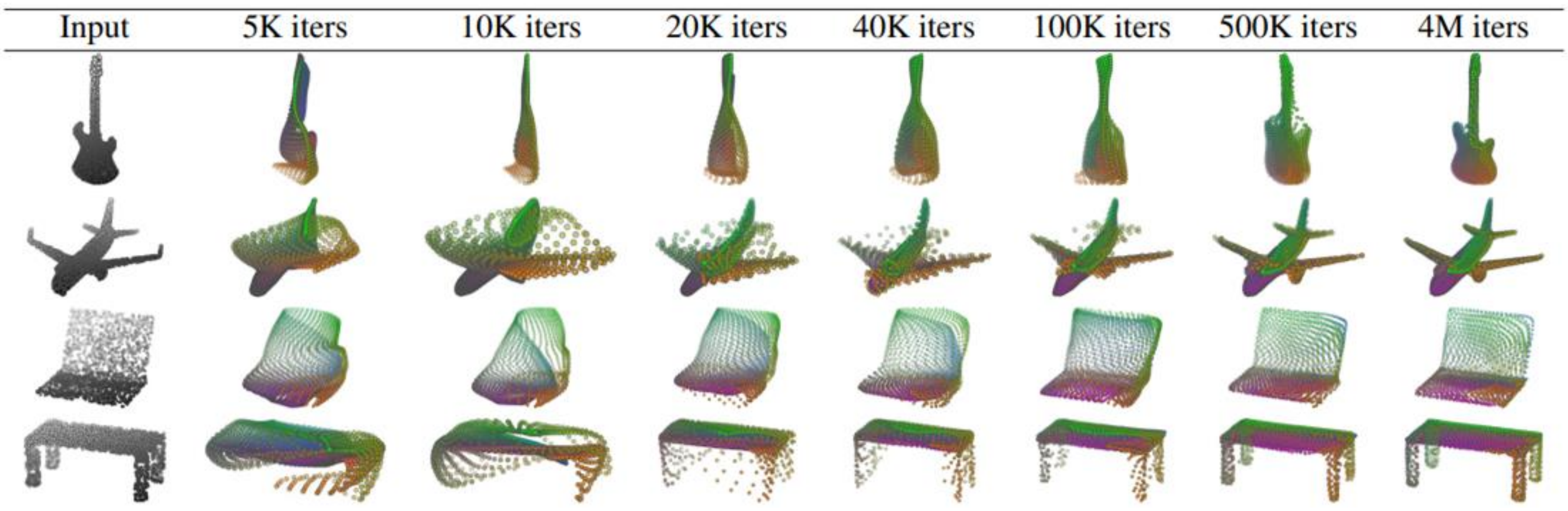


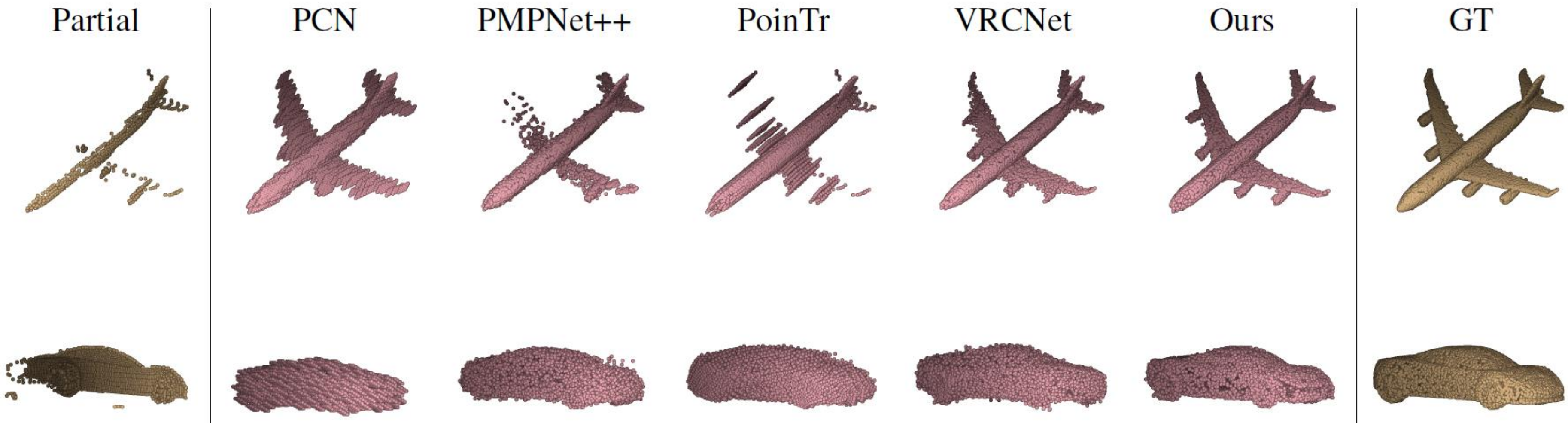
Table 2. Illustration of the training process. Random 2D manifolds gradually transform into the surfaces of point clouds.

FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, CVPR 2018

Extensions of FoldingNet

Point Cloud completion: complete partial point clouds

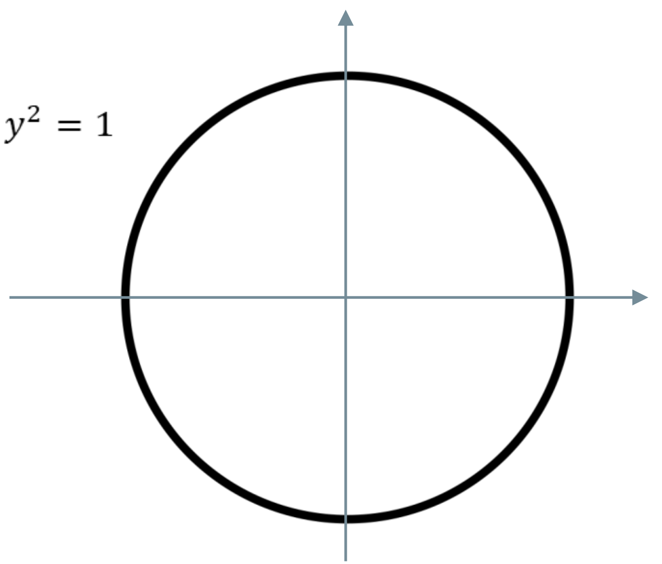
- **PCN**: Point Completion Network, 3DV 2018
- **VRCNet**: Variational Relational Point Completion Network, CVPR 2021
- **PoinTr**: Diverse Point Cloud Completion with Geometry-Aware Transformers, ICCV 2021
- **Variational Transformer** for Dense Point Cloud Semantic Completion, NeurIPS 2022 (VLLab @ NTU)



Implicit Representation

- Represent shapes as “function”
- Tell us whether a point is on the surface

$x^2 + y^2 = 1$

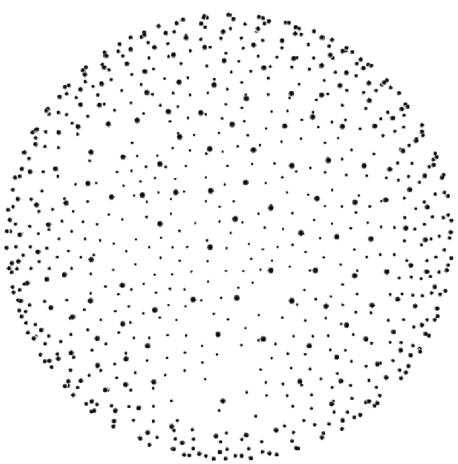


2D circle

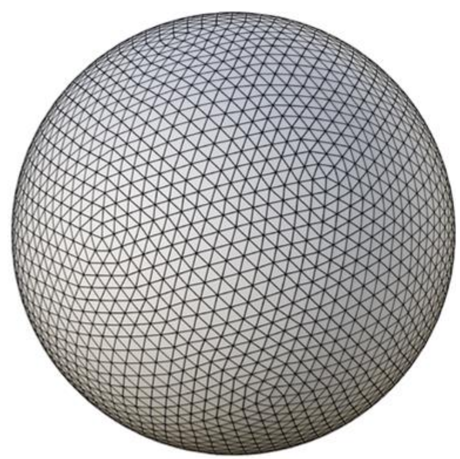
Q: Are these points on the circle?
(0, 1)
(1, 0)
(1, 1)
(0, 0)

Implicit Representation

- Represent shapes as “function”
- Unit sphere: $f(x, y, z) = x^2 + y^2 + z^2 - 1$
  - Surface is the solution set of $f(.) = 0$

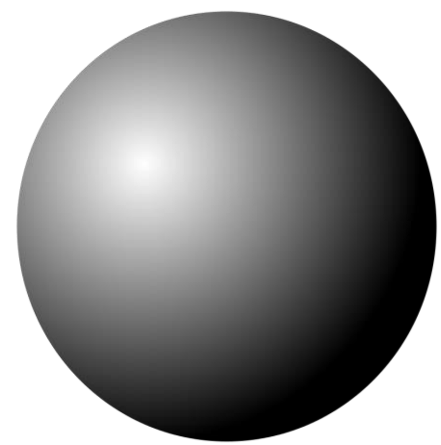


Point cloud



Mesh

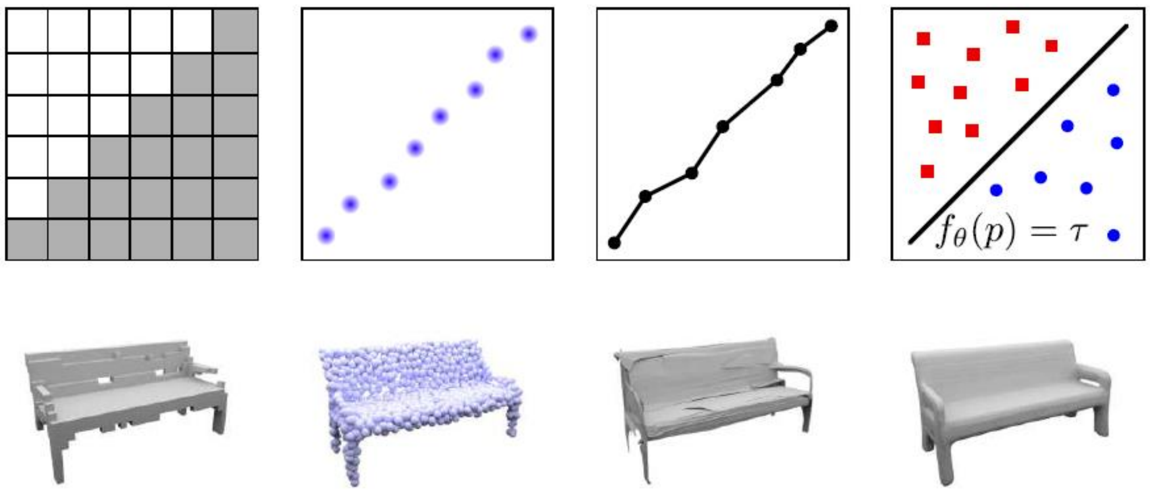
$x^2 + y^2 + z^2 = 1$



Implicit function

Occupancy Network

- Shape is a function that determines a point is inside/outside of it



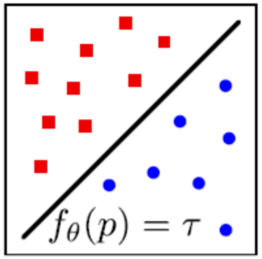
(a) Voxel (b) Point (c) Mesh (d) Ours

Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

Occupancy Network

- Make model learn to predict occupancy at every possible 3D point $p \in R^3$
- Think of occupancy function as a "classifier"
- Condition on object feature X

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0, 1]$$

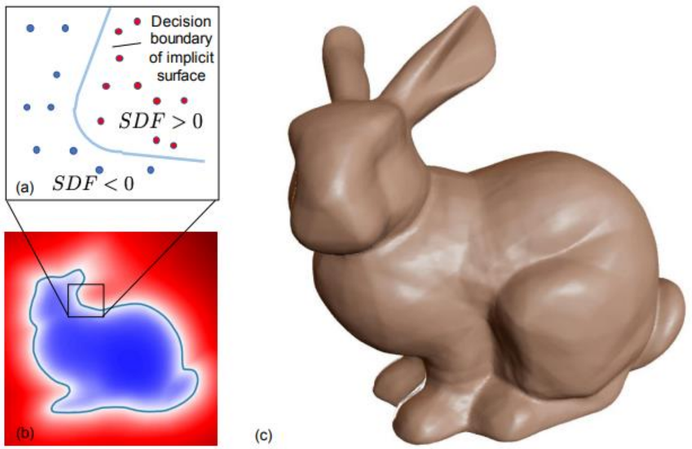


Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

Signed Distance Function

- Make model learn to predict **distance to surface** at every possible 3D point $p \in R^3$
- Think of signed distance function as a "regressor"
- Condition on object feature X

$$f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [R]$$

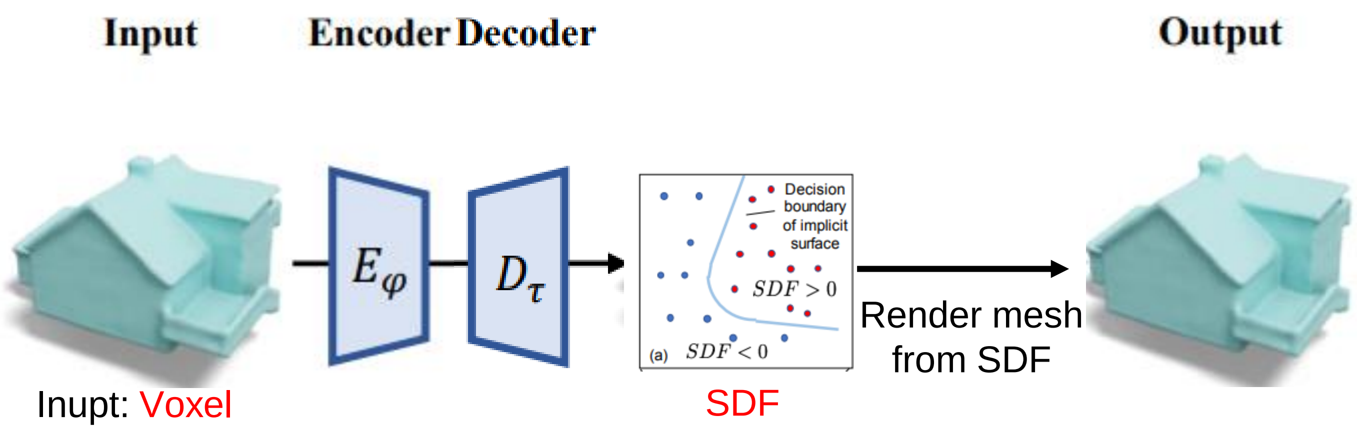


DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 2019

Extension on Signed Distance Function

SDFusion (CVPR 2023)

- Train an Auto Encoder for SDF (input voxel)

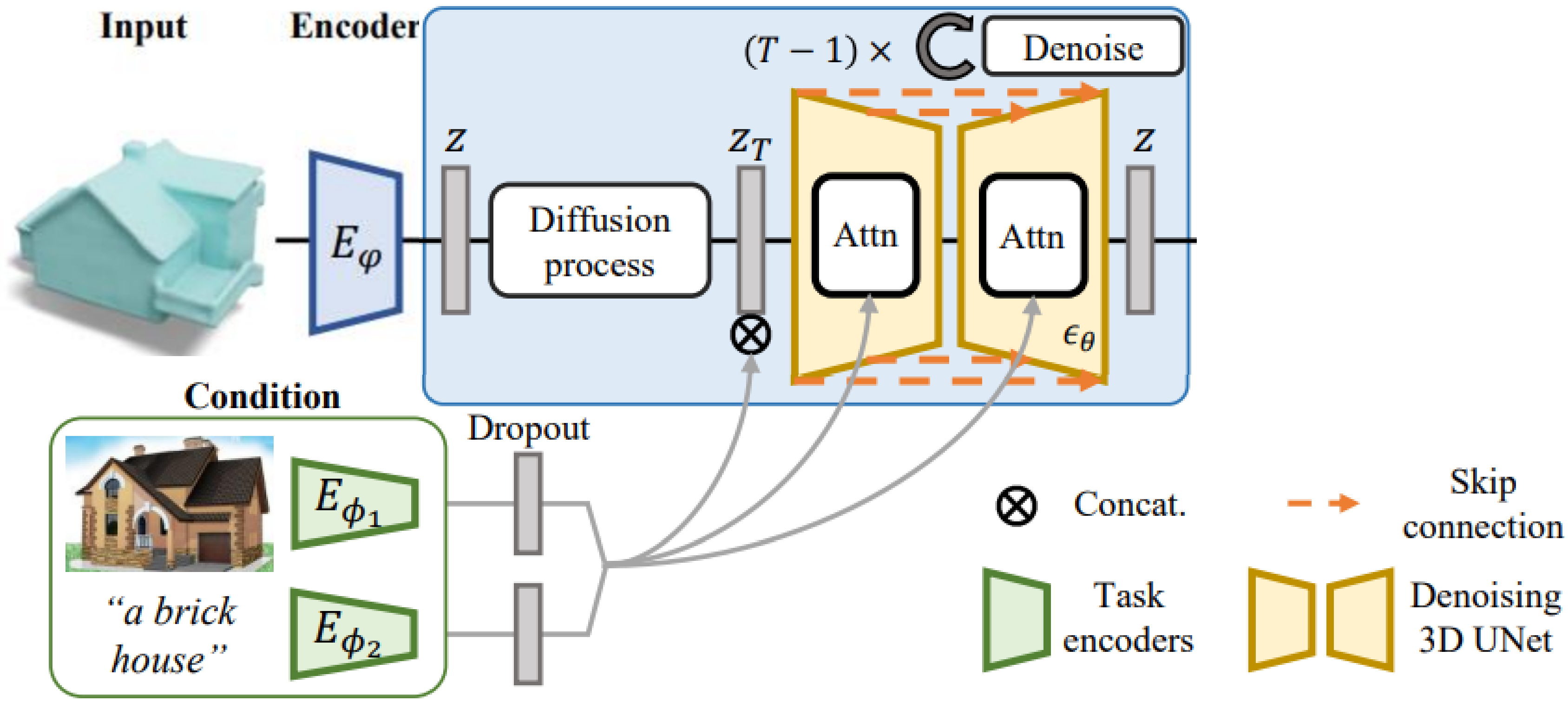


SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation, CVPR 2023

Extension on Signed Distance Function

SDFusion (CVPR 2023)

- Train an Auto Encoder for SDF (input voxel)
- Train a conditional LDM for latent vector z (text or image)

