GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields

Michael Niemeyer, Andreas Geiger
Max Planck Institute for Intelligent Systems, Tübingen
CVPR 2021, Best Paper Award
https://arxiv.org/abs/2011.12100

Presenter: Minho Park

GIRAFFE Overview

- **Compositional** generative neural **feature** fields

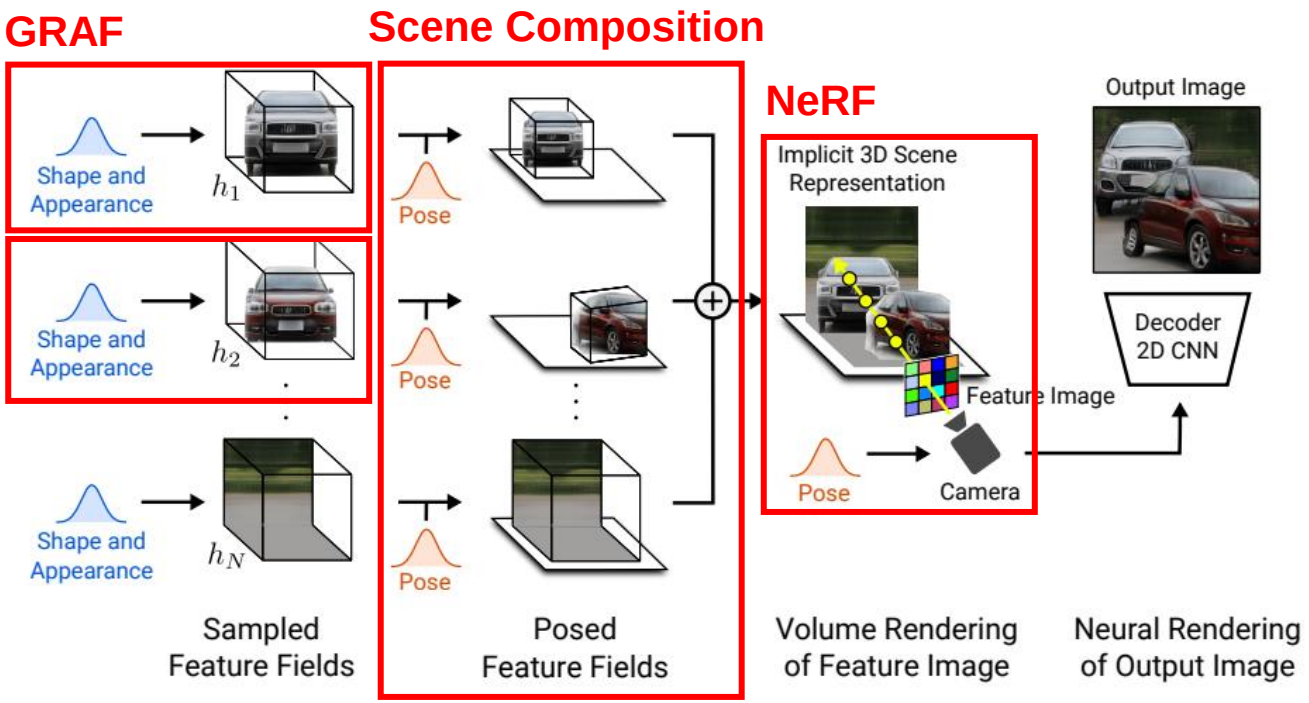


Figure 1: Overview. We represent scenes as compositional generative neural feature fields.

Demos

- https://m-niemeyer.github.io/project-pages/giraffe/index.html

NeRF

- Objective

Fig. 1: We present a method that optimizes a continuous 5D neural radiance field representation (volume density and view-dependent color at any continuous location) of a scene from a set of input images.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934

NeRF Overview

- Implicit Neural Representation: One network for one scene.

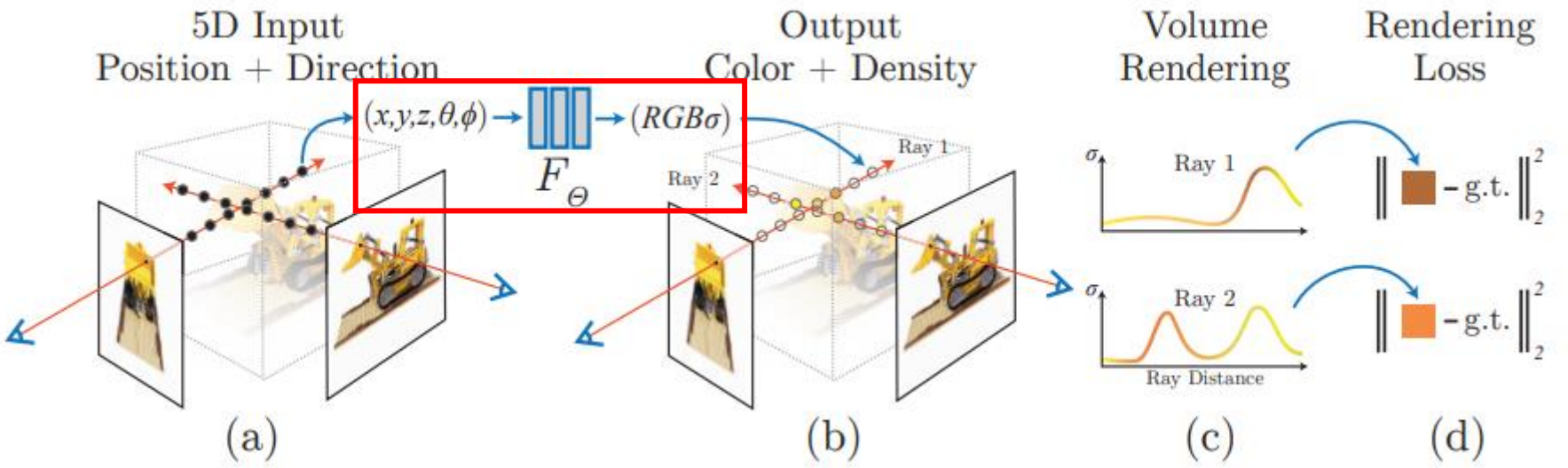


Fig. 2: An overview of our neural radiance field scene representation and differentiable rendering procedure.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934

