Awesome Conditional 3D Generation

Tianhang Cheng

Update: 2023.11.29

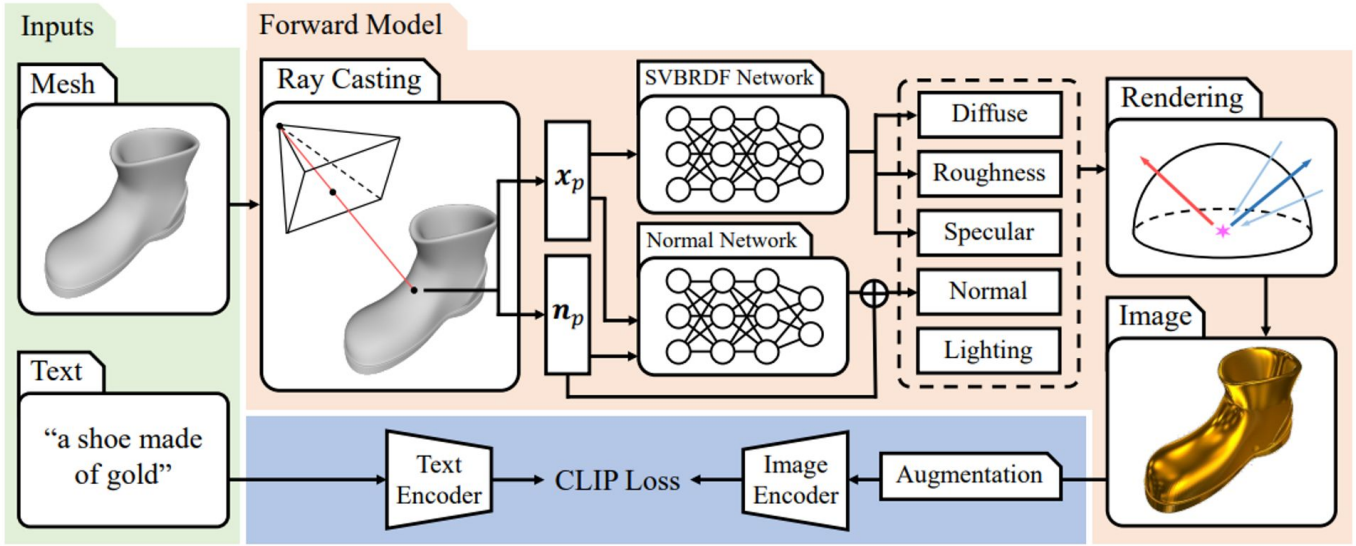
Category

https://github.com/yyeboah/Awesome-Text-to-3D

https://github.com/topics/text-to-3d

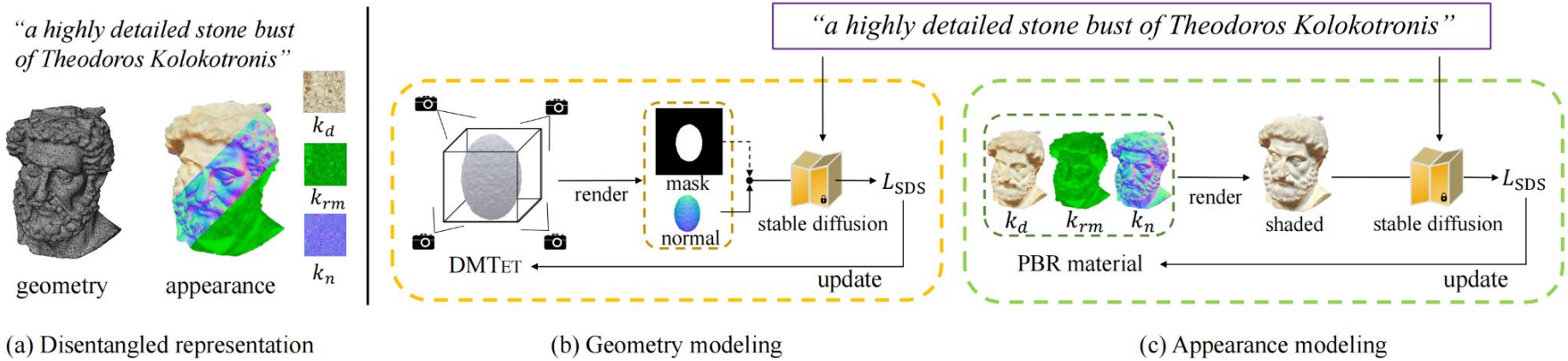
A. **Input**
   Text, Image, Mesh, DMTeT
B. **Output**
   3D model, PBR Material, styled output, segmentation
C. **Representation**
   Point, 3D Gaussian Splattings, Mesh, NeRF
D. **Optimization**
   Intermediate representation: multiview image/normal
   Diffusion: SDS, VSD, SJC, NFSD, ISM, etc.
   GAN: Descriminator
   Forward: Triplane, cost-volume
   Other: GTP+Blender, etc.

A.Input: Mesh



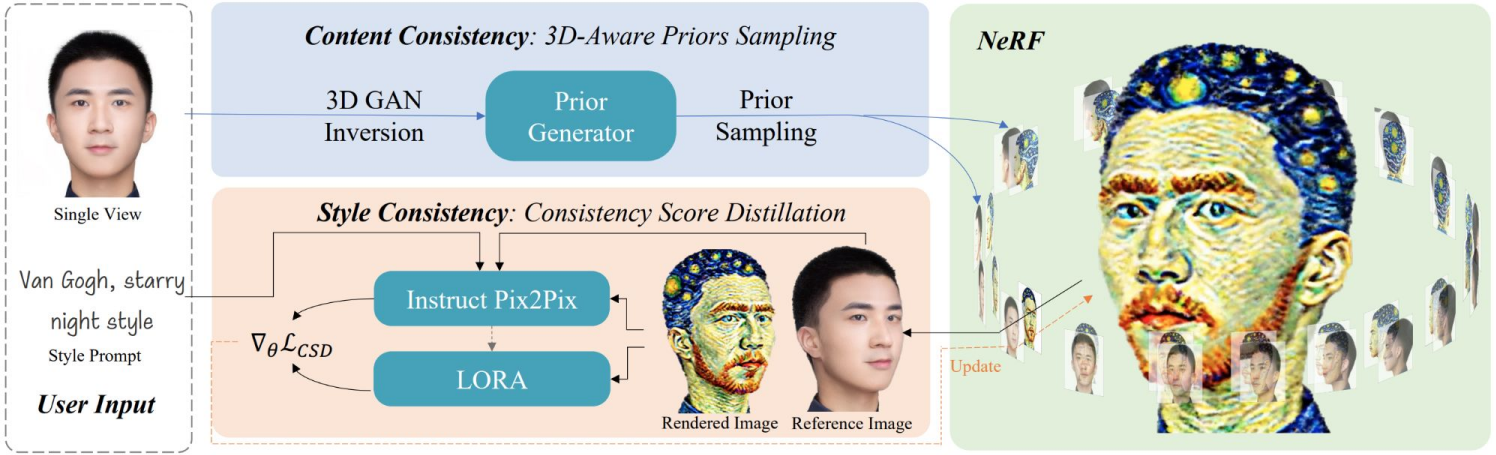
TANGO: Text-driven Photorealistic and Robust 3D Stylization via Lighting Decomposition

