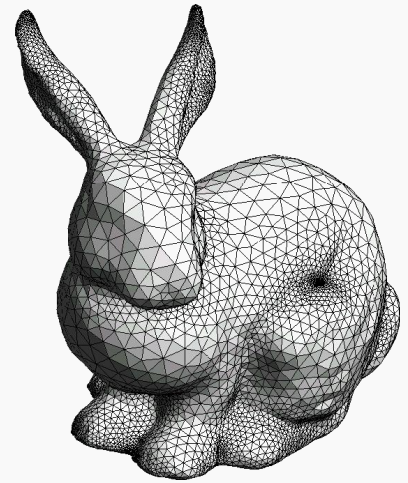
基于SDS的三维生成方法

复旦大学
顾淳
2024年6月13日

三维生成



任务：根据给定prompt生成相应三维模型（Text-to-3D, Image-to-3D）

直接生成三维模型：

- 生成模型：多步扩散（3D Diffusion）
- 回归模型：一次推理（Large Reconstruction Model）

利用预训练二维模型生成三维模型（每个prompt需要重新训练）：

- 在训练中逐渐从二维模型的先验知识中蒸馏出三维模型（SDS）
- 二维模型生成多视角图片，用重建算法得到三维模型

资料库：

三维生成：
https://github.com/hitcslj/Awesome-AIGC-3D

3DGS：
https://github.com/MrNeRF/awesome-3D-gaussian-splatting
https://github.com/yangjiheng/3DGS_and_Beyond_Docs

内容大纲

一 基本的三维表征介绍（NeRF，3DGS，DMTet）

二 SDS与不同三维表征的结合

三 扩散模型先验知识与SDS的改良

四 基于SDS的三维生成优缺点以及其他三维生成方法

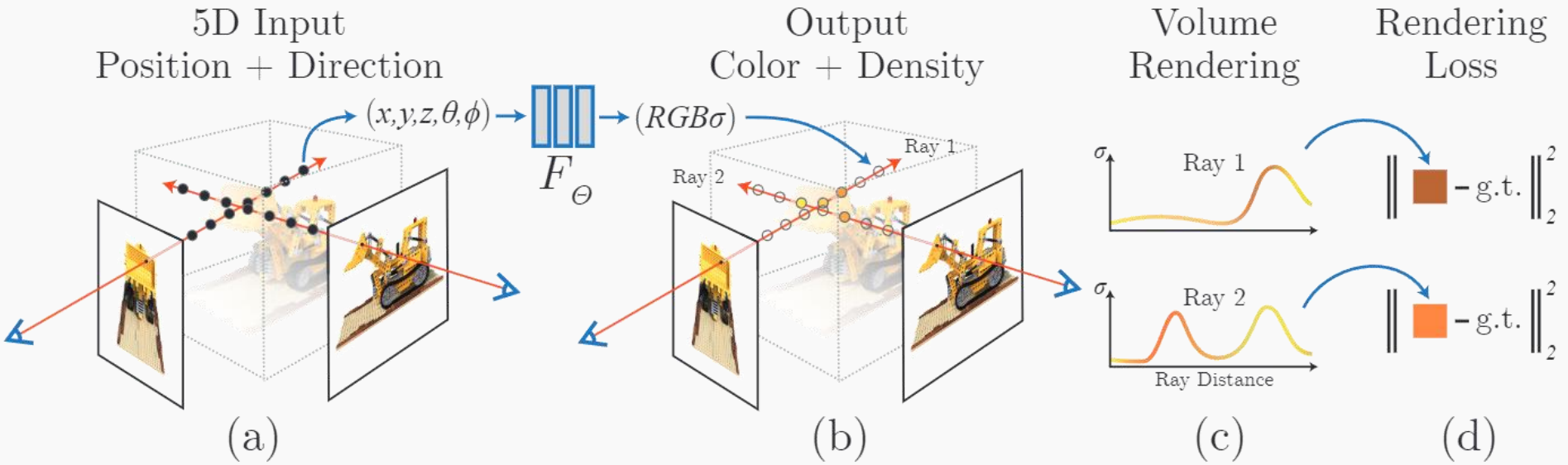
一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

Neural Radiance Field (NeRF)

每次渲染以一个像素为基本单元：

1. 从像素射出光线
2. 在光线上采样
3. 采样点通过MLP得到颜色、密度
4. volume rendering得到最终像素的颜色

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right),$$



一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

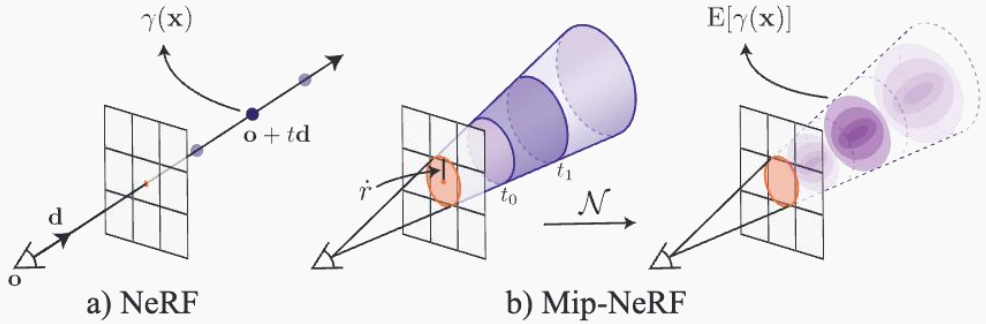
Neural Radiance Field (NeRF)

NeRF的改进

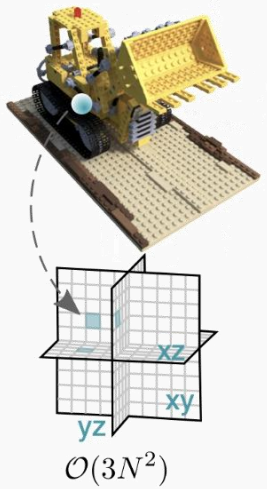
从渲染质量上：抗锯齿（MipNeRF等）

从训练/渲染速度上：更快的查询速度（InstantNGP，Triplane等）

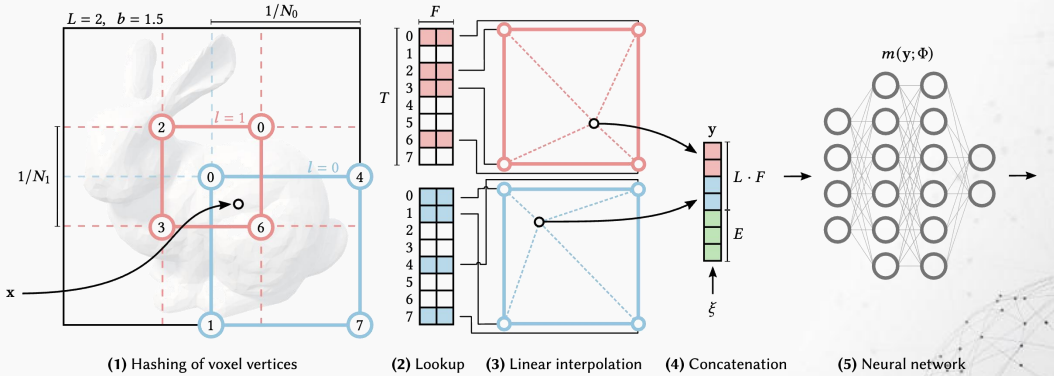
MipNeRF



Triplane



InstantNGP



一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

3D Gaussian Splatting (3DGS)

场景由很多个三维高斯椭球表示，每个高斯包含以下性质：均值（位置），协方差矩阵，不透明度，颜色

高斯在三维空间中的影响： $G(x) = e^{-\frac{1}{2}(x)^T\Sigma^{-1}(x)}$

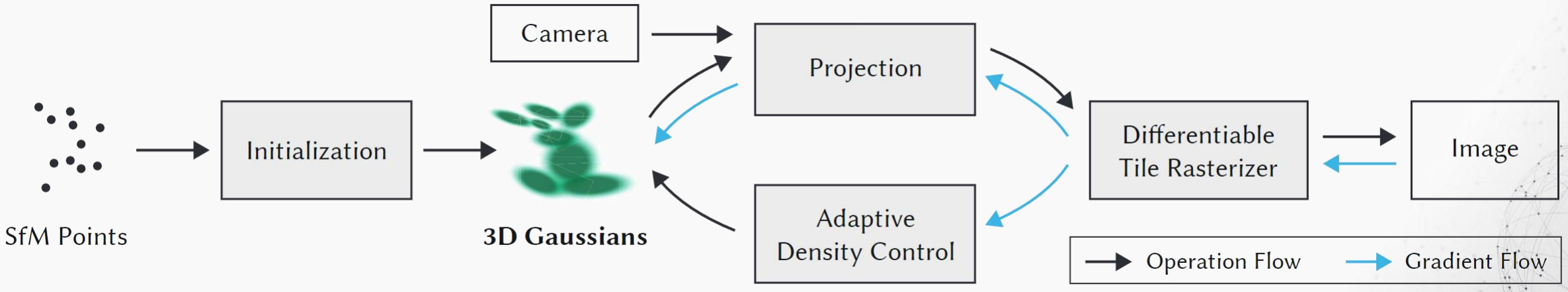
协方差矩阵： $\Sigma = RSS^TR^T$

每次渲染以一张图片为基本单元：

1. 根据相机常数把三维高斯投影到二维相机平面： $\Sigma' = JW\,\Sigma\,W^TJ^T$

（对于相机平面上的一个像素来说，可能有多个高斯落在它上面）

2. 根据这些高斯的深度排序，并按照对这个像素的影响进行alpha-blending得到最终像素的颜色： $C = \sum_{i\in N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j),$



一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

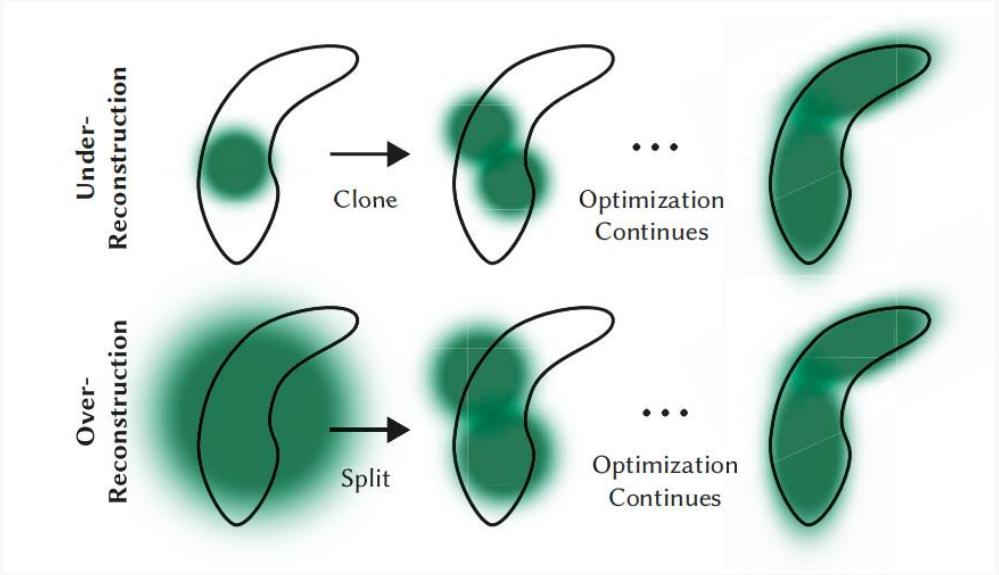
3D Gaussian Splatting (3DGS)

优化流程

Algorithm 1 Optimization and Densification
w, h: width and height of the training images

$M \leftarrow$ SfM Points ▷ Positions
$S, C, A \leftarrow$ InitAttributes() ▷ Covariances, Colors, Opacities
$i \leftarrow 0$ ▷ Iteration Count
while not converged **do**
  $V, \hat{I} \leftarrow$ SampleTrainingView() ▷ Camera V and Image
  $I \leftarrow$ Rasterize(M, S, C, A, V) ▷ Alg. 2
  $L \leftarrow Loss(I, \hat{I})$ ▷ Loss
  $M, S, C, A \leftarrow$ Adam(∇L) ▷ Backprop & Step
  if IsRefinementIteration(i) **then**
    for all Gaussians (μ, Σ, c, α) **in** (M, S, C, A) **do**
      if $\alpha < \epsilon$ or IsTooLarge(μ, Σ) **then** ▷ Pruning
        RemoveGaussian()
      end if
      if $\nabla_p L > \tau_p$ **then** ▷ Densification
        if $\|S\| > \tau_S$ **then** ▷ Over-reconstruction
          SplitGaussian(μ, Σ, c, α)
        else ▷ Under-reconstruction
          CloneGaussian(μ, Σ, c, α)
        end if
      end if
    end for
  end if
  $i \leftarrow i + 1$
end while