NeRF Volume Rendering

- Density and shown color

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma\big(r(t)\big)c(r(t), d)dt\,, \qquad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma\big(r(s)\big)ds\right)$$

To Discrete $\quad \hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i\,, \qquad \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$

Traditional alpha compositing α

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934

NeRF Volume Rendering

- Hierarchical volume sampling

$$\hat{C}(r) = \sum_{i=1}^{N} \underbrace{T_i(1 - \exp(-\sigma_i \delta_i))}_{w_i} c_i \,, \qquad \text{where } T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

- Optimize two networks: coarse and fine

$$\hat{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$$

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934

NeRF Optimizing

- Positional Encoding

Significantly improve performance in detail

$$\gamma(p) = \sin(2^0\pi p), \cos(2^0\pi p), \cdots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)$$

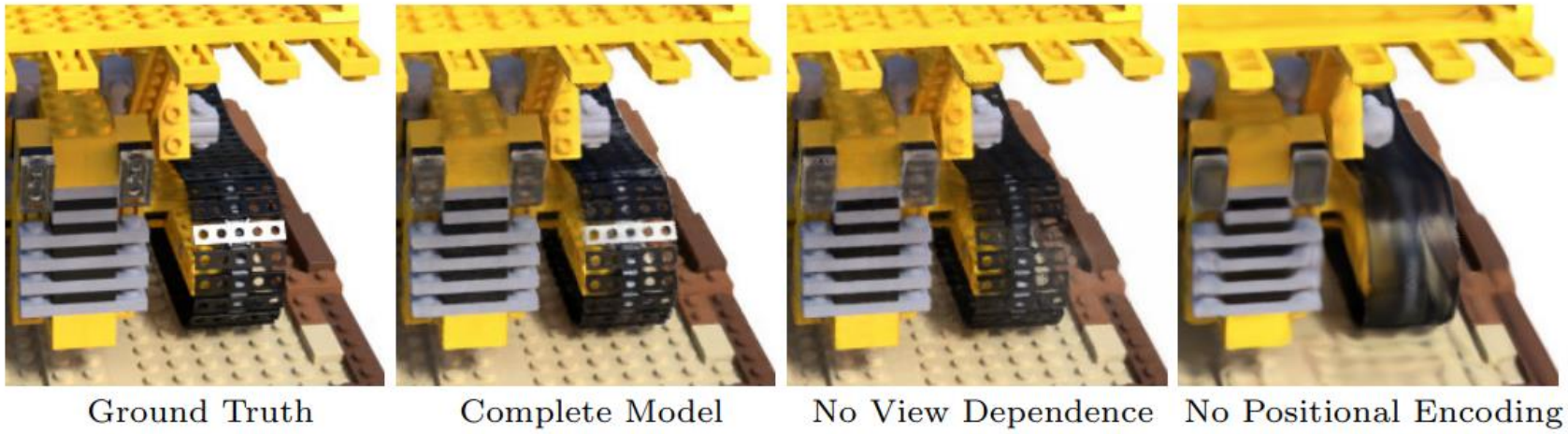


Fig. 4: Here we visualize how our full model benefits from representing view dependent emitted radiance and from passing our input coordinates through a high-frequency positional encoding.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934
Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall et al., Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, https://bmild.github.io/fourfeat/index.html

GRAF Objective

- 3D-Aware Generative Model using Radiance Fields

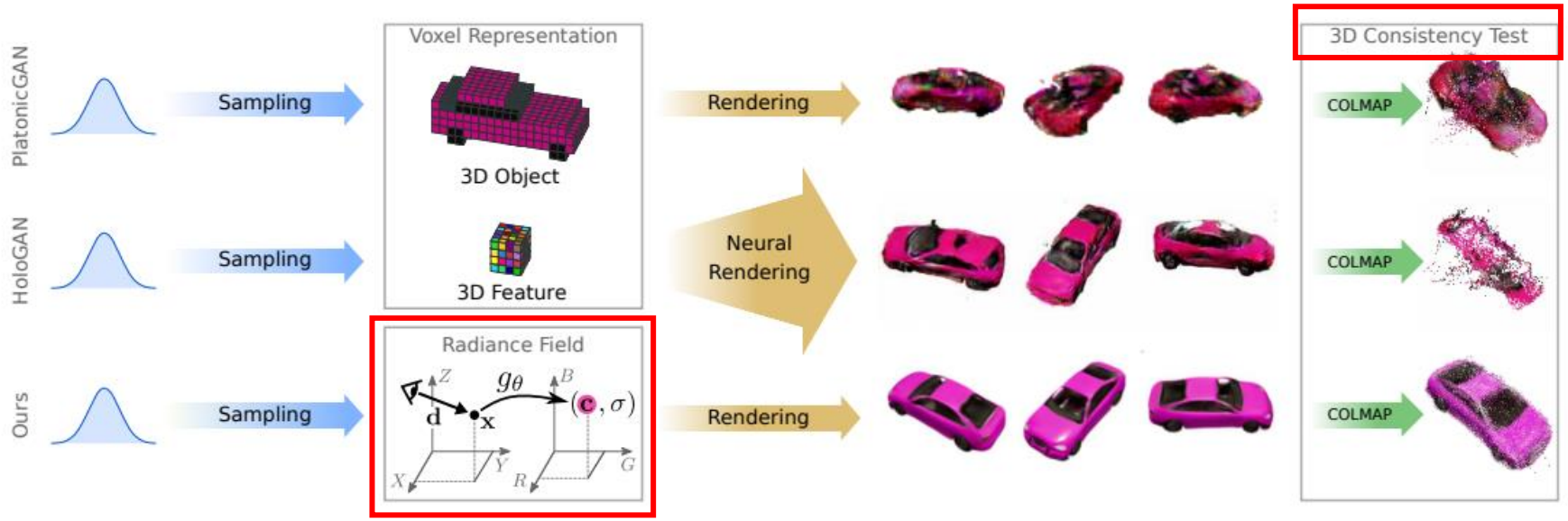


Figure 1: Motivation.

COLMAP

- COLMAP is a general-purpose Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline.