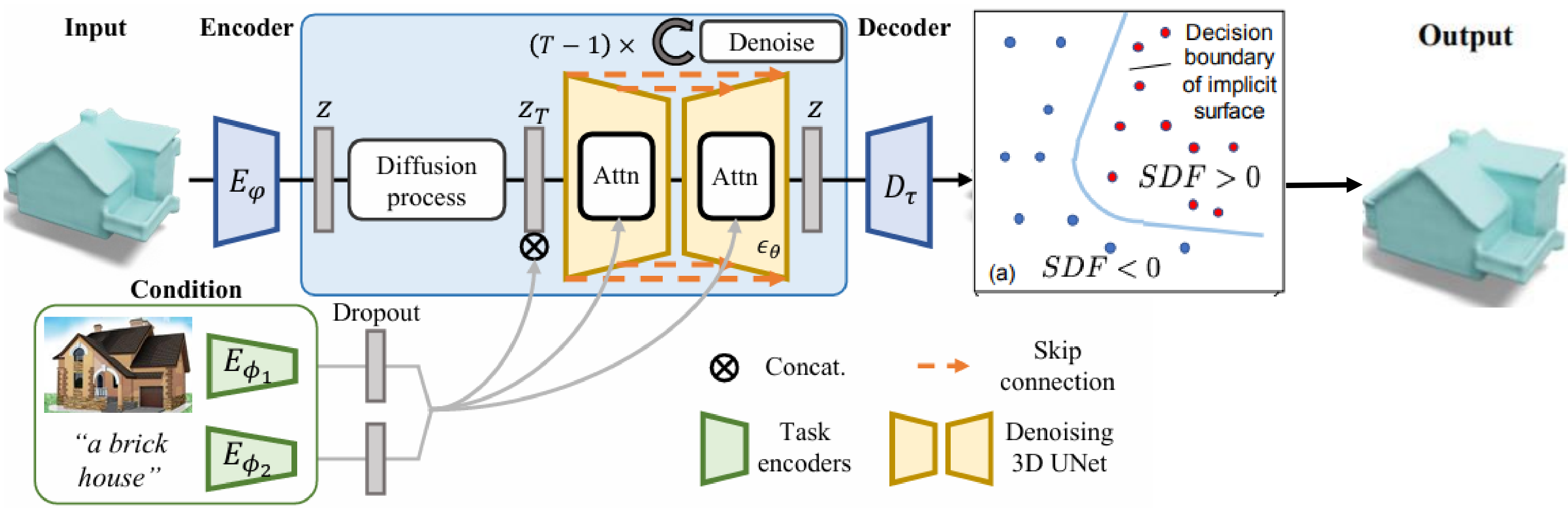


SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation, CVPR 2023

Extension on Signed Distance Function

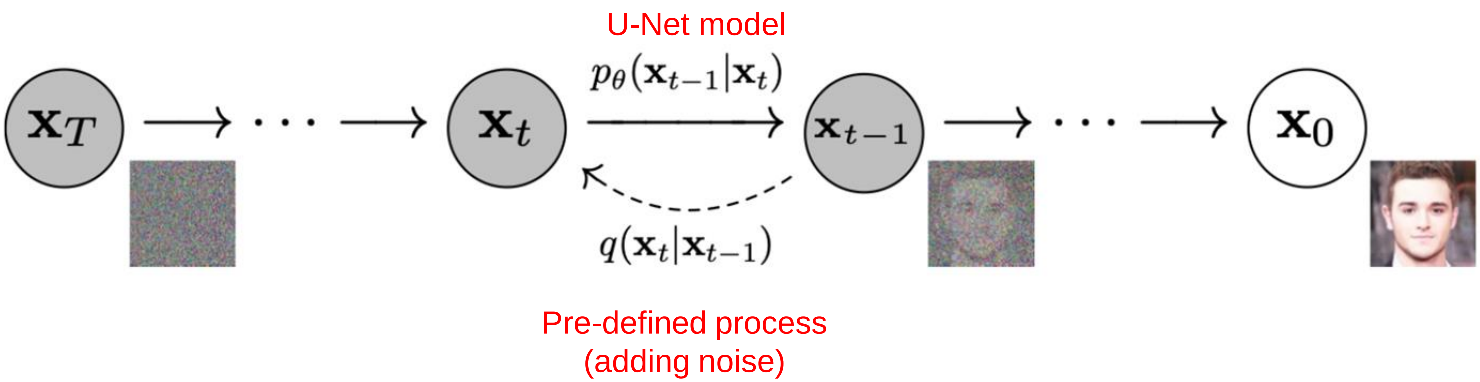
SDFusion (CVPR 2023)

- Train an Auto Encoder for SDF (input voxel)
- Train a conditional LDM for latent vector z (text or image)



SDFusion: Multimodal 3D Shape Completion, Reconstruction, and Generation, CVPR 2023

Recap: Diffusion model (intuitively)



- Can be viewed as denoising from a Gaussian noise image
- Each step makes little progress of denoising (total about 1000 steps)
- Output image of each step can be seen as the **original image** combining with a **noise** using specific ratio
- The process can also be seen as predicting the **added noise**

Implicit Representation (Occupancy, SDF)

Strength

- Flexible shape topology
- Arbitrary resolution
- Few model parameters

Weakness

- Require post-processing to get mesh
- Cannot handle complex scene

Extensions of Occupancy, SDF

Text-to-3D Generation

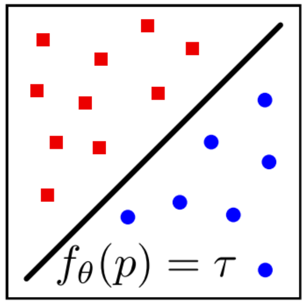
- **Diffusion-SDF:** Text-To-Shape via Voxelized Diffusion (CVPR 2023)
- **Diffusion-SDF**: Conditional Generative Modeling of Signed Distance Functions (CVPR 2023)
- Learning Shape-Color Diffusion Priors for Text-Guided 3D Object Generation (accepted to TMM 2024 Sept.)(VLLab @ NTU)
- **GraphDreamer**: Compositional 3d scene synthesis from scene graphs (CVPR 2024)

What to Cover Today?

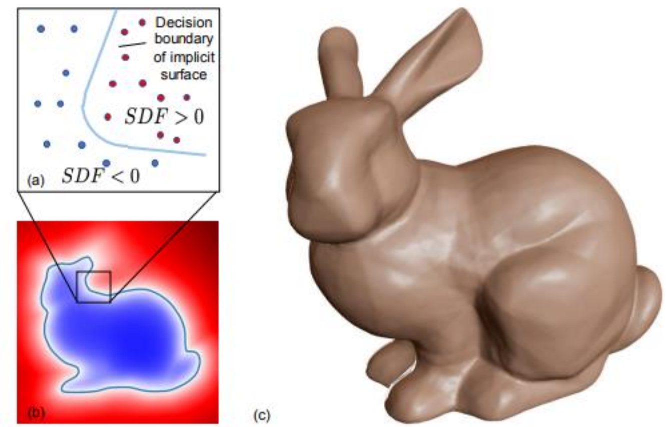
- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian Splatting

Recap:
Neural Networks as a Continuous Shape Representation

Occupancy Networks
(Mescheder et al. 2019)
(x,y,z) -> occupancy



Deep SDF
(Park et al. 2019)
(x,y,z) -> distance



Pros: Compact and expressive parameterization

Cons: Limited rendering, difficult to optimize

Slide credit: Jon Barron, cs598dwh

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Many slides from Jon Barron and cs598dwh (UIUC)

Ben Mildenhall*

UC Berkeley

Pratul Srinivasan*

UC Berkeley

Matt Tancik*

UC Berkeley

Jon Barron

Google Research

Ravi Ramamoorthi

UC San Diego

Ren Ng

UC Berkeley

Slide credit: cs598dwh

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Problem: Novel view synthesis (NVS)

Inputs: sparsely sampled images of a scene

Outputs: *new* views of the same scene

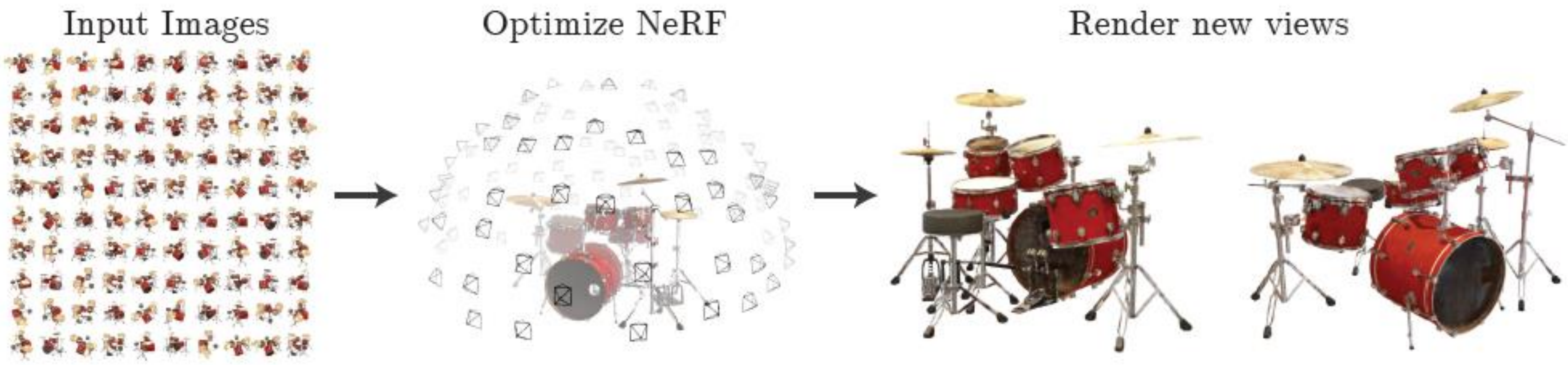
tancik.com/nerf

Slide credit: Jon Barron

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

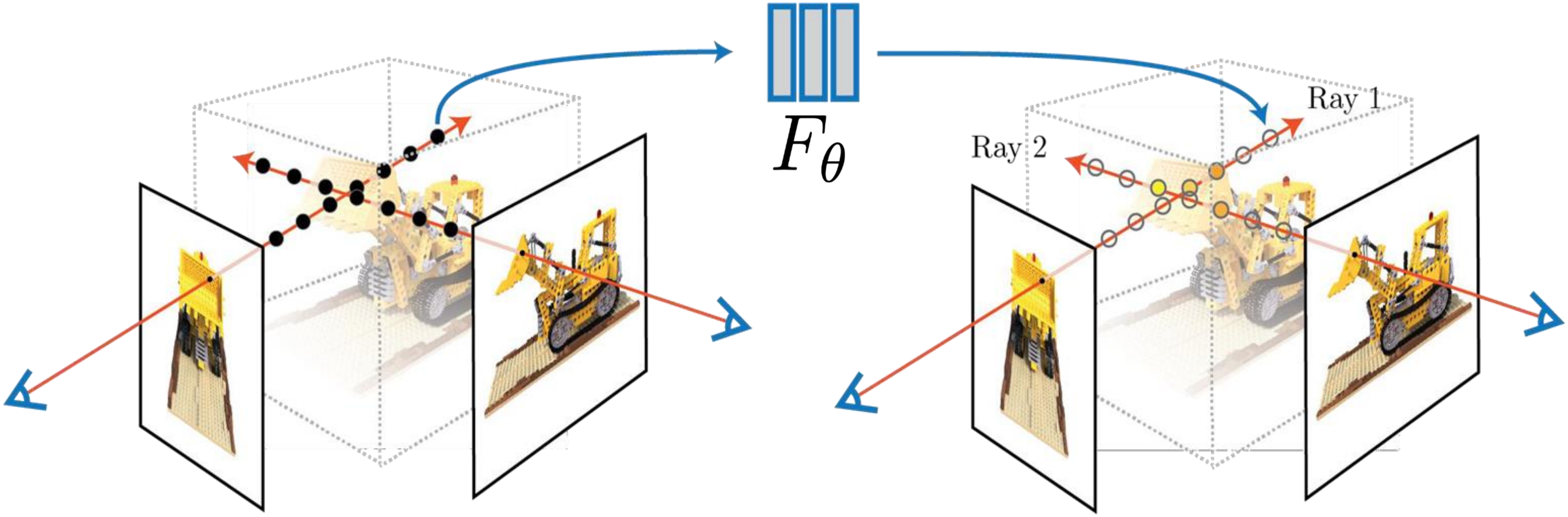
NeRF (Neural randiance field)

- Goal: learn 3D representation, and perform **novel view synthesis**
- Input: multi-view images + camera poses
- Output: 3D representation (neural radiance field)



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

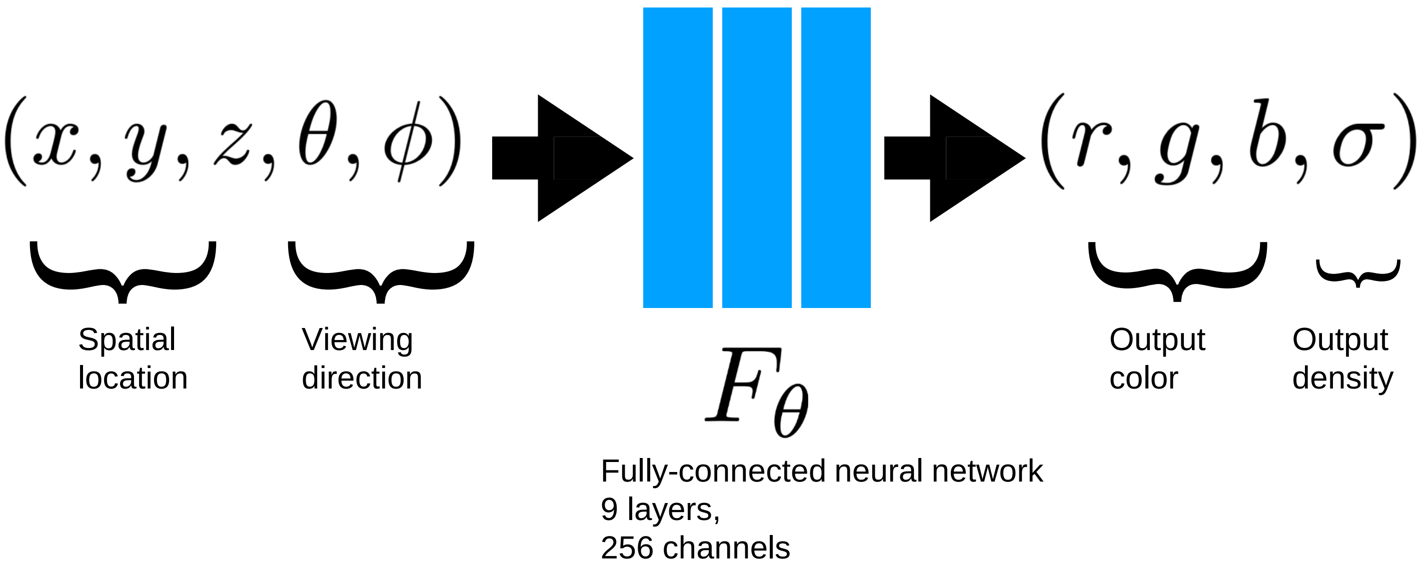
Generate views with traditional volume rendering



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

NeRF (Neural randiance field)



Slide credit: Jon Barron

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

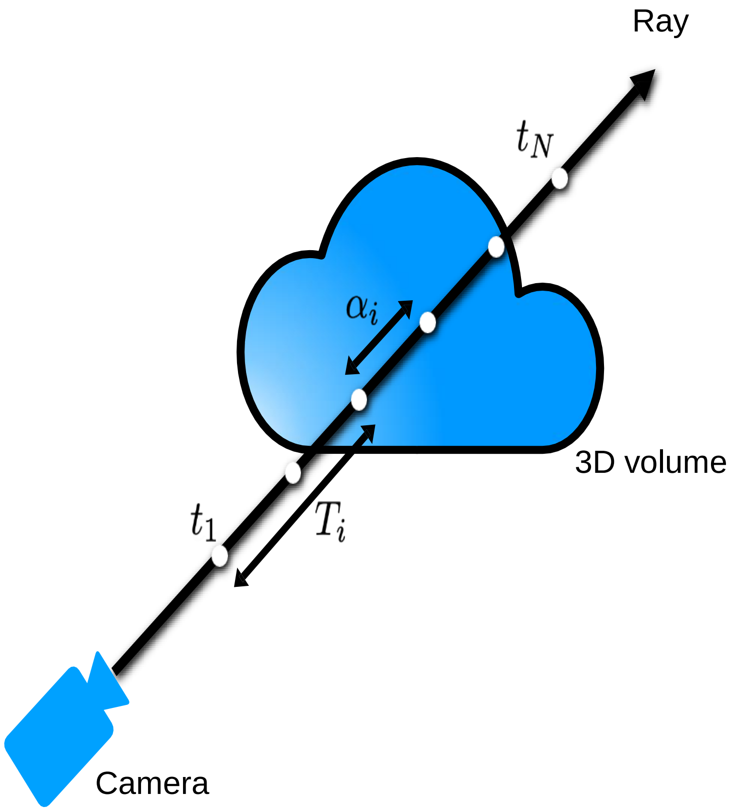
Generate views with traditional volume rendering

Rendering model for ray r(t) = o + td:

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

weights

colors



- How much light is blocked earlier along ray:

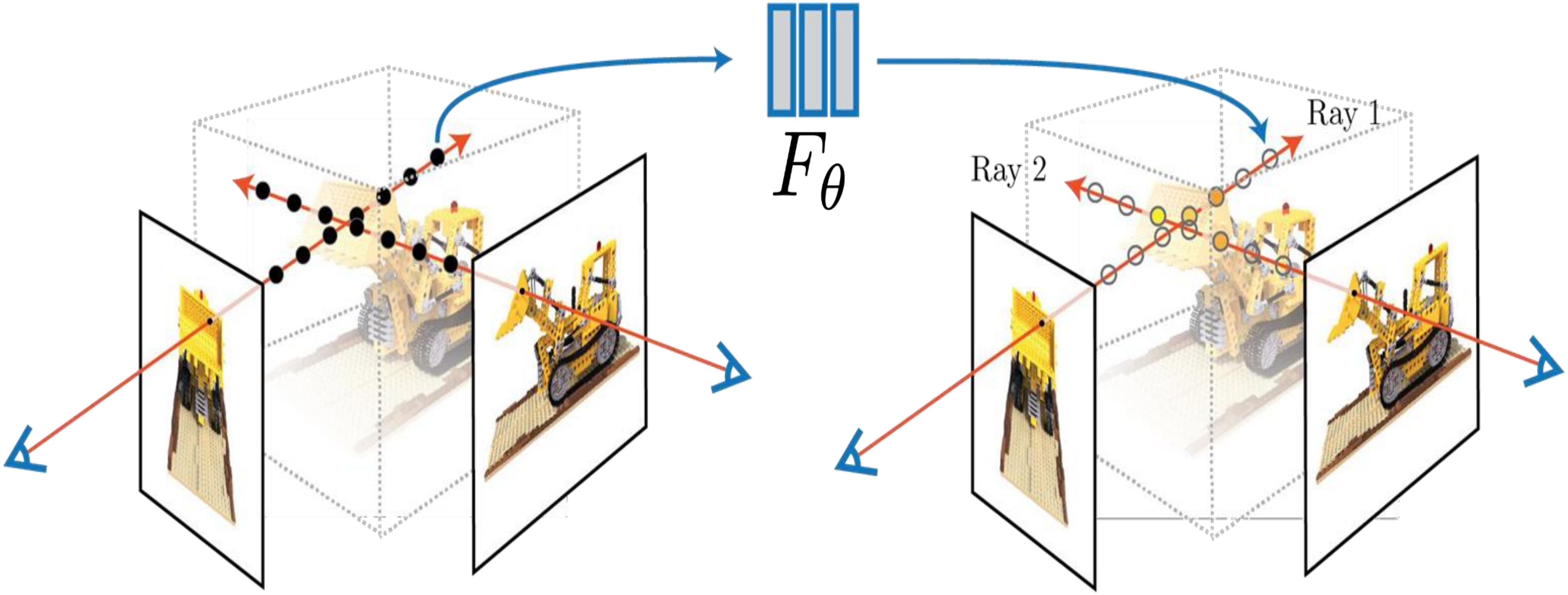
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

- How much light is contributed by ray segment *i*:

$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$

-Density * Distance Between Points

Optimize with gradient descent on rendering loss

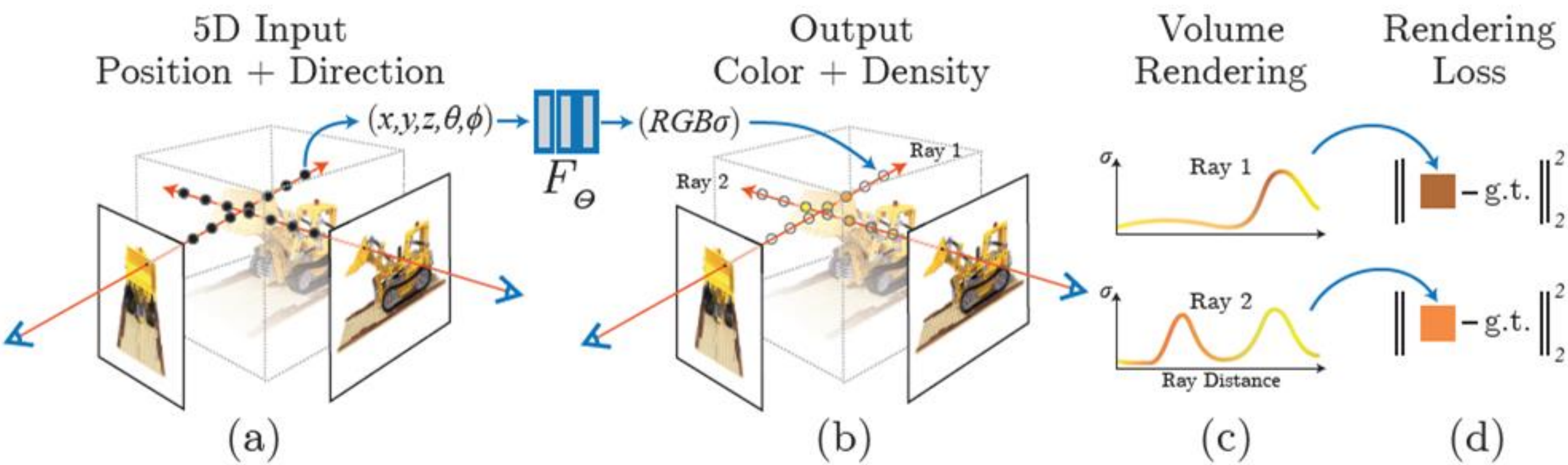


$$\min_{\theta} \sum_{i} || \, \mathrm{render}_i(F_\theta) - I_i \, ||^2$$

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

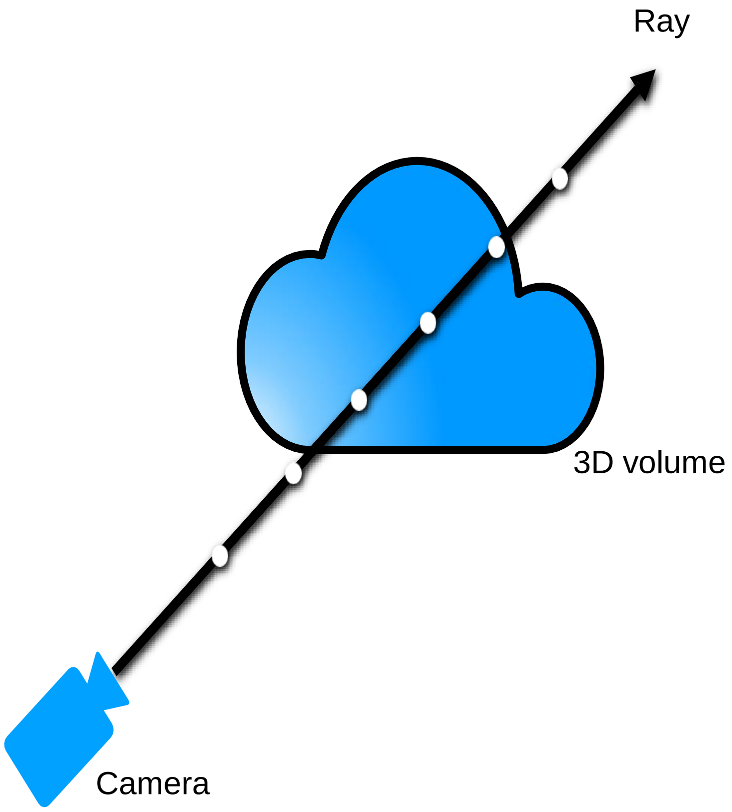
Training network to reproduce all input views of the scene



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Can we allocate samples more efficiently?
--Two pass rendering



NeRF: Representing Scenes as Neural Radiance Fields for
View Synthesis, ECCV 2020

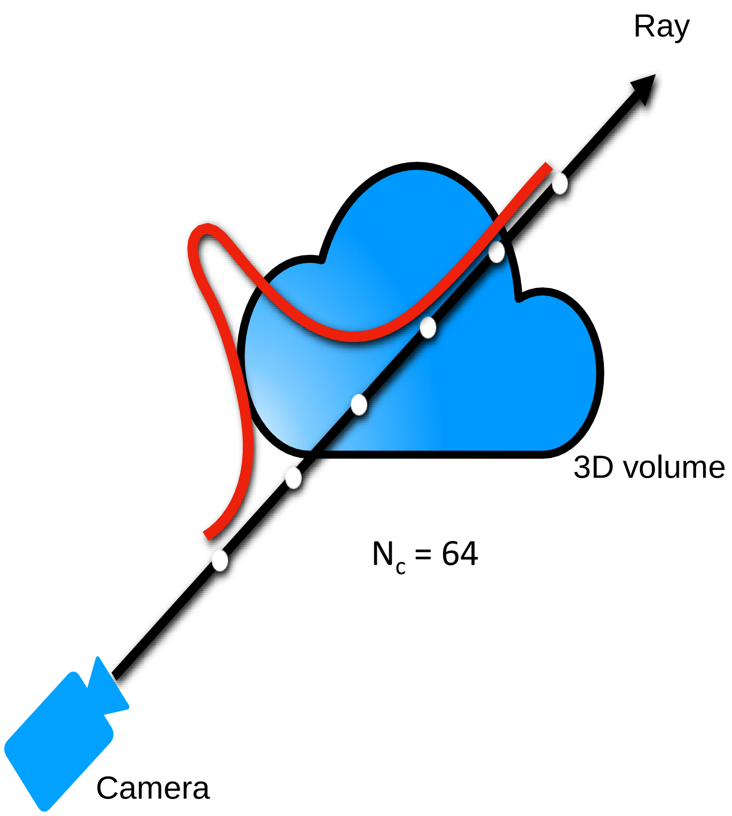
Slide credit: Jon Barron

Two pass rendering:coarse network

- Sparsely sample points along ray
- Serve as a coarse guidance

$$C \approx \sum_{i=1}^{N} \boxed{T_i \alpha_i} c_i$$

treat weights as probability distribution for new samples



Slide credit: Jon Barron

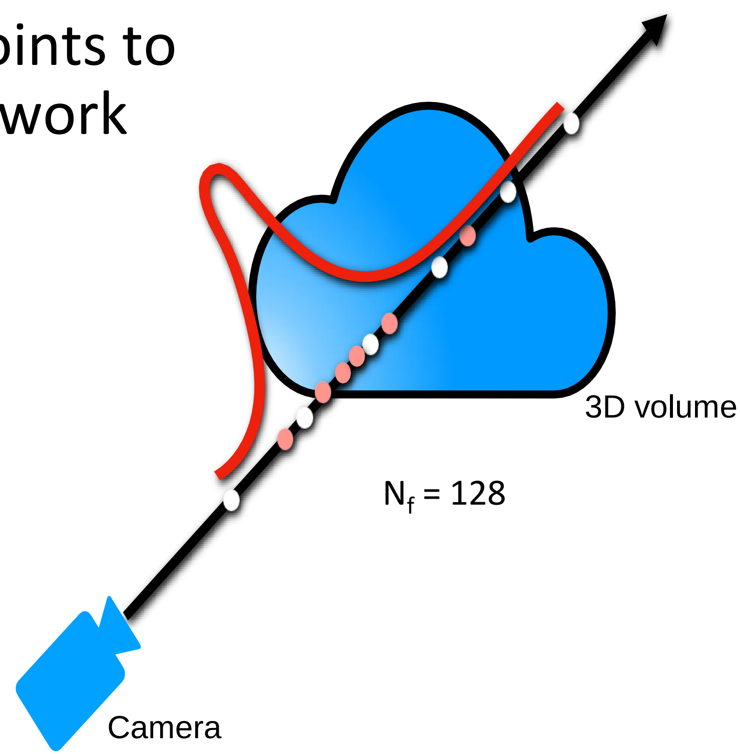
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Two pass rendering:fine network

- Use the coarse predicted density to resample new points along ray
- Together compute all N_c + N_f points to calculate final color for fine network

$$C \approx \sum_{i=1}^{N} \boxed{T_i \alpha_i} c_i$$

treat weights as probability distribution for new samples



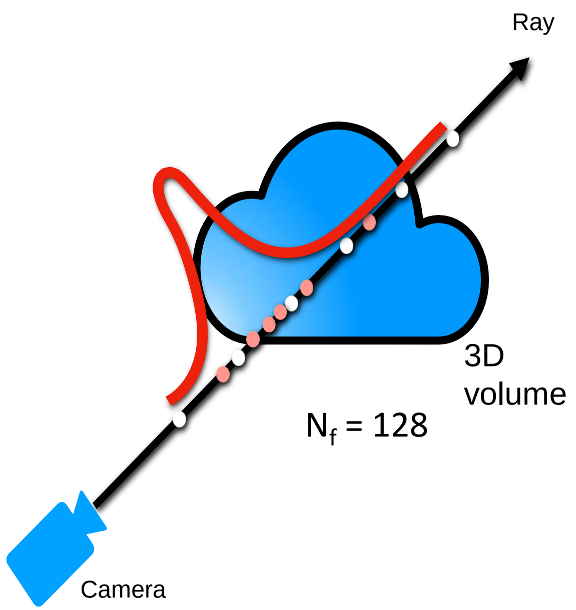
$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$ (coarse + fine)

Slide credit: Jon Barron

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Two pass rendering: optimization

- Optimize coarse network and fine network together
- Only use the prediction of fine network when rendering a new scene



$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left[\left\| \boxed{\hat{C}_c(\mathbf{r})} - C(\mathbf{r}) \right\|_2^2 + \left\| \boxed{\hat{C}_f(\mathbf{r})} - C(\mathbf{r}) \right\|_2^2 \right]$$ (coarse + fine)

predicted color from coarse network

predicted color from fine network

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Positional encoding

NeRF (Naive)

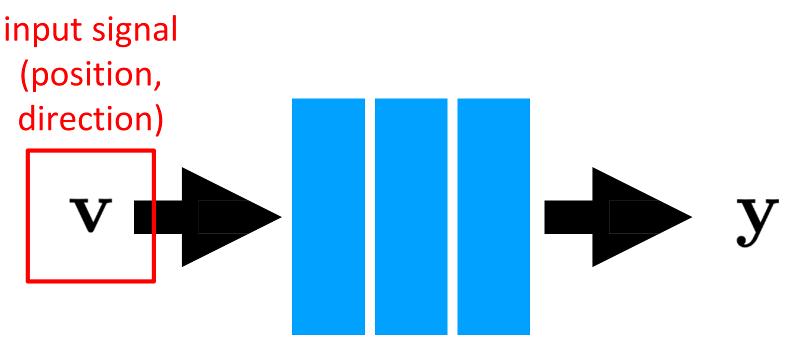
NeRF (with positional encoding)

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Positional encoding

Naive



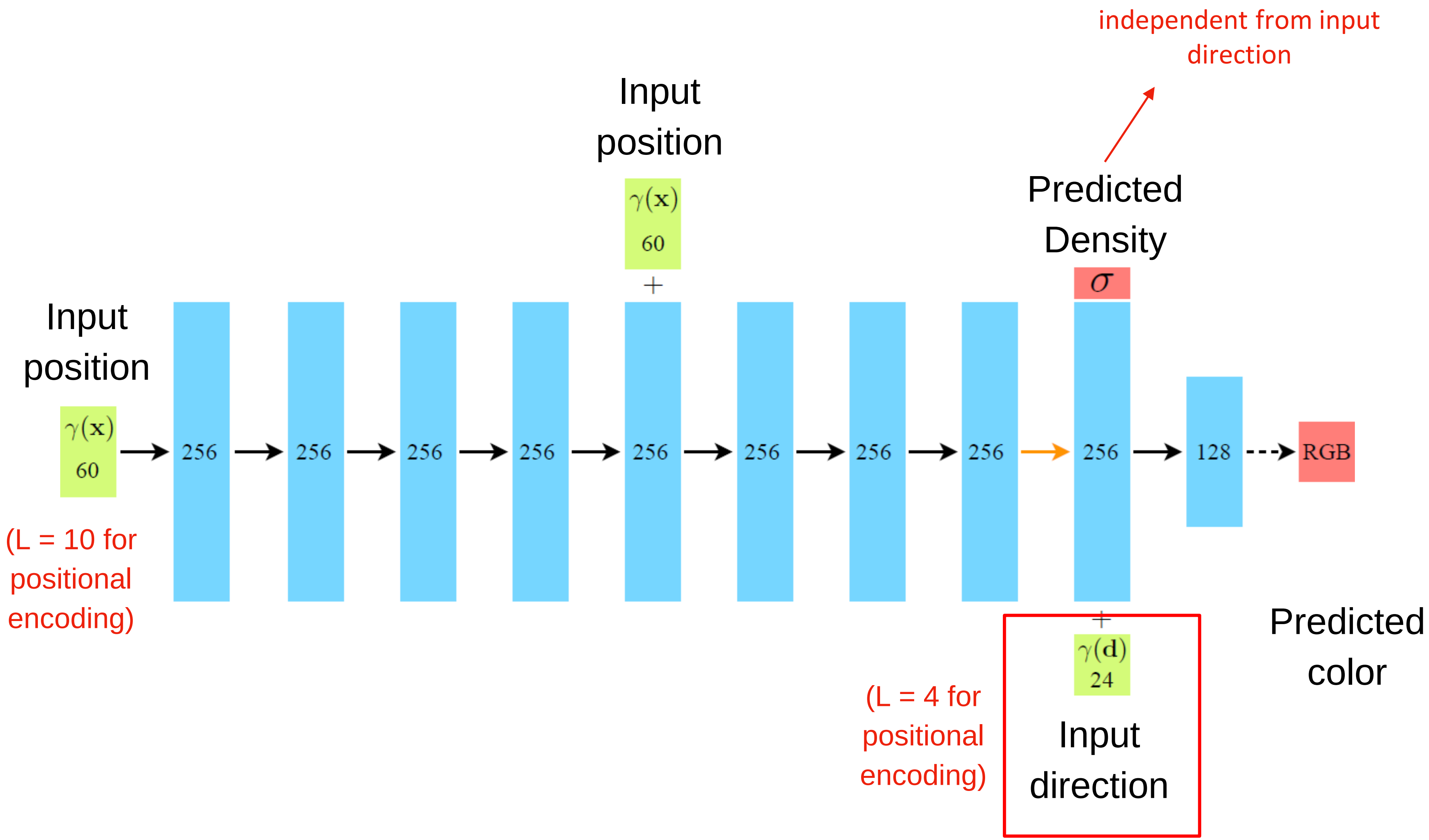
$$\begin{pmatrix} \sin(\mathbf{v}), \cos(\mathbf{v}) \\ \sin(2\mathbf{v}), \cos(2\mathbf{v}) \\ \sin(4\mathbf{v}), \cos(4\mathbf{v}) \\ \dots \\ \sin(2^{L-1}\mathbf{v}), \cos(2^{L-1}\mathbf{v}) \end{pmatrix}$$

Positional encoding

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Slide credit: Jon Barron

Network Structure

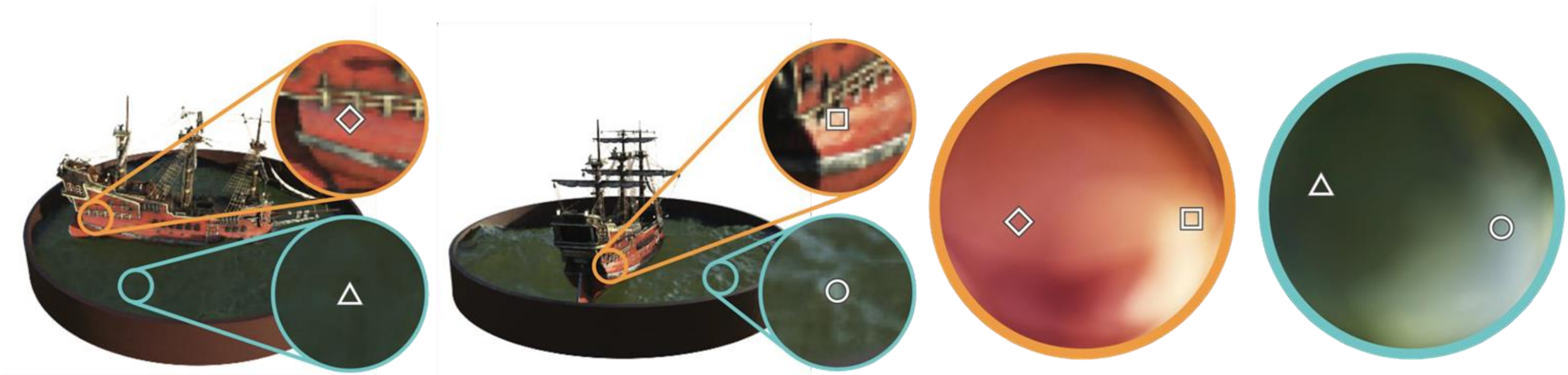


Slide credit: Jon Barron

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Viewing directions as input

- The specular reflection (or other changes influenced by lighting) varies across different views



(a) View 1 (b) View 2 (c) Radiance Distributions

Slide credit: Jon Barron

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Viewing directions as input

- The rendered color changes as the viewing direction
- L: image plane change with viewing direction
- R: fixing image plane while the viewing direction feeded to NeRF changes


NeRF: Representing Scenes as Neural Radiance Fields for
View Synthesis, ECCV 2020

Viewing directions as input

- Another example

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Depth (geometry) Estimation

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

$$d_i$$

Distance from the points to camera

- The predicted density indicates the object surface
- The estimated depth perfectly shows the geometry of foreground object

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Depth (geometry) Estimation

- Another example

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Depth (geometry) Estimation

- By correctly estimate the depth of the scene, virtual objects are possible to interact with the real scene

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

NeRF: strength & weakness

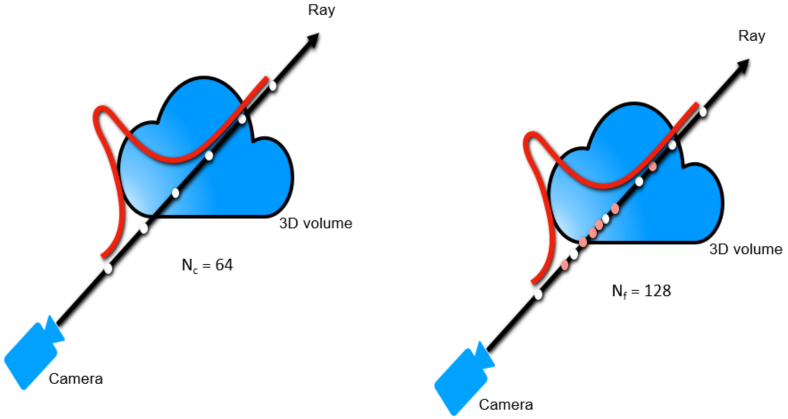
Strength

- Photo-realistic texture
- Do not require 3D ground truth
- View-dependent effect

Weakness

- Only fit single scene
- Require much posed images
- Time-consuming rendering (30s per frame)

<- Fatal for real-time applications !!



Extensions of NeRF

NeRF Acceleration, Generalization

- Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains (NeurIPS 2020) -> explain why positional encoding works
- **pixelnerf:** Neural radiance fields from one or few images (CVPR 2021)
- **KiloNeRF**: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs (ICCV 2021)
- Instant Neural Graphics Primitives with a Multiresolution Hash Encoding (SIGGRAPH 2022)
- **NeurMiPs**: Neural Mixture of Planar Experts for View Synthesis (CVPR 2022) (VLLab @ NTU)
- Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction (CVPR 2022)
- **GSNeRF**: Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding (CVPR 2024) (VLLab @ NTU)

What to Cover Today?

- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian Splatting

3D Gaussian Splatting for Real-Time Radiance Field Rendering

SIGGRAPH 2023

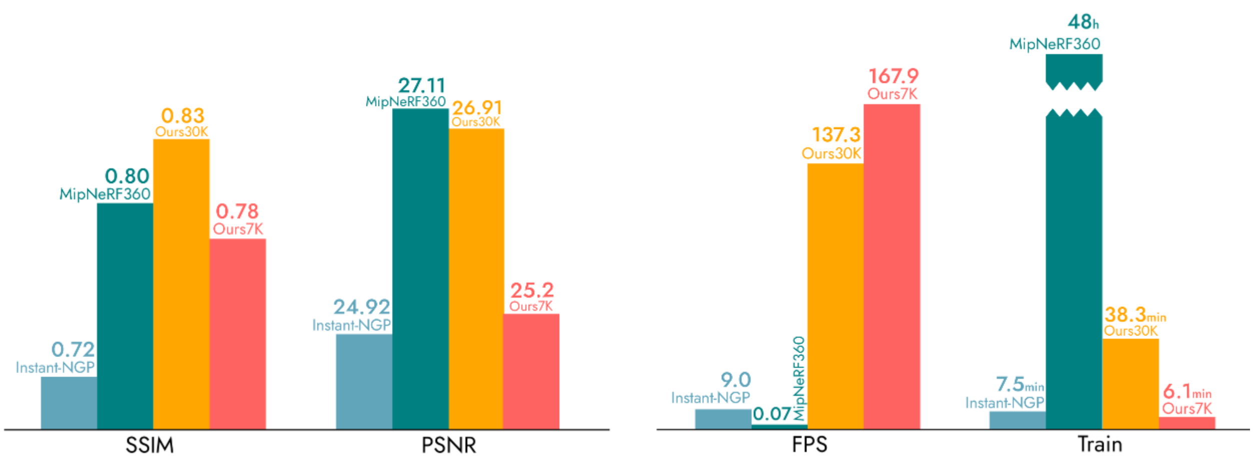
(ACM Transactions on Graphics)

Bernhard Kerbl* [1,2] Georgios Kopanas* [1,2] Thomas Leimkühler[3] George Drettakis[1,2]

* Denotes equal contribution

[1]Inria [2]Université Côte d'Azur [3]MPI Informatik

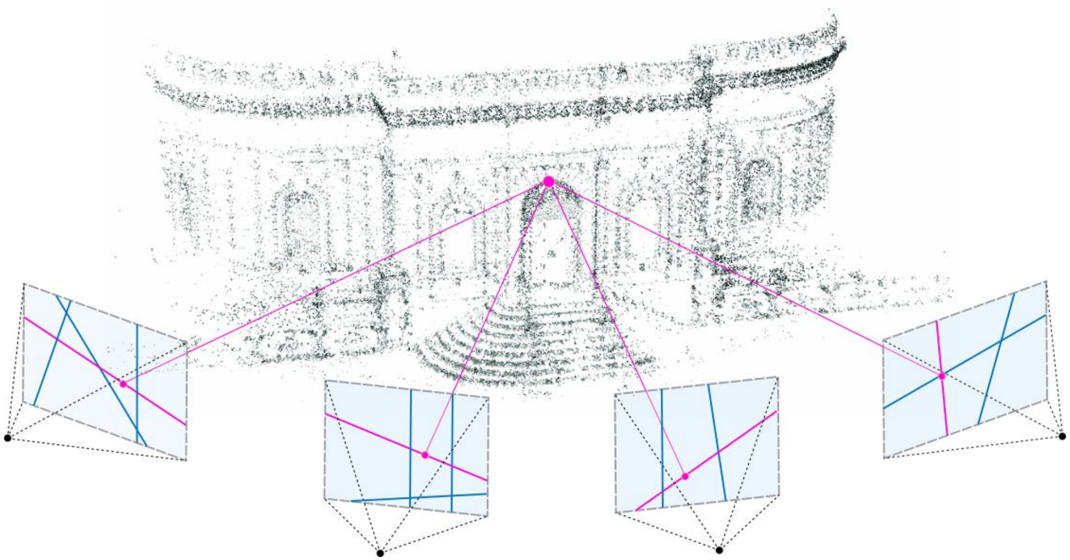
1 Inria 2 UNIVERSITÉ CÔTE D'AZUR 3 max planck institut informatik



3D Gaussian Splatting for Real-Time Radiance Field Rendering [Link]

How to make renderings faster?

- Borrow the idea from point cloud
  - Can be super fast using **rasterization** for rendering
  - Only preserves regions containing objects



https://www.linkedin.com/pulse/structure-from-motion-manish-joshi/

Method – When old meets new

NeRF

Gaussian Splatting

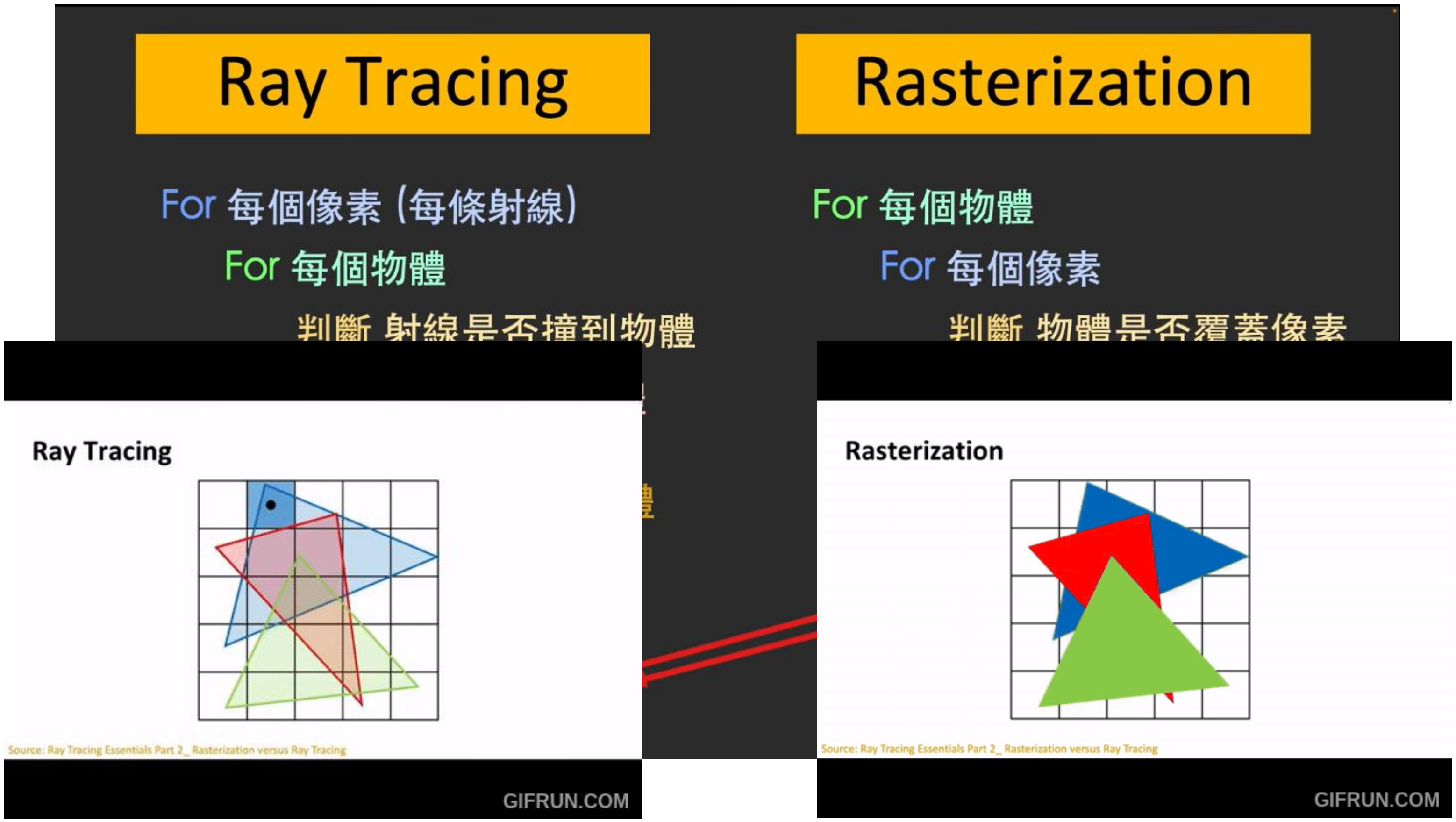
Slide credit: AI 甘安捏

3D Gaussian Splatting for Real-Time Radiance Field Rendering [Link]

Method – When old meets new

NeRF

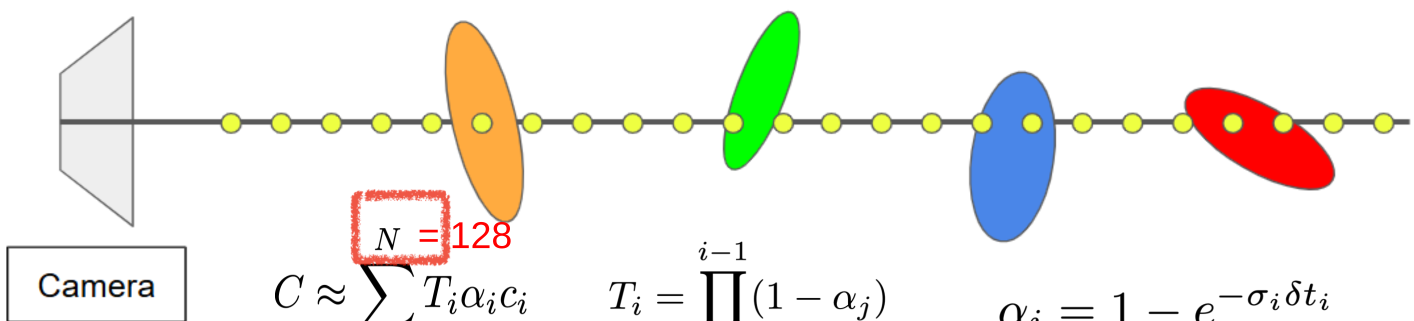
Gaussian Splatting



Slide credit: AI 甘安捏

3D Gaussian Splatting for Real-Time Radiance Field Rendering Link

Method – When old meets new

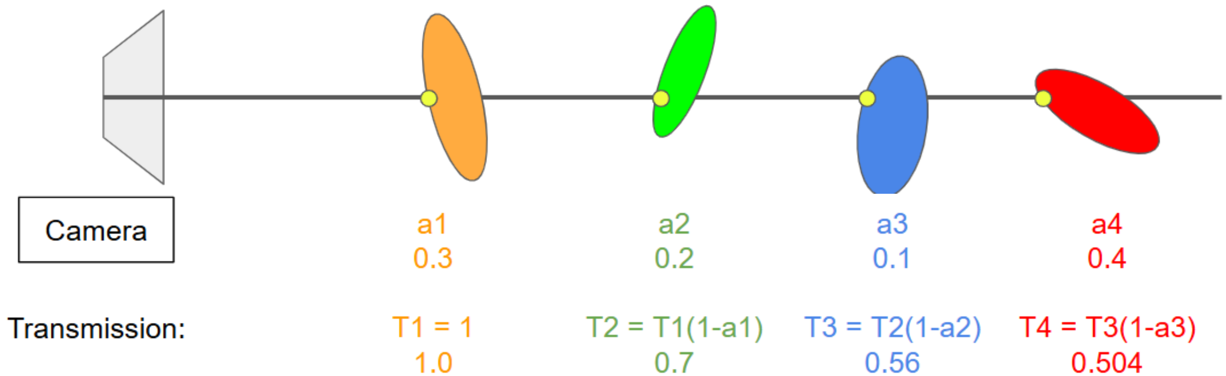


$$C \approx \sum_{i=1}^{\boxed{N}=128} T_i \alpha_i c_i \qquad T_i = \prod_{j=1}^{i-1}(1-\alpha_j) \qquad \alpha_i = 1 - e^{-\sigma_i \delta t_i}$$

For NeRF-based methods, despite there are only four primitives, they still require intensive samples in empty space

Slide credit: 陳楚融

Method – When old meets new



$$C = \sum_{i=1}^{N=4} T_i \alpha_i c_i = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1} 1 - \alpha_j \qquad \boldsymbol{\alpha_i = \mathcal{G}(\mathbf{x_{2D}}) \cdot o_i}$$

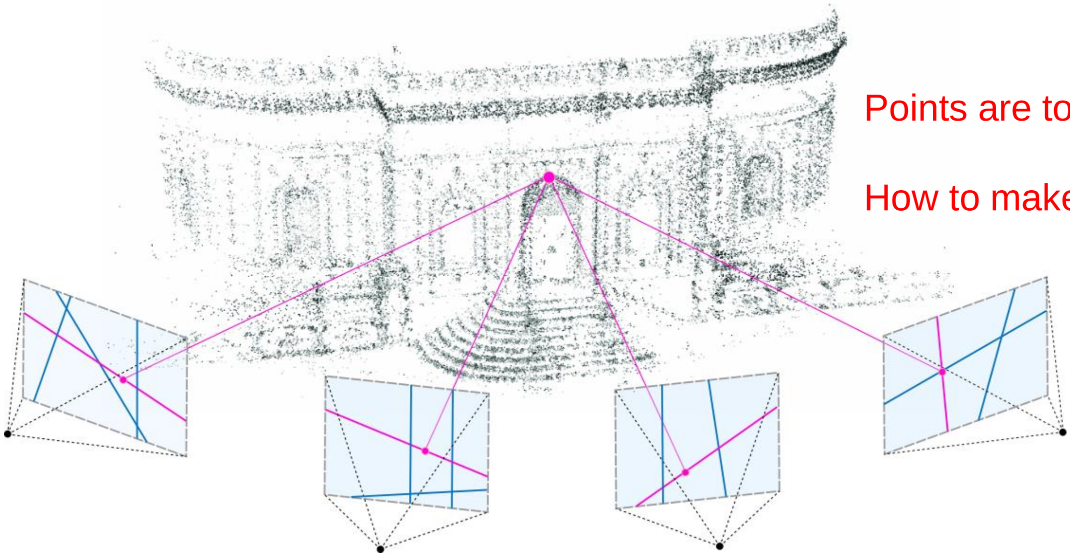
For Rasterization, only 4 times of calculation for 4 object surface encountered

Slide credit: 陳楚融

3D Gaussian Splatting for Real-Time Radiance Field Rendering Link

How to make renderings faster?

- Borrow the idea from point cloud
  - Can be super fast using **rasterization** for rendering
  - Only preserves regions containing objects



Points are too sparse !! (no volume)

How to make it work??

https://www.linkedin.com/pulse/structure-from-motion-manish-joshi/

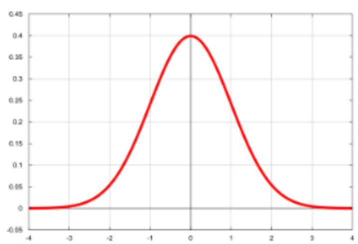
Method – How to solve sparsity problem of point cloud?

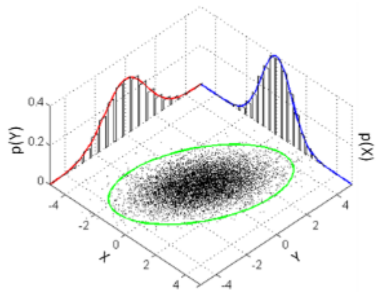
Recall Gaussian distribution

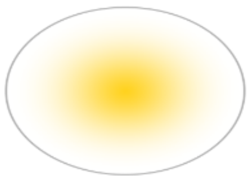
$$\mathcal{G}(\mathbf{x}-\mu) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

μ is the mean, $\mathbf{\Sigma}$ is the covariance matrix

Resemble ellipsoid







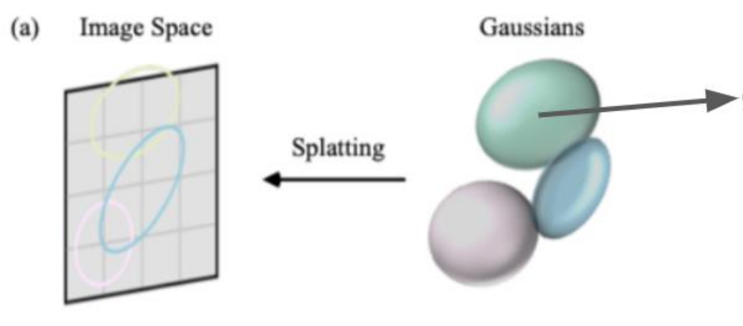
Slide credit: 陳楚融

3D Gaussian Splatting for Real-Time Radiance Field Rendering Link

Method – How to solve sparsity problem of point cloud?

Use numerous 3D Gaussians distributed in the space, each having volume, direction and color

→ Solve the discontinuity problem of point-based method



$$\mathcal{G}(\mathbf{x}-\mu) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boxed{\mu)^T\Sigma^{-1}}(\mathbf{x}-\mu)\right)$$

mean, covariance, color, opacity

Slide credit: 陳楚融

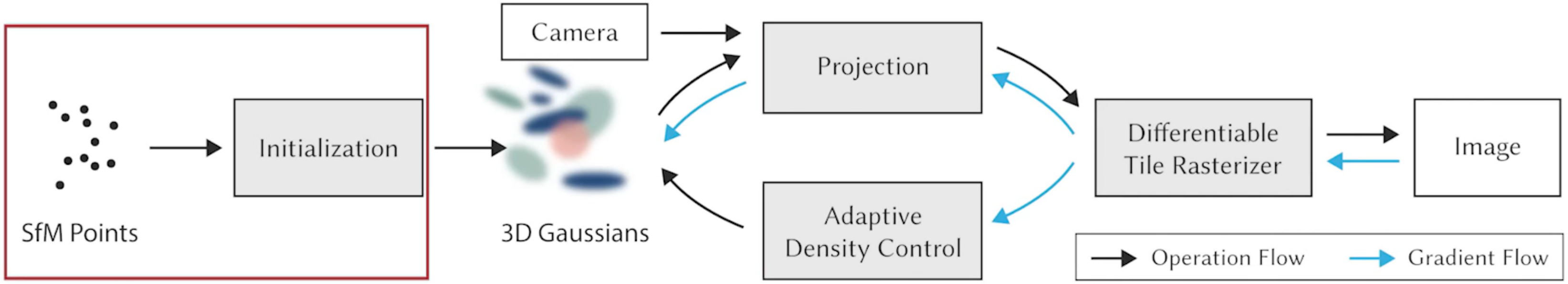
3D Gaussian Splatting for Real-Time Radiance Field Rendering Link

Method – When old meets new

- Structure from Motion:
  - from multi-view image to sparse point cloud

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Photo credit: colmap

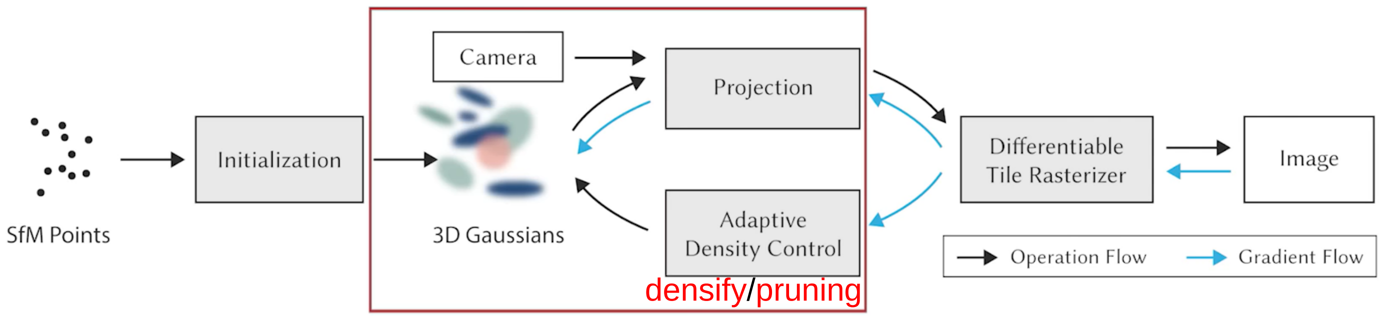


We start with a set of cameras and a point cloud provided by Structure from Motion during calibration.

