DreamFusion

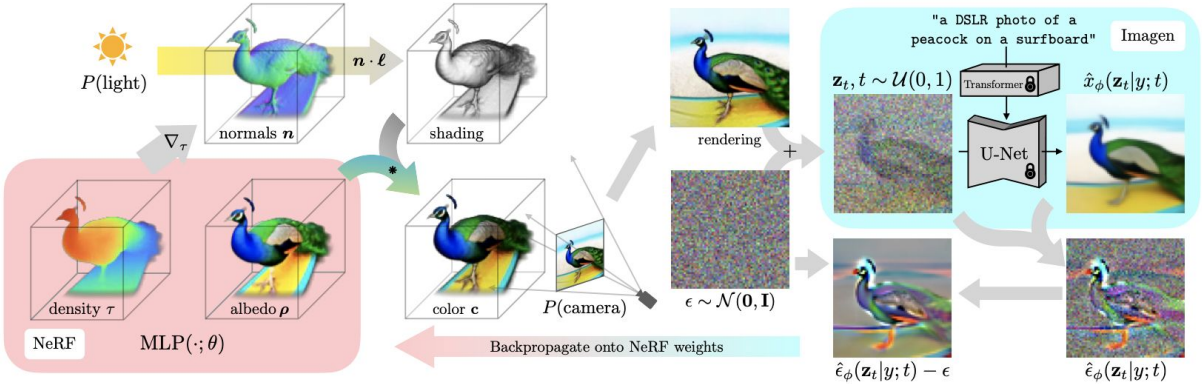
Лебедюк Вероника

Постановка задачи

задача: генерация 3d сцены по текстовому запросу

проблема: мало размеченных 3d данных

решение: научиться использовать text2image диффузионную модель и NeRF для генерации 3d сцены



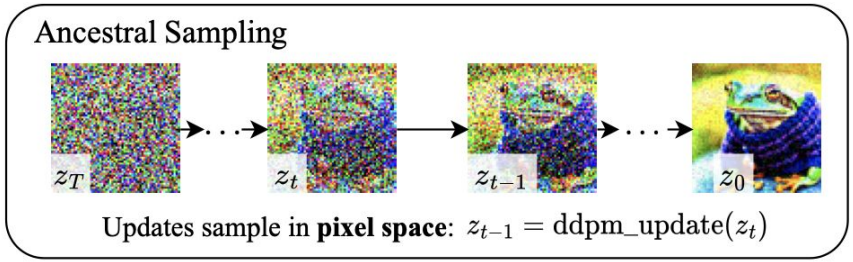
Score Distillation Sampling

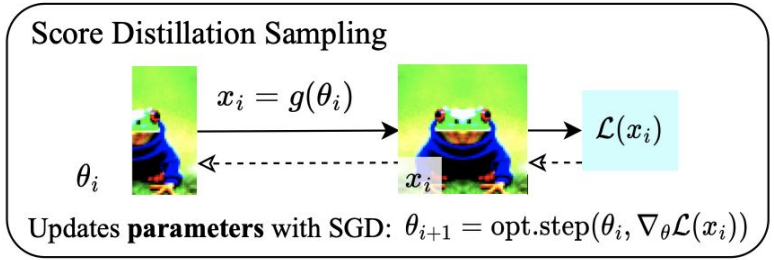
Задача: оптимизировать параметры 3d пространства по 2d сэмплам

Решение: differentiable image parameterization (DIP):

- θ – 3d параметры
- g(θ) – дифференцируемый генератор изображений (рендерер)

Score Distillation Sampling: оптимизация θ так, чтобы изображения g(θ) были похожи на результат 2d модели





Score Distillation Sampling

Пробуем считать $\theta^* = \arg\min_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x} = g(\theta))$

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

СЛОЖНО

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t,\epsilon}\left[w(t) (\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta} \right]$$

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) = \nabla_\theta \mathbb{E}_t \left[\sigma_t / \alpha_t w(t) \text{KL}(q(\mathbf{z}_t | g(\theta); y, t) \| p_\phi(\mathbf{z}_t; y, t)) \right]$$

weighted probability density distillation loss

Score Distillation Sampling

```
params = generator.init()
opt_state = optimizer.init(params)
diffusion_model = diffusion.load_model()
for nstep in iterations:
  t = random.uniform(0., 1.)
  alpha_t, sigma_t = diffusion_model.get_coeffs(t)
  eps = random.normal(img_shape)
  x = generator(params, <other arguments>...) # Get an image observation.
  z_t = alpha_t * x + sigma_t * eps # Diffuse observation.
  epshat_t = diffusion_model.epshat(z_t, y, t) # Score function evaluation.
  g = grad(weight(t) * dot(stopgradient[epshat_t - eps], x), params)
  params, opt_state = optimizer.update(g, opt_state) # Update params with optimizer.
return params
```