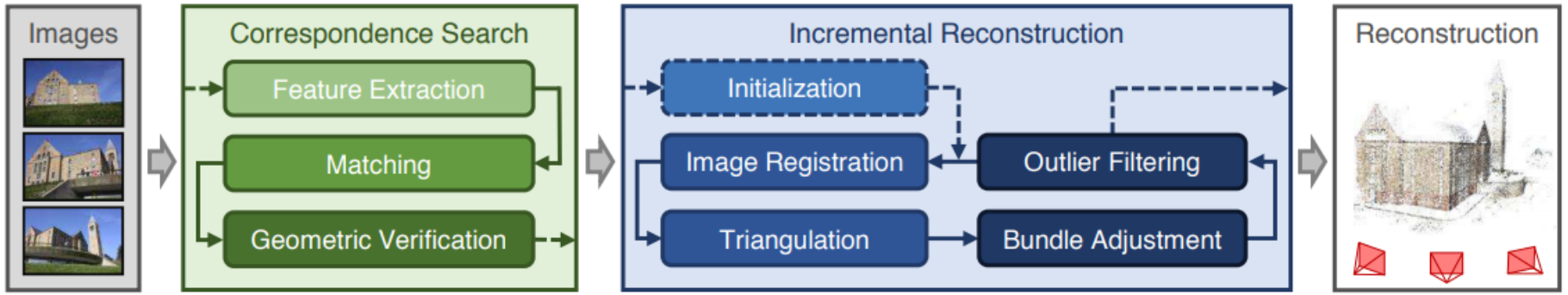


Figure 2. Incremental Structure-from-Motion pipeline.

Johannes L. Schönberger, COLMAP, https://colmap.github.io/
Johannes L. Schönberger and Jan-Michael Frahm, Structure-from-Motion Revisited, https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf

GRAF Overview

- Generative Radiance Fields

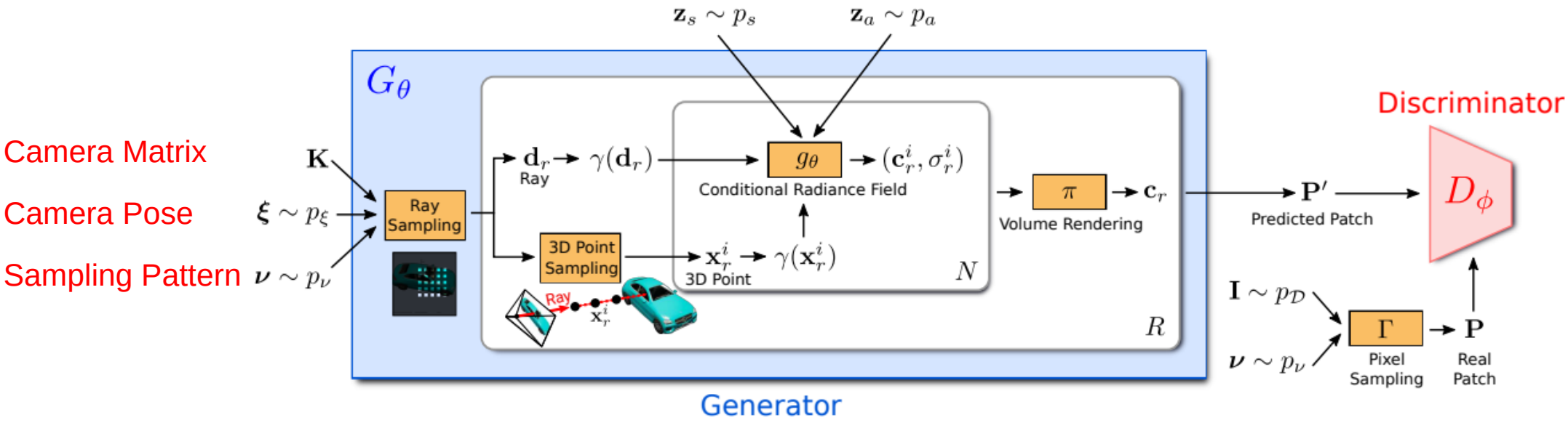


Figure 2: Generative Radiance Fields

GRAF Ray Sampling

- Camera Matrix: $K = \#\,\text{of patches}$
- Camera Pose: $\xi = [R|t] \sim p_\xi$
- Sampling Pattern: $(u, s) = (\text{center}, \text{scale})$

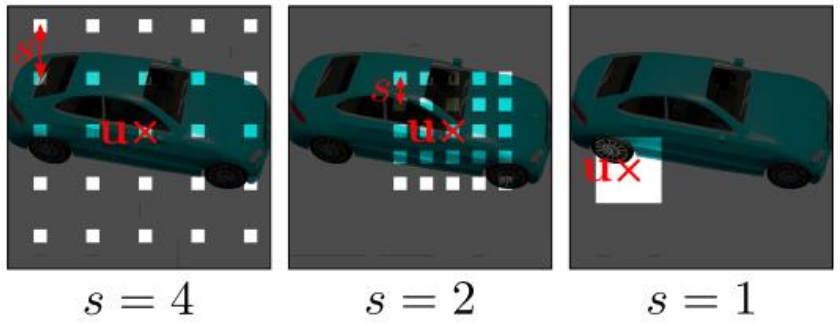


Figure 3: Ray Sampling.

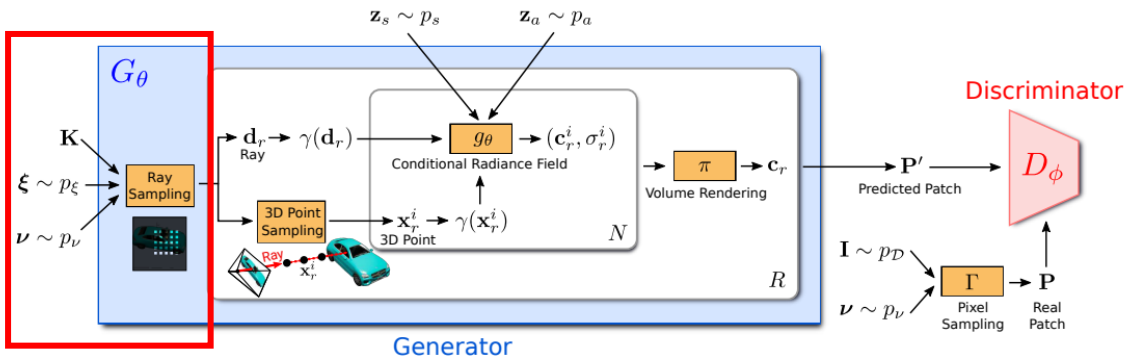


Figure 2: Generative Radiance Fields.

