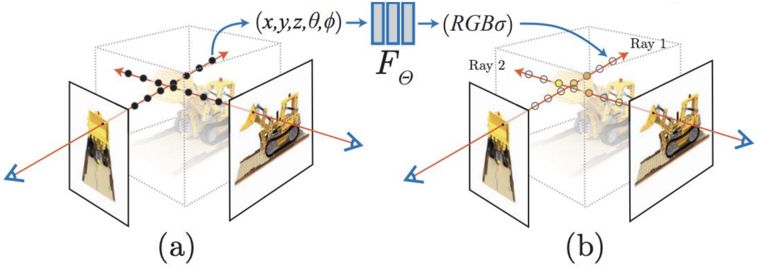
DreamFusion: Text-to-3D using 2D Diffusion

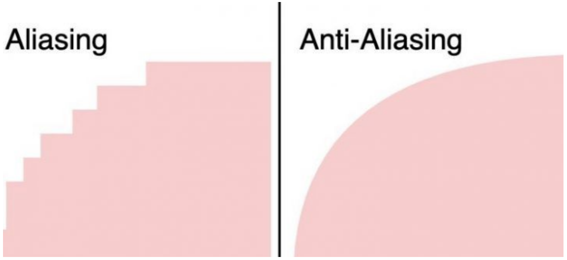
Plan

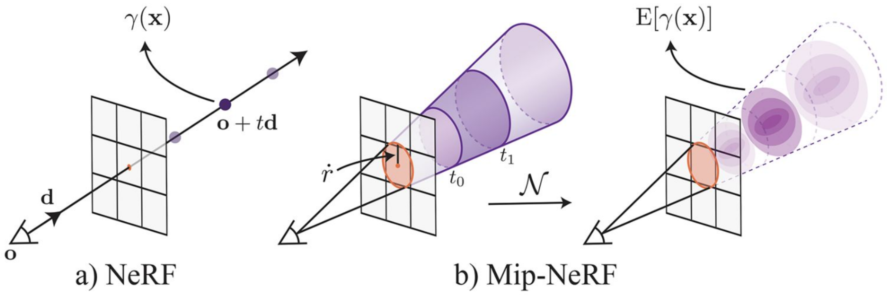
- Refresher
- **Goal**, problems
- Other approaches
- **Function Loss**
- **DreamFusion Algorithm**
- Comparison

Refresher: NeRF

- images to 3D object
- mip-NeRF 360
  - reduce aliasing
  - penalty for filling in empty spaces

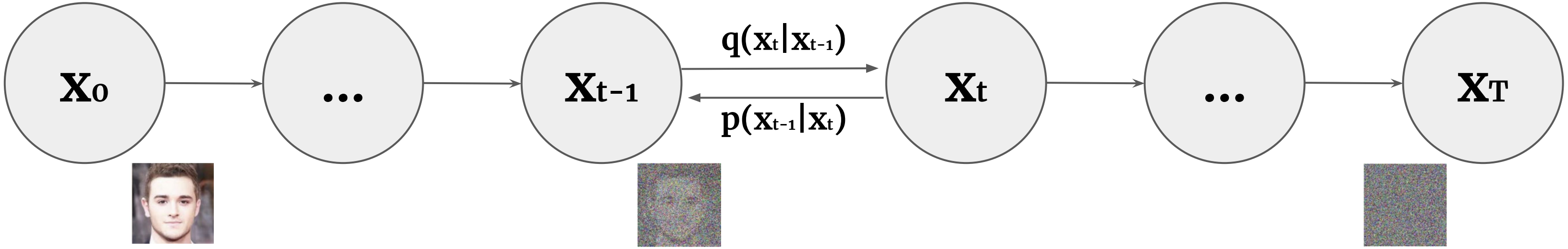






Refresher: Diffusion Model

- Forward process (q): image-to-noise
- Reverse process (p): noise-to-image



Goal

- From *text* to *3D scene*

"fries and hamburger"