Score Distillation Sampling

Результат:

- получили способ обучать параметры 3d модели так, чтобы их рендеры с разных ракурсов выглядели как хорошие изображения
- просто в реализации
- **не требуется модификация диффузионной модели**

NeRF

используем модификацию **mip-NeRF 360**

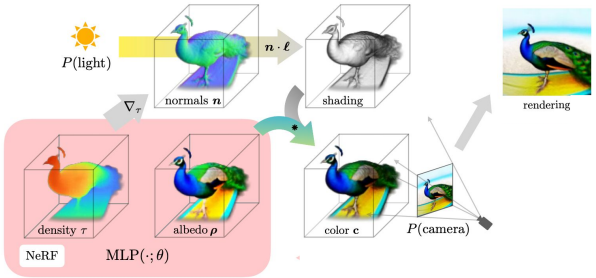
обучается multilayer perceptron (MLP) $(\tau, \boldsymbol{\rho}) = \mathrm{MLP}(\boldsymbol{\mu}; \theta)$

Особенности рендера:

- Обычный NeRF: MLP выдает цвет с тенями относительно лучей из точки обзора
- Модификация: MLP выдает **цвет материала** , а добавление теней происходит отдельно

$$\mathbf{c} = \boldsymbol{\rho} \circ (\boldsymbol{\ell}_\rho \circ \max(0, \boldsymbol{n} \cdot (\boldsymbol{\ell} - \boldsymbol{\mu}) / \|\boldsymbol{\ell} - \boldsymbol{\mu}\|) + \boldsymbol{\ell}_a)$$

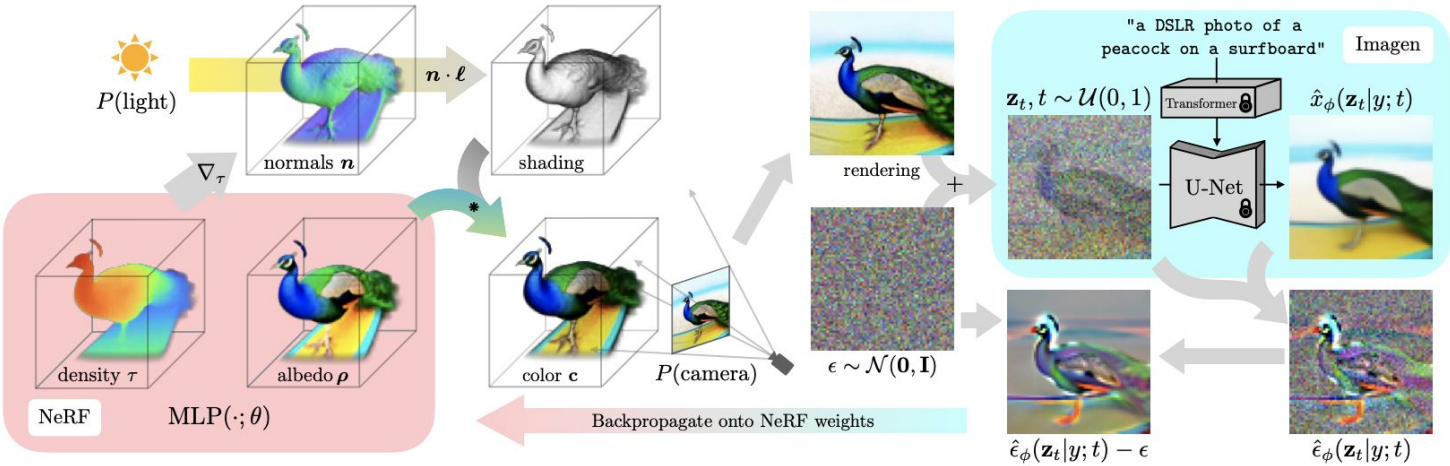
  - случайная замена цвета материала на белый во время обучения помогает избежать генерации плоской сцены