GRAF Conditional Radiance Field

- g_θ in the overview

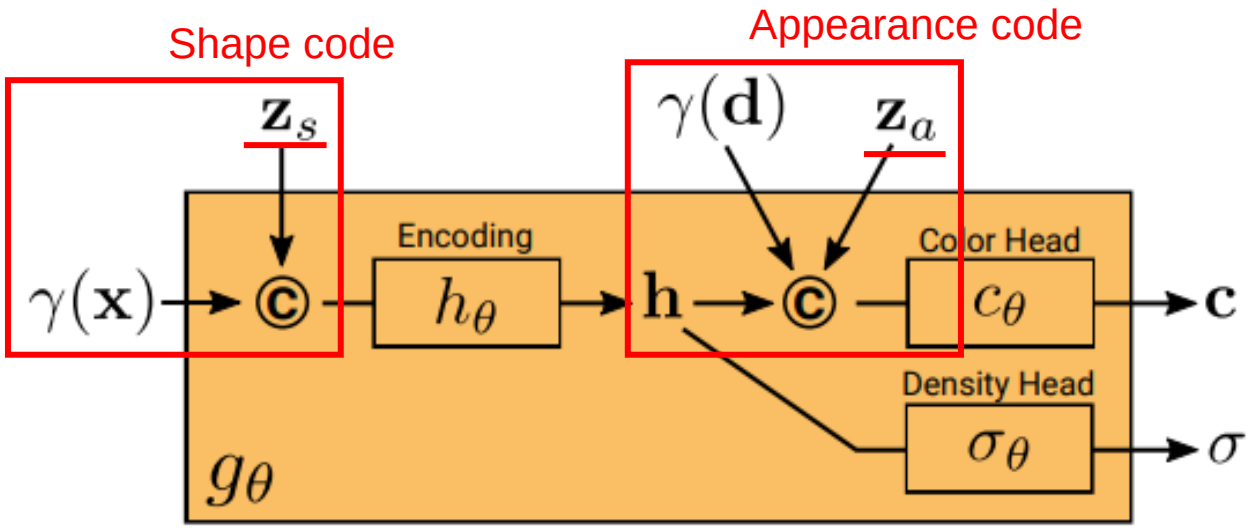


Figure 4: Conditional Radiance Field.

GRAF Volume Rendering

- Same as NeRF
- Hierarchical volume sampling

$$\hat{C}(r) = \sum_{i=1}^{N} \underbrace{T_i(1 - \exp(-\sigma_i \delta_i))}_{w_i} c_i \,, \qquad \text{where } T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

- Optimize two networks: coarse and fine

$$\hat{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$$

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934

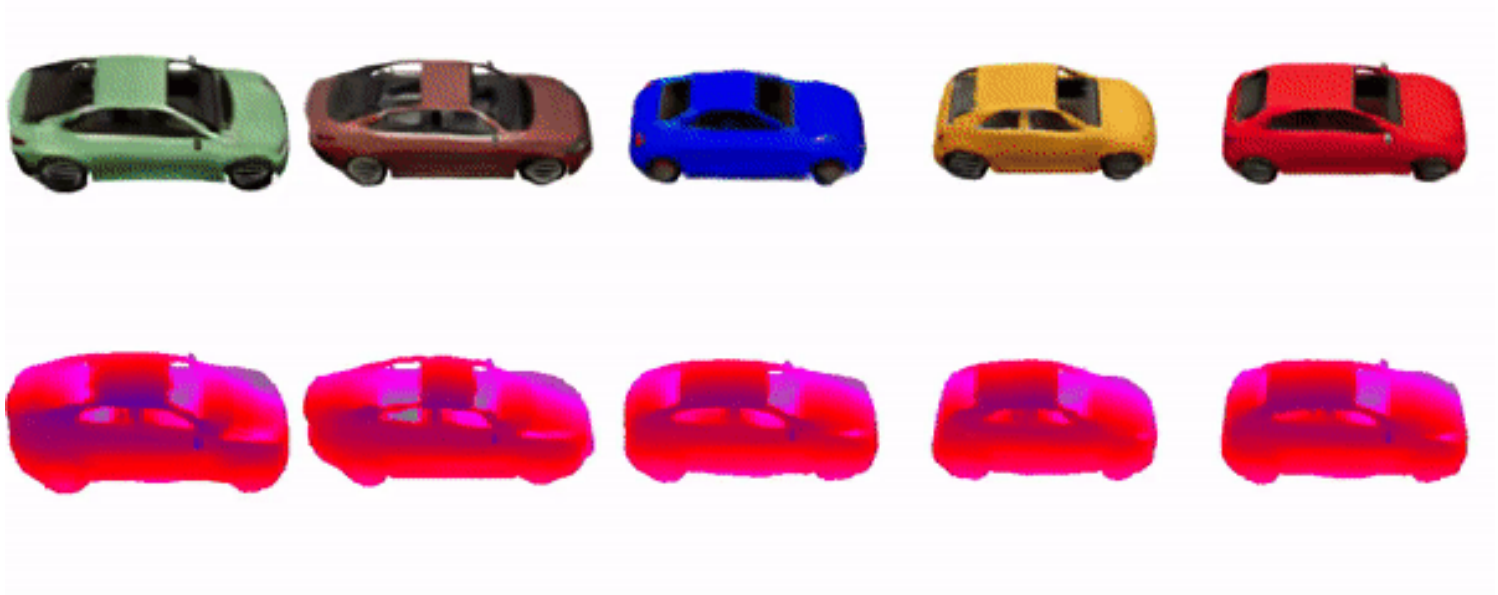
GRAF Results

- Change shape and appearance codes

Katja Schwarz, Yiyi Liao et al., GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis, https://arxiv.org/abs/2007.02442
Katja Schwarz, Generative Radiance Fields for 3D-Aware Image Synthesis, https://autonomousvision.github.io/graf/