B.Output: PBR Material



Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation

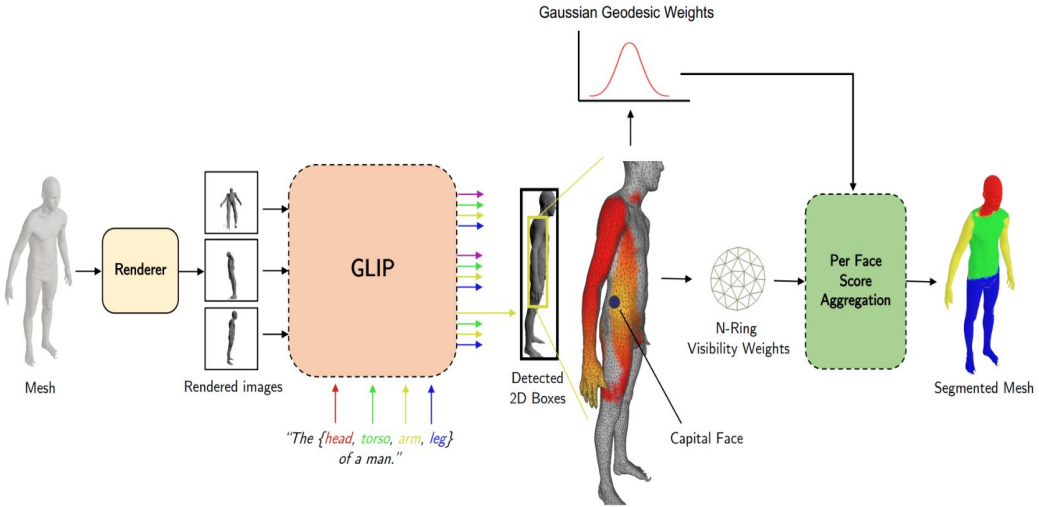
B.Output: styled output



STYLEDREAMER: MAKE YOUR 3D STYLE AVATAR FROM A SINGLE VIEW WITH CONSISTENCY SCORE DISTILLATION

B.Output: segmentation

zero-shot segmentation from mesh



SATR: Zero-Shot Semantic Segmentation of 3D Shapes

C. Representation: Overview

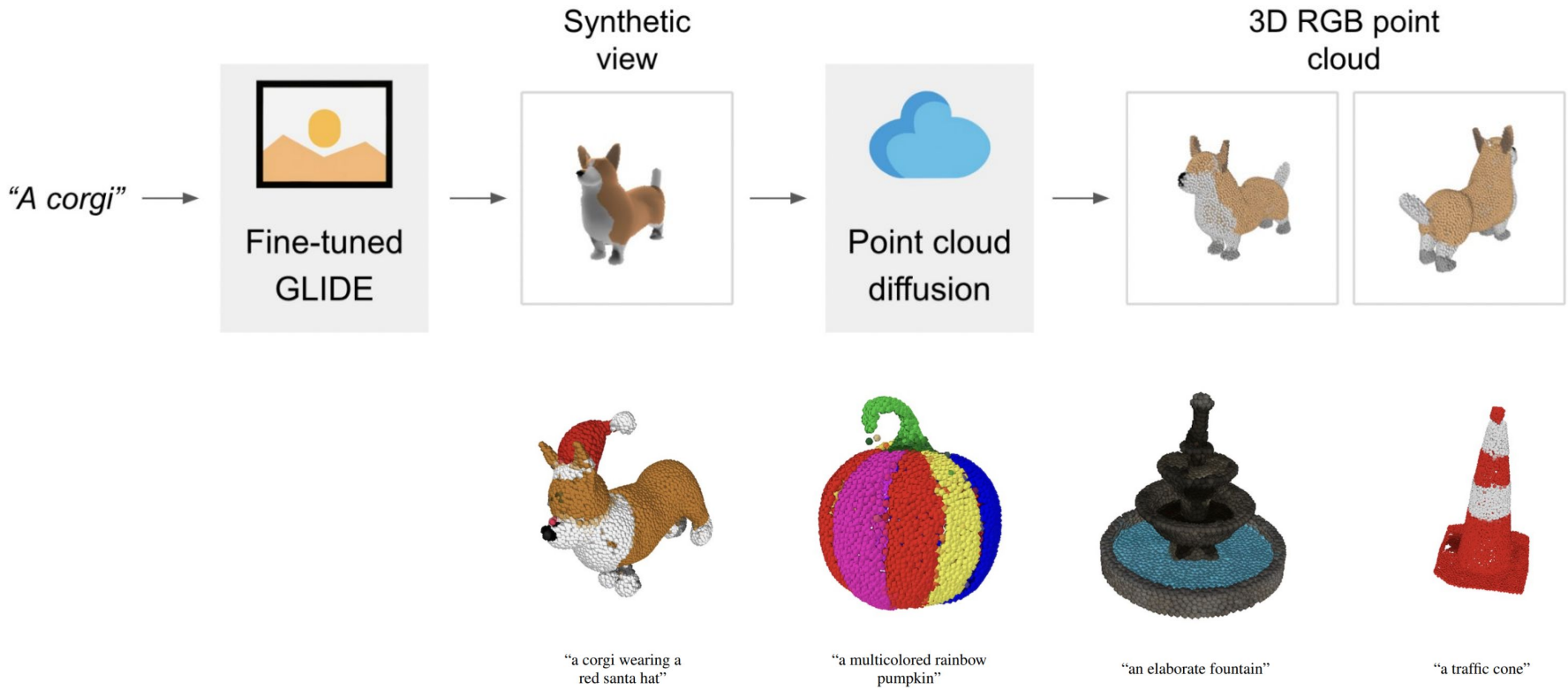
Point, 3D Gaussian Splattings, Mesh, NeRF, etc.

Table 1: Comparison of text-conditioned 3D generation methods on efficiency and applications.

Method	Latency	Device	Generation Category	Downstream Task
DreamFields [37]	1.2h	8×TPUv4	Multi-Category	Generation
DreamFusion [65]	1.5h	4×TPUv4	Multi-Category	Generation
CLIP-Mesh [58]	30min	V100	Multi-Category	Generation
CLIP-Sculptor [77]	0.9s	V100	Multi-Category	Generation
Point·E (40M) [59]	25s	V100	Multi-Category	Generation
ShapeCrafter [24]	-	-	Single-Category	Generation, Editing
LION [99]	27s	V100	Single-Category	Generation, Completion
SDFusion [9]	-	-	Single-Category	Generation, Completion

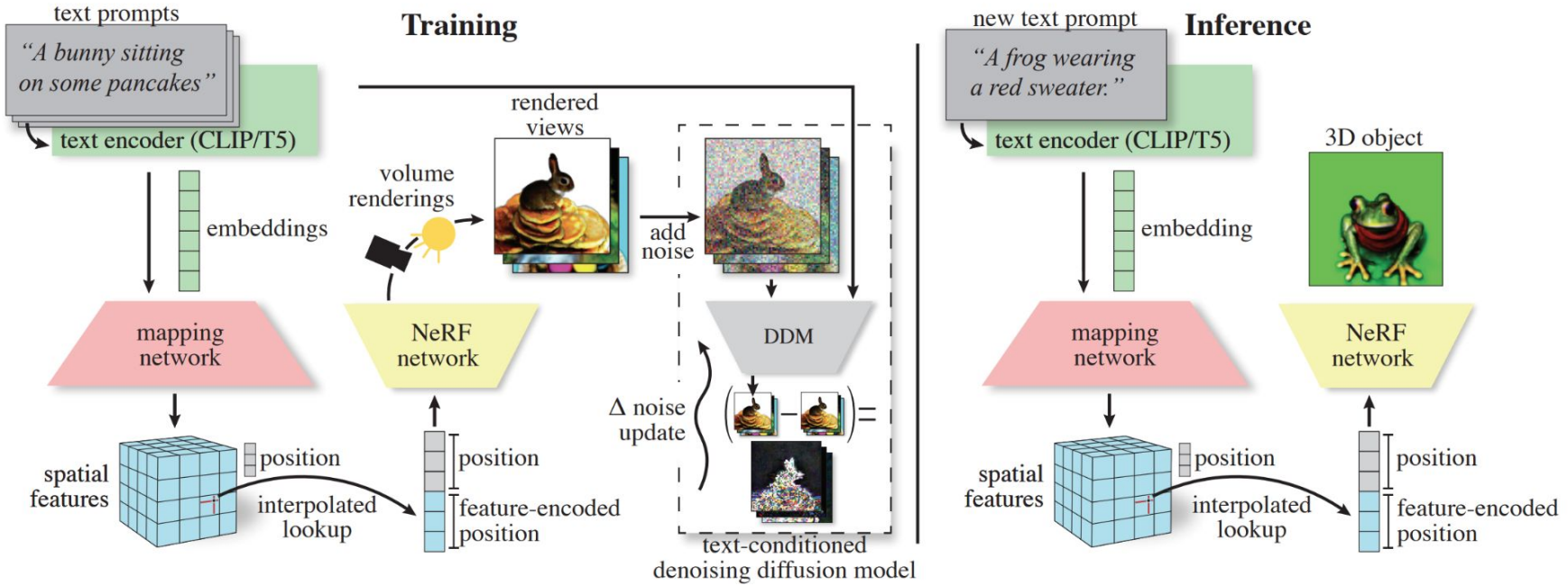
figure from *VPP : Efficient Conditional 3D Generation via Voxel-Point Progressive Representation*

C.Representation: Point



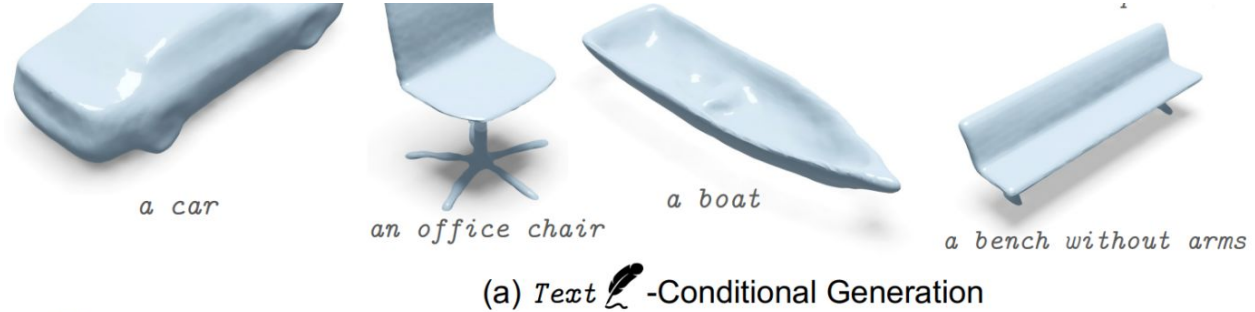
Point·E: A System for Generating 3D Point Clouds from Complex Prompts