Densification

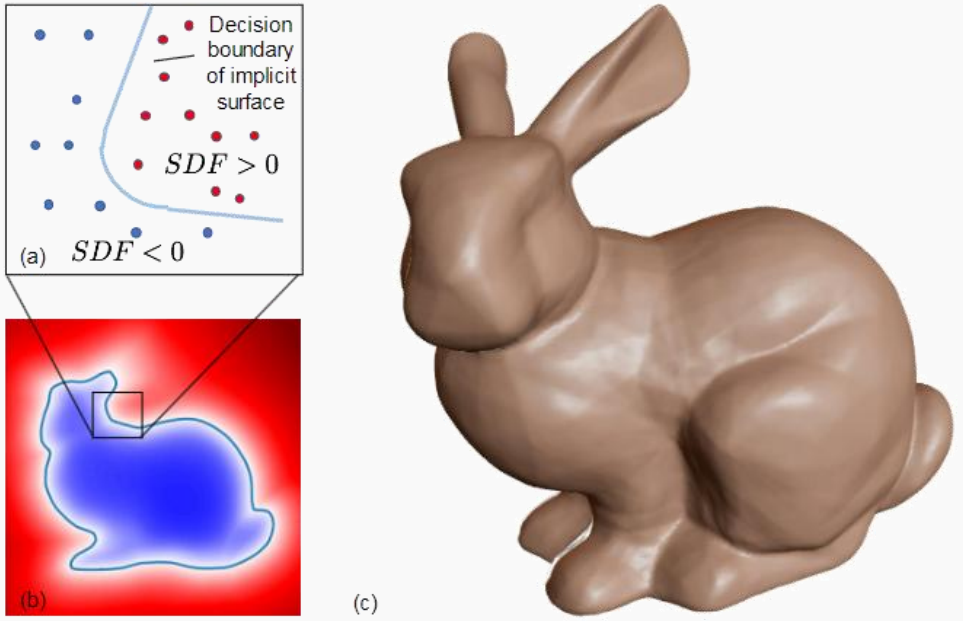


一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

Deep Marching Tetrahedra (DMTet)

Signed Distance Field (SDF)

Surface: $\mathcal{S}_\theta = \{\boldsymbol{x} \in \mathbb{R}^3 \mid f(\boldsymbol{x}; \theta) = 0\}$

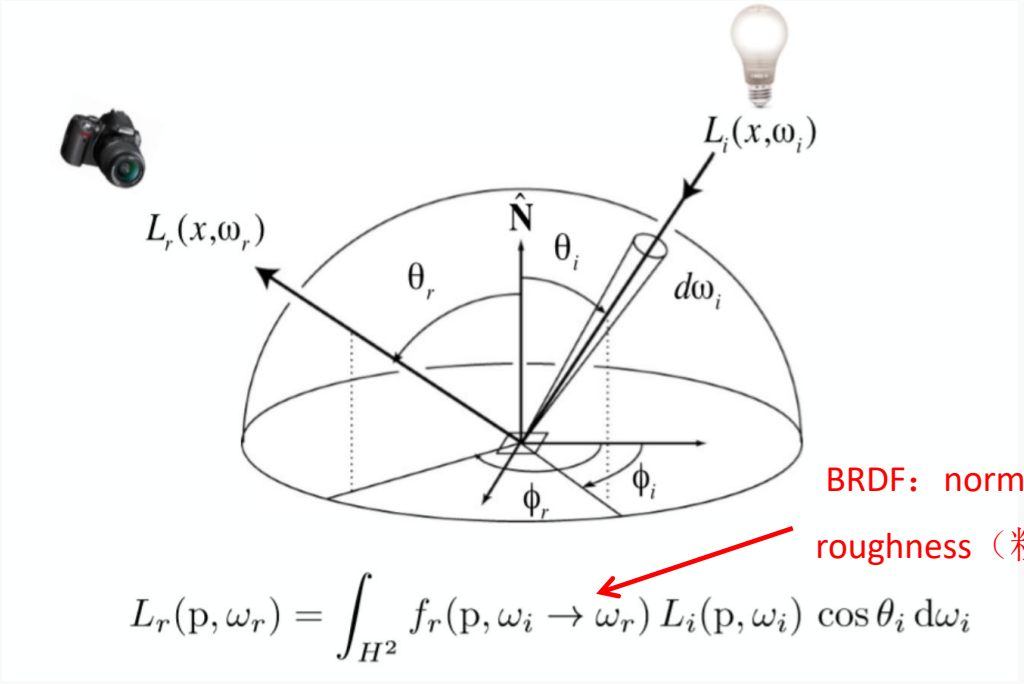


DeepSDF

一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

Deep Marching Tetrahedra (DMTet)

Physically Based Rendering (PBR)



BRDF：normal（法线），albedo（反照率），roughness（粗糙程度），metallic（金属质感）

Rendering equation

https://zhuanlan.zhihu.com/p/145410416

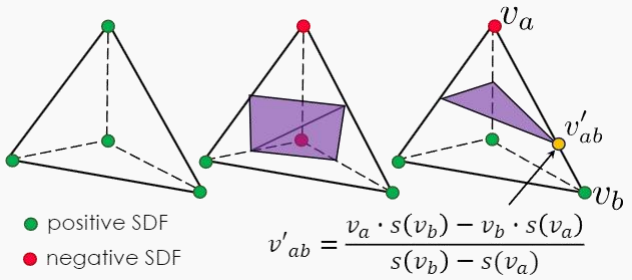
一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

Deep Marching Tetrahedra (DMTet)

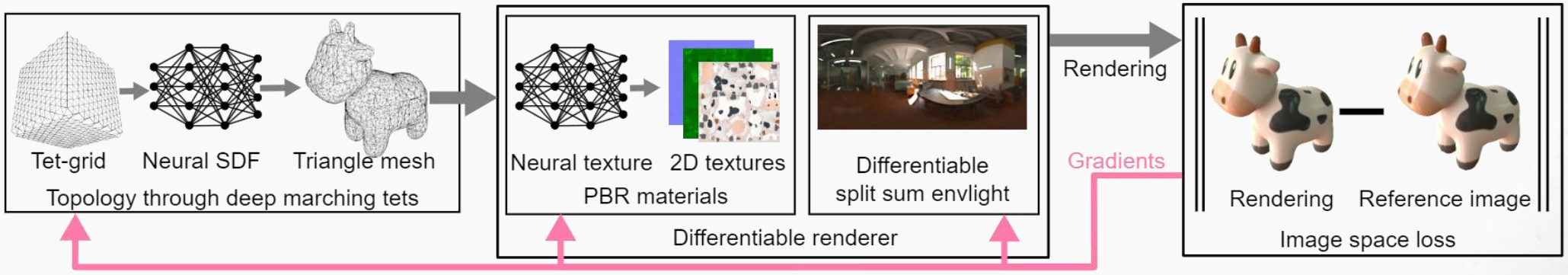
Nvdiffrec

场景中有一个四面体网格（Tet-grid），每个四面体的顶点与附近的四面体相互连接

每次渲染以一张图片为基本单元：



1. 在Signed Distance Field (SDF)中查询网格顶点的SDF值
2. 通过Marching Tetrahedra算法从四面体网格中抽取出mesh
3. 根据相机参数光栅化，算出相机每个像素对应在mesh上的位置
4. 根据这些位置在材质的网络中查询材质：albedo，roughness，metallic等
5. 用上面得到的材质图合成最后的基于物理的渲染（PBR）

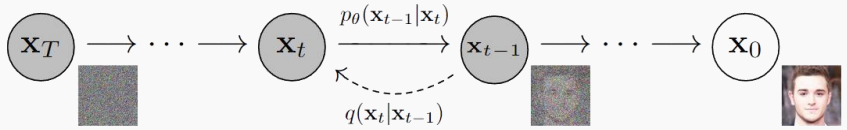


一. 基本的三维表征介绍（NeRF，3DGS，DMTet）

三种表征的对比

Representation	NeRF	3DGS	DMTet
Precise mesh extraction			✓
Easy convergence	✓	✓	
Real-time rendering		✓	✓

二. SDS与不同三维表征的结合



DreamFusion

提出SDS，利用二维预训练扩散模型构造梯度，回传给渲染的图片

训练diffusion的损失函数：

$$\mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[w(t)\|\epsilon_\phi(\alpha_t\mathbf{x}+\sigma_t\epsilon;t)-\epsilon\|_2^2\right]$$

利用这个损失函数来更新渲染的图片：

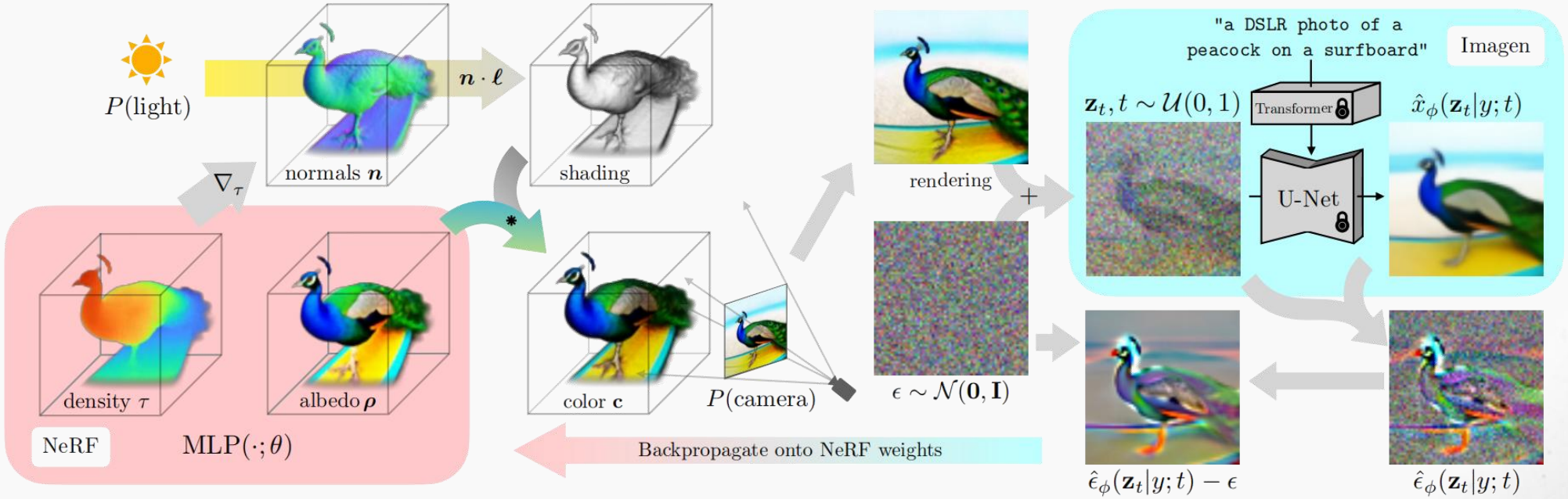
$$\nabla_\theta\mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}=g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t)\underbrace{(\hat{\epsilon}_\phi(\mathbf{z}_t;y,t)-\epsilon)}_{\text{Noise Residual}}\underbrace{\frac{\partial\hat{\epsilon}_\phi(\mathbf{z}_t;y,t)}{\mathbf{z}_t}}_{\text{U-Net Jacobian}}\underbrace{\frac{\partial\mathbf{x}}{\partial\theta}}_{\text{Generator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