DreamFusion

Goal

- From *text* to *3D scene*

"fries and hamburger"

DreamFusion

Problems

- Lack of labeled 3D *data* (text to 3D)
- Absence *architectures* for denoising 3D data

Other approaches

- PointFlow, Text2Shape, Point-Voxel Diffusion – use only 3D data
- GANs (PlatonicGAN, HoloGAN, StyleSDF) – not universal
- Data Fields – bad 3D objects

Function Loss

$$\mathcal{L}_{\text{Diff}}(\phi, x) = \mathbb{E}_{t,\epsilon\sim\mathcal{N}(0,I)}[w(t)||\hat{\epsilon}_\phi(x,t) - \epsilon||_2^2]$$

Function Loss

- $\mathcal{L}_{\text{Diff}}(\phi, x) = \mathbb{E}_{t,\epsilon\sim\mathcal{N}(0,I)}[w(t)||\hat{\epsilon}_\phi(x,t) - \epsilon||_2^2]$

$$x = g(\theta) =$$ 

$$(\theta)$$

Function Loss

- $\mathcal{L}_{\text{Diff}}(\phi, x) = \mathbb{E}_{t,\epsilon\sim\mathcal{N}(0,I)}[w(t)||\hat{\epsilon}_\phi(x,t) - \epsilon||_2^2]$
- $x = g(\theta) =$  (θ), x – image, θ – NeRF parameters
- $\theta^* = \text{argmin}_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta))$

Function Loss

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}[w(t) \underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}} \quad]$$

Function Loss

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}[w(t) \underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}} \underbrace{\frac{\partial \hat{\epsilon}(x)}{\partial x}}_{\text{U-Net Jacobian}} \qquad]$$

Function Loss

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}[w(t) \underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}} \underbrace{\frac{\partial \hat{\epsilon}(x)}{\partial x}}_{\text{U-Net Jacobian}} \underbrace{\frac{\partial x}{\partial \theta}}_{Generator Jacobian}]$$

Function Loss

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}[w(t) \underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}} \cancel{\underbrace{\frac{\partial \hat{\epsilon}(x)}{\partial x}}_{\text{U-Net Jacobian}}} \underbrace{\frac{\partial x}{\partial \theta}}_{Generator Jacobian}]$$

Function Loss (Score Distillation Sampling)

$$\nabla_\theta \mathcal{L}_{\mathrm{SDS}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}\Big[w(t) \underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}} \quad \underbrace{\frac{\partial x}{\partial \theta}}_{Generator Jacobian} \quad \Big]$$