C.Representation: Voxel NeRF

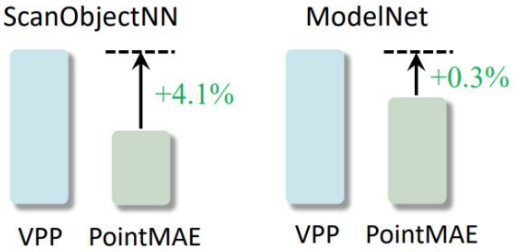


ATT3D: Amortized Text-to-3D Object Synthesis

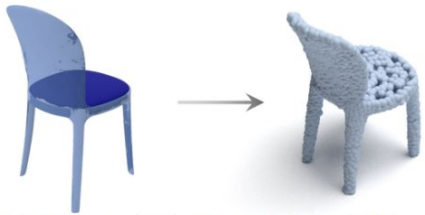
C.Representation: Voxel & Point



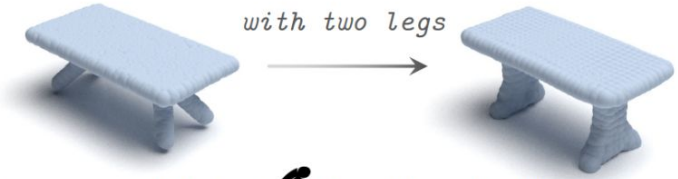
(a) Text -Conditional Generation



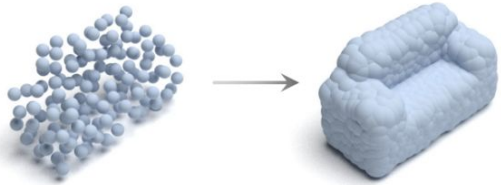
(b) Pretraining



(c) Image -Conditional Generation



(d) Text -Conditional Editing



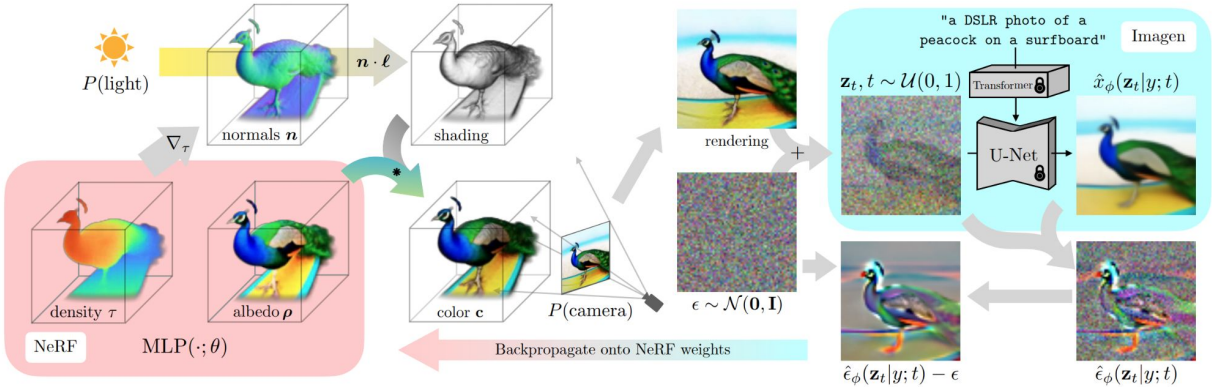
(e) Point Cloud Upsampling

VPP : Efficient Conditional 3D Generation via Voxel-Point Progressive Representation

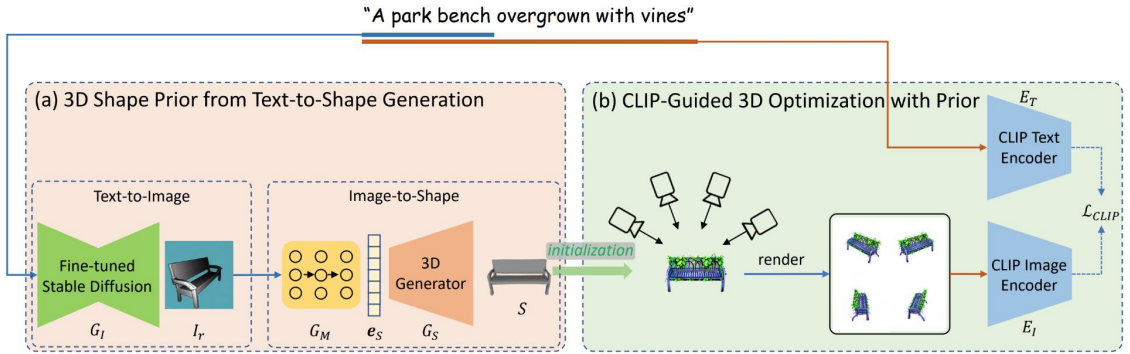
C.Representation: NeRF

NeRF (volume render)

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

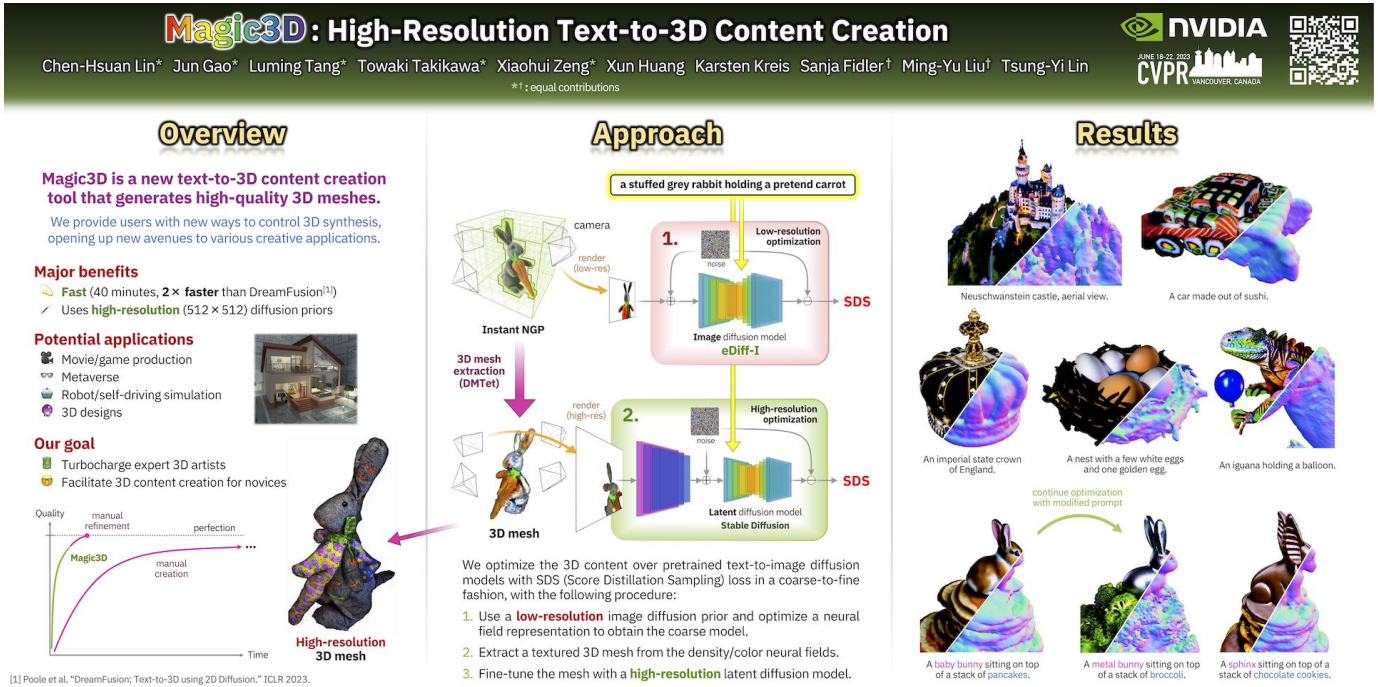


Dream3D: Zero-Shot Text-to-3D Synthesis Using 3D Shape Prior and Text-to-Image Diffusion Models