GRAF Results

- Generative radiance fields can produce a depth map to every RGB image



Katja Schwarz, Yiyi Liao et al., GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis, https://arxiv.org/abs/2007.02442
Katja Schwarz, Generative Radiance Fields for 3D-Aware Image Synthesis, https://autonomousvision.github.io/graf/

GIRAFFE Overview

- **Compositional** generative neural **feature** fields

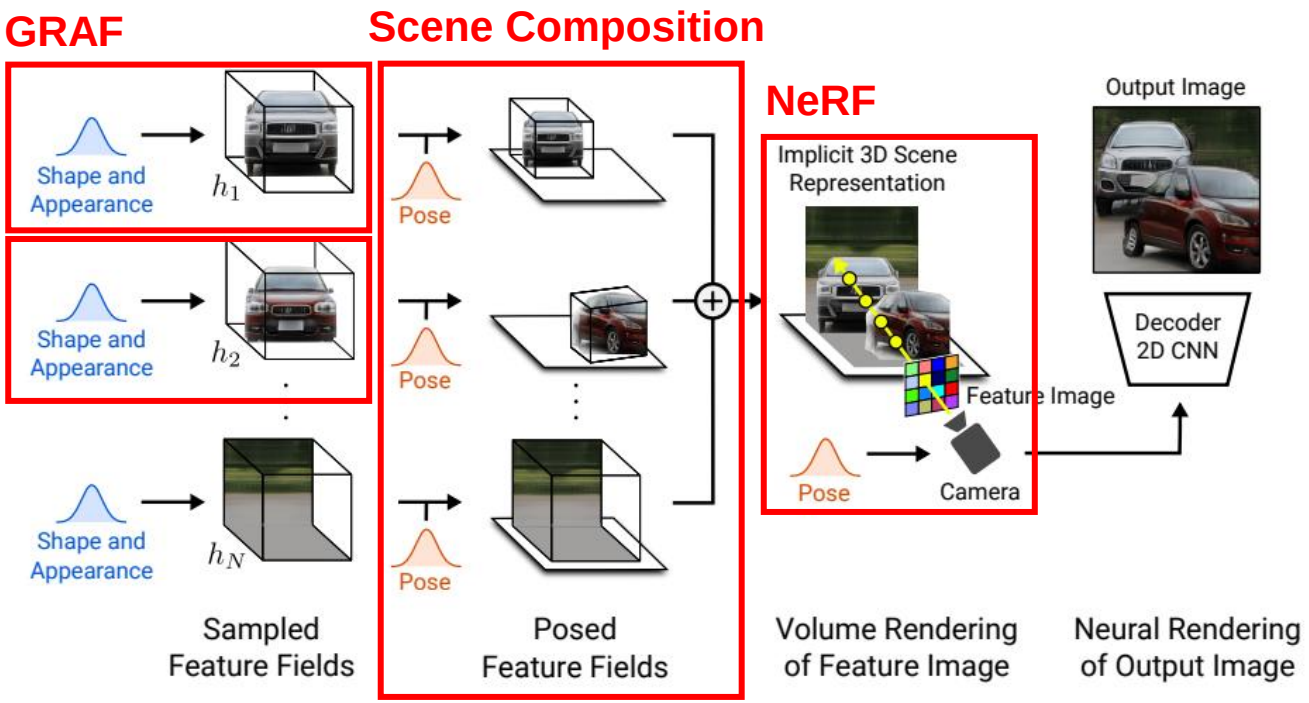


Figure 1: Overview. We represent scenes as compositional generative neural feature fields.

GIRAFFE Model

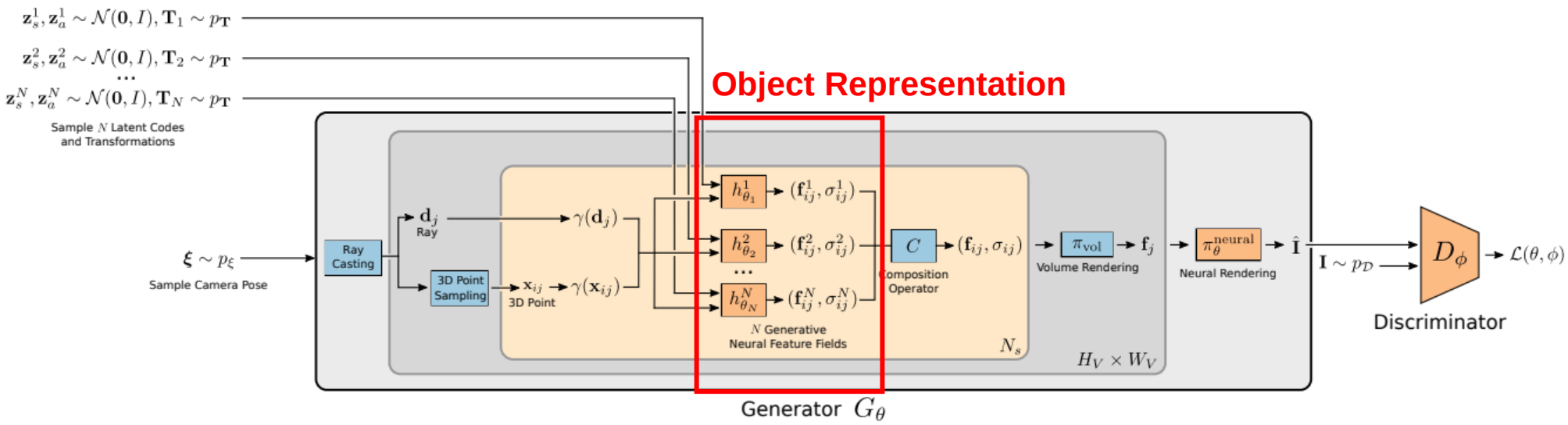


Figure 3: GIRAFFE.