Алгоритм DreamFusion

DIP – NeRF, text2image diffusion model – Imagen

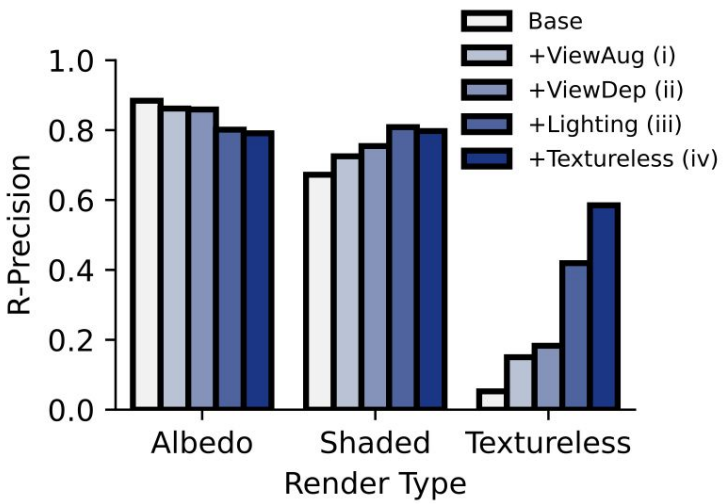
Для каждого запроса инициализируется и обучается свой NeRF

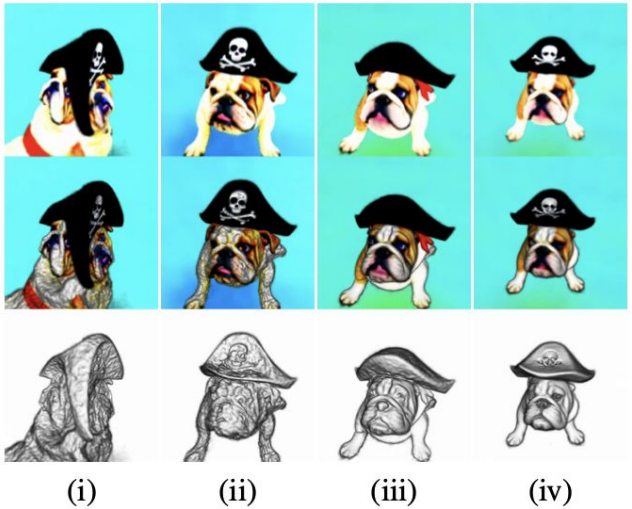


Алгоритм DreamFusion

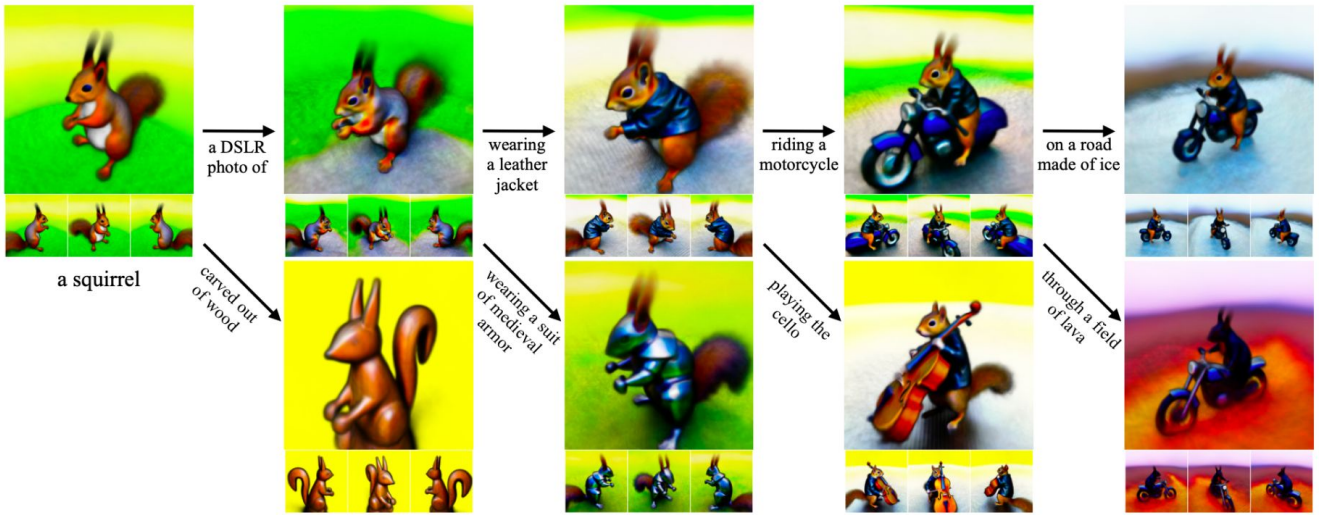
1. случайно выбираем позицию камеры и освещения
2. рендеринг NeRF модели с тенями
3. считаем градиент SDS loss
    a. view-dependent conditioning
4. обновляем параметры NeRF