省去unet Jacobian项：

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

Classifier Free Guidance：

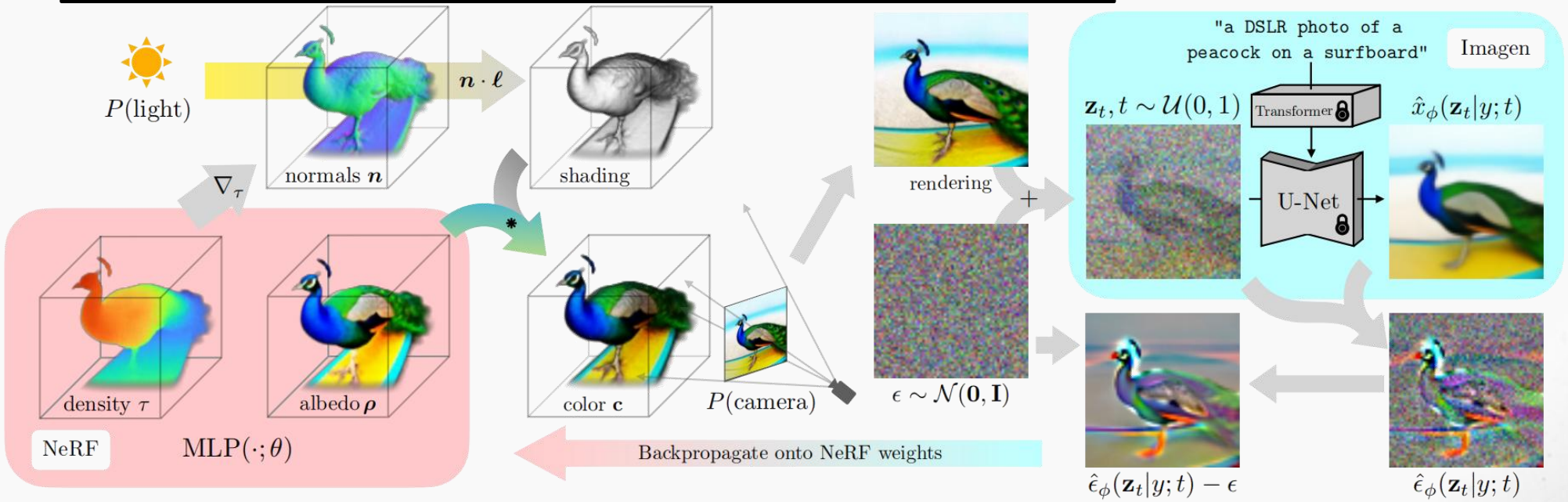
$$\hat{\epsilon}_\phi(\mathbf{z}_t;y,t) = (1+\omega)\epsilon_\phi(\mathbf{z}_t;y,t) - \omega\epsilon_\phi(\mathbf{z}_t;t)$$



二. SDS与不同三维表征的结合

DreamFusion

```
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
```



二. SDS与不同三维表征的结合

DreamFusion

二. SDS与不同三维表征的结合

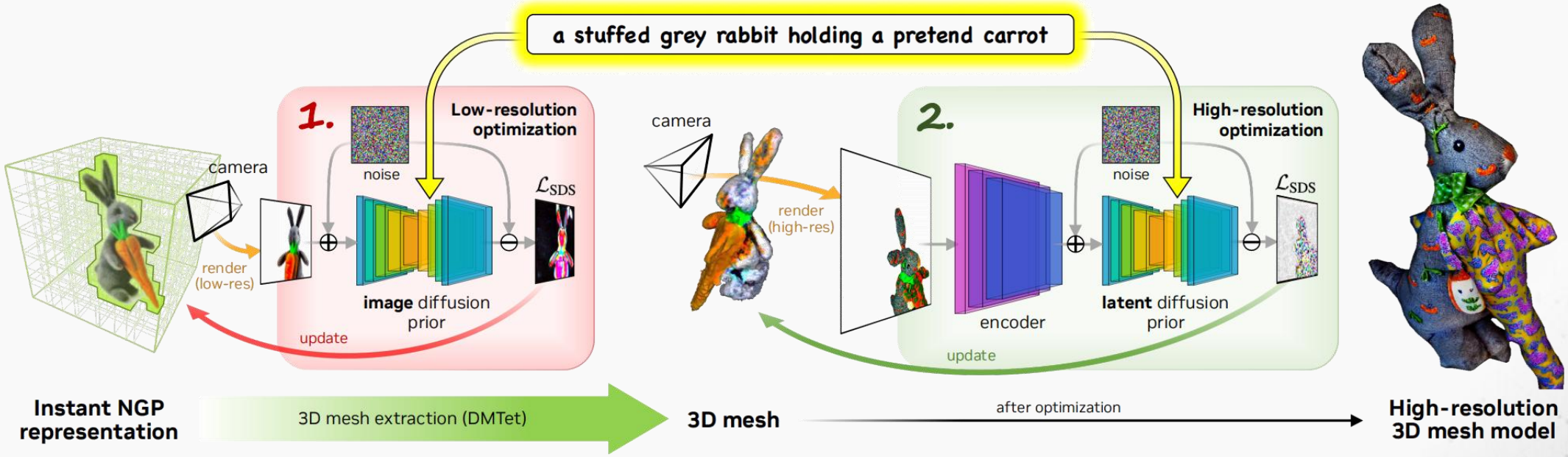
Magic3D　　提出two-stage训练策略，引入DMTet

针对DreamFusion中的两个问题：

1. MipNeRF速度慢
2. 来自Diffusion的监督分辨率低（64x64）

提出采取two-stage训练策略：

1. 先用SDS在64x64分辨率训练一个InstantNGP
2. 将InstantNGP转换为DMTet
3. 继续用SDS在512x512分辨率训练DMTet



二. SDS与不同三维表征的结合

Magic3D

a silver platter piled high with fruits

michelangelo style statue of an astronaut

a stuffed grey rabbit holding a pretend carrot

an iguana holding a balloon

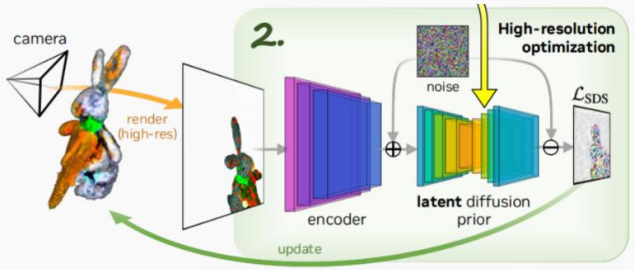
a beautiful dress made out of garbage bags