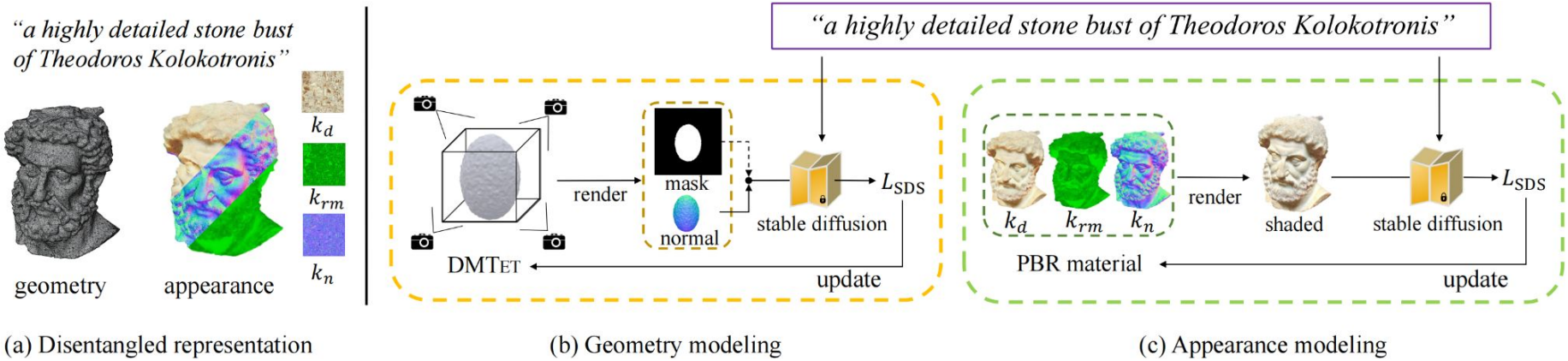
C.Representation: NeRF

NeRF (volume render) + instant ngp



Magic3D: High-Resolution Text-to-3D Content Creation

C.Representation: DMTeT

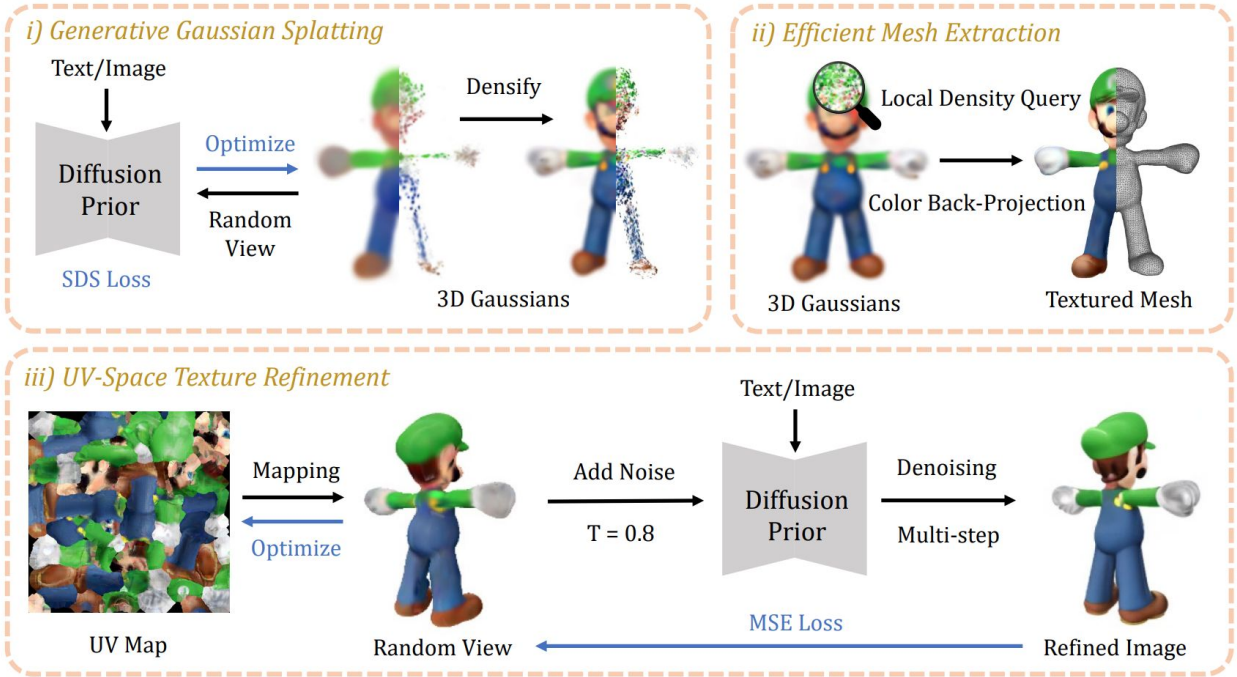


(a) Disentangled representation (b) Geometry modeling (c) Appearance modeling

Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation

C.Representation: Mesh

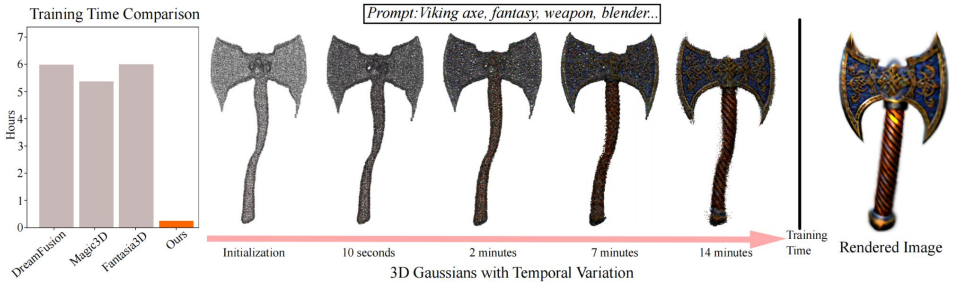
3dgs as mesh initilization

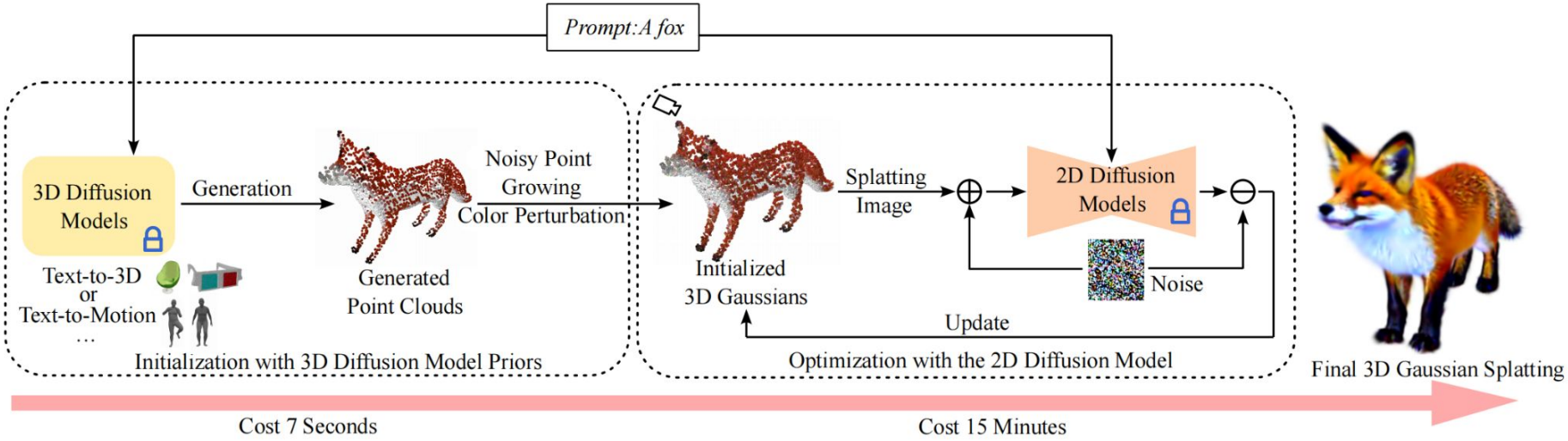


DREAMGAUSSIAN: GENERATIVE GAUSSIAN SPLATTING FOR EFFICIENT 3D CONTENT CREATION

C.Representation: 3DGS

promt -> point cloud, optimized by 2D diffusion

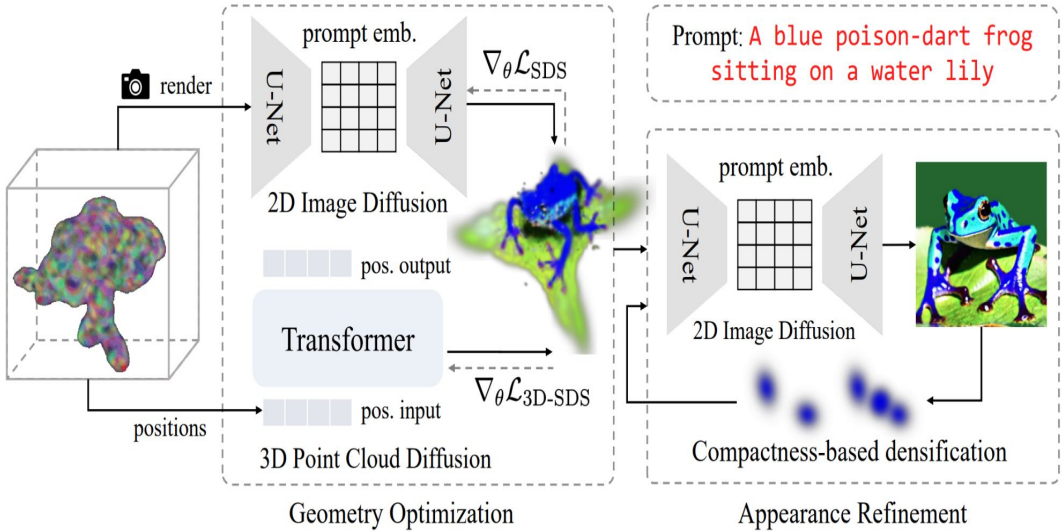


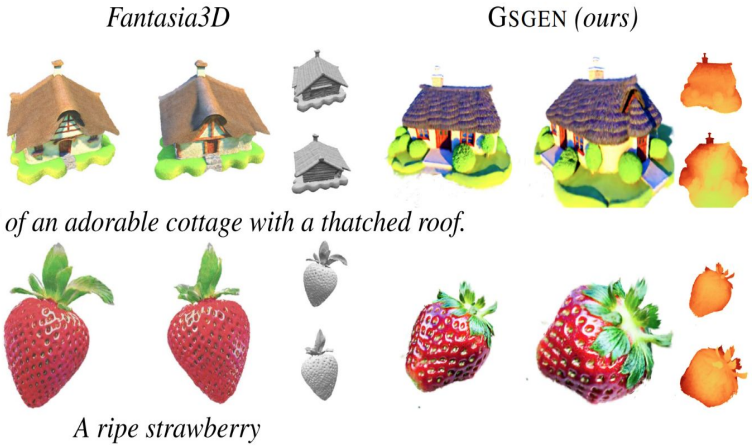


GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models

C.Representation: 3DGS

3DGS + 3D SDS Loss





TEXT-TO-3D USING GAUSSIAN SPLATTING

D.Optimization: Overview

- Multiview image/surface normal as intermediate representation

  Use (and finetune) pretrained generative model to generate multiview information from input; Then transfer multiview to 3d model (usually with post-processing if images are not perfect, like 3d conv)

- SDS, VSD, SJC, NFSD, ISM, GAN, etc.

  Use pretrained generative model to implicity modify the probability / adversarial training

- Forward model (reference is fast)