Object Representation

- T is sampled from a dataset-dependent distribution

$$T = \{s, t, R\}$$

$$k(x) = R \cdot \begin{bmatrix} s_1 & & \\ & s_2 & \\ & & s_3 \end{bmatrix} \cdot x + t$$

$$(\boldsymbol{\sigma}, \boldsymbol{f}) = \boldsymbol{h_\theta}\left(\boldsymbol{\gamma}\left(\boldsymbol{k^{-1}}(\boldsymbol{x})\right), \boldsymbol{\gamma}\left(\boldsymbol{k^{-1}}(\boldsymbol{d})\right), \boldsymbol{z_s}, \boldsymbol{z_a}\right)$$

GIRAFFE Model

$$C(x,d) = \left(\sigma, \frac{1}{\sigma}\sum_{i=1}^{N}\sigma_i f_i\right), \qquad where\ \sigma = \sum_{i=1}^{N}\sigma_i$$

$$f = \sum_{j=1}^{N}\tau_j \alpha_j f_j\,, \qquad \tau_j = \prod_{k=1}^{j-1}(1-\alpha_k)\,, \qquad \alpha_j = 1 - e^{-\sigma_j \delta_j}$$

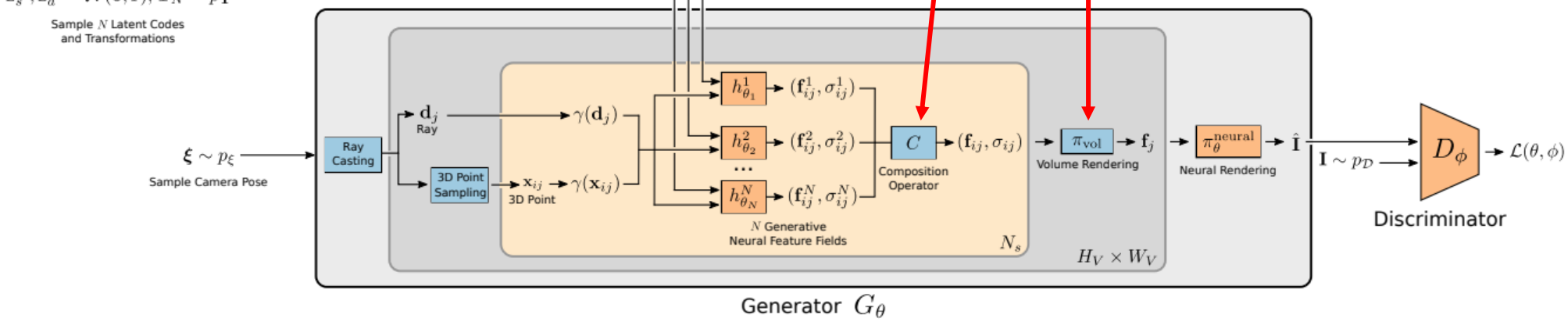


Figure 3: GIRAFFE.

GIRAFFE Model

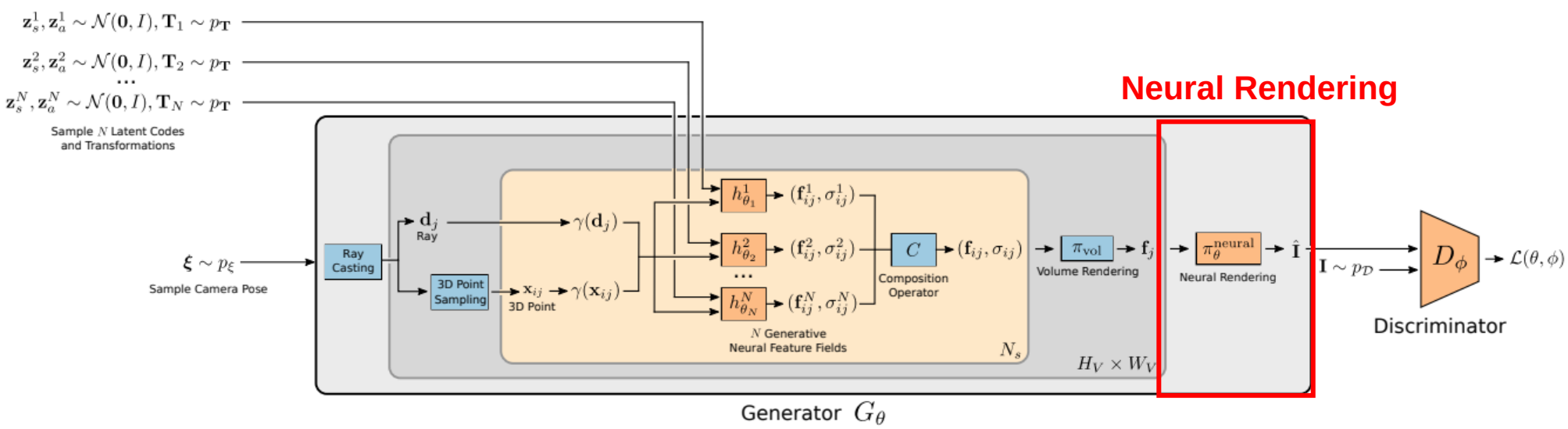


Figure 3: GIRAFFE.

Neural Rendering Operation

- Neural feature fields to RGB space
- $\mathbb{R}^{H_V, W_V, M_f} \rightarrow \mathbb{R}^{H \times W \times 3}$

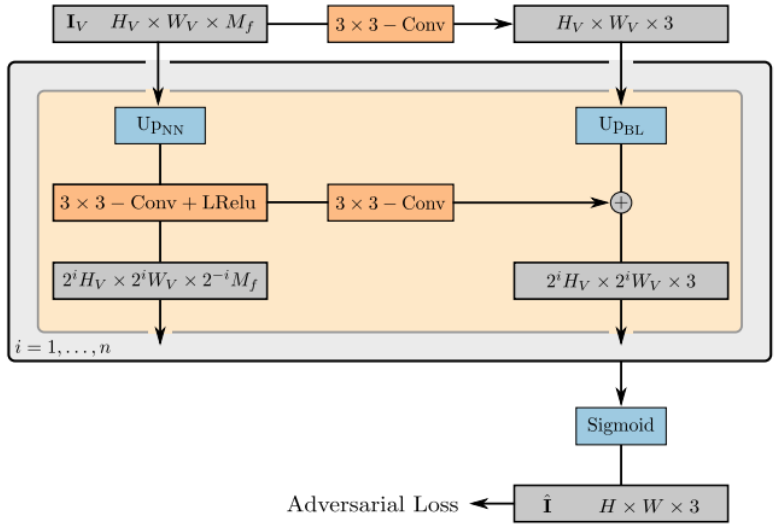


Figure 4: Neural Rendering Operator

Training

- Non-saturating GAN objective and R_1 gradient penalty

$$v(\theta,\phi) = \mathbb{E}_{z_s^i, z_a^i \sim \mathcal{N}, \xi \sim p_\xi, T_i \sim p_T}\left[f\left(D_\phi\left(G_\theta\left(\{z_s^i, z_a^i, T_i\}, \xi\right)\right)\right)\right] + \mathbb{E}_{I \sim p_D}\left[f\left(-D_\phi(I)\right) - \lambda\left\|\nabla D_\phi(I)\right\|\right]$$

where $f(t) = -\log(1+\exp(-t))$, $\lambda = 10$, and p_D indicates the data distribution

autonomousvision, giraffe, https://github.com/autonomousvision/giraffe
Lmescheder, GAN stability, https://github.com/LMescheder/GAN_stability