an imperial state crown of england

a blue poison-dart frog sitting on a water lily

neuschwanstein castle, aerial view

二. SDS与不同三维表征的结合



Fantasia3D

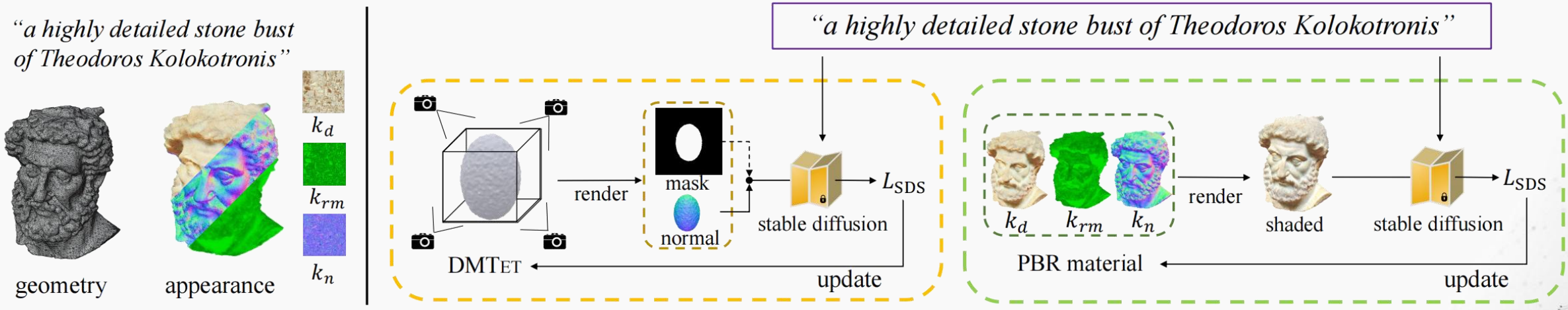
提出将几何和材质分开训练，并引入BRDF来建模材质

采取two-stage训练策略（先训几何，后训材质）：

1. 用SDS在512x512分辨率训练DMTet，以法线图（normal）为扩散模型的输入（在初始训练阶段采用latent encoding）

latent encoding：直接把normal和mask拼在一起，作为latent（省去了过encoder这一步）

2. 固定住几何，继续用SDS在512x512分辨率训练DMTet，用材质图合成最终渲染图像作为扩散模型的输入



(a) Disentangled representation (b) Geometry modeling (c) Appearance modeling

二. SDS与不同三维表征的结合

Fantasia3D

A vintage record player

An ice cream sundae

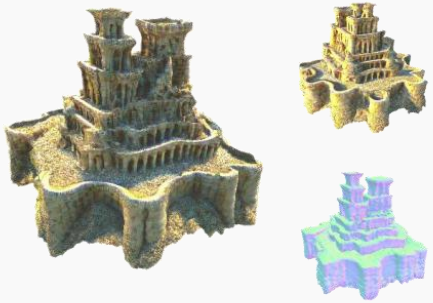
A red rotary telephone

A fresh cinnamon roll covered in glaze, high resolution

A delicious croissant

A golden goblet

The leaning tower of Pisa



A highly detailed sandcastle

A car made out of cheese

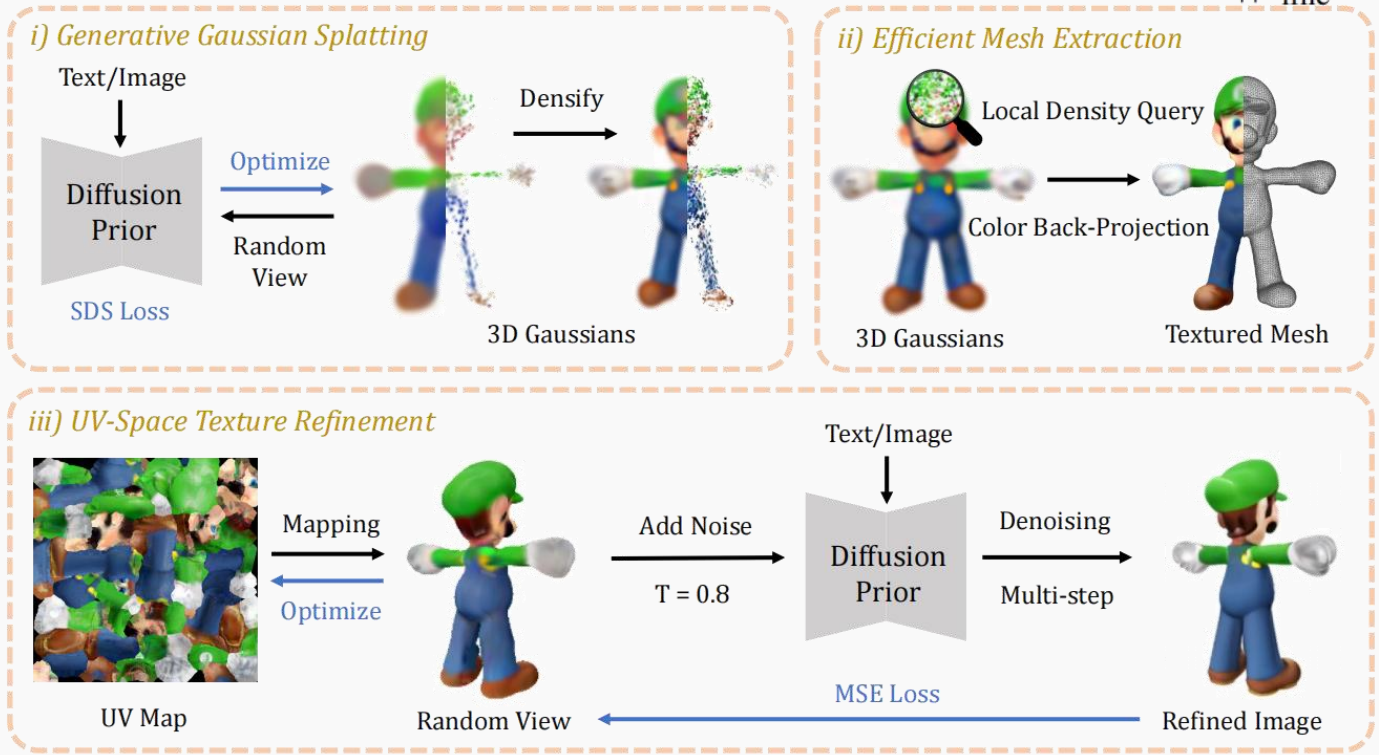
二. SDS与不同三维表征的结合

DreamGaussian　　引入3DGS，仅需单卡两分钟得到结果

采取two-stage训练策略：

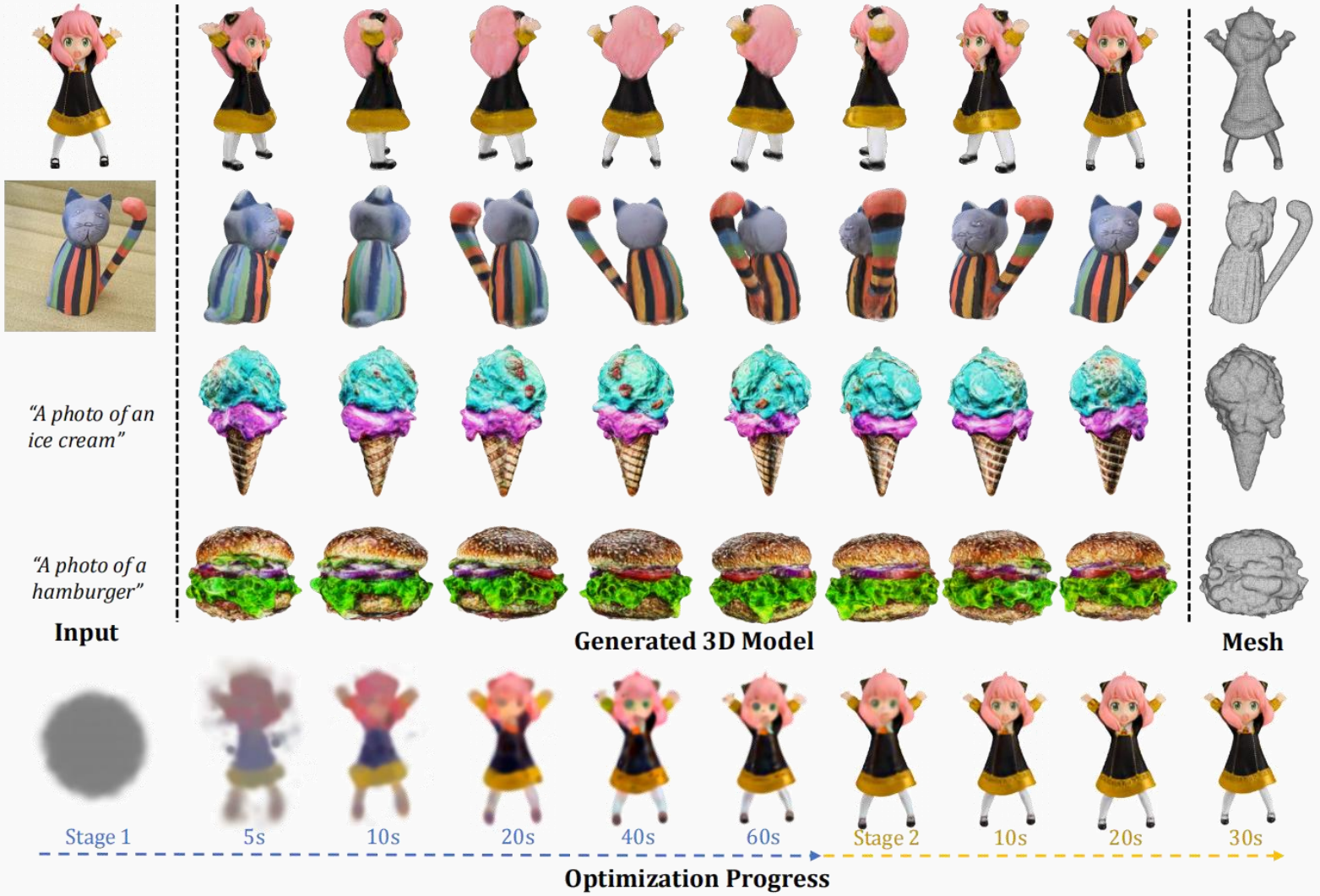
1. 先用SDS训练一个3DGS（比NeRF更容易收敛）
2. 从3DGS中抽取mesh：　$d(\mathbf{x}) = \sum_i \alpha_i \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{x_i})^T \Sigma_i^{-1}(\mathbf{x}-\mathbf{x_i}))$
3. 将渲染图片进行multi-step denoising，作为监督来继续优化材质：　$I^p_{\text{fine}} = f_\phi(I^p_{\text{coarse}} + \epsilon(t_{\text{start}}); t_{\text{start}}, c)$

$$\mathcal{L}_{\text{MSE}} = ||I^p_{\text{fine}} - I^p_{\text{coarse}}||_2^2$$



二. SDS与不同三维表征的结合

DreamGaussian



三. 扩散模型先验知识与SDS的改良

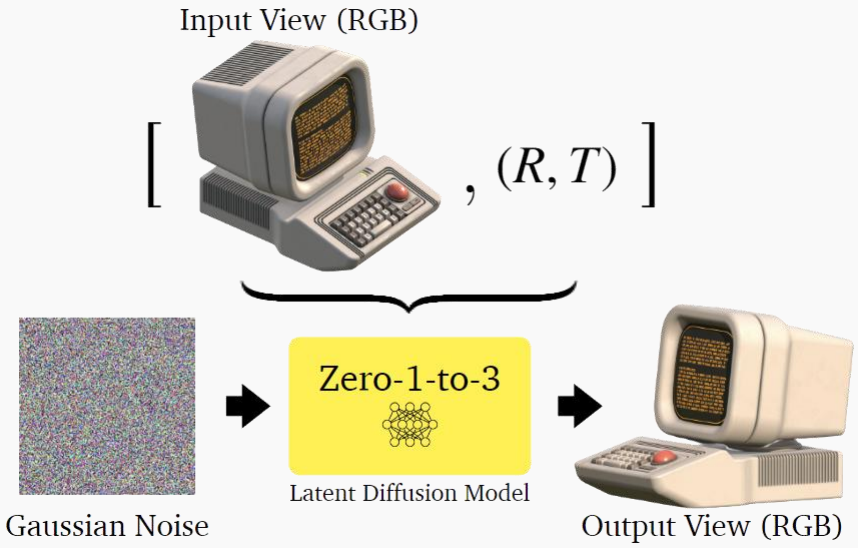
Zero-1-to-3

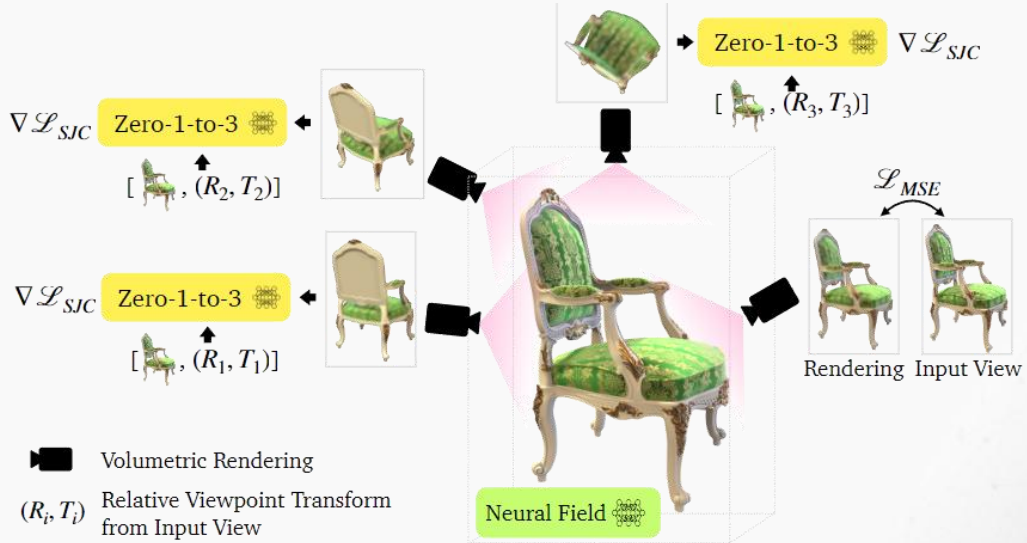
让扩散模型拥有3D awareness

需要在大规模数据集（Objaverse）上finetune。

在扩散模型的condition输入中加入相机参数

1. 给定一张图片
2. 希望得到图片中相应物体的另一个视角
3. 将给定的图片和另一个视角相对给定图片的视角的相对相机位姿拼在一起作为condition
4. 输出对于视角的图片



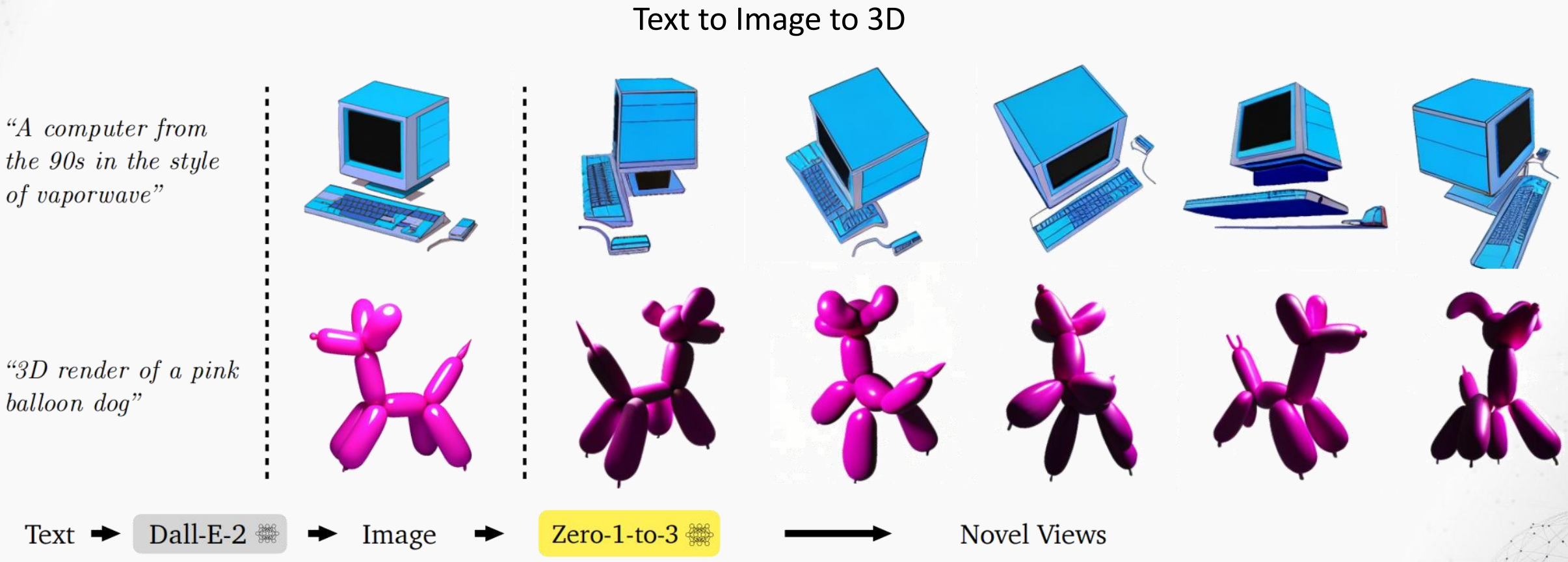


三. 扩散模型先验知识与SDS的改良

Zero-1-to-3

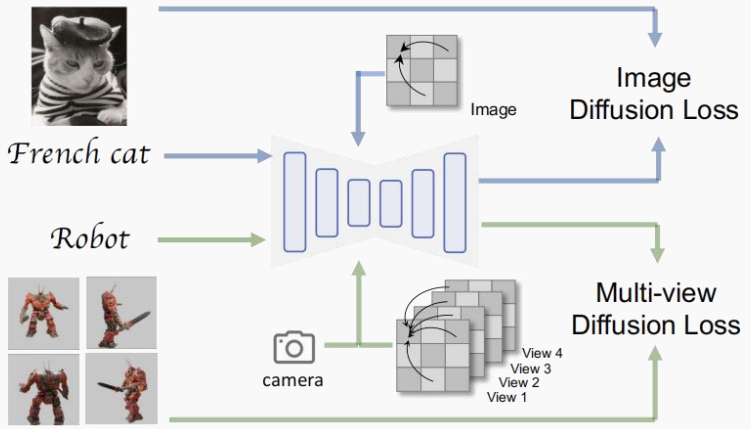
三. 扩散模型先验知识与SDS的改良

Zero-1-to-3



三. 扩散模型先验知识与SDS的改良

MVDream　　一次性输出一个物体的四个视角



在原本Stable Diffusion的基础上做了两点改动：

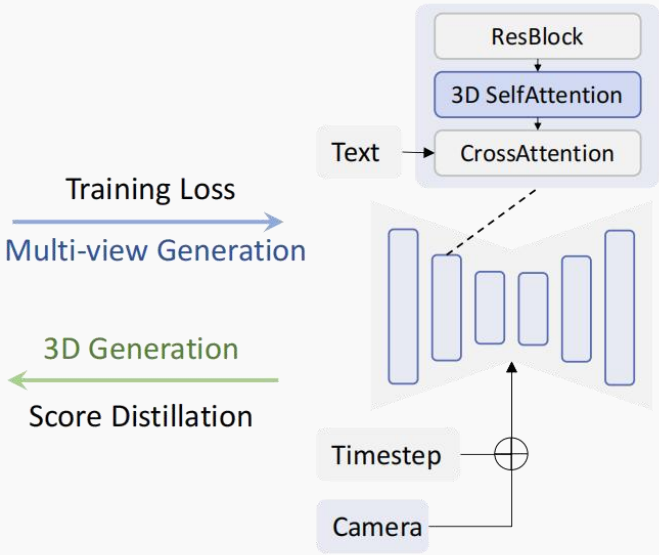
1. 将2D self-attention改为3D self-attention