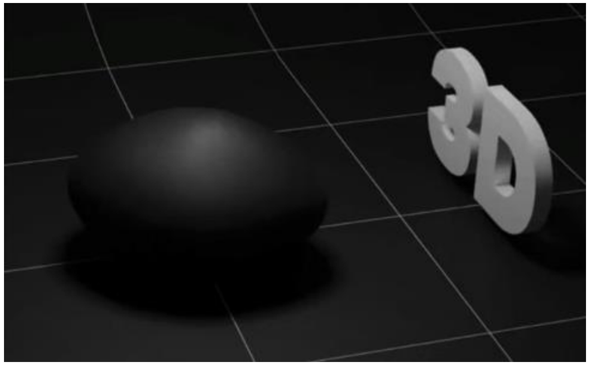
3D Gaussian Splatting for Real-Time Radiance Field Rendering [Link]()

Method

Overview of our Method



Next, we optimize the set of 3D Gaussians to represent the scene

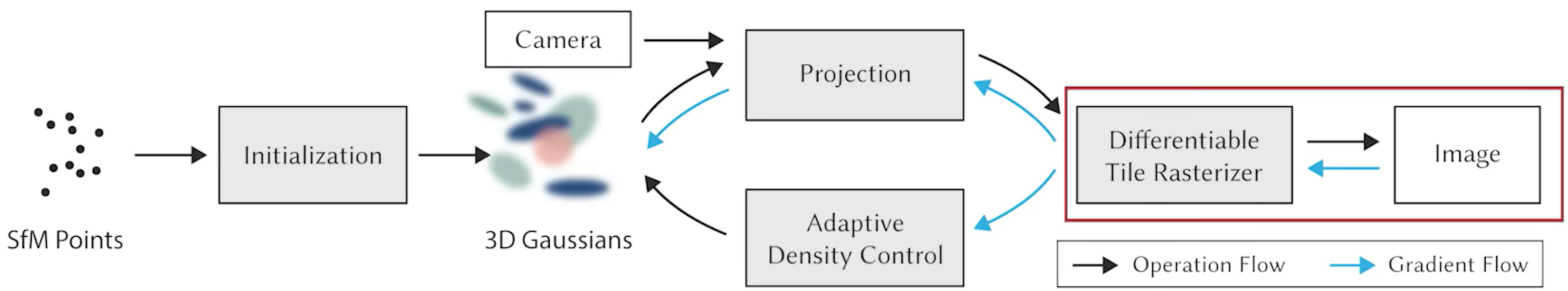


- The whole Gaussian Splat model have N Gaussians
  - Dynamically adjust by densify/pruning
- Each 3D Gaussians composed of four parameters:
  - position (x, y, z),
  - covariance (how it's stretched/scaled: 3x3),
  - color (RGB)
  - alpha (density)

3D Gaussian Splatting for Real-Time Radiance Field Rendering Link

Method

Overview of our Method



Finally, we render transparent anisotropic Gaussians and backpropagate the gradients to their properties.

3D Gaussian Splatting for Real-Time Radiance Field Rendering [Link]()

Result

- The render results

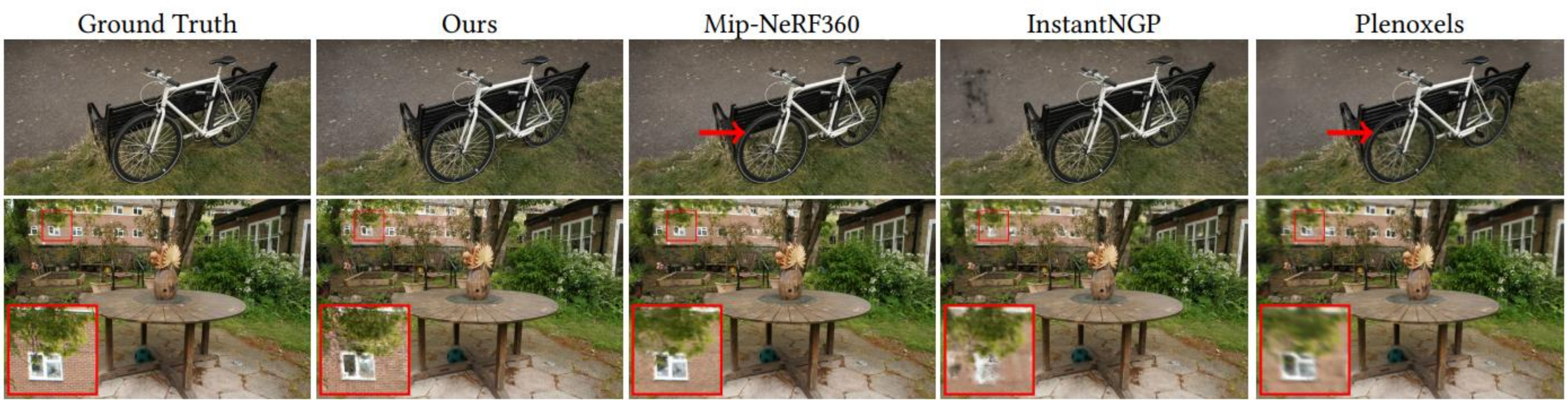
Result

- The render results if we set all 3D gaussians' alpha to 1, without transparency.

Result

- Better visual quality with order of magnitude rendering speed difference.

Dataset	Mip-NeRF360						Tanks&Temples						Deep Blending					
Method|Metric	SSIM↑	PSNR↑	LPIPS↓	Train	FPS	Mem	SSIM↑	PSNR↑	LPIPS↓	Train	FPS	Mem	SSIM↑	PSNR↑	LPIPS↓	Train	FPS	Mem
Plenoxels	0.626	23.08	0.463	25m49s	6.79	2.1GB	0.719	21.08	0.379	25m5s	13.0	2.3GB	0.795	23.06	0.510	27m49s	11.2	2.7GB
INGP-Base	0.671	25.30	0.371	5m37s	11.7	13MB	0.723	21.72	0.330	5m26s	17.1	13MB	0.797	23.62	0.423	6m31s	3.26	13MB
INGP-Big	0.699	25.59	0.331	7m30s	9.43	48MB	0.745	21.92	0.305	6m59s	14.4	48MB	0.817	24.96	0.390	8m	2.79	48MB
M-NeRF360	0.792†	27.69†	0.237†	48h	0.06	8.6MB	0.759	22.22	0.257	48h	0.14	8.6MB	0.901	29.40	0.245	48h	0.09	8.6MB
Ours-7K	0.770	25.60	0.279	6m25s	160	523MB	0.767	21.20	0.280	6m55s	197	270MB	0.875	27.78	0.317	4m35s	172	386MB
Ours-30K	0.815	27.21	0.214	41m33s	134	734MB	0.841	23.14	0.183	26m54s	154	411MB	0.903	29.41	0.243	36m2s	137	676MB



Pros and Cons

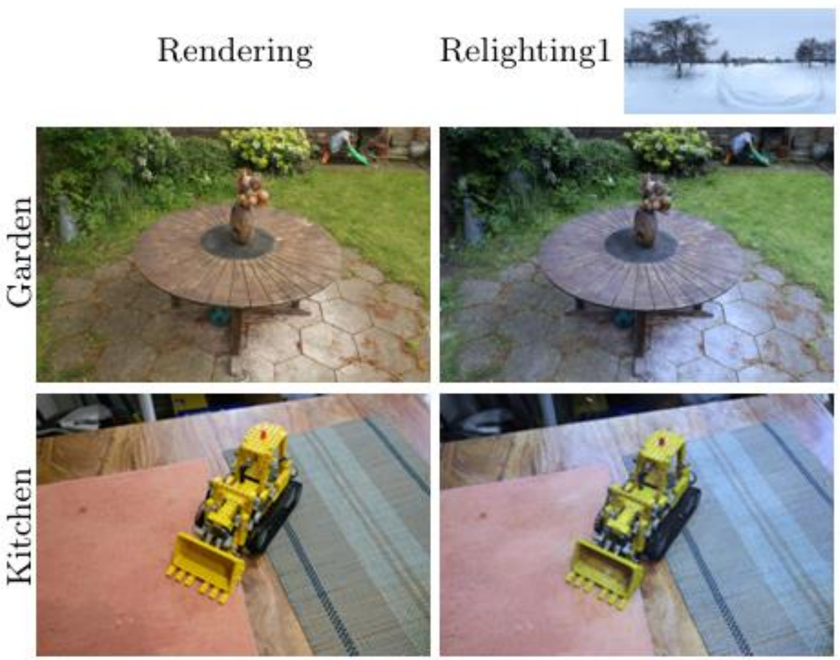
NeRF | Gaussian Splatting

Slide credit: AI 甘安捏

3D Gaussian Splatting for Real-Time Radiance Field Rendering [Link]

Extensions of 3DGS

- **Relightable 3D Gaussians**: Realistic Point Cloud Relighting with BRDF Decomposition and Ray Tracing (ECCV 2024)
- **Gaussian Grouping**: Segment and Edit Anything in 3D Scenes (ECCV 2024)



What to Cover Today?

- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian

DREAMFUSION:
TEXT-TO-3D USING 2D DIFFUSION

Ben Poole
Google Research

Ajay Jain
UC Berkeley

Jonathan T. Barron
Google Research

Ben Mildenhall
Google Research

2023 ICLR

https://dreamfusion3d.github.io/

Goal

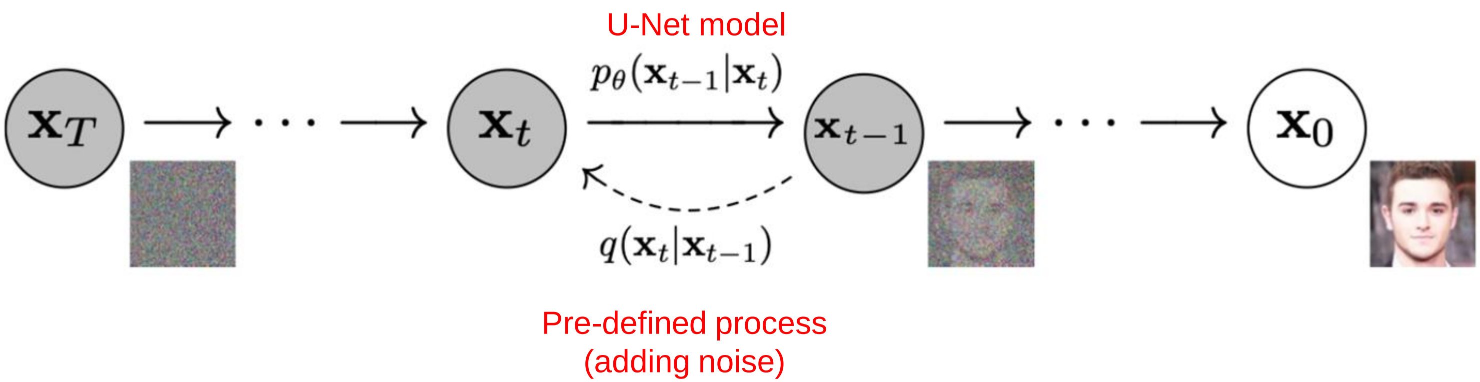

an orangutan making a clay bowl on a throwing wheel*


a raccoon astronaut holding his helmet†


a blue jay standing on a large basket of rainbow macarons*

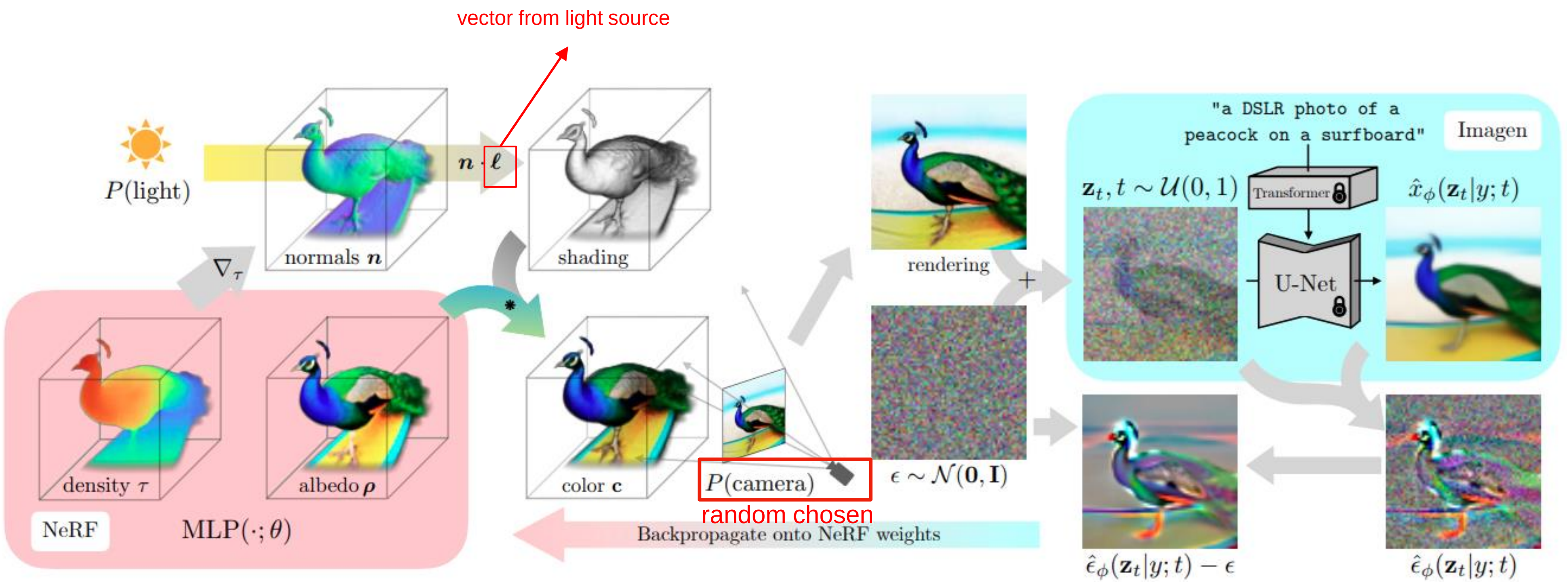
- Take description as input and generate corresponding 3D results (via 2D rendering)
- Without paired “text and 3D object”
- Combining NeRF and 2D text-to-image diffusion model

Recap: Diffusion model (intuitively)



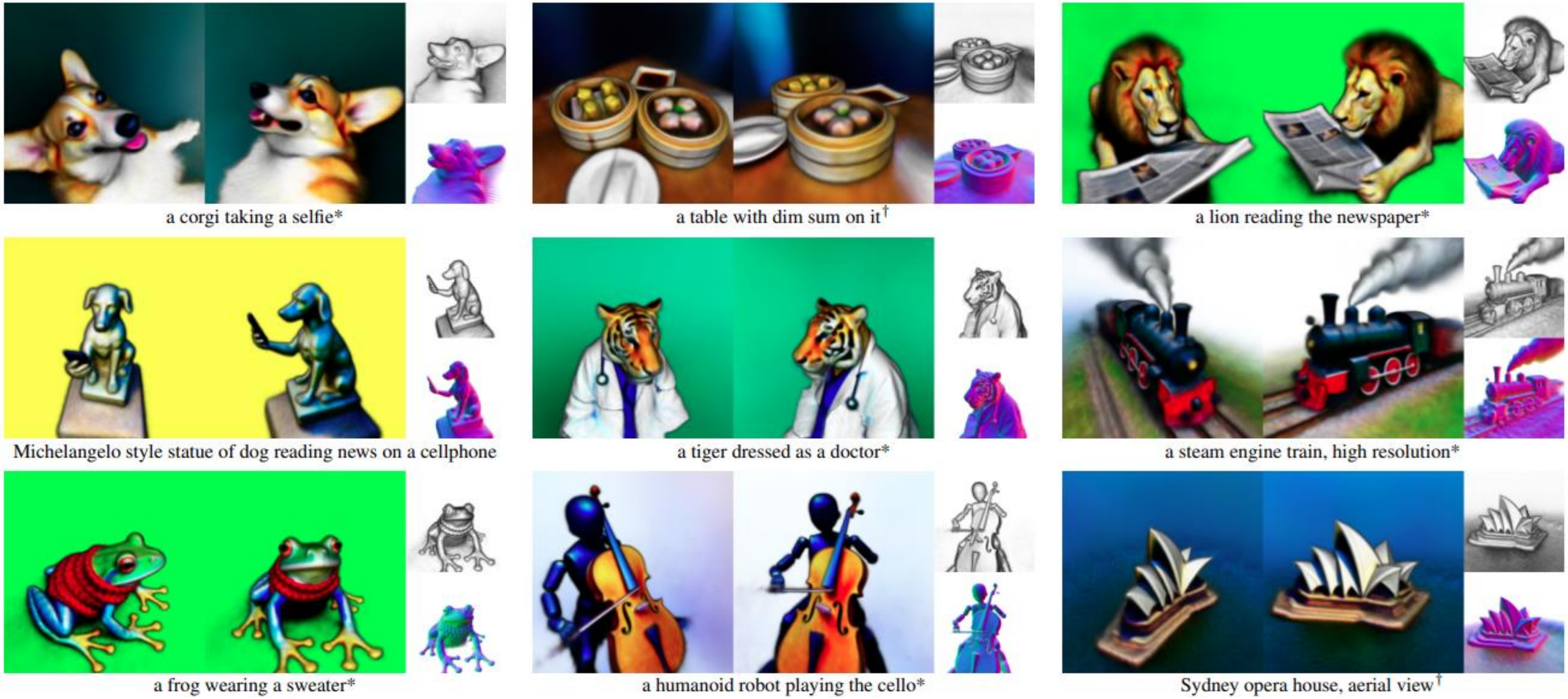
- Can be viewed as denoising from a Gaussian noise image
- Each step makes little progress of denoising (total about 1000 steps)
- Output image of each step can be seen as the **original image** combining with a **noise** using specific ratio
- The process can also be seen as predicting the **added noise**

Method



- The left part is a standard NeRF with shading condition
- Combine the rendered NeRF image with random noise to simulate a state of the text-to-image diffusion model
- The difference between the predicted noise and the inserted noise is treated as the rendering loss to guide NeRF

Result


a corgi taking a selfie*
a table with dim sum on it†
a lion reading the newspaper*
Michelangelo style statue of dog reading news on a cellphone
a tiger dressed as a doctor*
a steam engine train, high resolution*
a frog wearing a sweater*
a humanoid robot playing the cello*
Sydney opera house, aerial view†

Result

an all-utility vehicle driving across a stream[†]

a chimpanzee dressed like Henry VIII king of England*

a baby bunny sitting on top of a stack of pancakes[†]

a sliced loaf of fresh bread

a bulldozer clearing away a pile of snow*

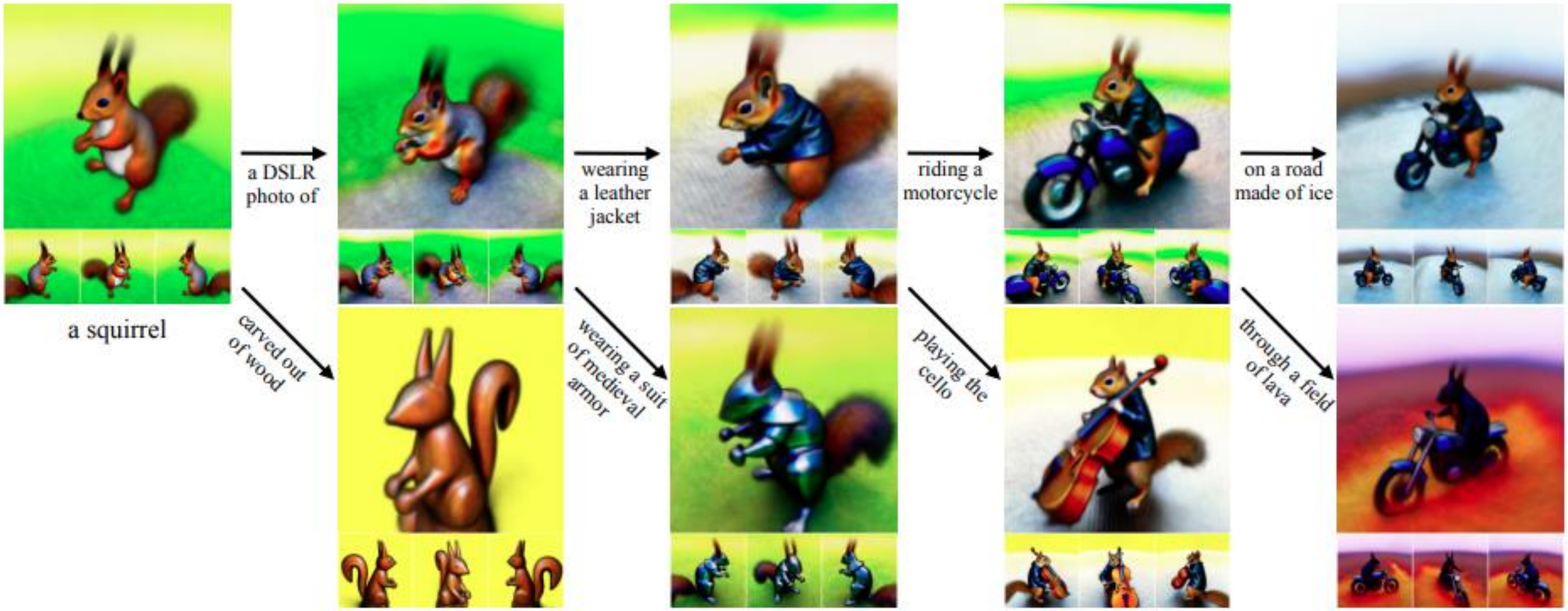
a classic Packard car*

zoomed out view of Tower Bridge made out of gingerbread and candy[‡]

a robot and dinosaur playing chess, high resolution*

a squirrel gesturing in front of an easel showing colorful pie charts

Result


a squirrel
a DSLR photo of
wearing a leather jacket
riding a motorcycle
on a road made of ice
carved out of wood
wearing a suit of medieval armor
playing the cello
through a field of lava