DreamFusion Algorithm

1. Choose text prompt: “a DSLR photo of peacock on a surfboard”

DreamFusion Algorithm

1. Choose text prompt: “a DSLR photo of peacock on a surfboard”

2. NeRF random initialization

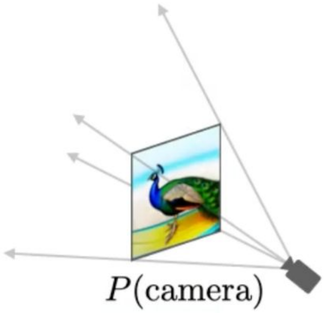
DreamFusion Algorithm

1. Choose text prompt: “a DSLR photo of peacock on a surfboard”

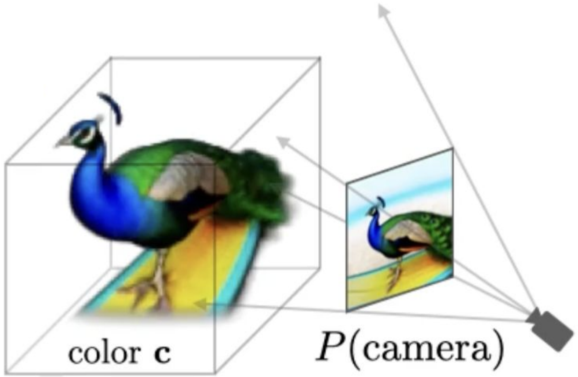
2. NeRF random initialization

3. Optimization

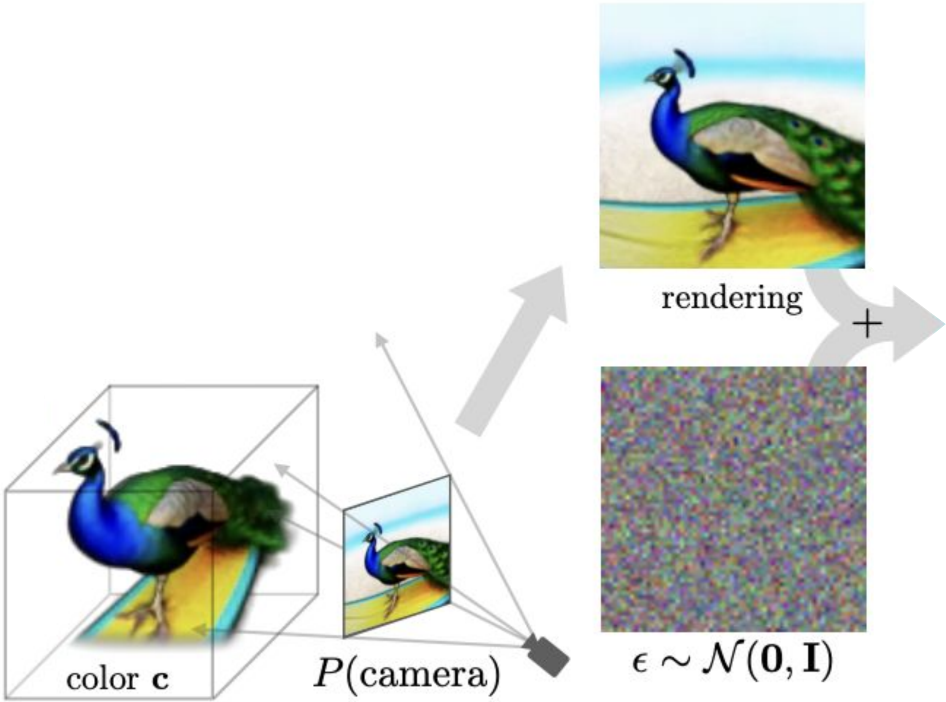
DreamFusion Algorithm



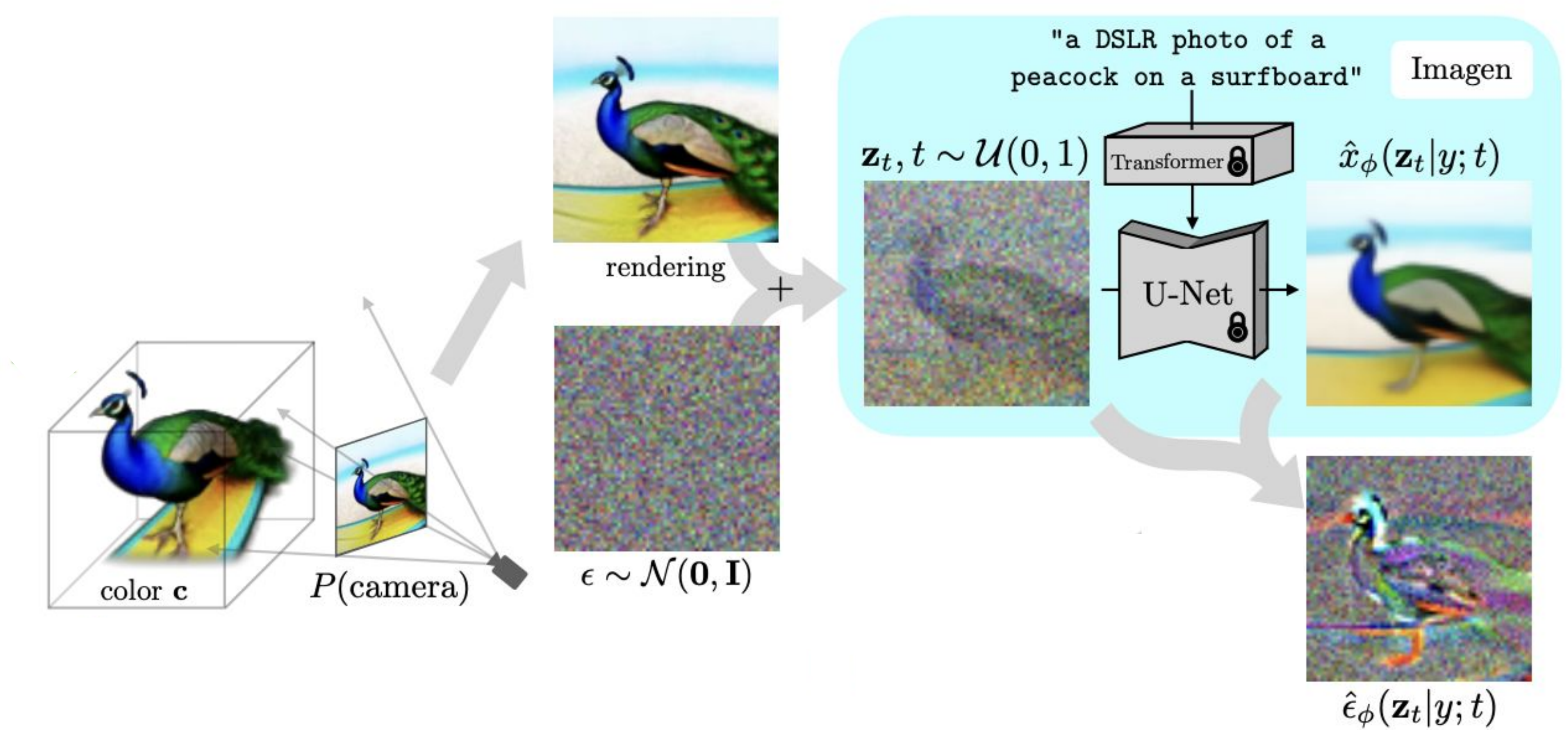
DreamFusion Algorithm