(B × F) × H × W × C ---> B × (F × H × W) × C

2. 加入camera embedding

3D model

Rendered images



Multi-view Diffusion UNet

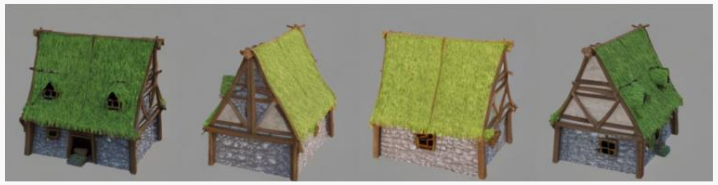
三. 扩散模型先验知识与SDS的改良

MVDream

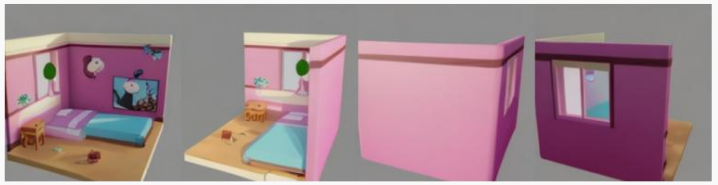
Diffusion生成的多视角图片

Zombie bust, terror, 123dsculpt, bust, zombie

Battletech Zeus with a sword!, tabletop, miniature, battletech, miniatures, wargames, 3d asset



Medieval House, grass, medieval, vines, farm, middle-age, medieval-house, stone, house, home, wood, medieval-decor, 3d asset



Isometric Slowpoke Themed Bedroom, fanart, pokemon, bedroom, assignment, isometric, pokemon3d, isometric-room, room-low-poly, 3d asset

An astronaut riding a horse

A bald eagle carved out of wood

A bull dog wearing a black pirate hat

a DSLR photo of a ghost eating a hamburger

三. 扩散模型先验知识与SDS的改良

MVDream

3D generation

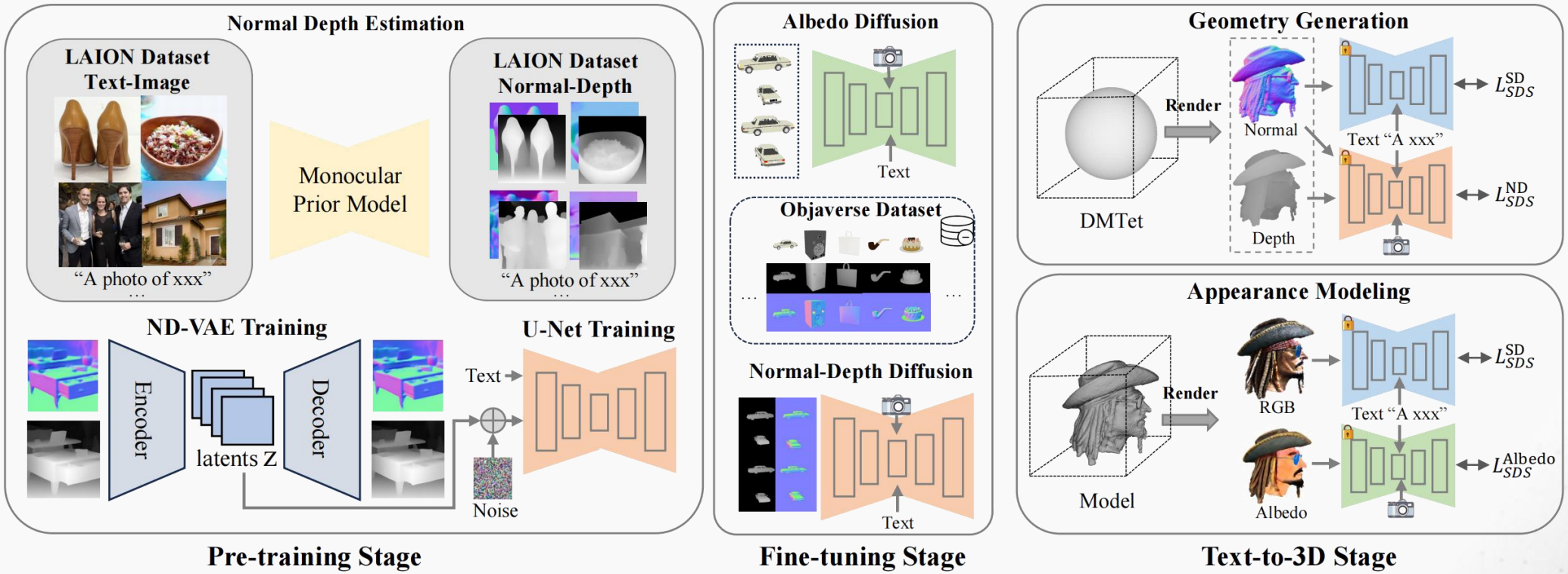
三. 扩散模型先验知识与SDS的改良

RichDreamer　　让扩散模型拥有关于normal/depth/albedo的先验

针对Fantasia3D中的问题：把normal当作SDS中扩散模型的输入

重新训练了两个多视角扩散模型用于SDS：

1. Normal-Depth Diffusion：根据文本生成normal，depth
2. Albedo Diffusion：根据文本生成albedo



三. 扩散模型先验知识与SDS的改良

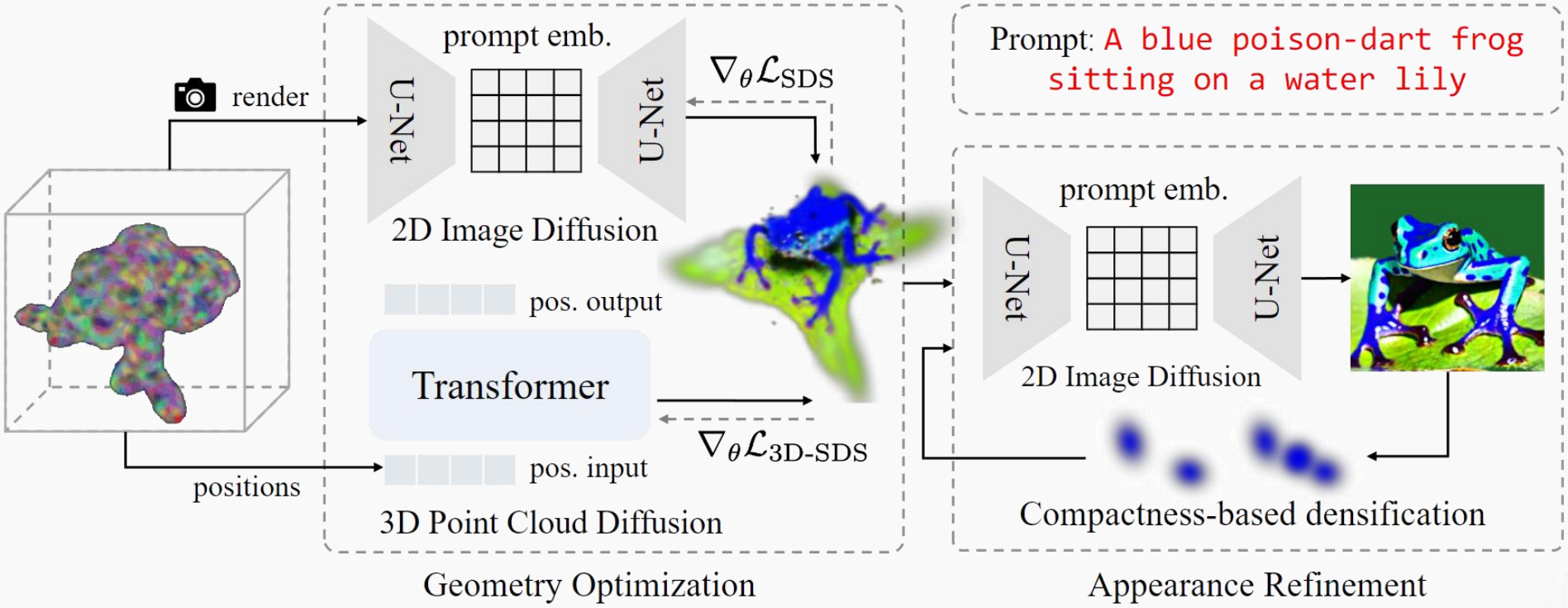
RichDreamer

三. 扩散模型先验知识与SDS的改良

GSGEN　　引入3DGS，并结合3D SDS

以3DGS为三维表征，在2D SDS的基础上，引入Point-E带来的3D SDS作为额外的监督信号

Point-E：三维点云扩散模型，根据文本输出三维点云



三. 扩散模型先验知识与SDS的改良

GSGEN

三. 扩散模型先验知识与SDS的改良

ProlificDreamer　　改进SDS

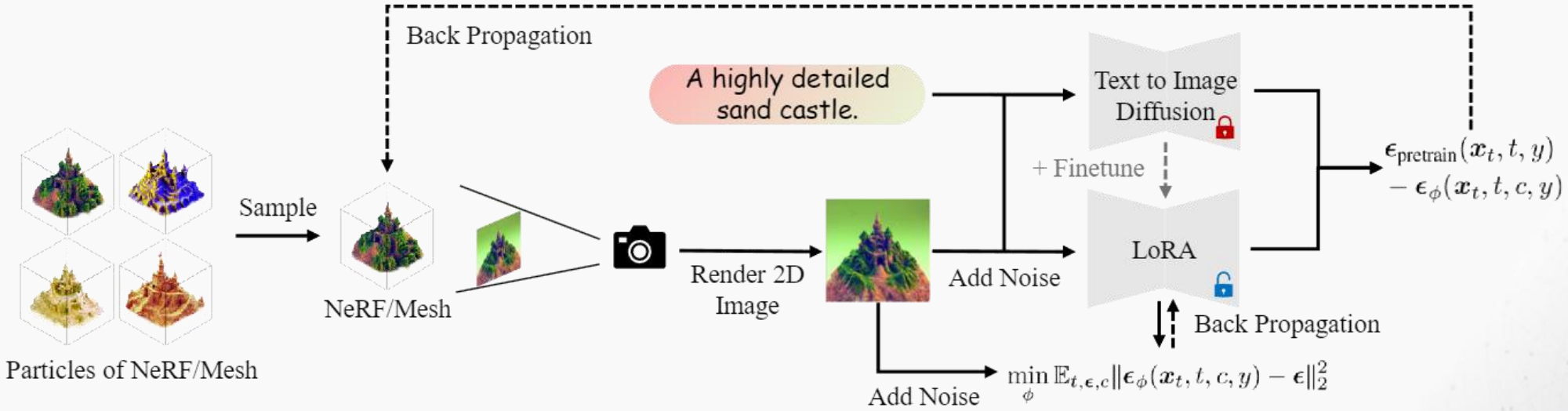
针对SDS存在的问题：过饱和，过平滑，缺少细节

提出Variational Score Distillation (VSD)，并同时优化多个场景

SDS：　$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta) \approx \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)(\boldsymbol{\epsilon}_{\text{pretrain}}(\boldsymbol{x}_t, t, y^c) - \boldsymbol{\epsilon})\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right]$