TPA3D: Triplane Attention for Fast Text-to-3D Generation

[1]National Taiwan University, [2]NVIDIA

[Project page](#)

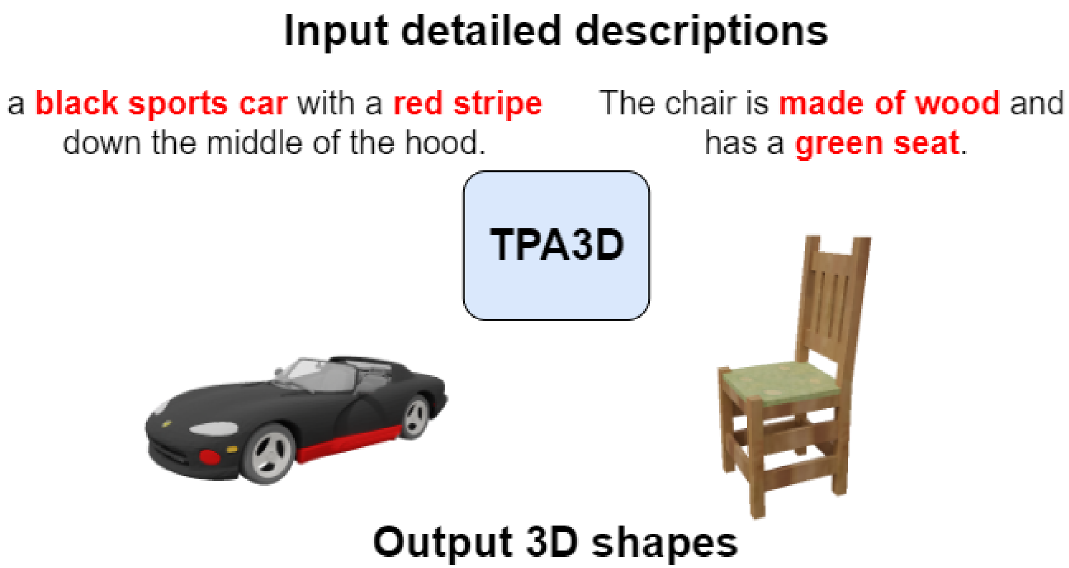



Bin-Shih Wu[1*] Hong-En Chen[1*] Sheng-Yu Huang[1] Yu-Chiang Frank Wang[1,2]

Slide credit: 吳彬世

Task

- Text-to-3D generation with detailed descriptions

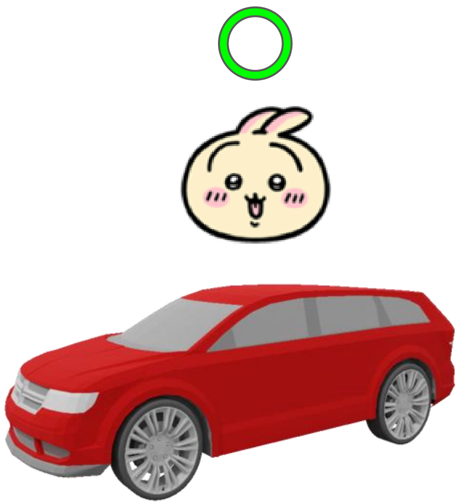


兄弟 細節一點

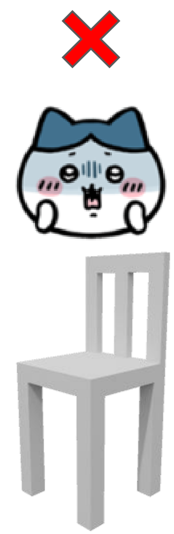
Slide credit: 吳彬世

Motivation

- Lack of supervision: Need large 3D dataset with paired **detailed descriptions**
- Lack of detailed: Omitted details for the texture and geometry in the text prompt



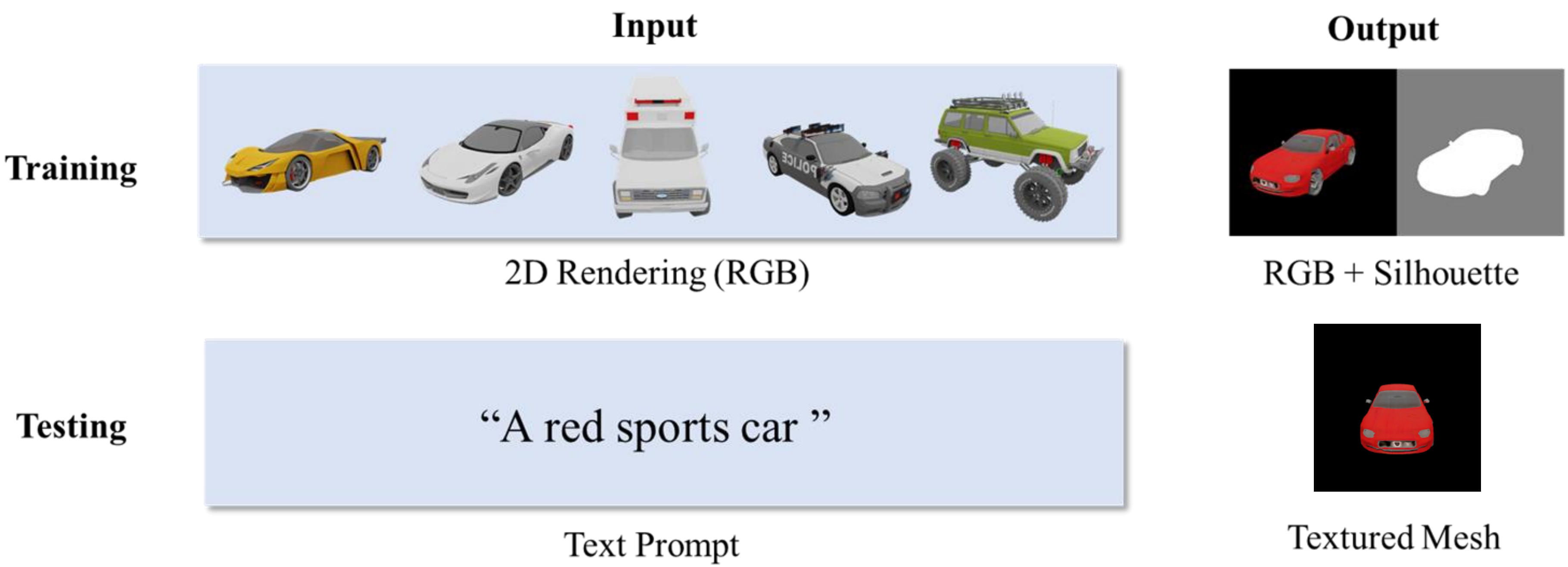
a red car



a white chair with
rounded back *and* ***wooden legs***

Problem Settings

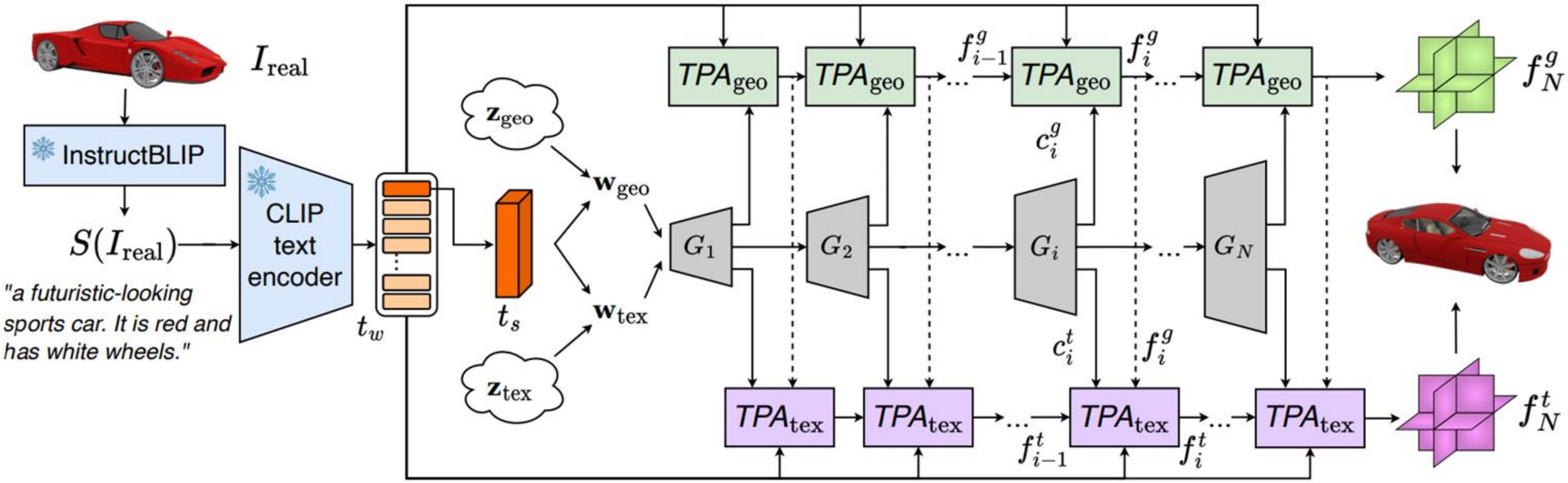
- Achieve text-guided 3D generation **without human-annotated text-3D pairs**



Slide credit: 吳彬世

Methods

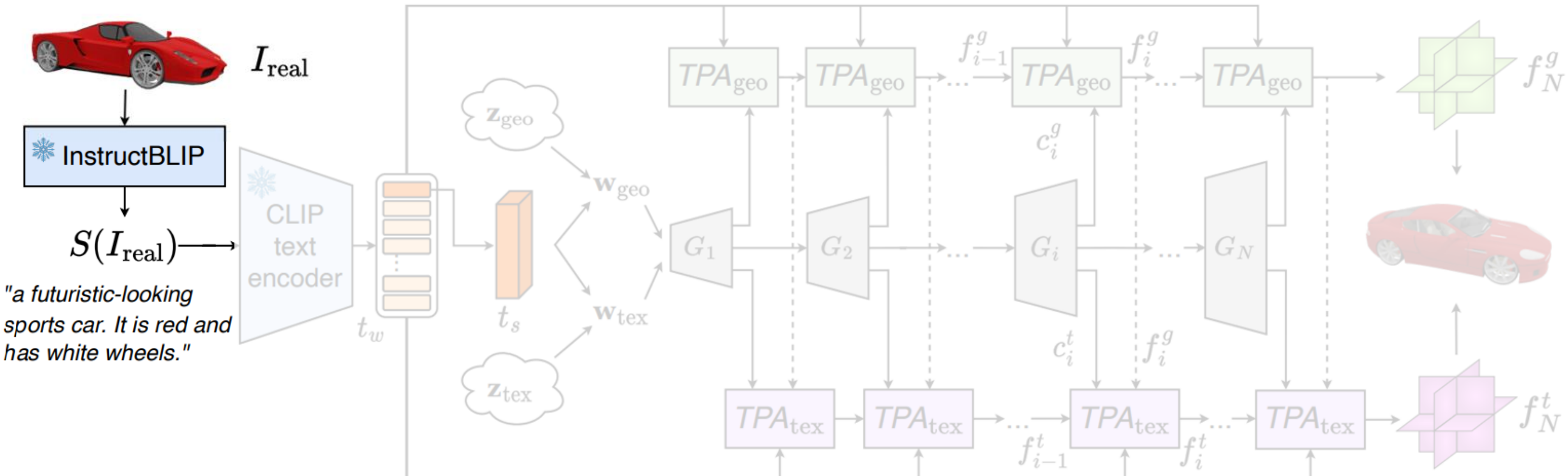
- Model overview



Slide credit: 吳彬世

Methods

- Pseudo caption generation



Slide credit: 吳彬世

Pseudo Caption Generation

InstructBLIP

[paper link](#)

- Given prompt: *“In the image, the background is black. Describe the design and appearance of the {category} in detail.”*
- Remove redundant information

a yellow sports car with a black stripe down the middle of its body.

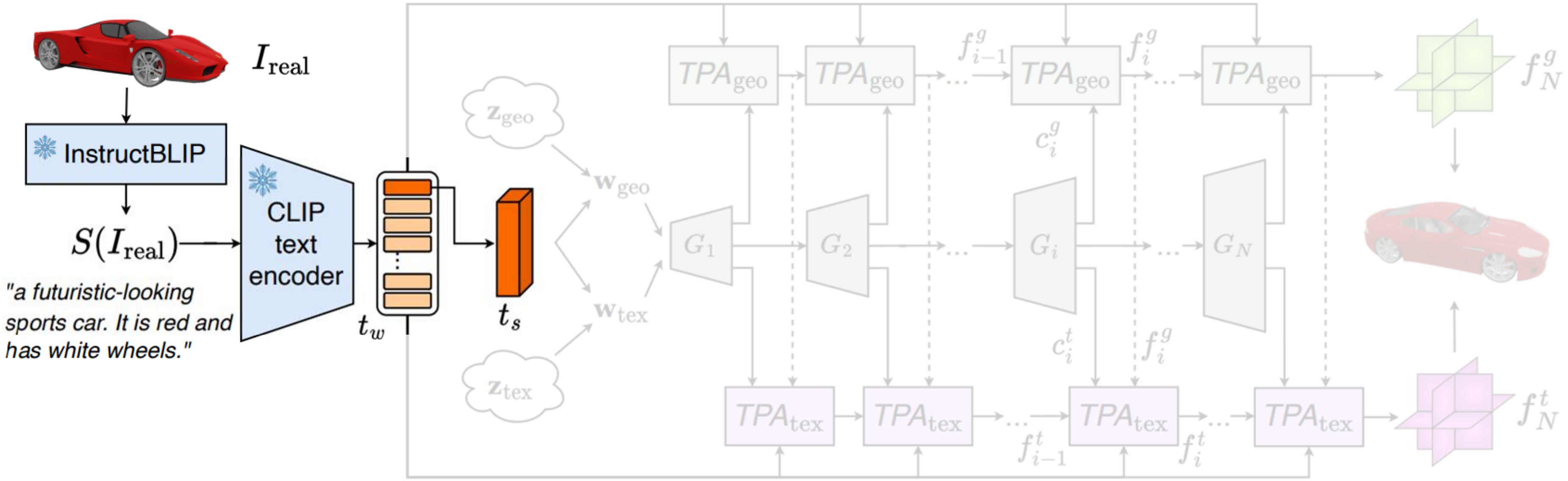
the chair is made of wood and has a brown leather seat.

a 1950s-style Harley Davidson. It has a red and white color scheme.

Slide credit: 吳彬世

Methods

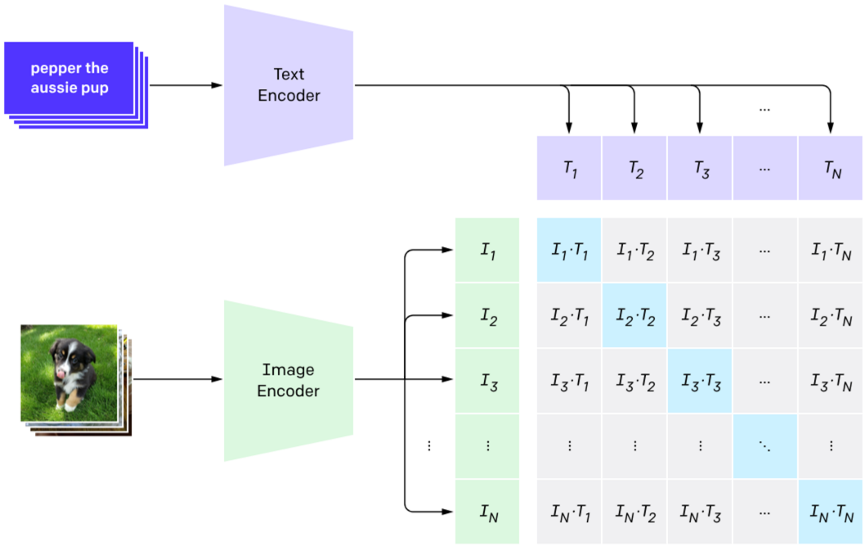
- Encode text prompt to word-level features & sentence-level features



Slide credit: 吳彬世

Encode Text Features

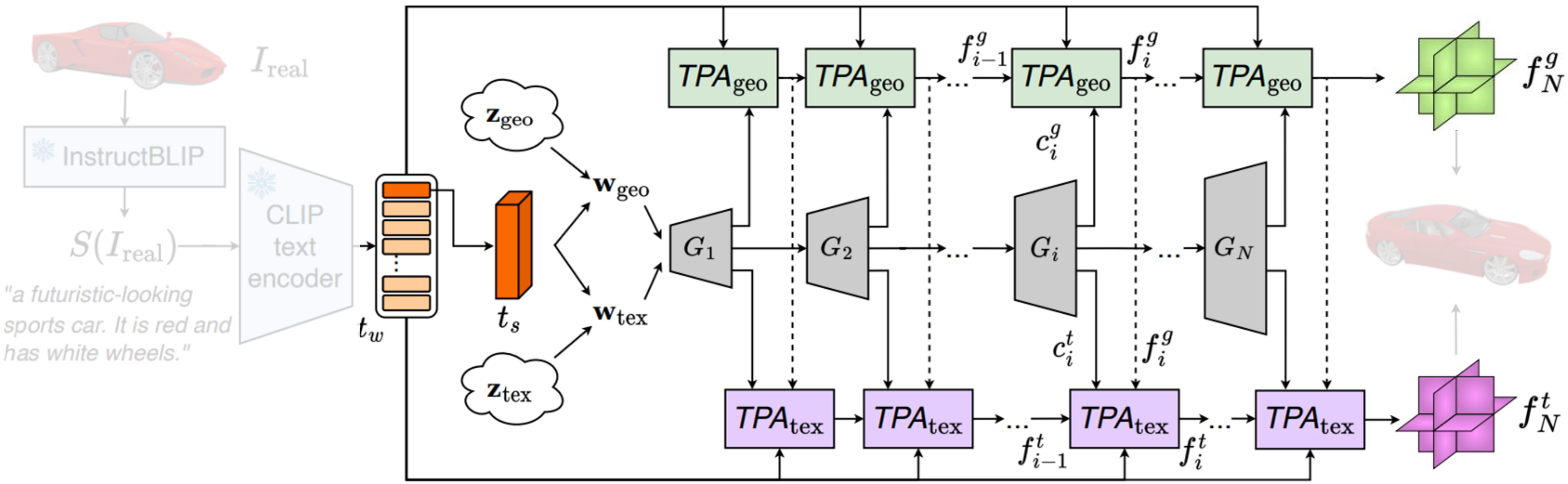
- CLIP (Contrastive Language-Image Pretraining)
- Word-level: second-last layer of VIT
- Sentence-level: after projection