Эксперименты

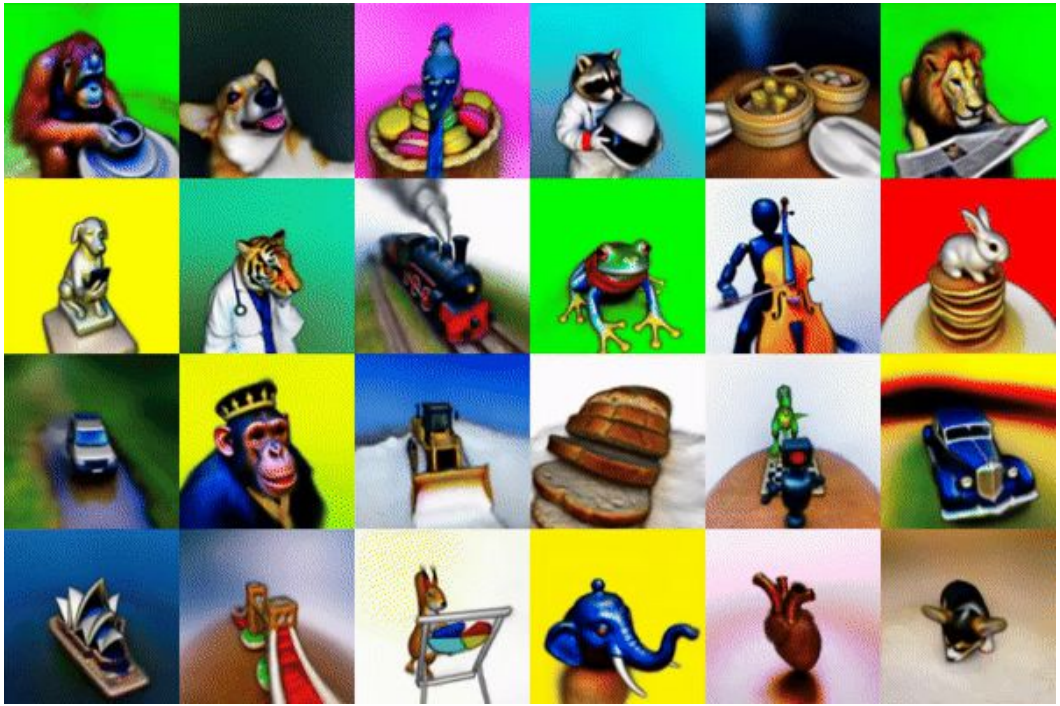




Эксперименты



Эксперименты



Сравнение с аналогами

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 (reimpl.)	68.3 **78.6**	– 1.3	74.2 (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**

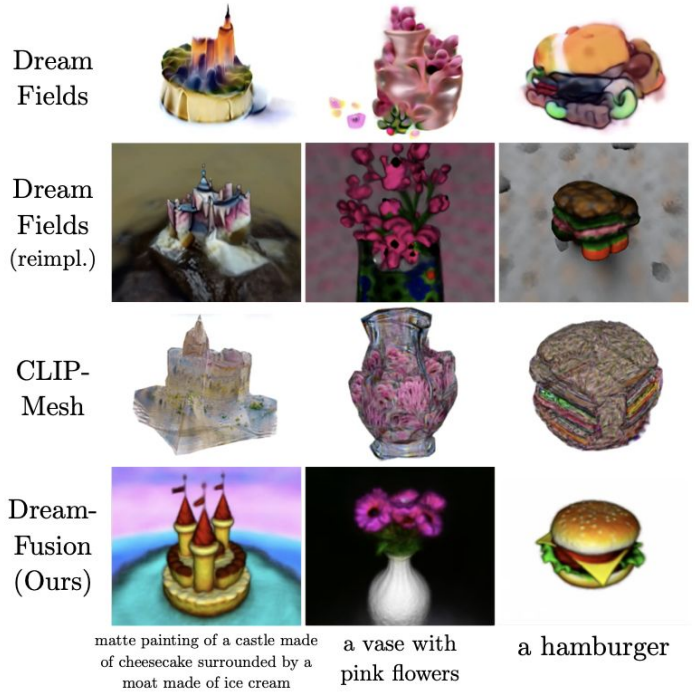


Figure 5: Qualitative comparison with baselines.

Итог

DreamFusion:

- предложили:
  - Score Distillation Sampling для оптимизации параметров NeRF
  - новый способ рендеринга, чтобы уметь получать модель в разном освещении

- обучает 3d модель по текстовому запросу
- не нуждается в 3d данных для обучения
- не требуется модификация диффузионной модели

Источники

https://arxiv.org/pdf/2209.14988.pdf

https://dreamfusion3d.github.io