  Generate triplane, cost-volume, 3D U-Net

D.Optimization: Diffusion: SJC

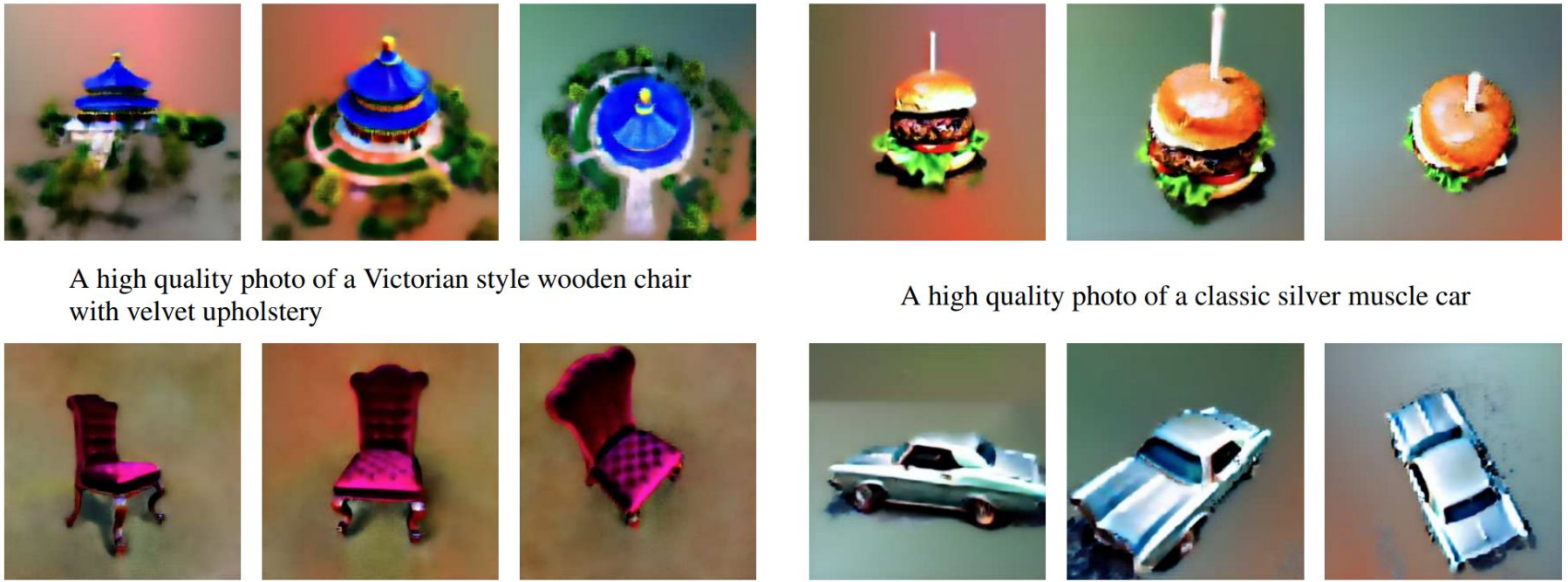
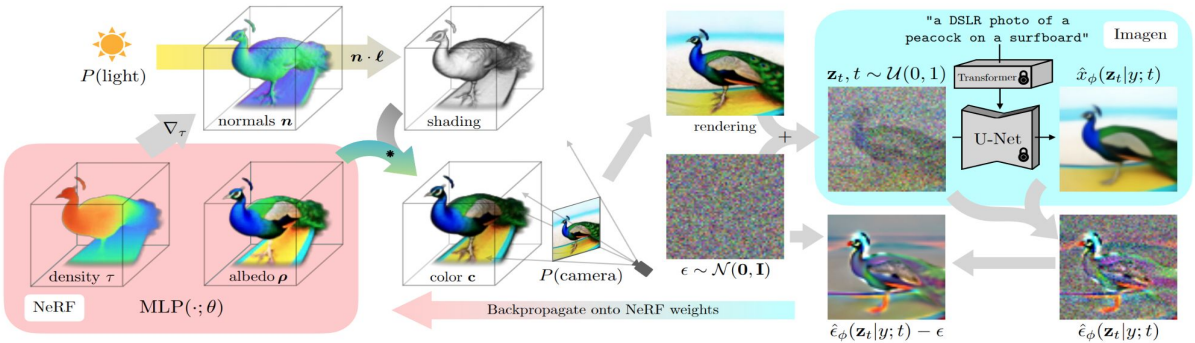


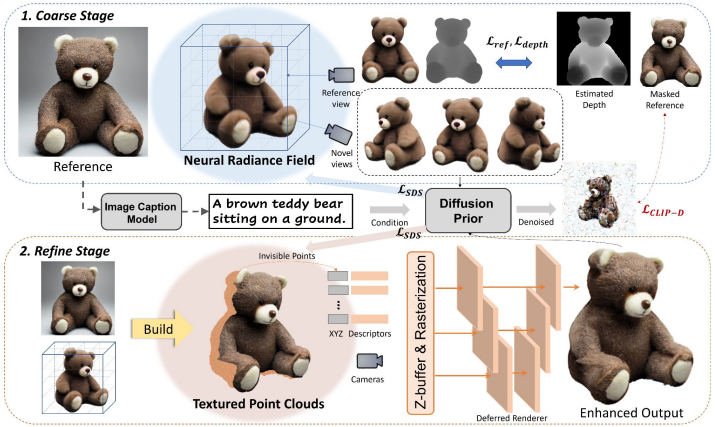
Figure 1. Results for text-driven 3D generation using Score Jacobian Chaining with Stable Diffusion as the pretrained model.

Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation

D.Optimization: Diffusion: SDS



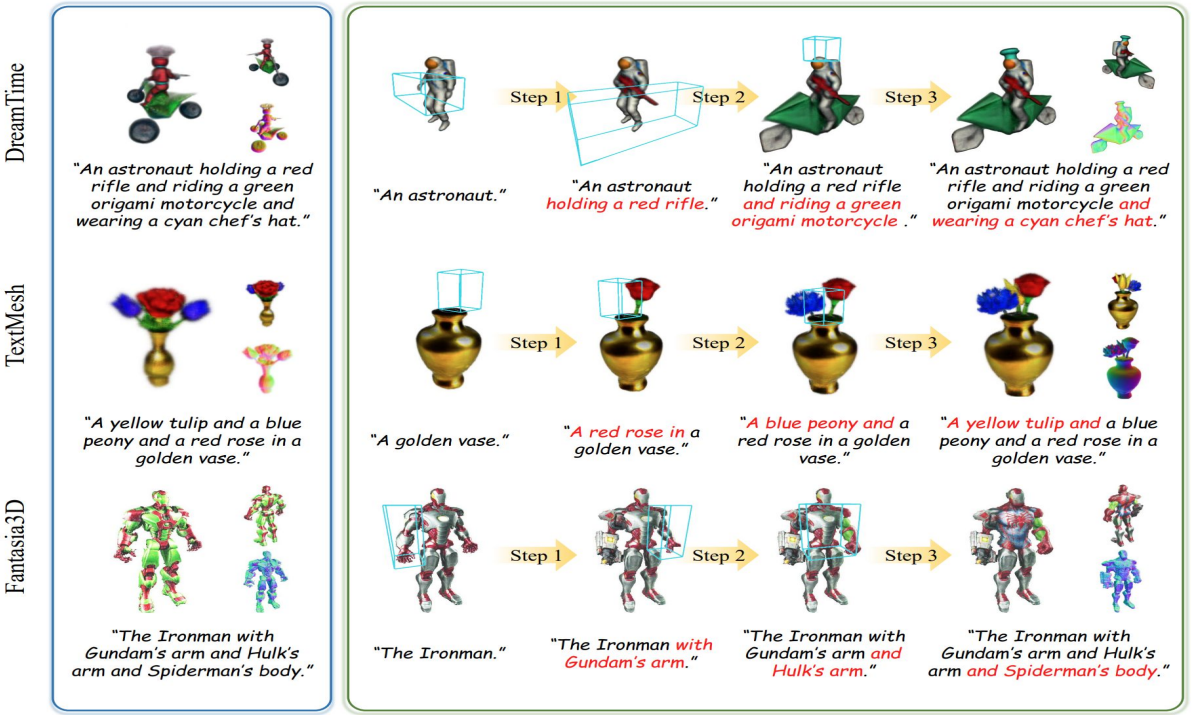
DREAMFUSION: TEXT-TO-3D USING 2D DIFFUSION



depth, clip, sds as supervision. Not perform very well on back face

Make-It-3D: High-Fidelity 3D Creation from A Single Image with Diffusion Prior

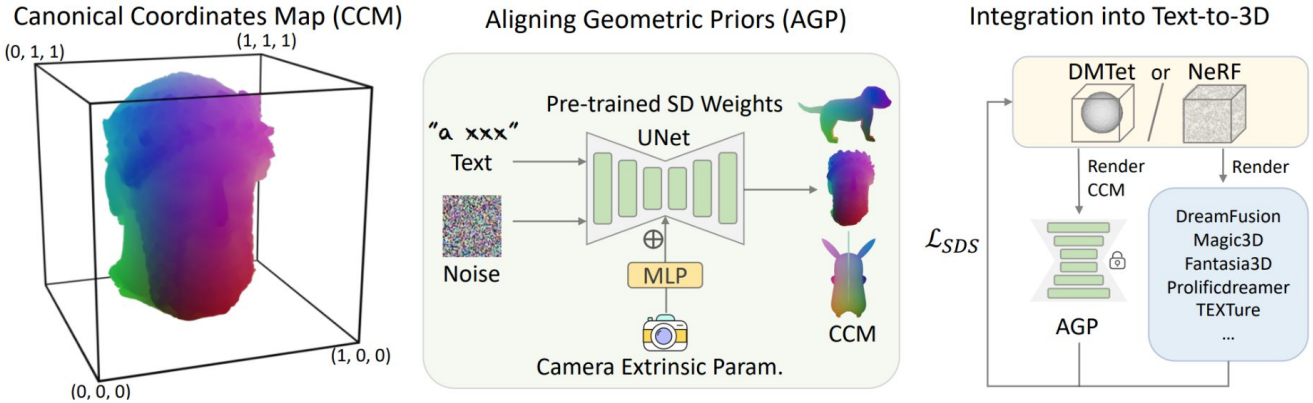
D.Optimization: Diffusion: (progressive) SDS



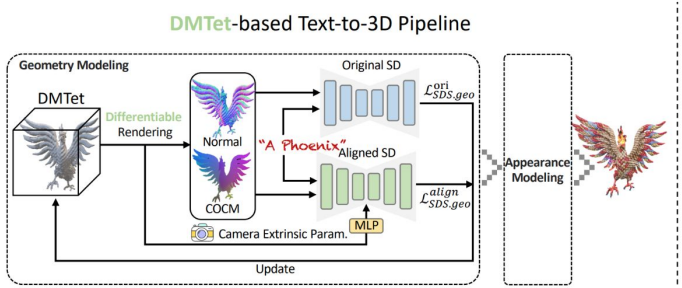
(a) Generating Directly (b) Generating with Progressive3D