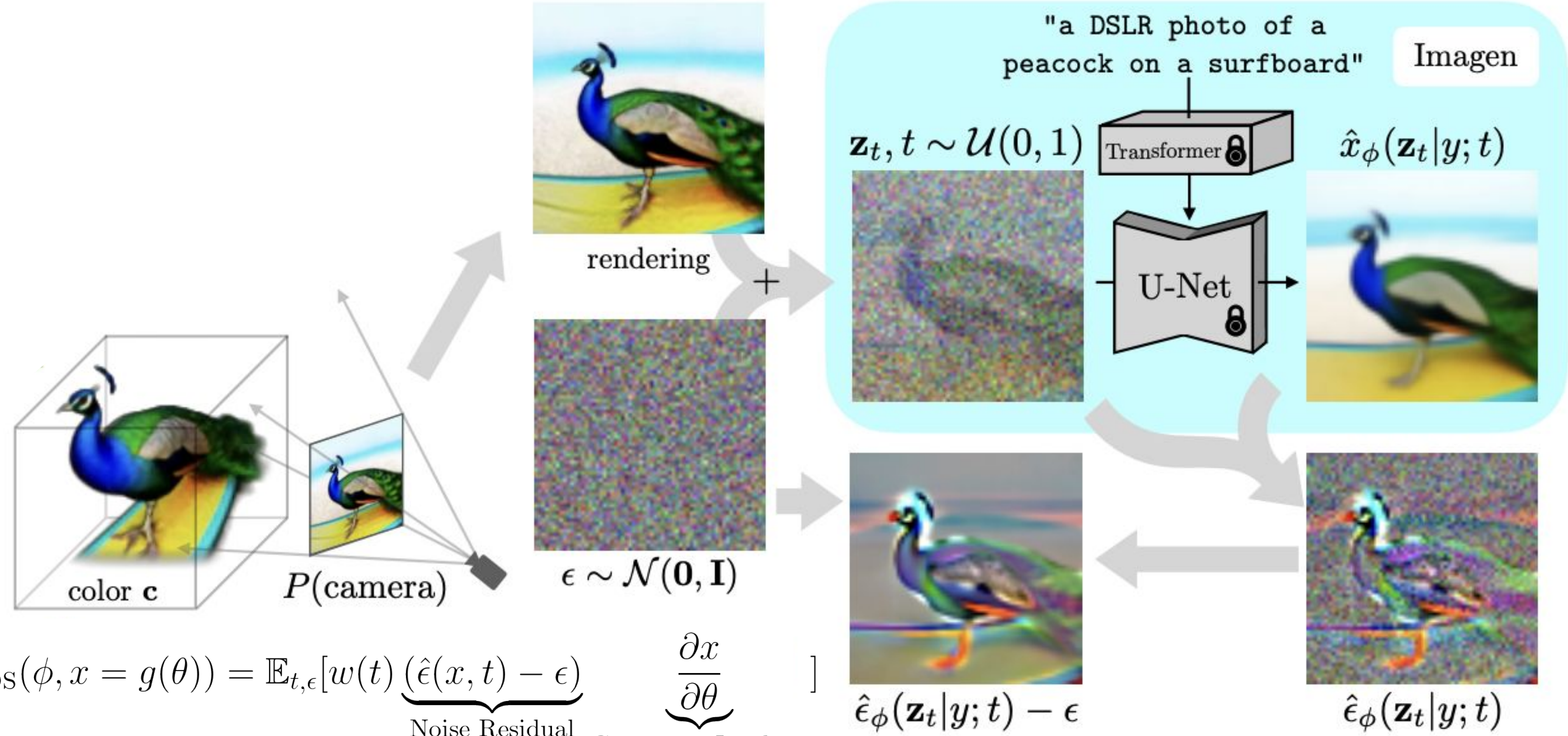
DreamFusion Algorithm



DreamFusion Algorithm

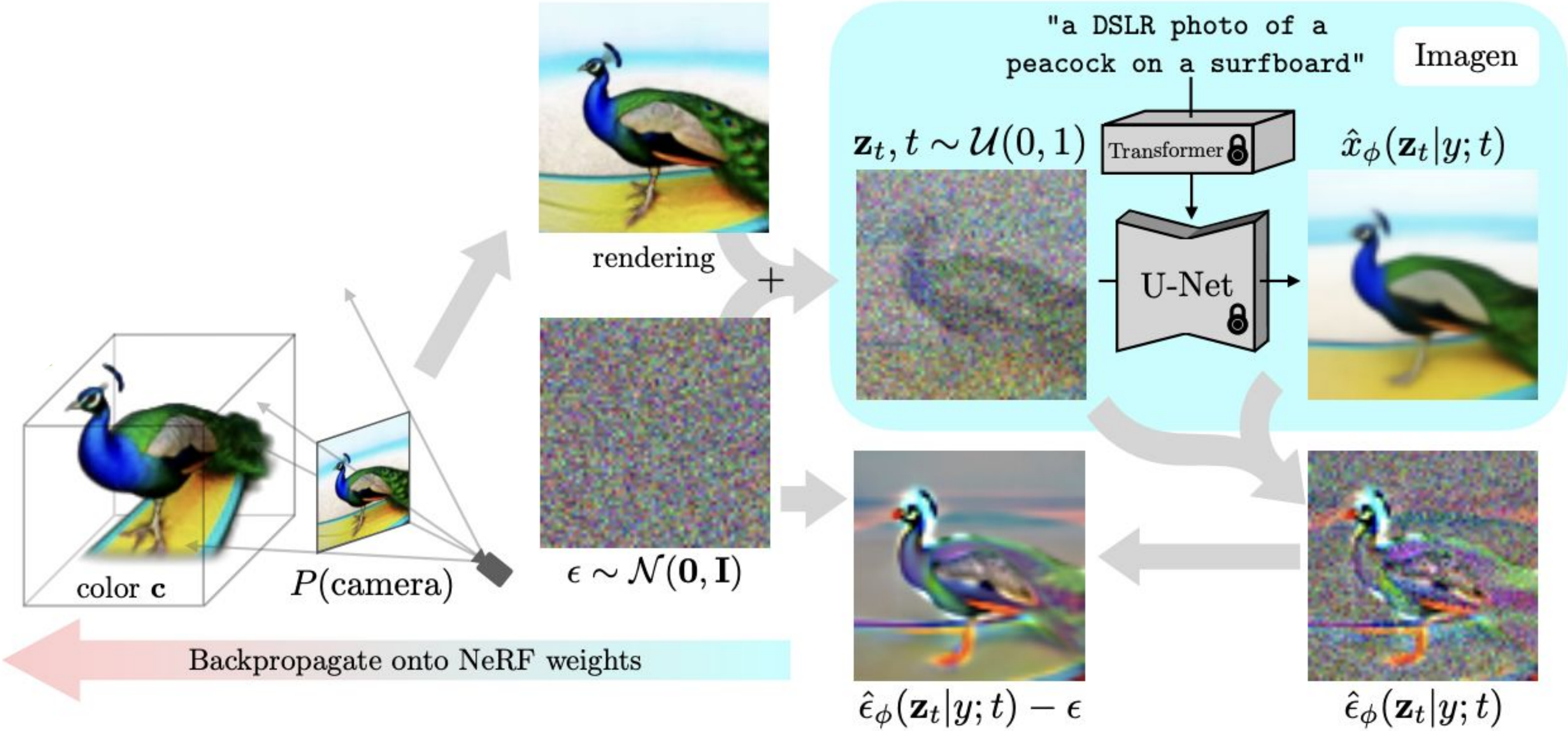


DreamFusion Algorithm

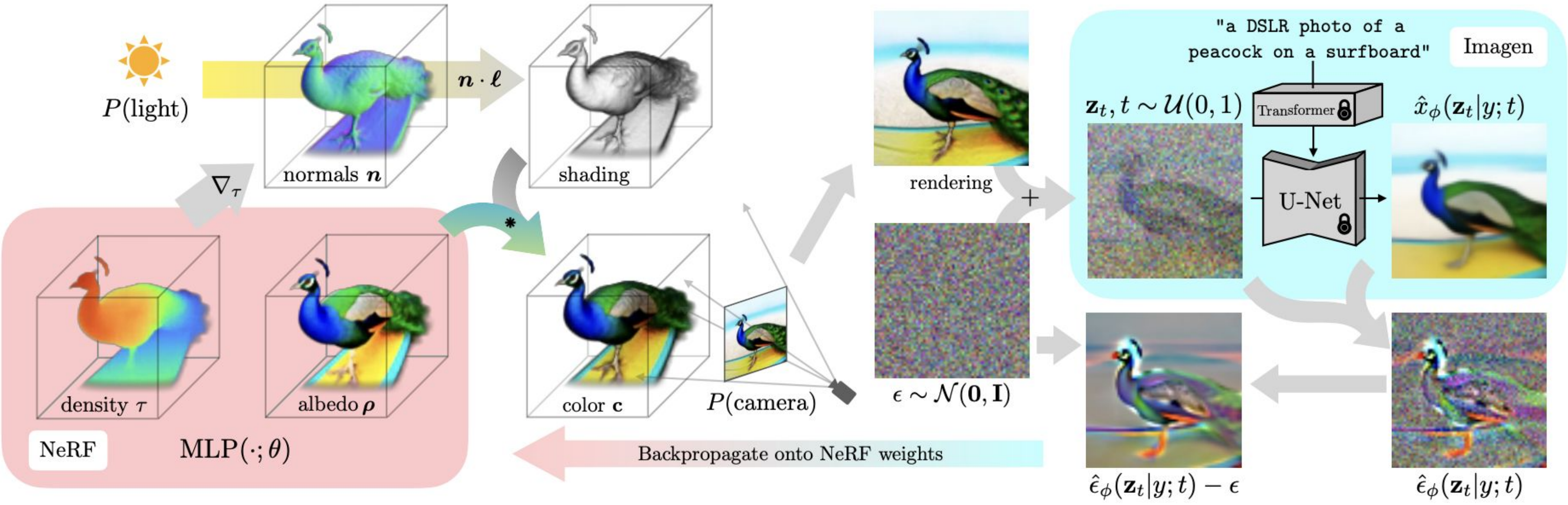


$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x = g(\theta)) = \mathbb{E}_{t,\epsilon}\Big[w(t)\,\underbrace{(\hat{\epsilon}(x,t) - \epsilon)}_{\text{Noise Residual}}\;\underbrace{\frac{\partial x}{\partial \theta}}_{Generator Jacobian}\Big]$$

DreamFusion Algorithm



DreamFusion Algorithm

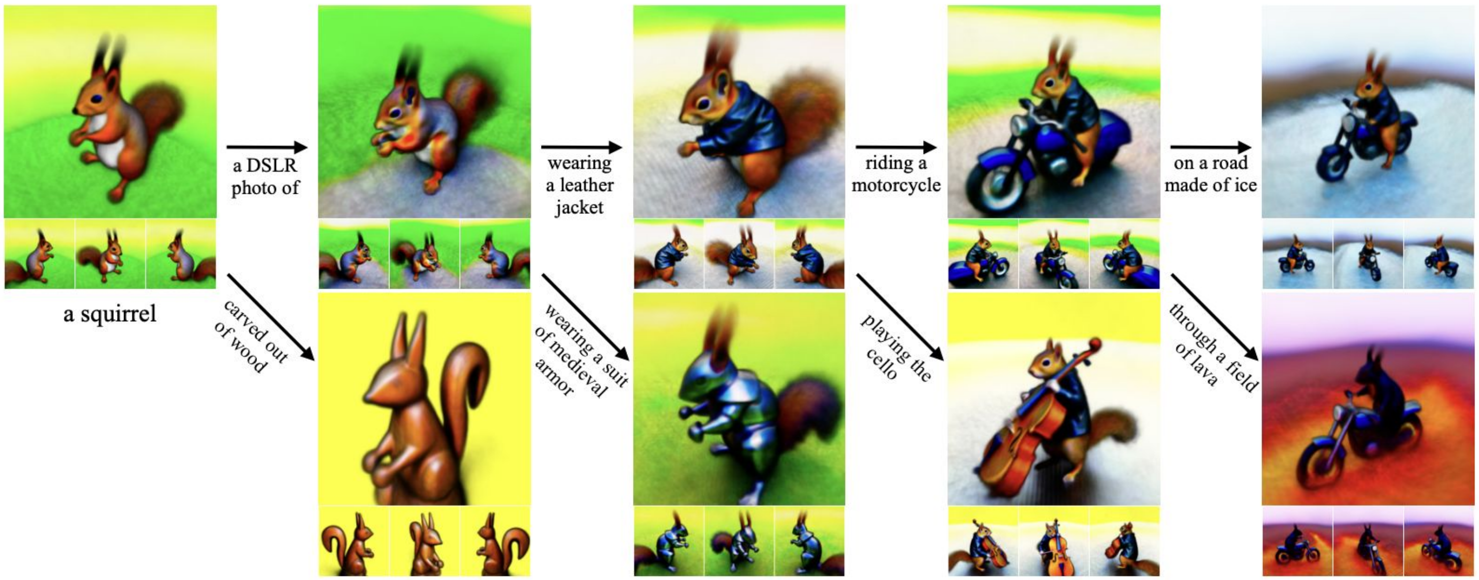


DreamFusion Algorithm

1. Choose text prompt: “a DSLR photo of peacock on a surfboard”
2. NeRF random initialization
3. Optimization:
    a. Random camera position
    b. Render Image (make photo)
    c. Add Noise
    d. Predict Noise (w. Diffusion Model)
    e. Backpropagate