Slide credit: 吳彬世

Methods

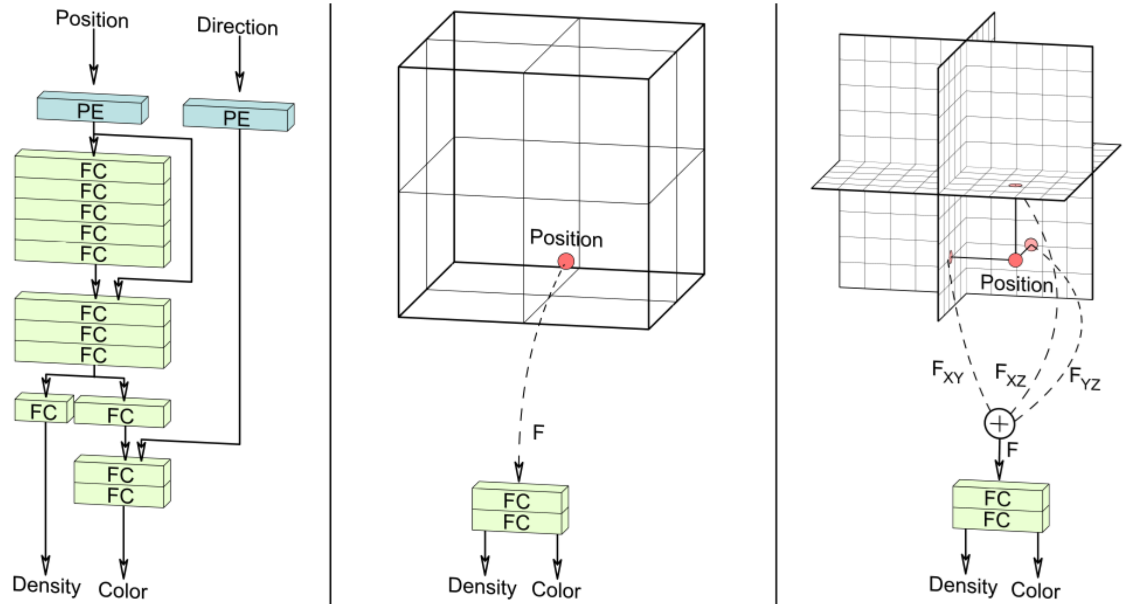
- Generate triplane features



Slide credit: 吳彬世

Triplane Features

- Use 2D planes to learn 3D features (efficient)

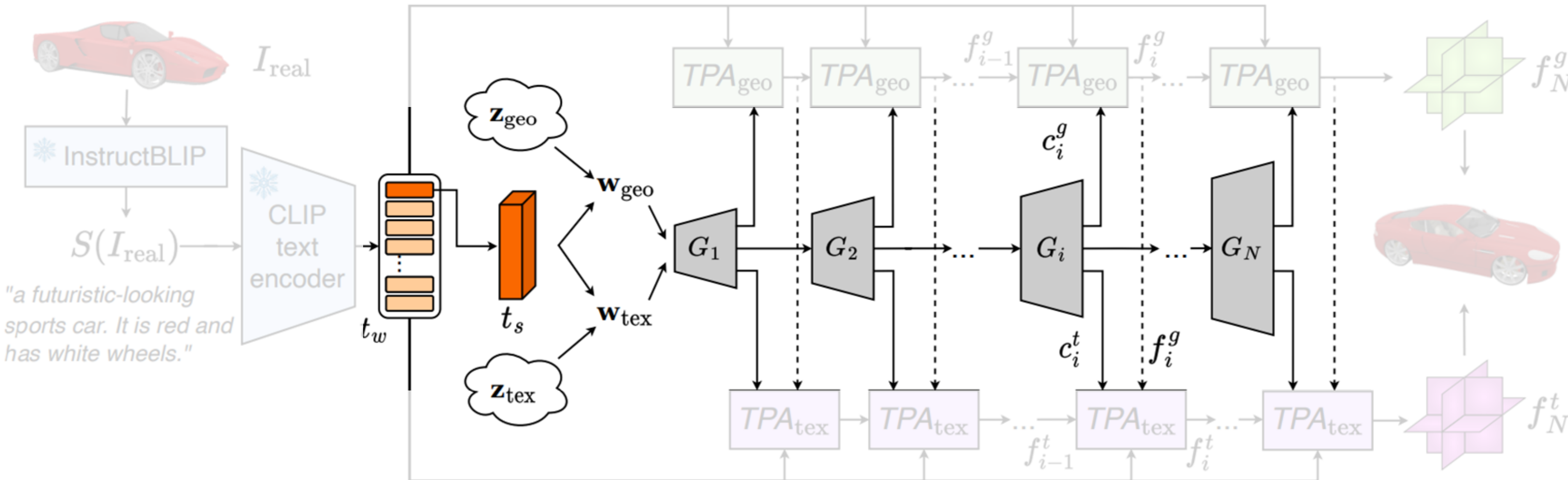


(a) NeRF (Implicit) (b) Voxels (Explicit or Hybrid) (c) Ours (Hybrid)

ref: EG3D

Methods

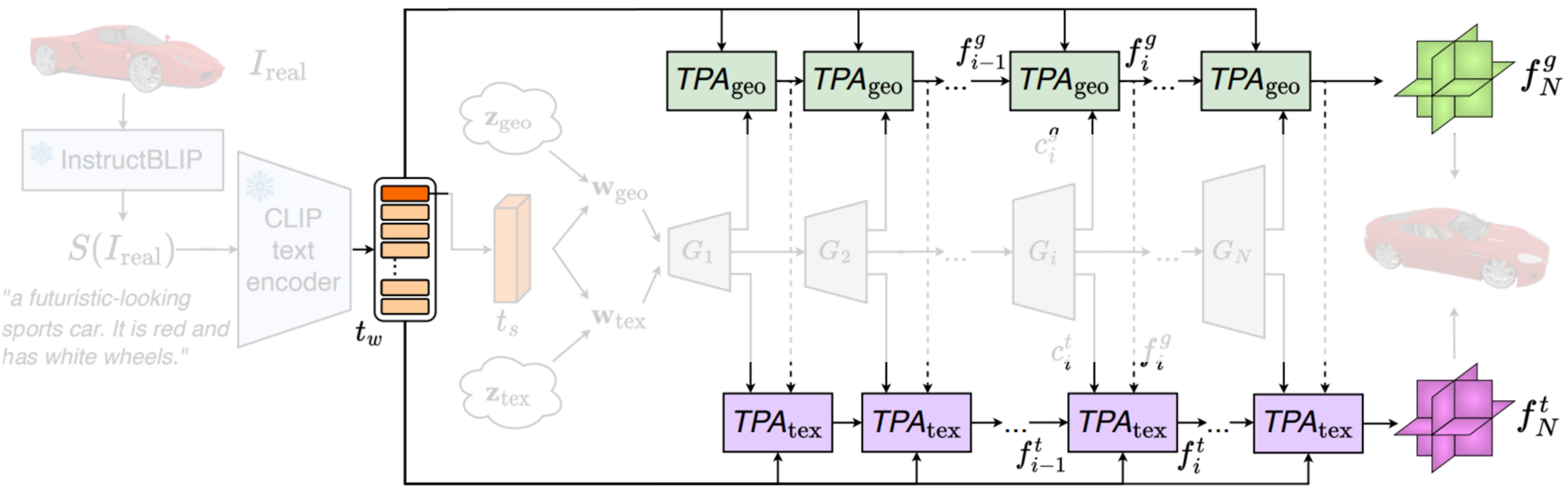
- Generate sentence-level triplane features



Slide credit: 吳彬世

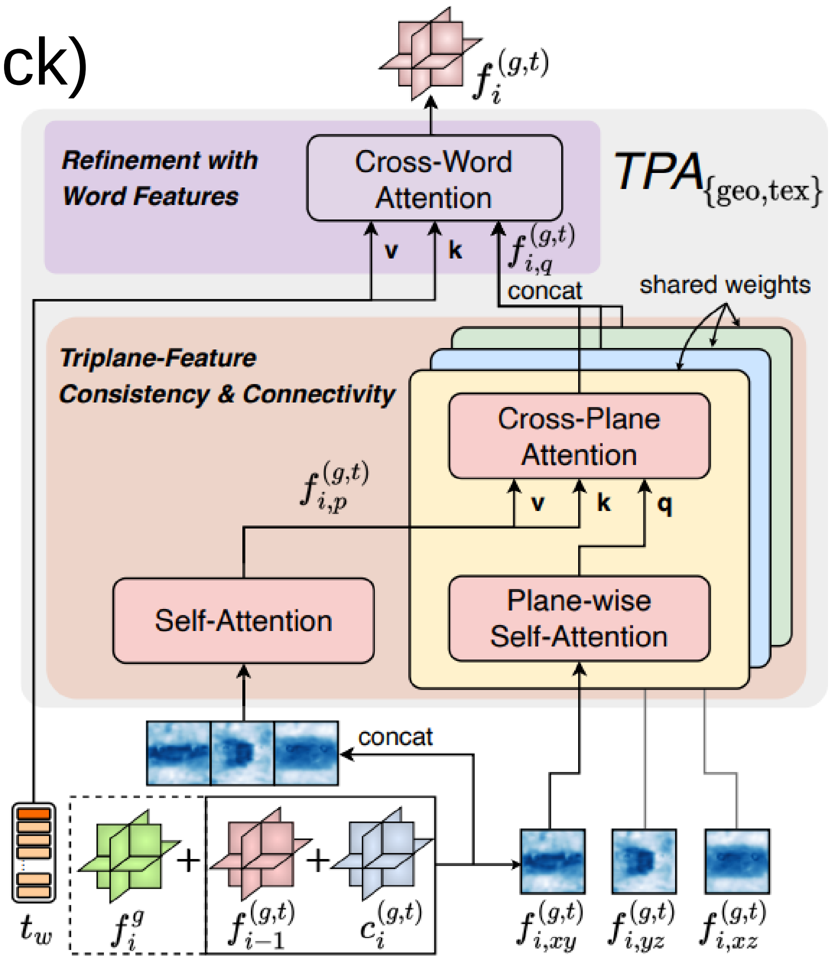
Methods

- Generate word-level triplane features



Triplane Attention Block (TPA Block)

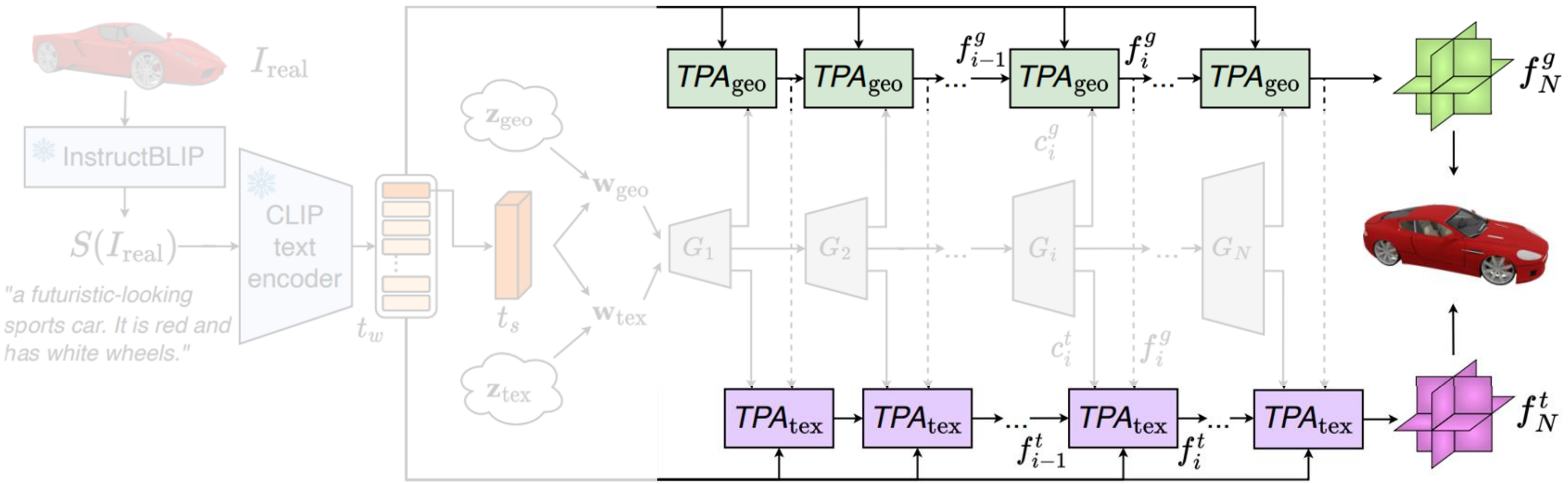
- Plane-wise Self-Attention
  - Intra-plane consistency
  - Extract plane-wise content features
- Cross-Plane Attention
  - Inter-plane connectivity
  - Ensure multi-aspect correspondence across different planes
- Cross-Word Attention
  - Word-level refinement
  - Incorporate word-level information into triplane features



Slide credit: 吳彬世

Methods

- Predict SDF values, deformations, and colors



Slide credit: 吳彬世

Training Objectives

- Text-guided discriminators
  - Use camera pose & text features as condition
  - On both rendered RGB images & silhouette masks
- Mismatching loss
  - Use negative pairs to increase discriminative ability
- CLIP loss
  - CLIP similarity score

$$\mathcal{L} = \mathcal{L}(D_{\mathrm{rgb}}, G) + \mathcal{L}(D_{\mathrm{mask}}, G) + \mathcal{L}_{\mathrm{mis}} + \mathcal{L}_{\mathrm{clip}}$$

Slide credit: 吳彬世

Experiments

- Qualitative comparison

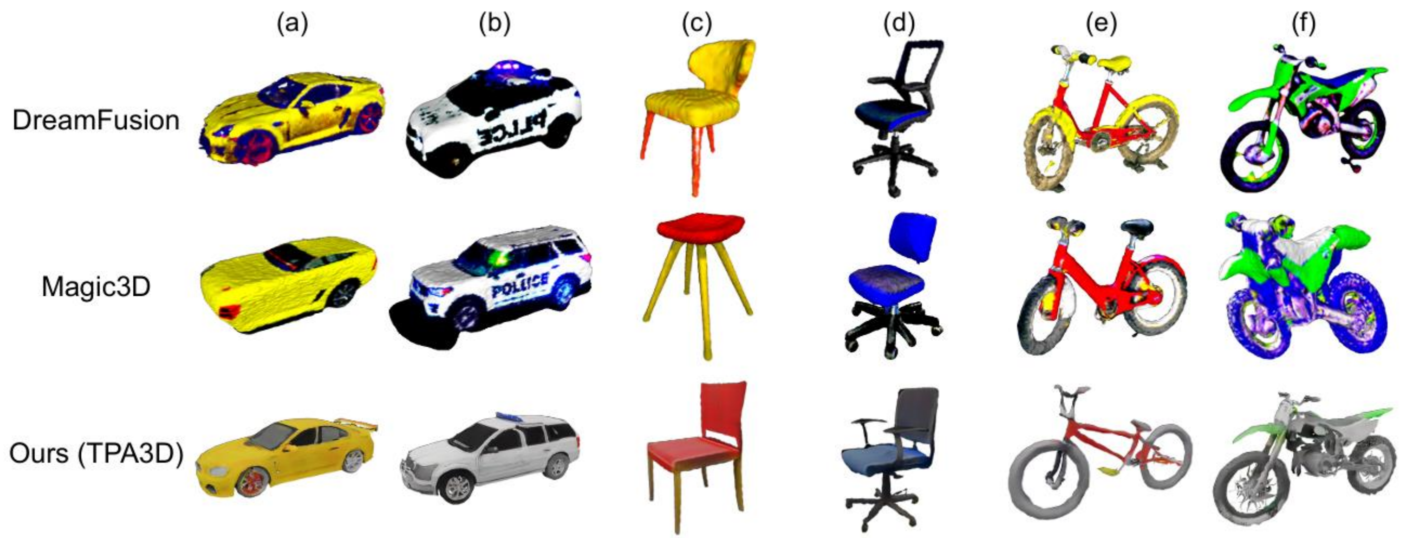


Fig. 6: Qualitative comparisons with SDS-based methods. Each column takes a unique text prompt of (a)*"a yellow sports car with red wheel and tinted window"*, (b)*"a white SUV with a blue police light on top of it"*, (c)*"the chair has a red seat and yellow legs"*, (d)*"a black office chair with a blue seat"*, (e)*"a red bicycle with yellow pedals"*, and (f)*"a green and white dirt bike"*.

Slide credit: 吳彬世

Experiments

- Inference time comparison

Method	Device	Output	Time
DreamFusion [40]	TPUv4 machine	Rendering	90 min
Magic3D [31]	NVIDIA A100 x8	Rendering	40 min
TITG3SG [32]	Telsa V100-32G	Voxel	2.21 sec
TAPS3D [55]	Telsa V100-32G	Rendering	0.05 sec
TAPS3D [55]	Telsa V100-32G	Mesh	1.03 sec
Ours (TPA3D)	Telsa V100-32G	Rendering	0.09 sec
Ours (TPA3D)	Telsa V100-32G	Mesh	2.87 sec

Slide credit: 吳彬世

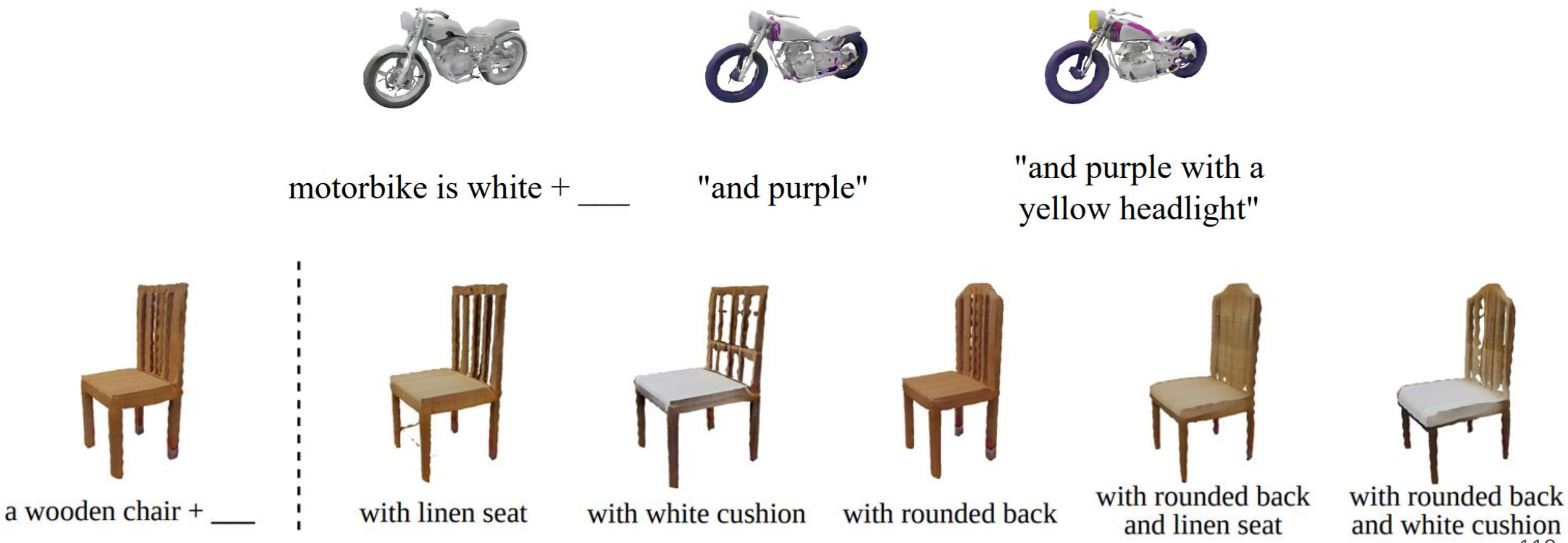
Experiments

- Interpolation

Slide credit: 吳彬世

Experiments

- Incremental manipulation



Slide credit: 吳彬世

More references about further topics of 3D supervision-free text-to-3D

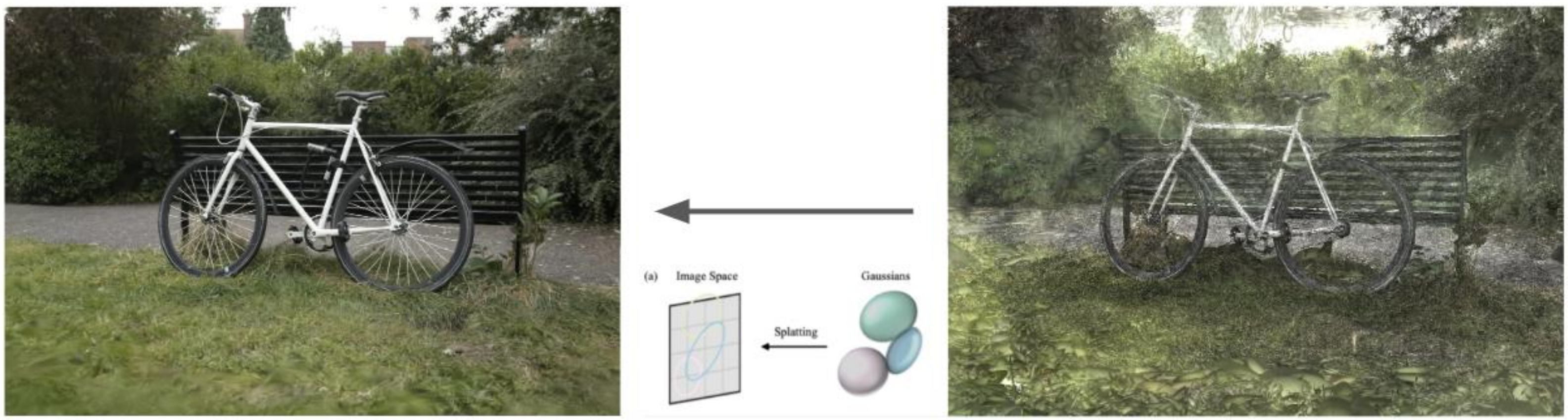
- **Magic3D**: High-Resolution Text-to-3D Content Creation (CVPR 2023)
- **MVDream:** Multi-view Diffusion for 3D Generation
- **TAPS3D**: Text-Guided 3D Textured Shape Generation From Pseudo Supervision (CVPR 2023)
- **LucidDreamer**: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching (CVPR 2024)
- **GALA3D**: Towards Text-to-3D Complex Scene Generation via Layout-guided Generative Gaussian Splatting (ICML 2024)

What to Cover Today?