PROGRESSIVE3D: PROGRESSIVELY LOCAL EDITING FOR TEXT-TO-3D CONTENT CREATION WITH COMPLEX SEMANTIC PROMPTS

D.Optimization: Diffusion: (aligned) SDS



align 2D diffusion prior by output CCM



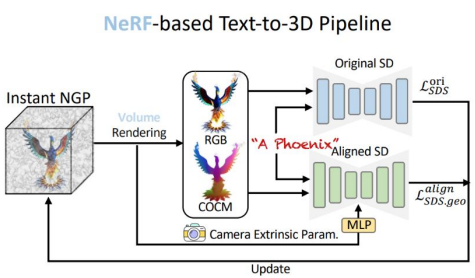
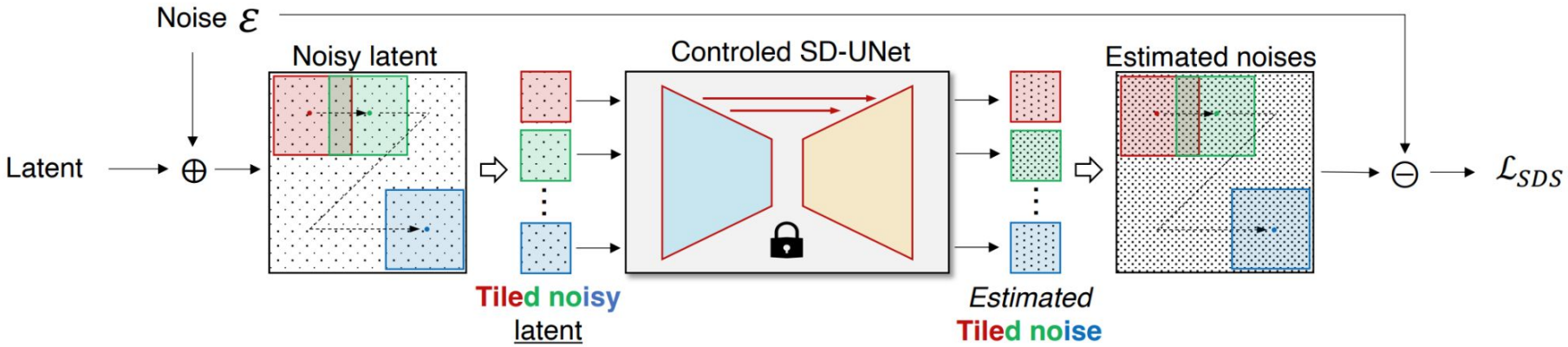


Figure 3: Seamless integration of our AGP in various text-to-3D pipelines.

Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d

D.Optimization: Diffusion: (multi-patch) SDS



multi-patch reduce unaligned problem of 2d prior

like the relation between *Triplane* vs *Sparse Tri-Vector Radiance Fields*

HD-Fusion: Detailed Text-to-3D Generation Leveraging Multiple Noise

D.Optimization: Diffusion: (constrain) SDS

constrain the semantic, geometry and saturation inconsistency in SDS

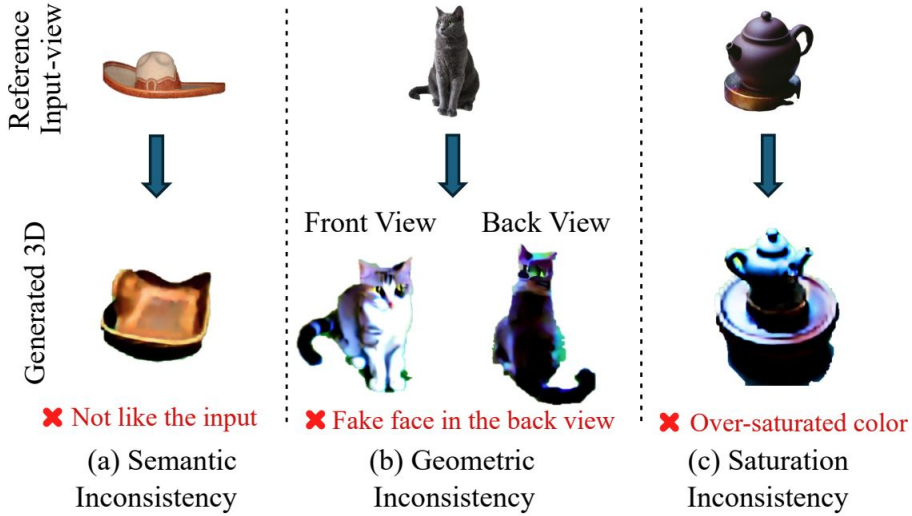
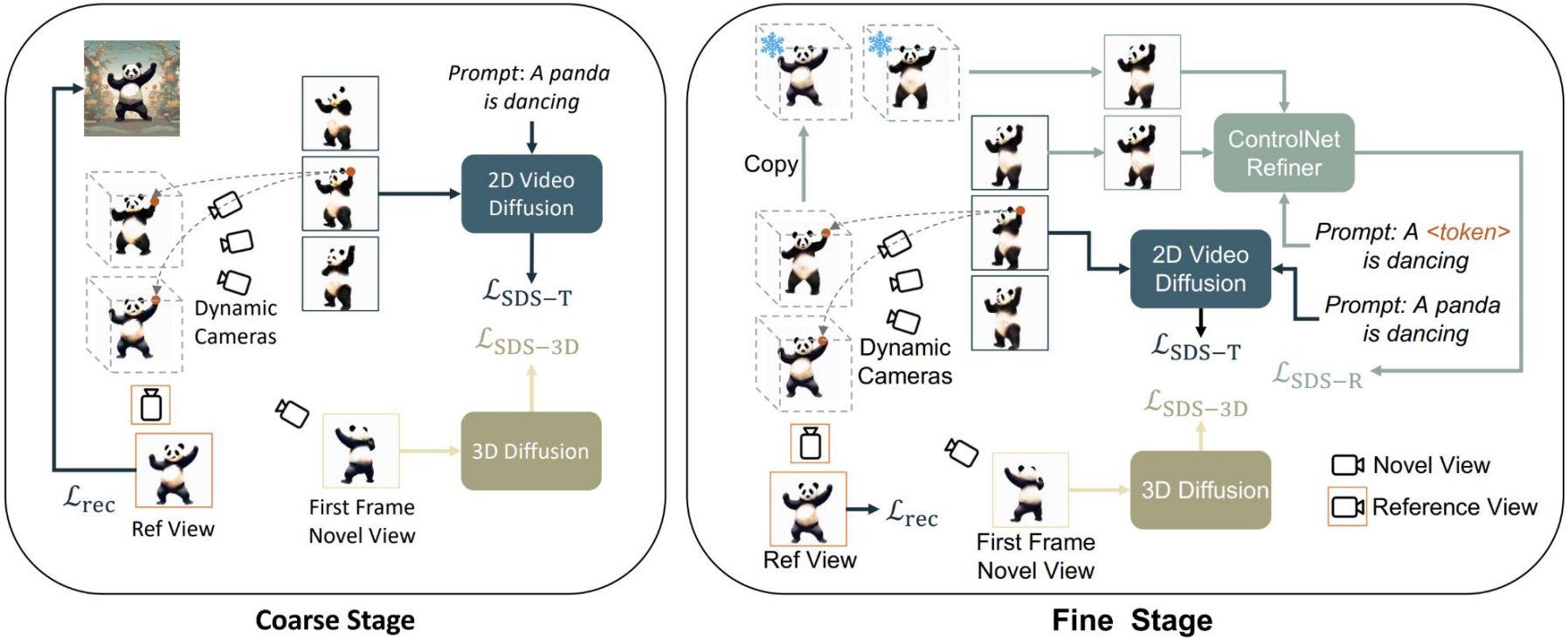