VSD：　$\nabla_\theta \mathcal{L}_{\text{VSD}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)\left(\boldsymbol{\epsilon}_{\text{pretrain}}(\boldsymbol{x}_t, t, y^c) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, c, y)\right)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right]$

https://zhuanlan.zhihu.com/p/638855316



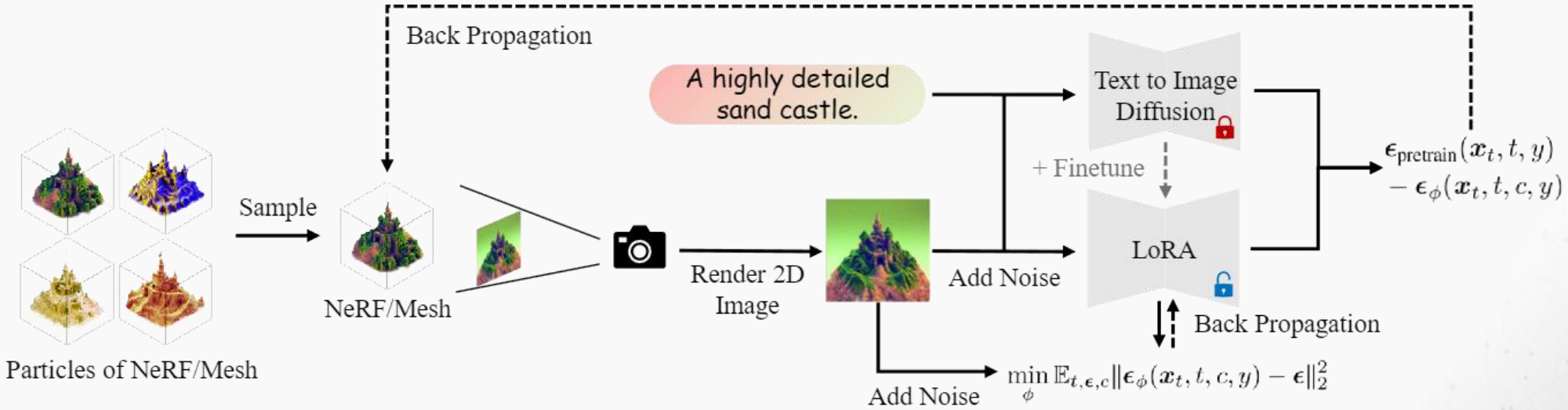
三. 扩散模型先验知识与SDS的改良

ProlificDreamer

Algorithm 1 Variational Score Distillation

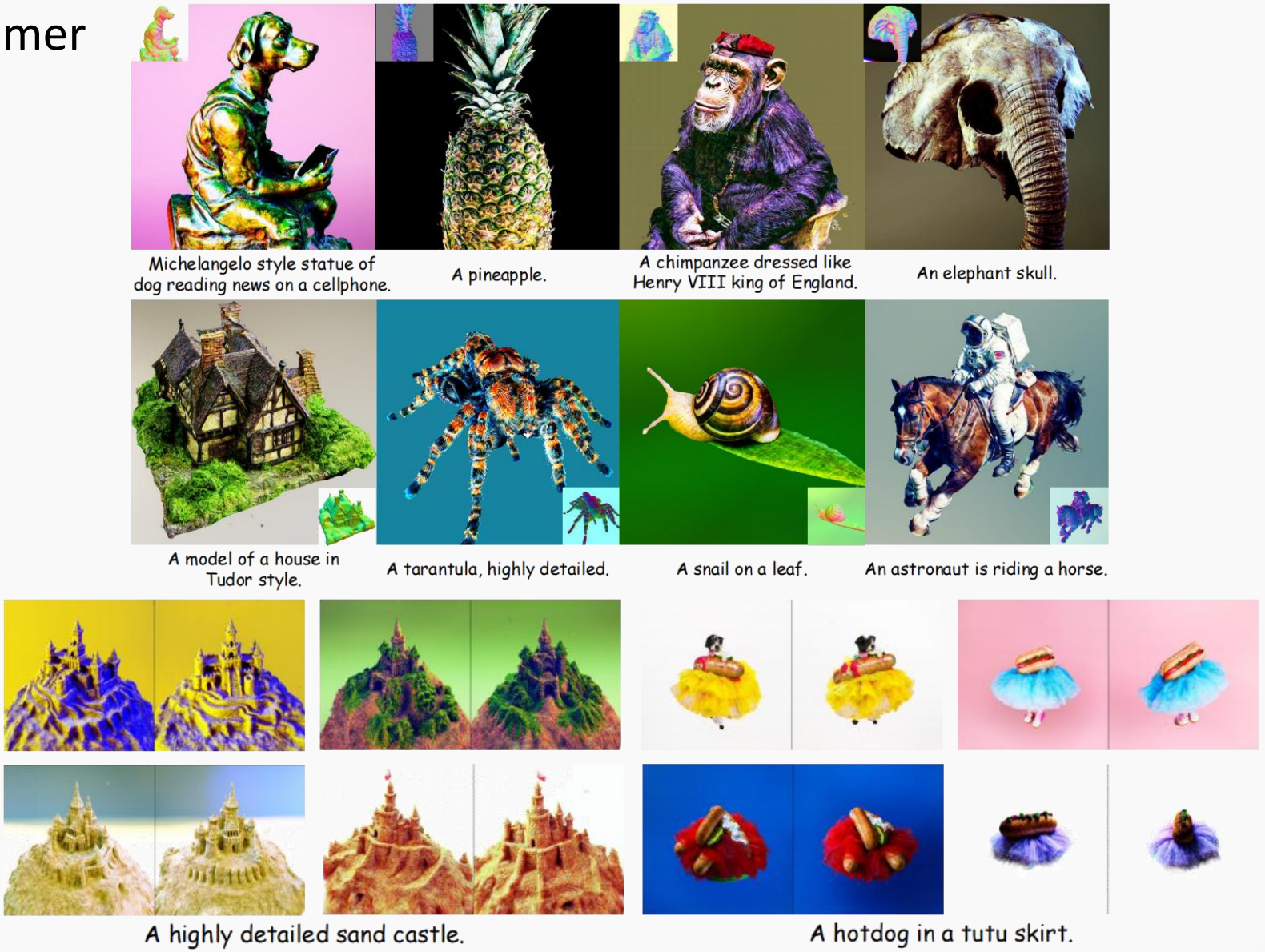
Input: Number of particles n (≥ 1). Large text-to-image diffusion model $\epsilon_{\text{pretrain}}$. Learning rate η_1 and η_2 for 3D structures and diffusion model parameters, respectively. A prompt y.

1: **initialize** n 3D structures $\{\theta^{(i)}\}_{i=1}^{n}$, a noise prediction model ϵ_ϕ parameterized by ϕ.
2: **while** not converged **do**
3: Randomly sample $\theta \sim \{\theta^{(i)}\}_{i=1}^{n}$ and a camera pose c.
4: Render the 3D structure θ at pose c to get a 2D image $\boldsymbol{x}_0 = \boldsymbol{g}(\theta, c)$.
5: $\theta \leftarrow \theta - \eta_1 \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)\left(\boldsymbol{\epsilon}_{\text{pretrain}}(\boldsymbol{x}_t, t, y^c) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, c, y)\right)\frac{\partial \boldsymbol{g}(\theta,c)}{\partial \theta}\right]$
6: $\phi \leftarrow \phi - \eta_2 \nabla_\phi \mathbb{E}_{t,\epsilon}\|\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, c, y) - \boldsymbol{\epsilon}\|_2^2$.
7: **end while**
8: **return**



三. 扩散模型先验知识与SDS的改良

ProlificDreamer



四. 基于SDS的三维生成优缺点以及其他三维生成方法

SDS的优缺点

优点：能够利用在大规模数据集上预训练过的二维扩散模型，可以生成质量非常高的三维模型

缺点：需要对每个prompt进行单独的训练，且多视角一致性依赖于扩散模型先验

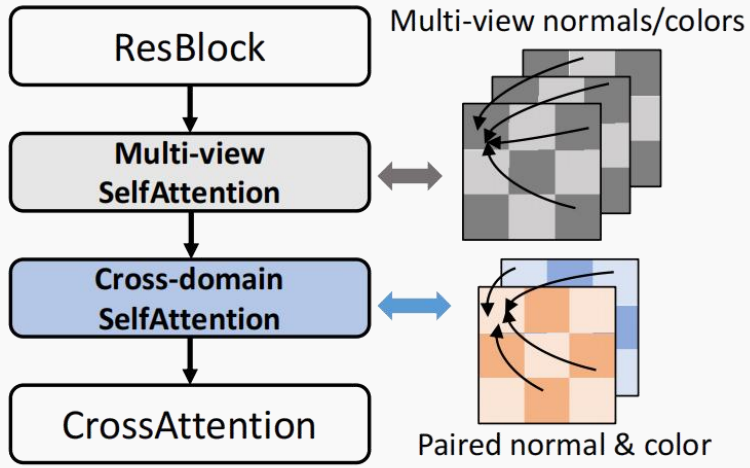
四. 基于SDS的三维生成优缺点以及其他三维生成方法

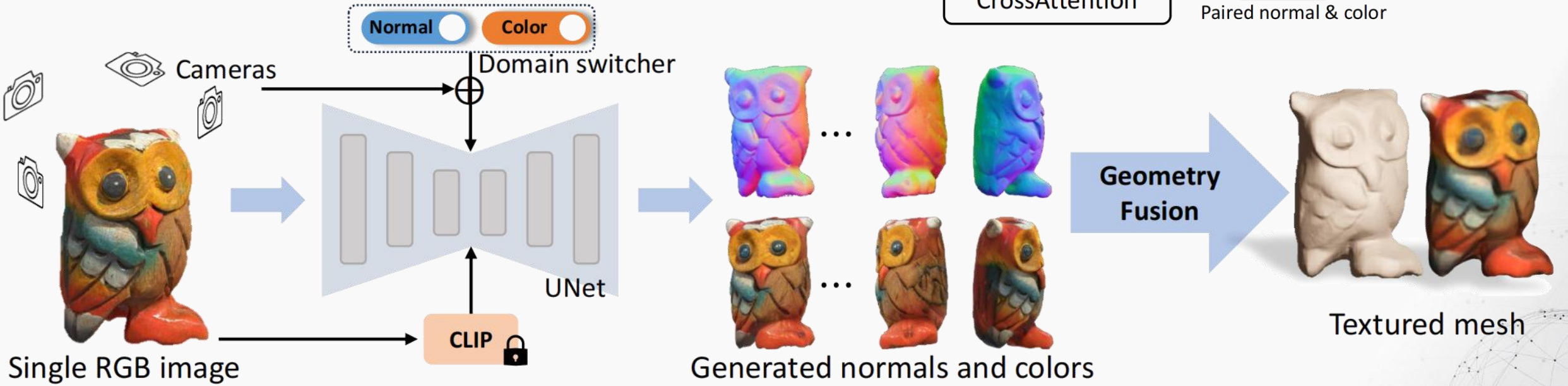
Wonder3D　　扩散模型生成多视角RGB/Normal图片用于重建

给定一张图片，生成对应物体的其他视角，用于重建

主要加入了两个模块：

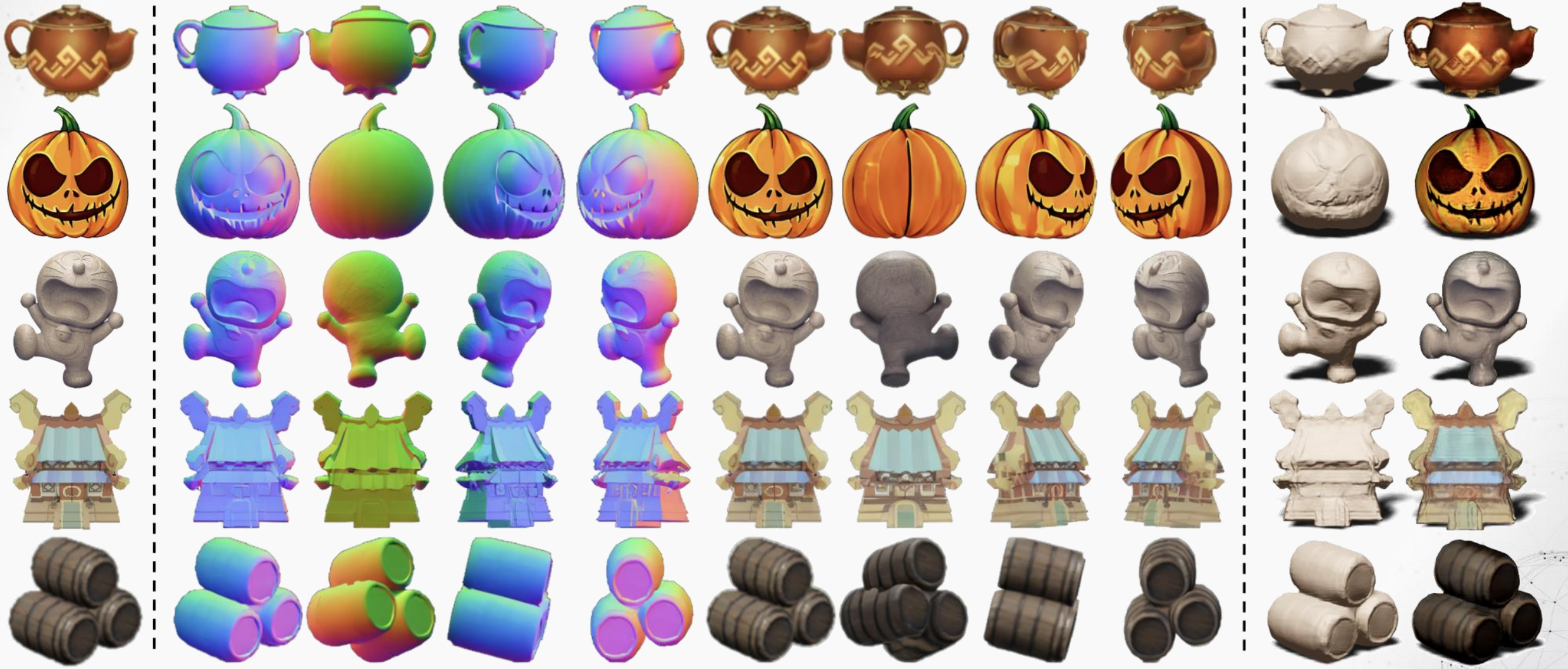
1. Multi-view Self-Attention：保证多视角一致性
2. Cross-domin Self-Attention：保证RGB和Normal一致