DreamFusion Algorithm

DreamFusion Algorithm



Random camera and light sampling

- Position (x, y, z)
- Elevation angle: [-10°, 90°]
- Azimut angle: [0°, 360°]
- Focal length: λw, $\lambda \in \mathcal{U}(0.7, 1.35)$
- Add view to prompt (“overhead view”, “front view”, etc.)

Rendering

- Image size 64x64
- Sample random type:

Illuminated color render

textureless render

rendering w/o shading

Rendering

Diffusion Model

- Imagen (diffusion model)
- Images 64x64
- Text embeddings from LLM T5-XXL

"*a* small cactus wearing a straw hat and neon sunglasses in the Sahara desert."

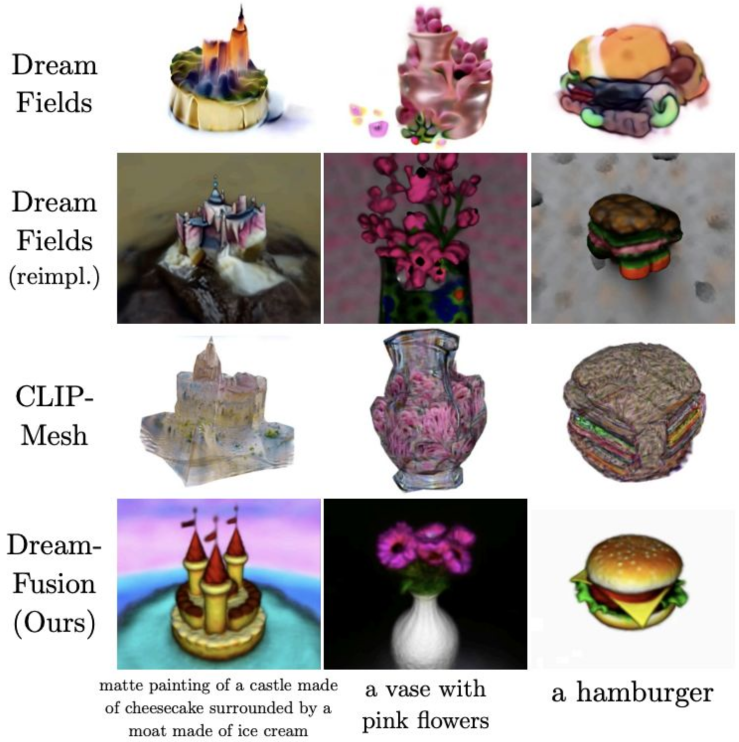
Imagen

Comparison

Method	R-Precision ↑					
	CLIP B/32		CLIP B/16		CLIP L/14	
	Color	Geo	Color	Geo	Color	Geo
GT Images	77.1	–	79.1	–	–	–
Dream Fields	68.3	–	74.2	–	–	–
(reimpl.)	**78.6**	1.3	(99.9)	(0.8)	**82.9**	1.4
CLIP-Mesh	67.8	–	75.8	–	74.5†	–
DreamFusion	75.1	**42.5**	**77.5**	46.6	79.7	**58.5**

Comparison

Method	R-Precision ↑					
	CLIP B/32		CLIP B/16		CLIP L/14	
	Color	Geo	Color	Geo	Color	Geo
GT Images	77.1	–	79.1	–	–	–
Dream Fields	68.3	–	74.2	–	–	–
(reimpl.)	**78.6**	1.3	(99.9)	(0.8)	**82.9**	1.4
CLIP-Mesh	67.8	–	75.8	–	74.5†	–
DreamFusion	75.1	**42.5**	**77.5**	46.6	79.7	**58.5**



Disadvantages

- Over smoothed results
- Lack of details
- 3D object painted on flat surface

The End.