Experiments

Background

Only object

Color-coded
object alpha maps

Synthesized image

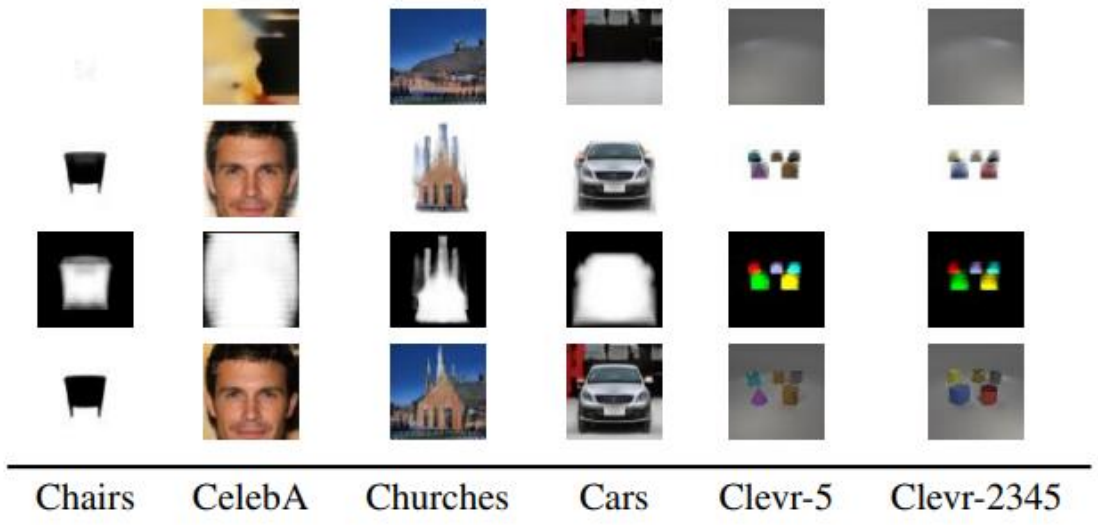


Figure 5: **Scene Disentanglement.** From top to bottom, we show only backgrounds, only objects, color-coded object alpha maps, and the final synthesized images at 64^2 pixel resolution. Disentanglement emerges without supervision, and the model learns to generate plausible backgrounds although the training data only contains images with objects.

Experiments

	Cats	CelebA	Cars	Chairs	Churches
2D GAN [58]	18	15	**16**	59	19
Plat. GAN [32]	318	321	299	199	242
BlockGAN [64]	47	69	41	41	28
HoloGAN [63]	27	25	17	59	31
GRAF [77]	26	25	39	34	38
Ours	**8**	**6**	**16**	**20**	**17**

Table 1: **Quantitative Comparison.** We report the FID score (↓) at 64^2 pixels for baselines and our method.

	CelebA-HQ	FFHQ	Cars	Churches	Clevr-2
HoloGAN [63]	61	192	34	58	241
w/o 3D Conv	33	70	49	66	273
GRAF [77]	49	59	95	87	106
Ours	**21**	**32**	**26**	**30**	**31**

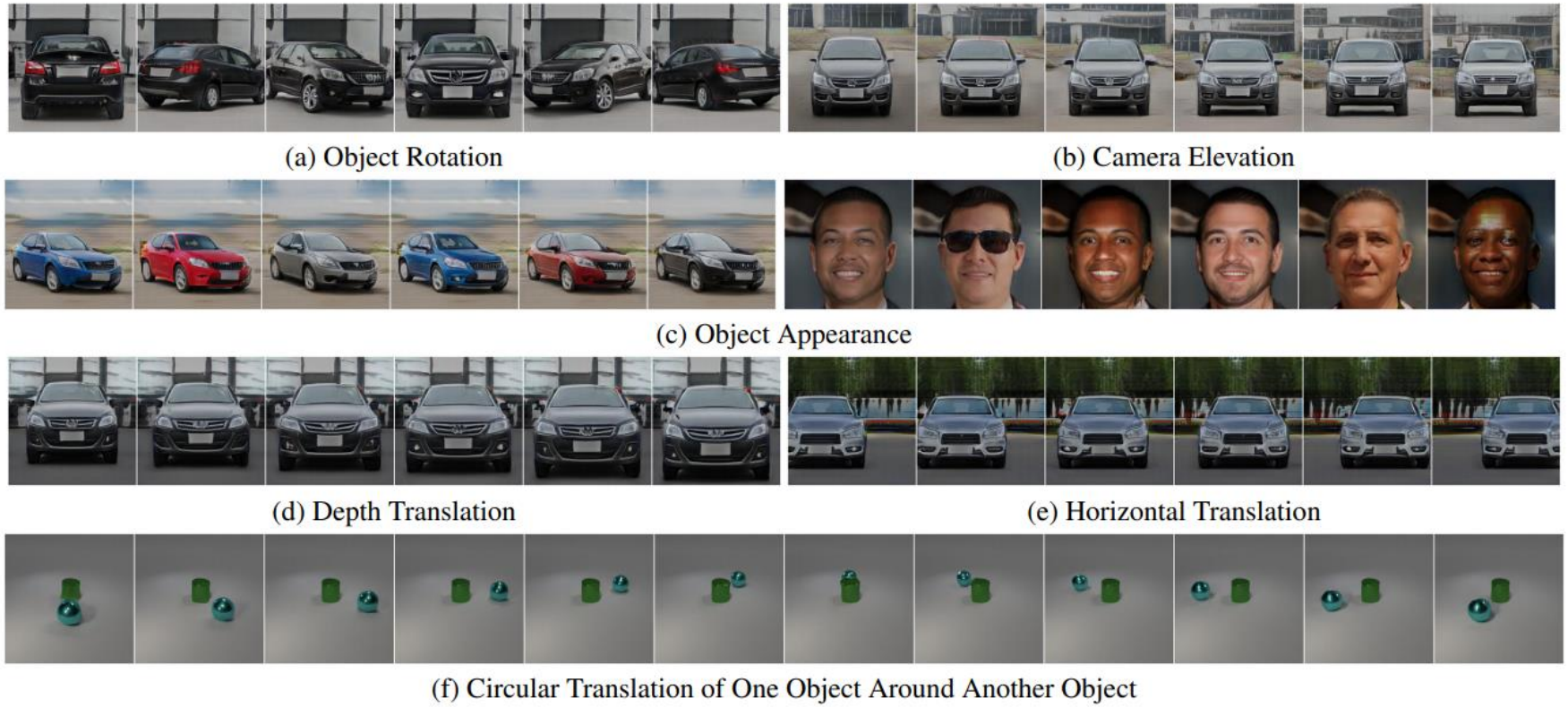
Table 2: **Quantitative Comparison.** We report the FID score (↓) at 256^2 pixels for the strongest 3D-aware baselines and our method.