Figure 1: **Inconsistency issues.** (a) The semantic inconsis-

Chasing Consistency in Text-to-3D Generation from a Single Image

D.Optimization: Diffusion: multiple diffusion

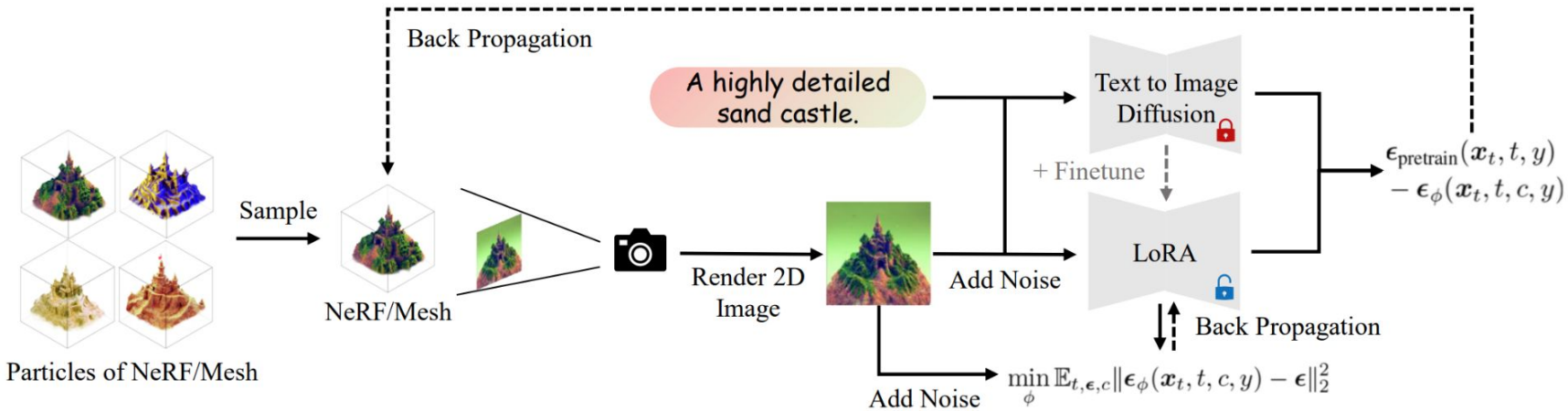
4D



Animate124: Animating One Image to 4D Dynamic Scene

D.Optimization: Diffusion: VDS

model variant probability by training several nerf “particle” together



DREAMGAUSSIAN: GENERATIVE GAUSSIAN SPLATTING FOR EFFICIENT 3D CONTENT CREATION

D.Optimization: Diffusion: NFSD

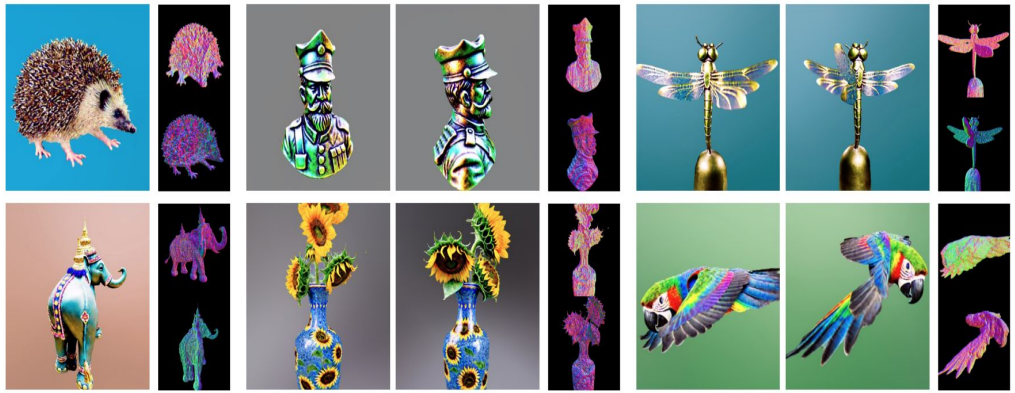


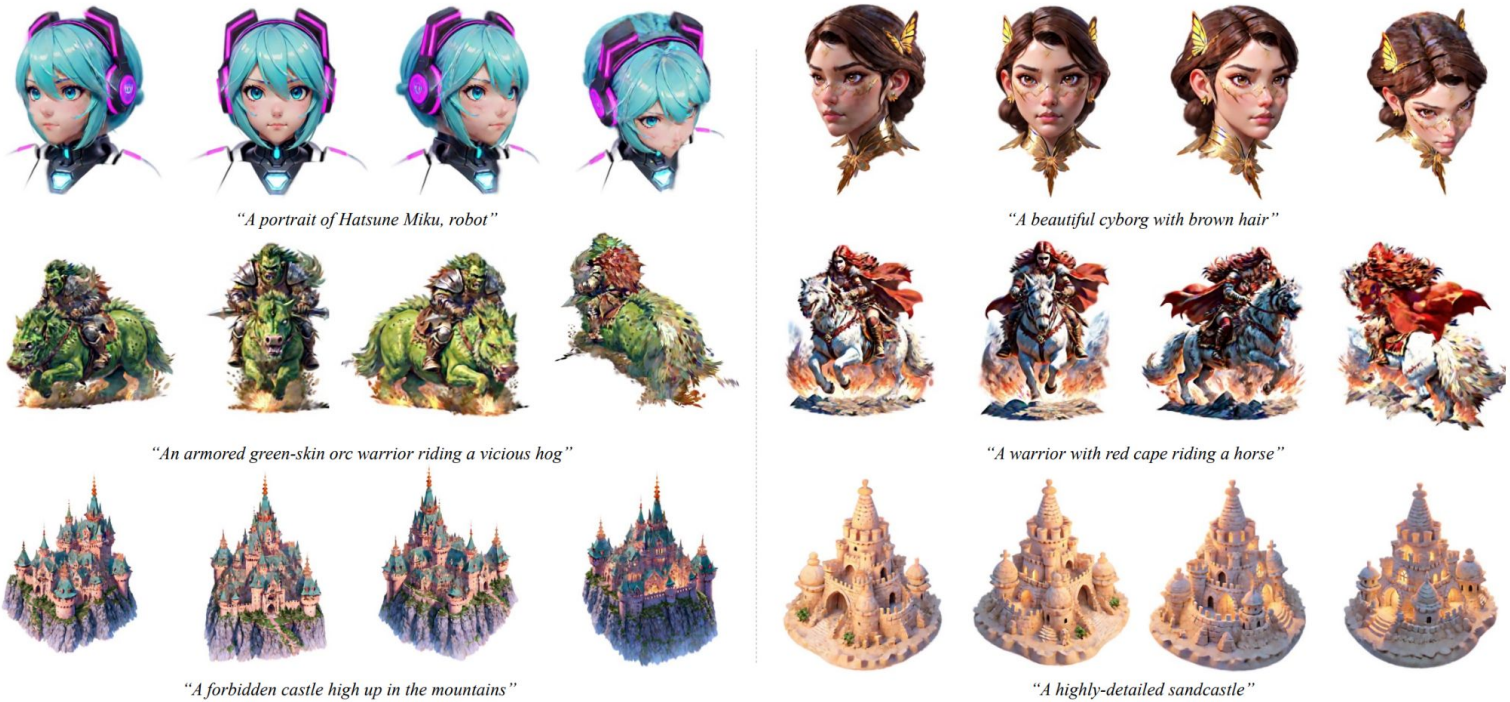
Figure 6: NeRFs optimized with NFSD.