四. 基于SDS的三维生成优缺点以及其他三维生成方法

Wonder3D



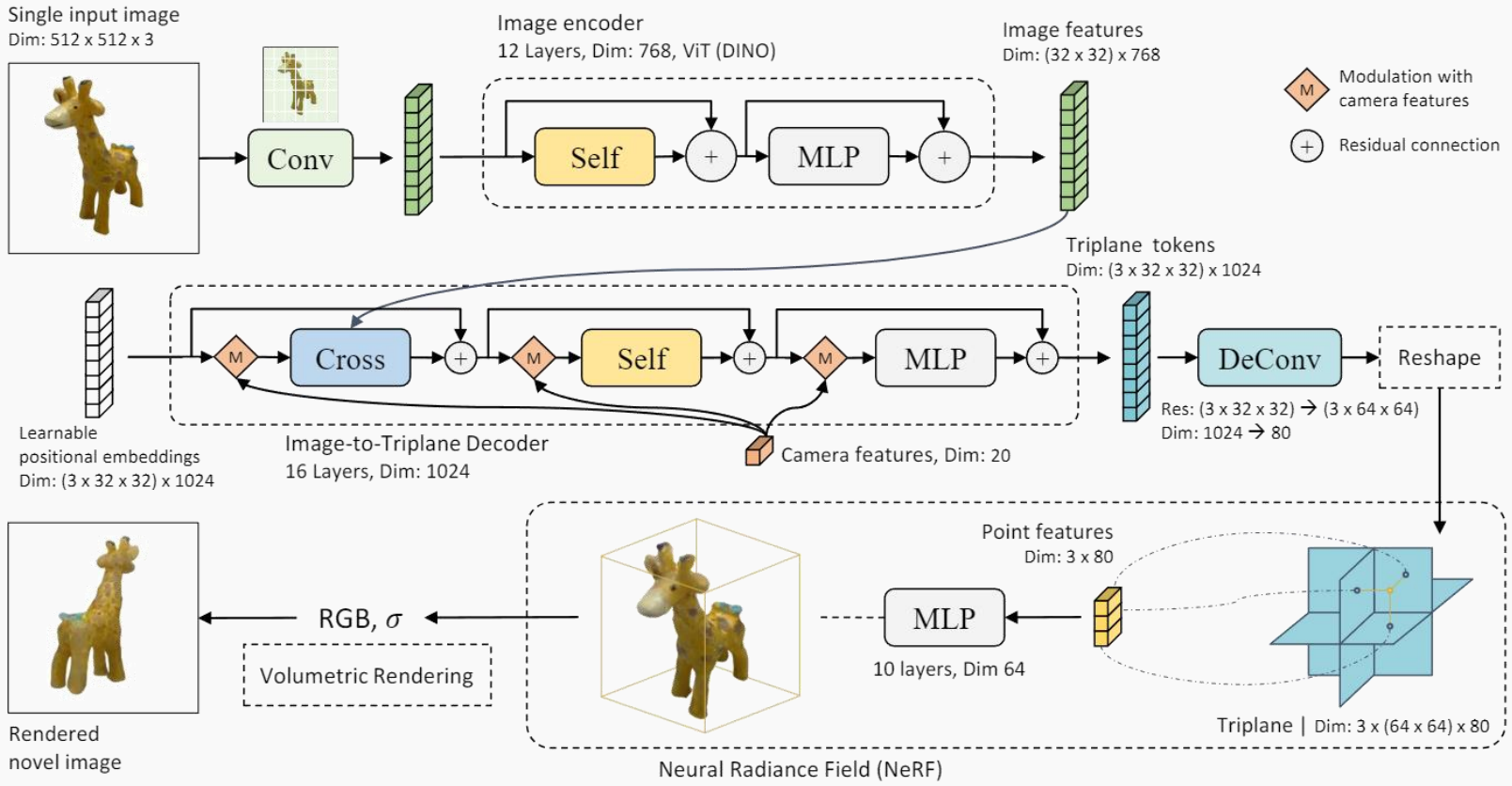
四. 基于SDS的三维生成优缺点以及其他三维生成方法

LRM　输入给定图片，一次推理得到三维表征（triplane）

推理流程：

1. 输入图片过预训练encoder（DINO）得到image features。

2. Image-to-Triplane Decoder将image features、camera features、positional embeddings转化为triplane



四. 基于SDS的三维生成优缺点以及其他三维生成方法

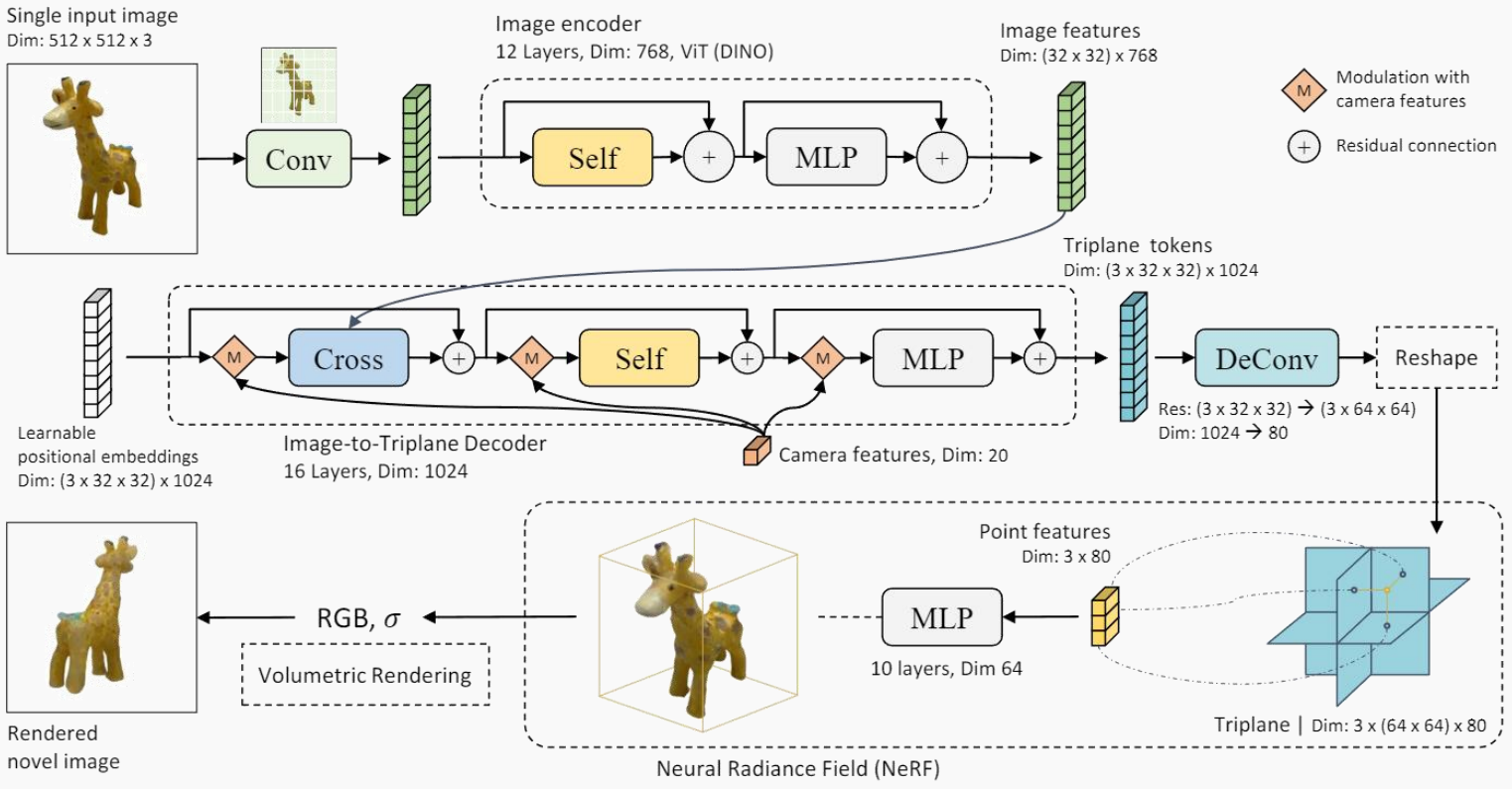
LRM　输入给定图片，一次推理得到三维表征（triplane）

Modulation with camera features:

$$\gamma, \beta = \mathrm{MLP}^{\mathrm{mod}}(\tilde{\boldsymbol{c}})$$
$$\mathrm{ModLN_c}(\boldsymbol{f}_j) = \mathrm{LN}(\boldsymbol{f}_j) \cdot (1+\gamma) + \beta$$

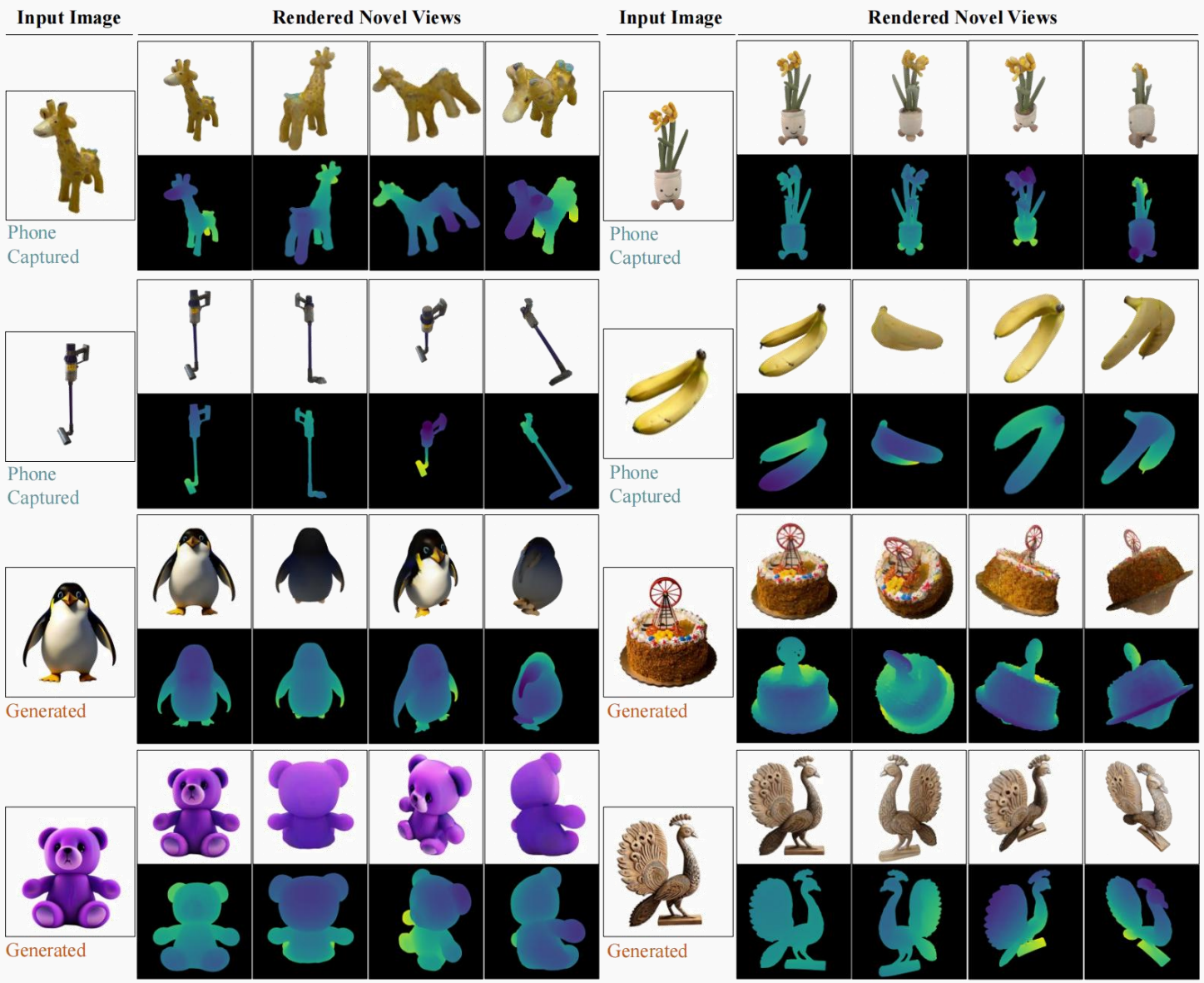
Transformer Layers:

$$\boldsymbol{f}_j^{cross} = \mathrm{CrossAttn}(\mathrm{ModLN}_c(\boldsymbol{f}_j^{in}); \{\boldsymbol{h}_i\}_{i=1}^n) + \boldsymbol{f}_j^{in}$$
$$\boldsymbol{f}_j^{self} = \mathrm{SelfAttn}(\mathrm{ModLN}_c(\boldsymbol{f}_j^{cross}); \{\mathrm{ModLN}_c(\boldsymbol{f}_j^{cross})\}_j) + \boldsymbol{f}_j^{cross}$$
$$\boldsymbol{f}_j^{out} = \mathrm{MLP}^{tfm}(\mathrm{ModLN}_c(\boldsymbol{f}_j^{self})) + \boldsymbol{f}_j^{self}$$



四. 基于SDS的三维生成优缺点以及其他三维生成方法

LRM



四. 基于SDS的三维生成优缺点以及其他三维生成方法

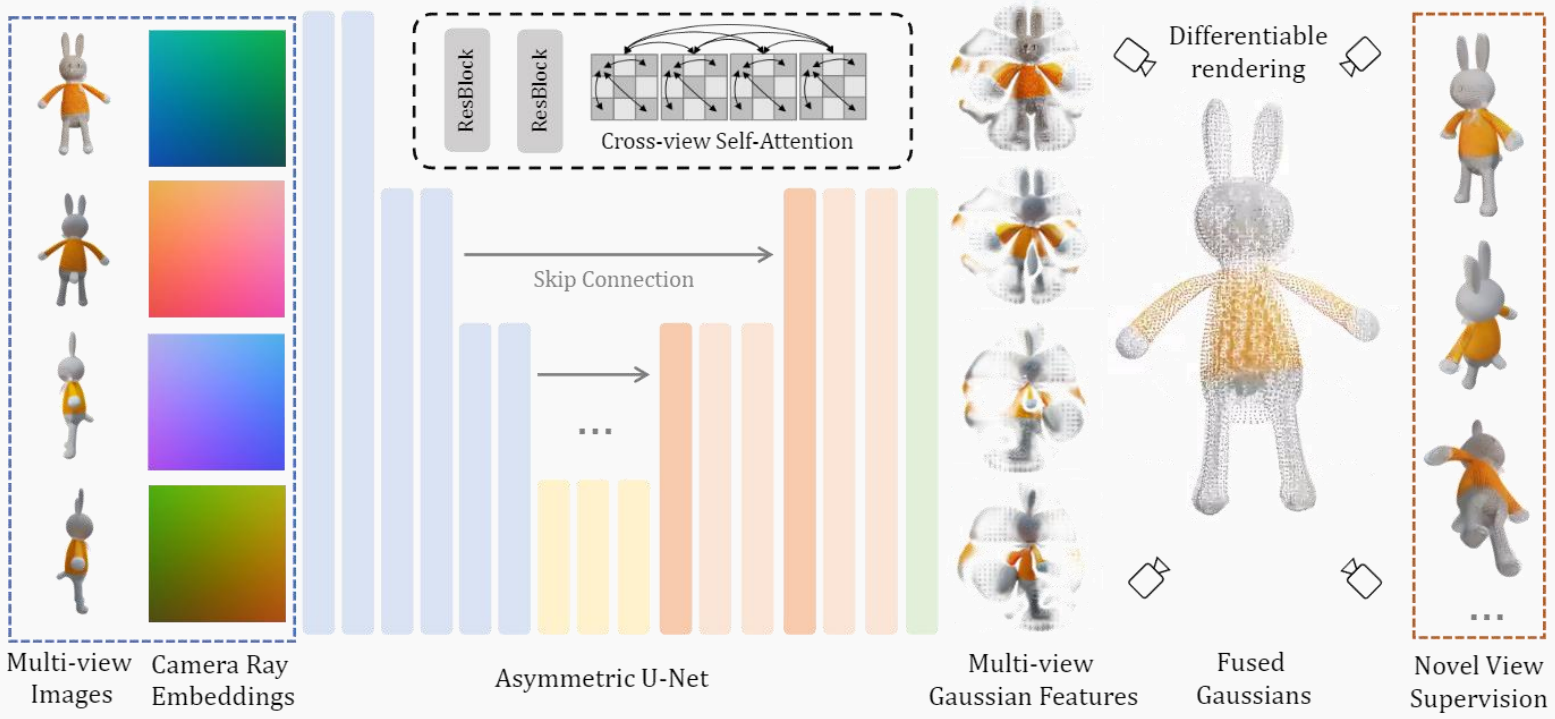
LGM　输入多视角图片，一次推理得到多视角高斯特征图

推理流程：

1. 输入多视角图片（拼接上camera ray embeddings）

2. 输出多视角高斯特征图，每个像素14维特征，代表一个高斯点的所有属性：均值，scale，rotation，opacity，RGB

3. 将所有视角的高斯点组合起来，构成整个场景



四. 基于SDS的三维生成优缺点以及其他三维生成方法

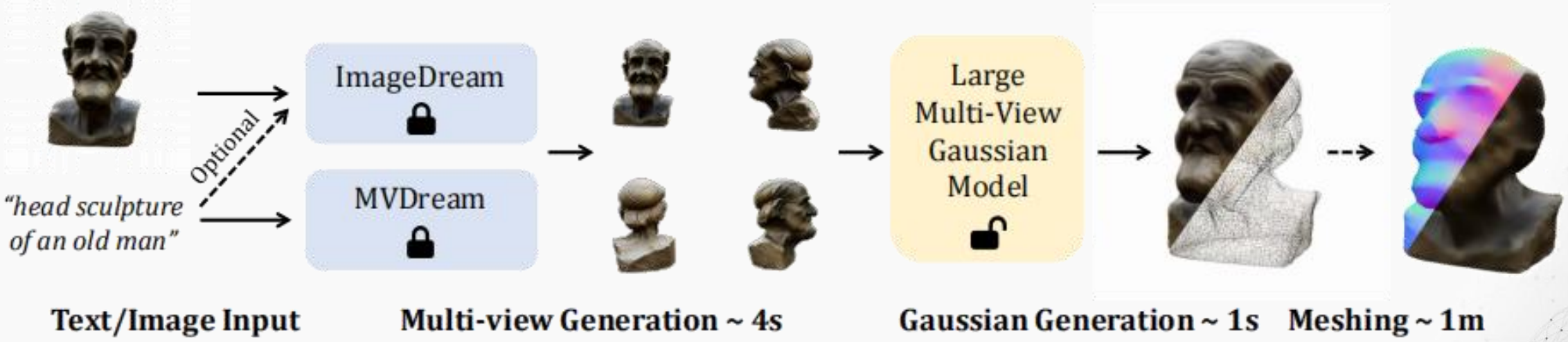
LGM　输入多视角图片，一次推理得到多视角高斯特征图

从文本/单张图像生成Mesh的流程：

1. 多视角二维扩撒模型根据文本/图像生成多视角图像

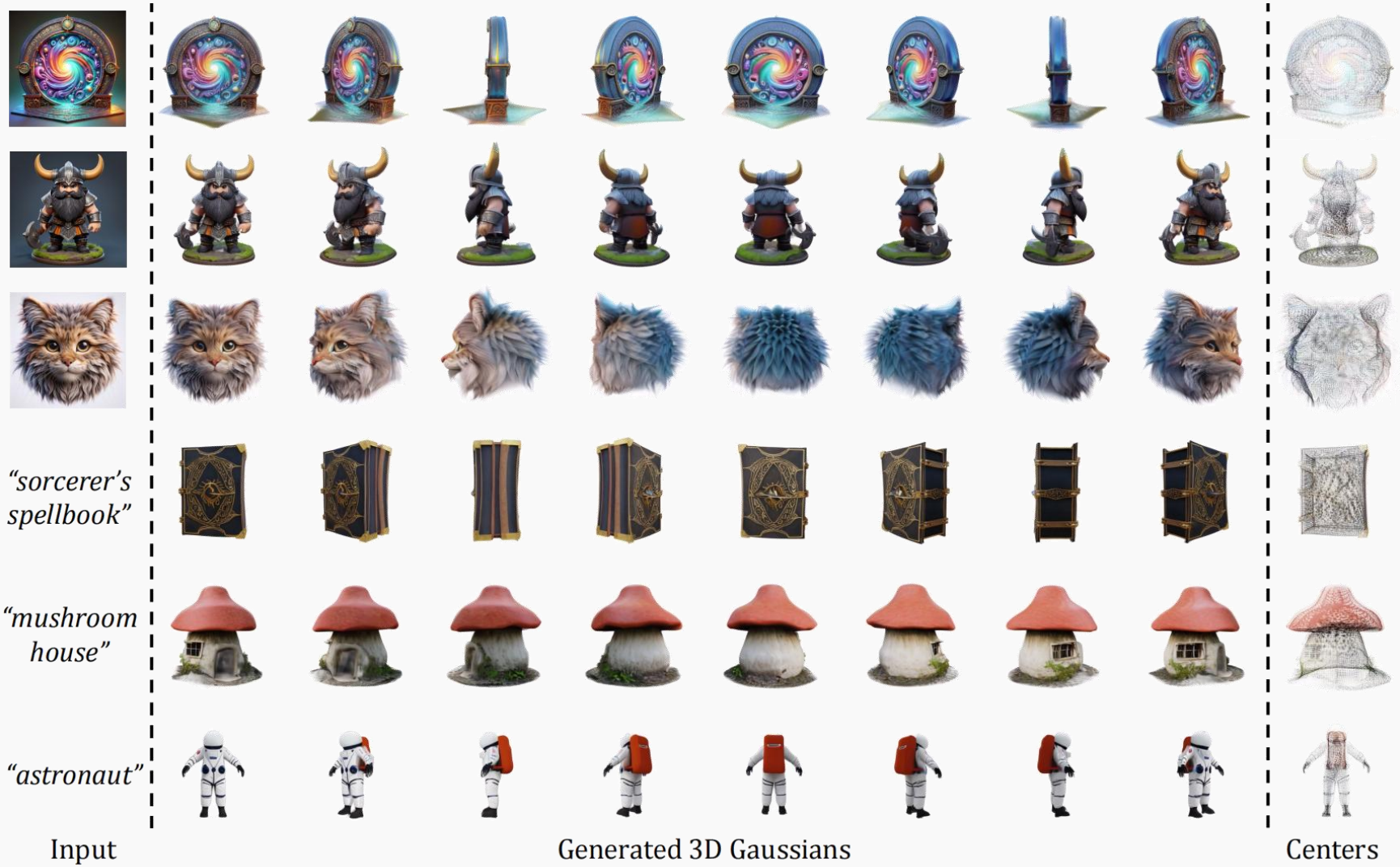
2. 多视角图像通过LGM生成表示场景的3DGS

3. 从3DGS中提取Mesh



四. 基于SDS的三维生成优缺点以及其他三维生成方法

LGM



谢谢

THANK YOU