- Introduction to 3D Vision
- Part I: Traditional 3D Representation
  - Perception
  - 3D Reconstruction
- Part II: Recent 3D Representation
  - Neural Radiance Fields
  - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
  - Text-to-3D without 3D supervision
  - 4D Gaussian Splatting

Recap 3DGS

Represent a static 3D scene by many 3D Gaussians with learnable size, position, color, etc.



How about dynamic scenes? (given videos from multiple views)

Slide credit: 陳楚融

4D-Rotor Gaussian Splatting: Towards Efficient Novel View Synthesis for Dynamic Scenes

Yuanxing Duan[*]
Peking University
China
mjdyx@pku.edu.cn

Fangyin Wei[*]
Princeton University
USA
fwei@princeton.edu

Qiyu Dai
Peking University
China
State Key Laboratory of General AI
China
qiyudai@pku.edu.cn

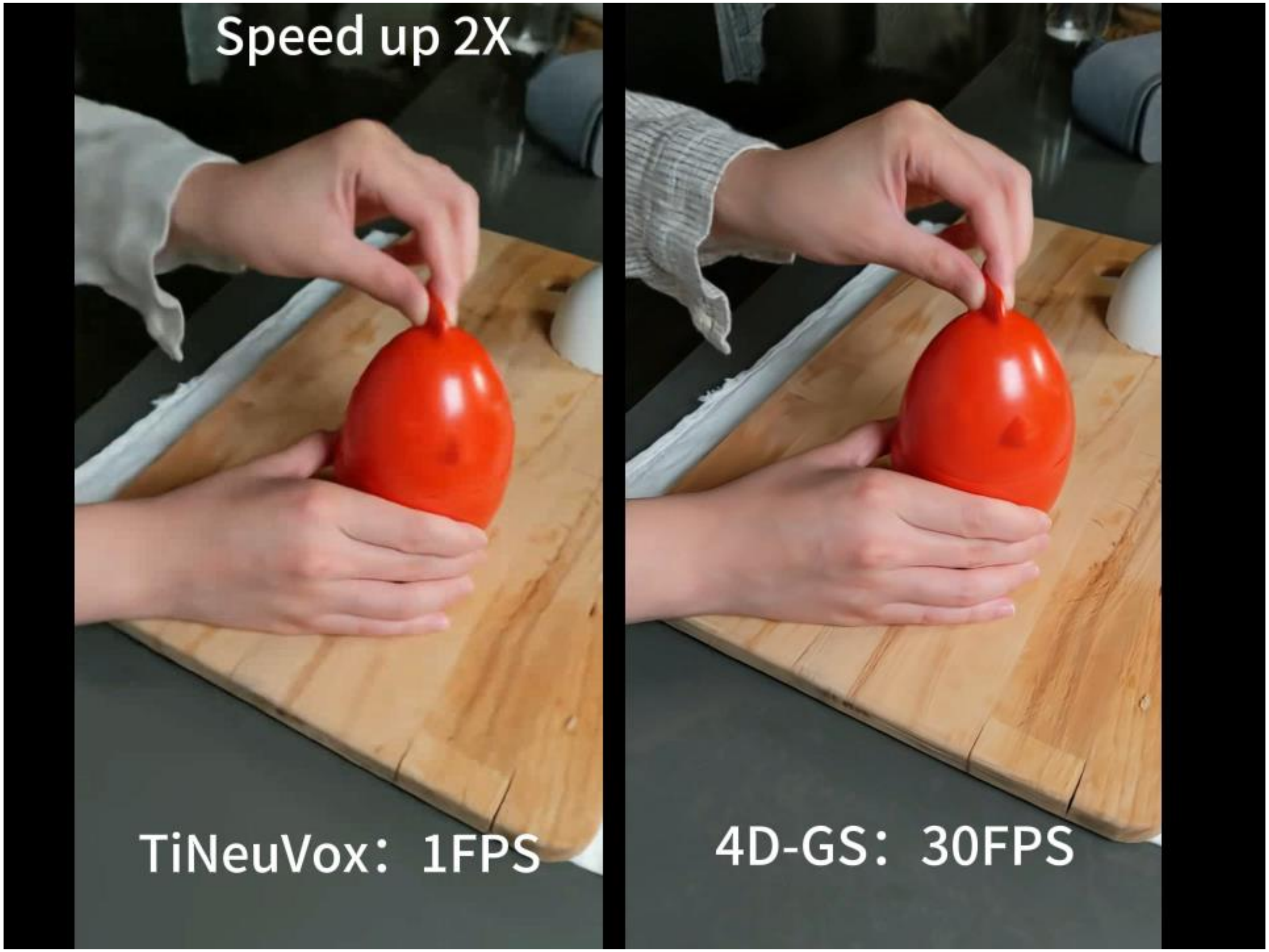
Yuhang He
Peking University
China
2100014725@stu.pku.edu.cn

Wenzheng Chen[†]
Peking University
China
NVIDIA
Canada
wenzhengchen@pku.edu.cn

Baoquan Chen[†]
Peking University
China
State Key Laboratory of General AI
China
baoquan@pku.edu.cn

Dynamic Video Input | The Proposed Method | HyperReel [Attal et al. 2023] | Evaluation on PSNR *vs.* FPS

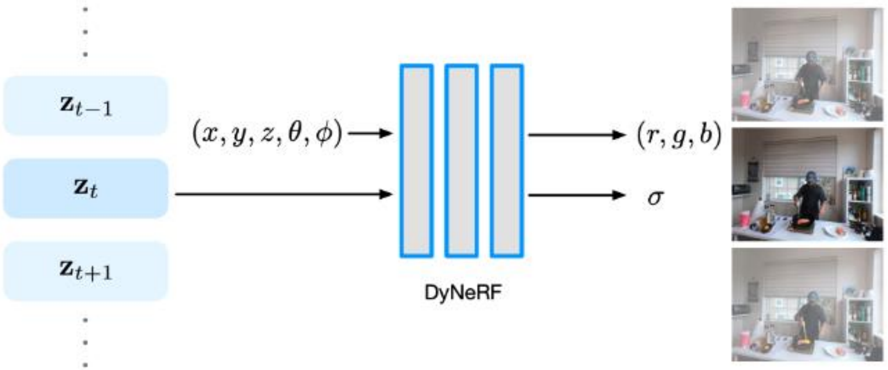
SIGGRAPH 2024



https://guanjunwu.github.io/4dgs/

Previous Methods using NeRF

Add additional temporal dimension or a latent vector conditioned on time to the input of MLP



Why using NeRF is not appropriate?

Slide credit: 陳楚融

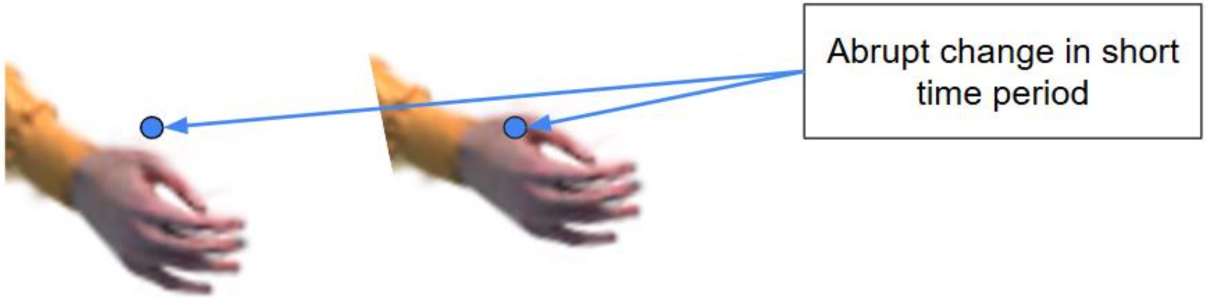
Previous Methods using NeRF

Add additional temporal dimension or a latent vector conditioned on time to the input of MLP

Challenges:

Inherited from NeRF, both training and inference are very slow

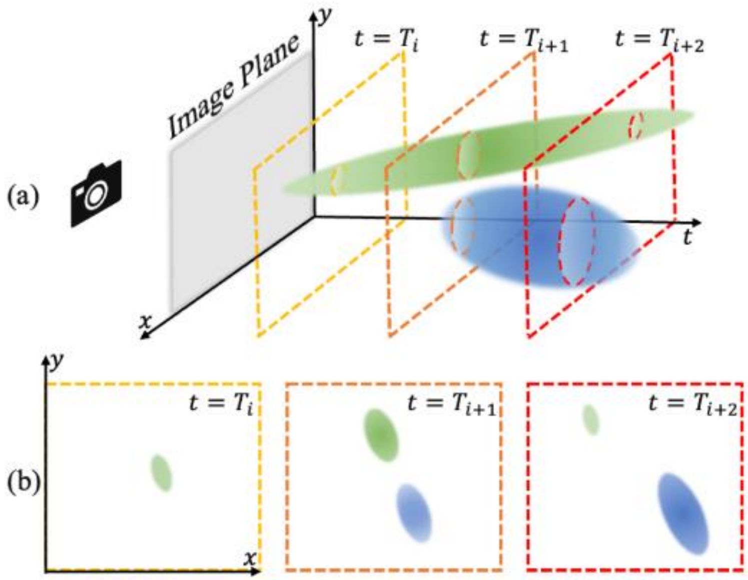
Hard to solve the complexities introduced by temporal-spatial entanglement



Slide credit: 陳楚融

Methods: 4D GS

Directly lift 3D Gaussians to 4D space (simple and intuitive)

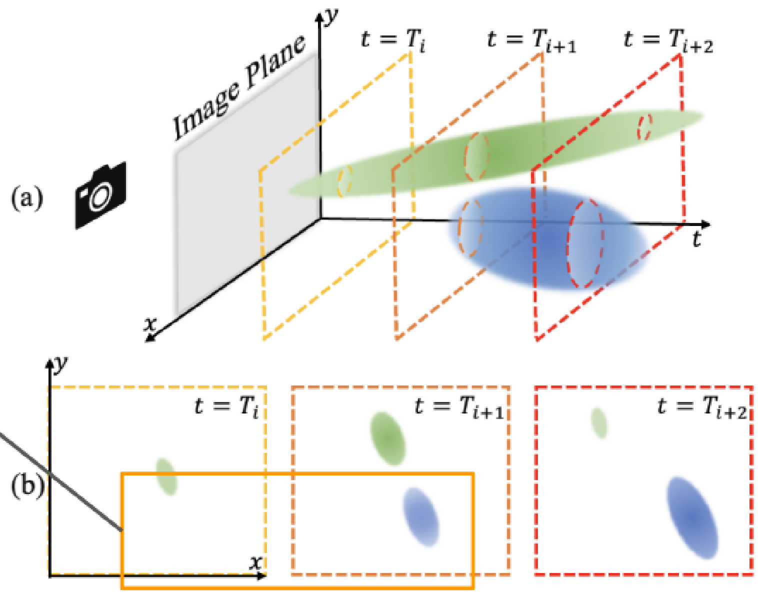


Simplified to 3D for illustration

Slide credit: 陳楚融

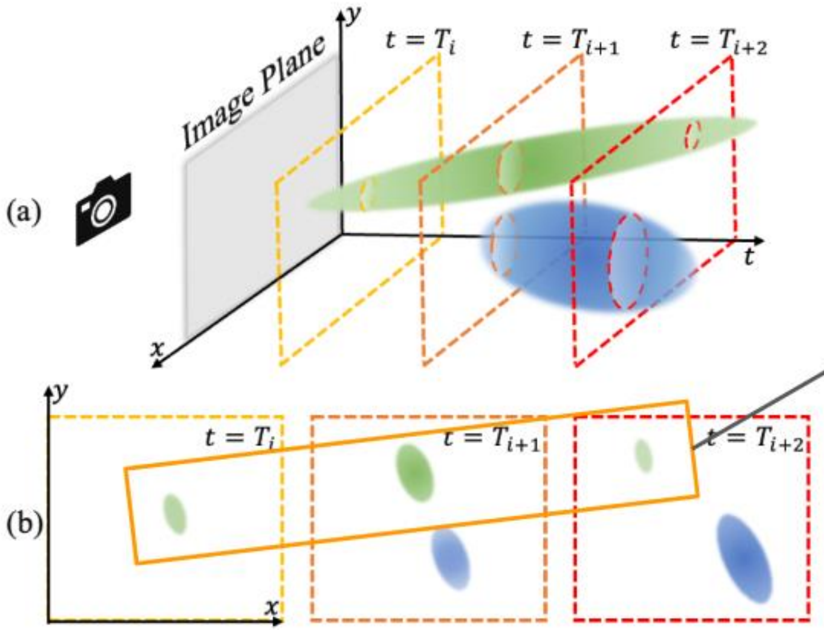
Methods: 4D GS

Naturally capable of dealing with objects that suddenly appear or disappear



Slide credit: 陳楚融

Methods: 4D GS



Can also represent linear movement without additional properties

Slide credit: 陳楚融

Methods: 4D GS

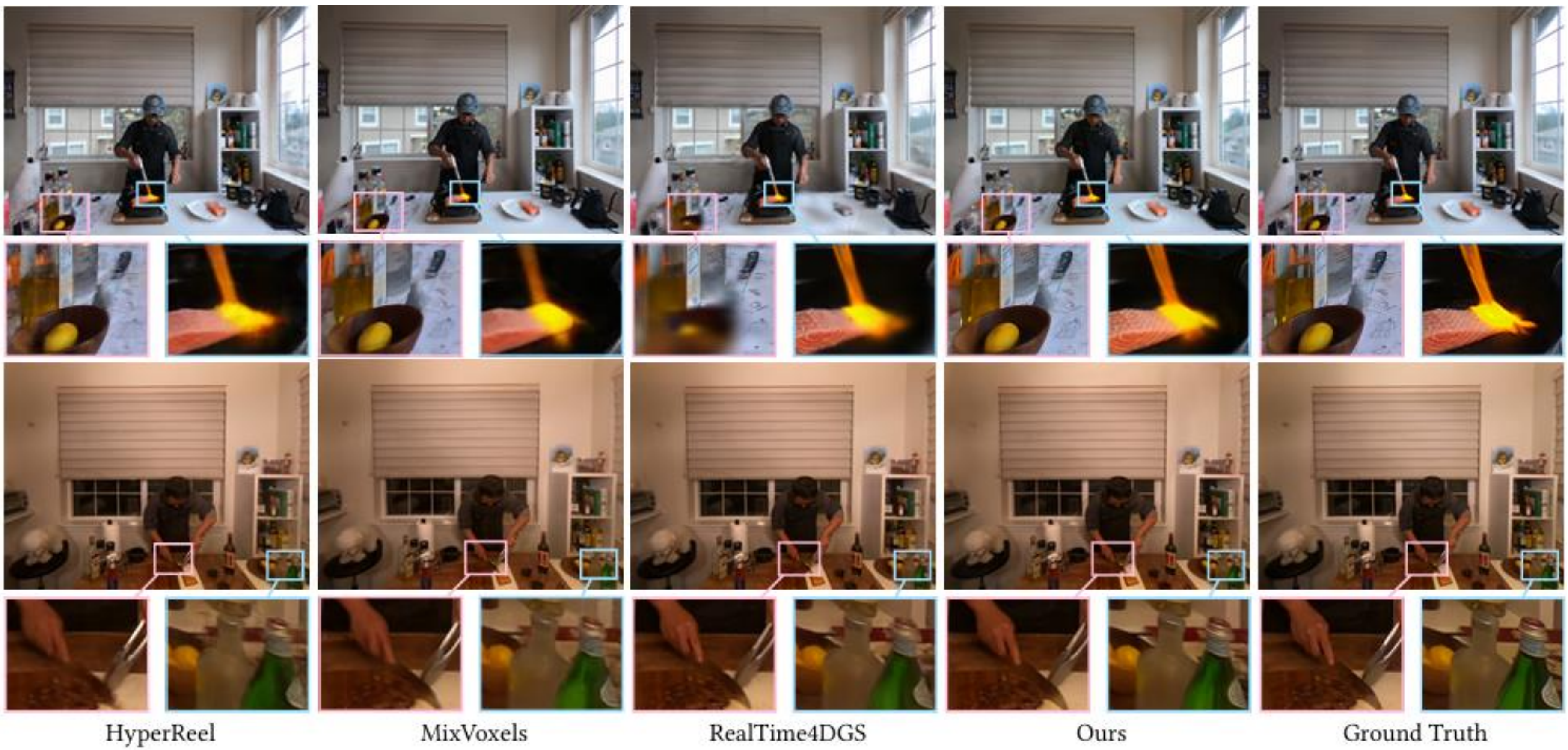
To ensure that nearby Gaussians have similar motion, the speed of close Gaussians are regularized to be close

$$L_{\text{consistent4D}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{s}_i - \boxed{\frac{1}{K} \sum_{j \in \Omega_i} \mathbf{s}_j} \right\|_1$$

Average speed of nearby Gaussians

Slide credit: 陳楚融

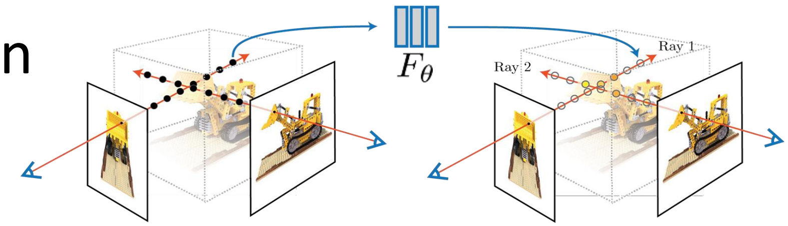
Results



Slide credit: 陳楚融

What We've Covered Today?

- Introduction to 3D Vision
- Part I: Traditional 3D Representation
- Part II: Recent 3D Representation
    - Neural Radiance Fields
    - 3D Gaussian Splatting
- Advanced Topics About NeRF & 3DGS
    - Text-to-3D without 3D supervision
    - Text-to-4D



a blue jay standing on a large basket of rainbow macarons*