2D GAN	Plat. GAN	BlockGAN	HoloGAN	GRAF	Ours
1.69	381.56	4.44	7.80	0.68	0.41

Table 3: **Network Parameter Comparison.** We report the number of generator network parameters in million.

Wandb, FID Score, https://wandb.ai/wandb_fc/korean/reports/-Frechet-Inception-distance-FID-GANs---Vmlldzo0MzQ3Mzc

Experiments



(a) Object Rotation

(b) Camera Elevation

(c) Object Appearance

(d) Depth Translation

(e) Horizontal Translation

(f) Circular Translation of One Object Around Another Object

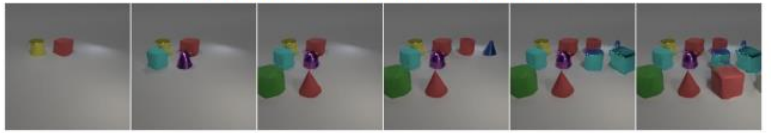
Figure 7: **Controllable Scene Generation at 256^2 Pixel Resolution.** Controlling the generated scenes during image synthesis: Here we rotate or translate objects, change their appearances, and perform complex operations like circular translations.

Experiments

Figure 10: **Neural Renderer.** We change the background while keeping the foreground object fixed for our method at 256^2 pixel resolution. Note how the neural renderer realistically adapts the objects' appearances to the background.

(a) Increase Depth Translation

(b) Increase Horizontal Translation



(c) Add Additional Objects (Trained on Two-Object Scenes)

(d) Add Additional Objects (Trained on Single-Object Scenes)

Figure 8: **Generalization Beyond Training Data.** As individual objects are correctly disentangled, our model allows for generating out of distribution samples at test time. For example, we can increase the translation ranges or add more objects than there were present in the training data.

Experiments

(a) 0° Rotation for Axis-Aligned Positional Encoding [61]

(b) 0° Rotation for Random Fourier Features [82]

Figure 11: **Canonical Pose.** In contrast to random Fourier features [82], axis-aligned positional encoding (1) encourages the model to learn objects in a canonical pose.

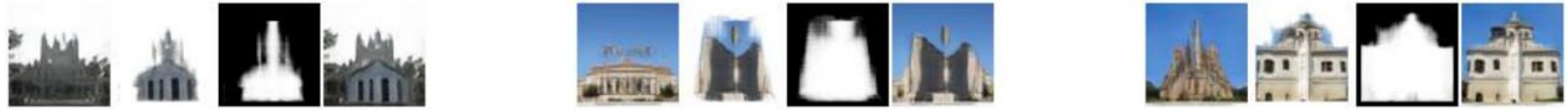
$$(\sigma, f) = h_\theta\big(\gamma\big(k^{-1}(x)\big), \gamma\big(k^{-1}(d)\big), z_s, z_a\big)$$

Positional Encoding: We use axis-aligned positional encoding for the input point and viewing direction (Eq. 1). Surprisingly, this encourages the model to learn canoncial representations as it introduces a bias to align the object axes with highest symmetry with the canonical axes which allows the model to exploit object symmetry (Fig. 11).

Limitations

Figure 12: **Dataset Bias.** Eye and hair rotation are examples for dataset biases: They primarily face the camera, and our model tends to entangle them with the object rotation.

Limitations



(a) Disentanglement Failure on *Churches*.

(b) Disentanglement Failure on *CompCars*.

Figure 12: **Disentanglement Failures.** For *Churches*, the background sometimes contains a church, and for *CompCars*, the object sometimes contains background parts or vice versa. We attribute these to mismatches between the assumed uniform distributions over object and camera poses and their real distributions, and identify learning them instead as interesting future work.

Michael Niemeyer et al., Supplementary Material for GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields, http://www.cvlibs.net/publications/Niemeyer2021CVPR_supplementary.pdf

References

- Michael Niemeyer et al., GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields, http://www.cvlibs.net/publications/Niemeyer2021CVPR.pdf
- Michael Niemeyer et al., Supplementary Material for GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields, http://www.cvlibs.net/publications/Niemeyer2021CVPR_supplementary.pdf
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, https://arxiv.org/abs/2003.08934
- Katja Schwarz, Yiyi Liao et al., GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis, https://arxiv.org/abs/2007.02442
- Johannes L. Schönberger and Jan-Michael Frahm, Structure-from-Motion Revisited, https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf
- Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall et al., Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, https://arxiv.org/abs/2006.10739

References

- Johannes L. Schönberger, COLMAP, https://colmap.github.io/
- Lmescheder, GAN stability, https://github.com/LMescheder/GAN_stability
- Michael Niemeyer, Generative Neural Scene Representations | 3D Representation Seminar, https://www.youtube.com/watch?v=scnXyCSMJF4