: 2D image generation with SDS and NFSD. We

NOISE-FREE SCORE DISTILLATION

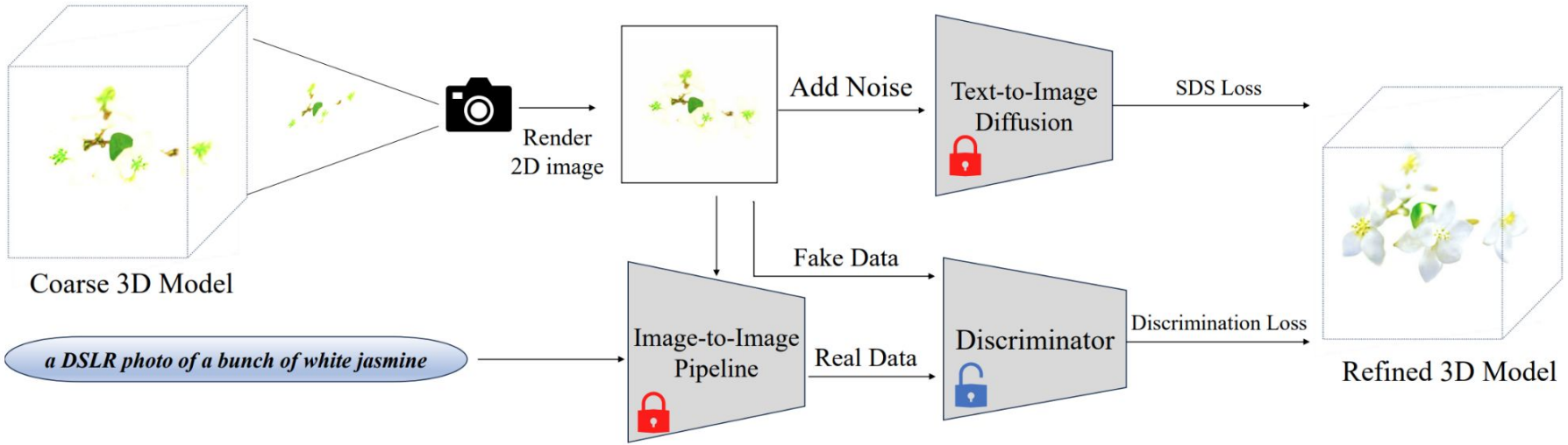
D.Optimization: Diffusion: ISM

Interval Score Matching



LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching

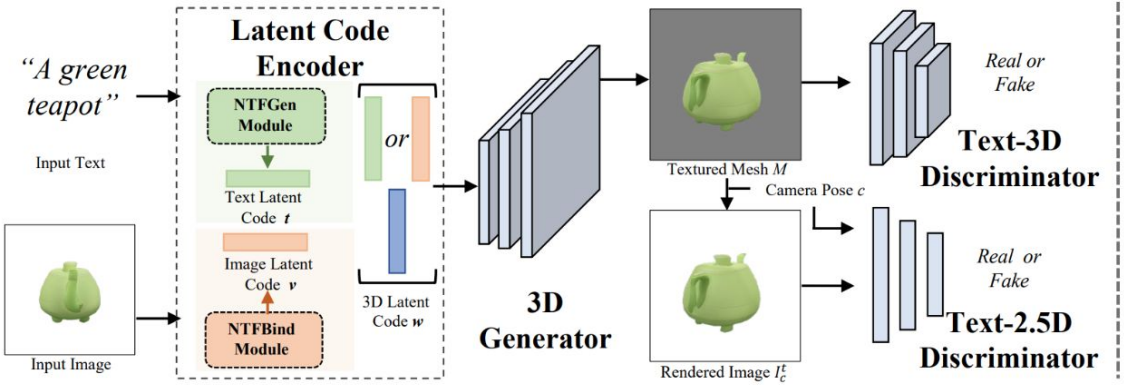
D.Optimization: GAN: Descriminator

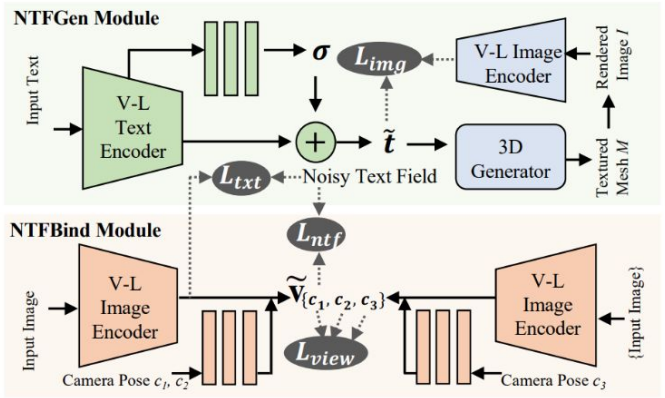


IT3D: Improved Text-to-3D Generation with Explicit View Synthesis

D.Optimization: GAN: (multi-modal) Descriminator

text3D discriminator and a text-2.5D discriminator





IT3D: Improved Text-to-3D Generation with Explicit View Synthesis

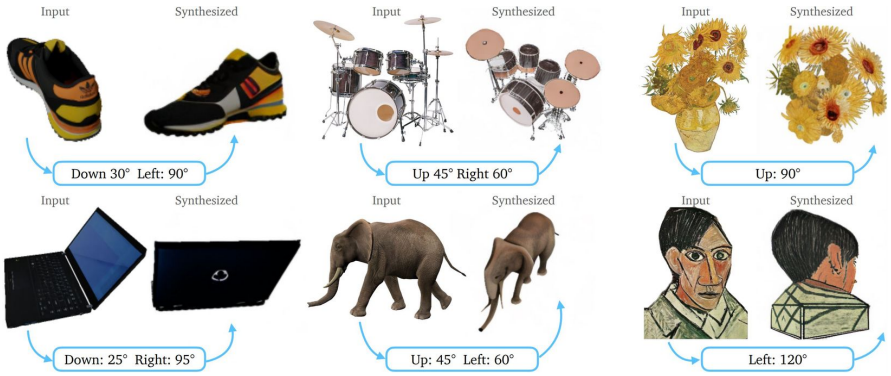
(a) Text-to-3D

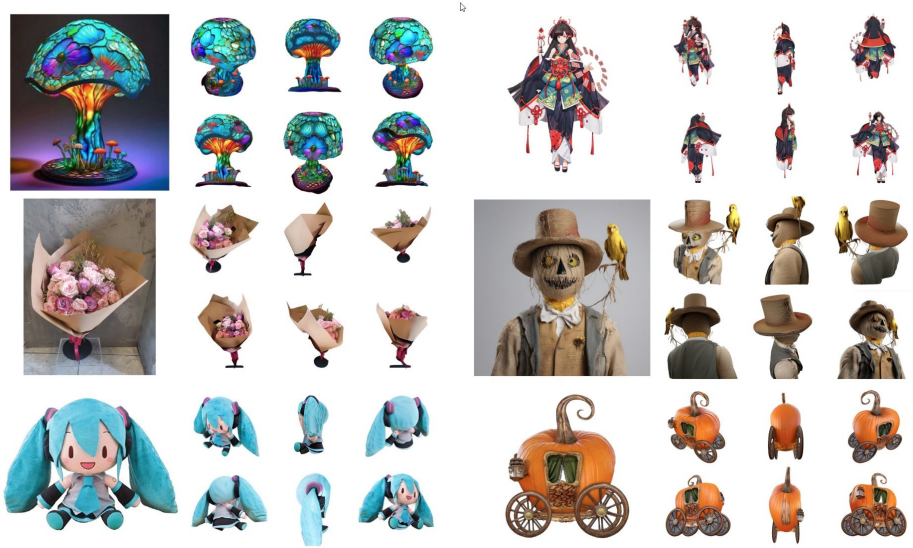
(b) Image-to-3D

D.Optimization: Intermediate representation: multiview image

finetune 2D diffusion model to get a pose-dependent model

finetune 2D diffusion model to generate more consistent multiview image





Zero-1-to-3: Zero-shot One Image to 3D Object

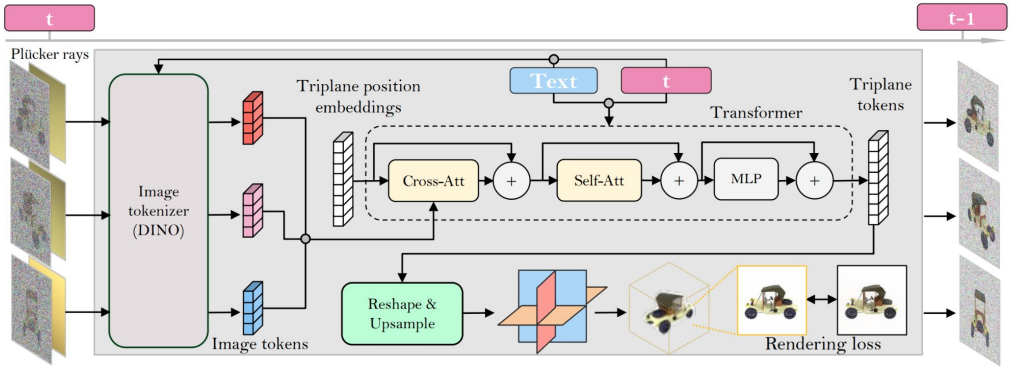
Zero123++: a Single Image to Consistent Multi-view Diffusion Base Model

D.Optimization: Intermediate representation: multiview image

finetune 2D diffusion model to get a multiview RGB images

https://mv-dream.github.io/index.html

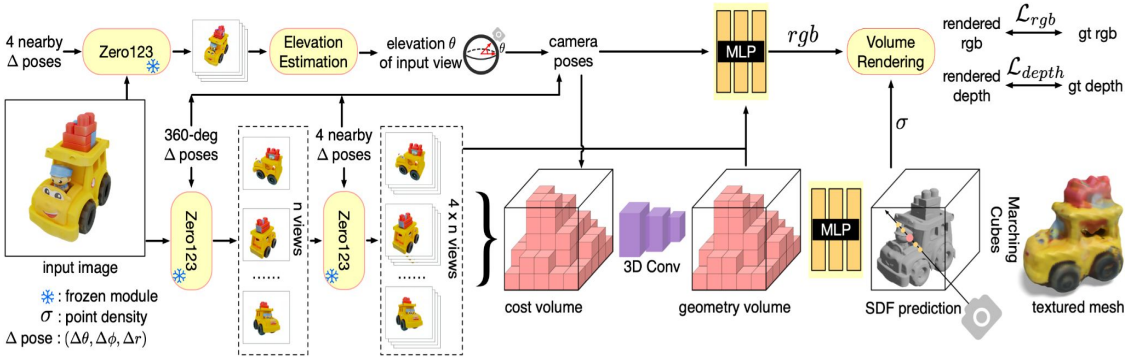
MVDream: Multi-view Diffusion for 3D Generation



DMV3D: DENOISING MULTI-VIEW DIFFUSION USING 3D LARGE RECONSTRUCTION MODEL

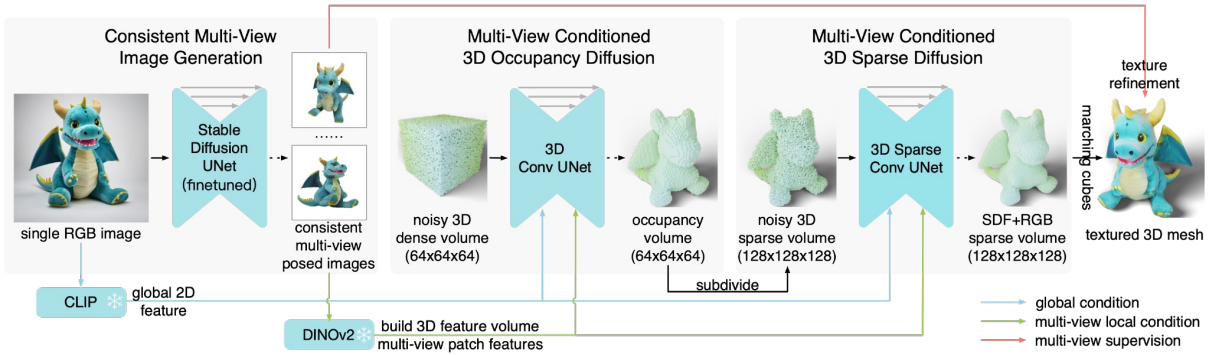
D.Optimization: Intermediate representation: multiview image

add cost-volume based on zero123 to mitigate the in-consistency



One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization

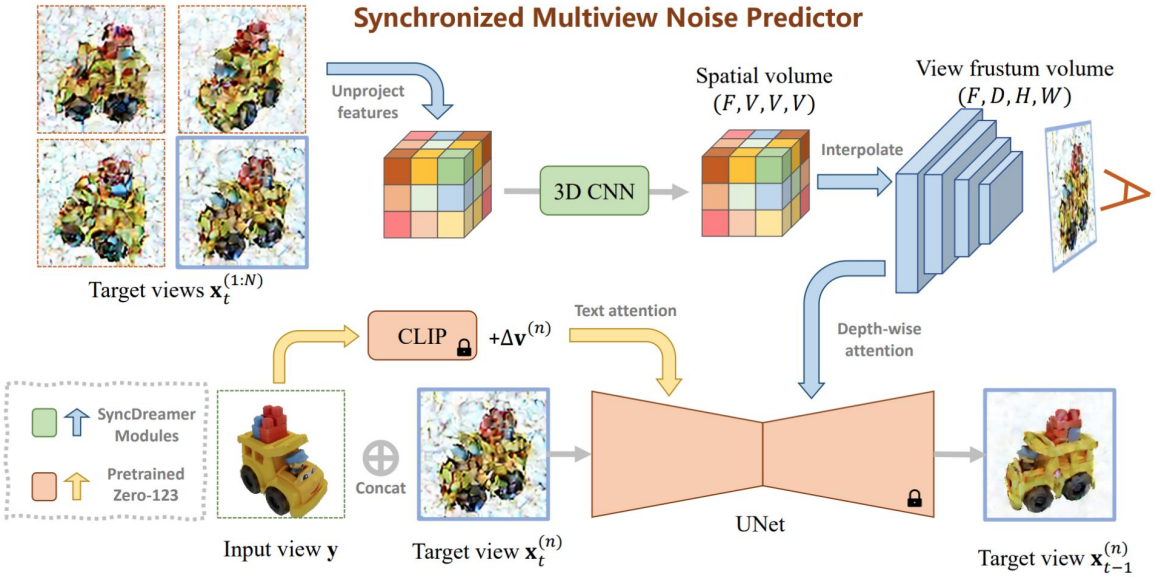
use better diffusion model and feature volume



One-2-3-45++: Fast Single Image to 3D Objects with Consistent Multi-View Generation and 3D Diffusion

D.Optimization: Intermediate representation: multiview image

constrain multi-view feature by unprojected to a volume

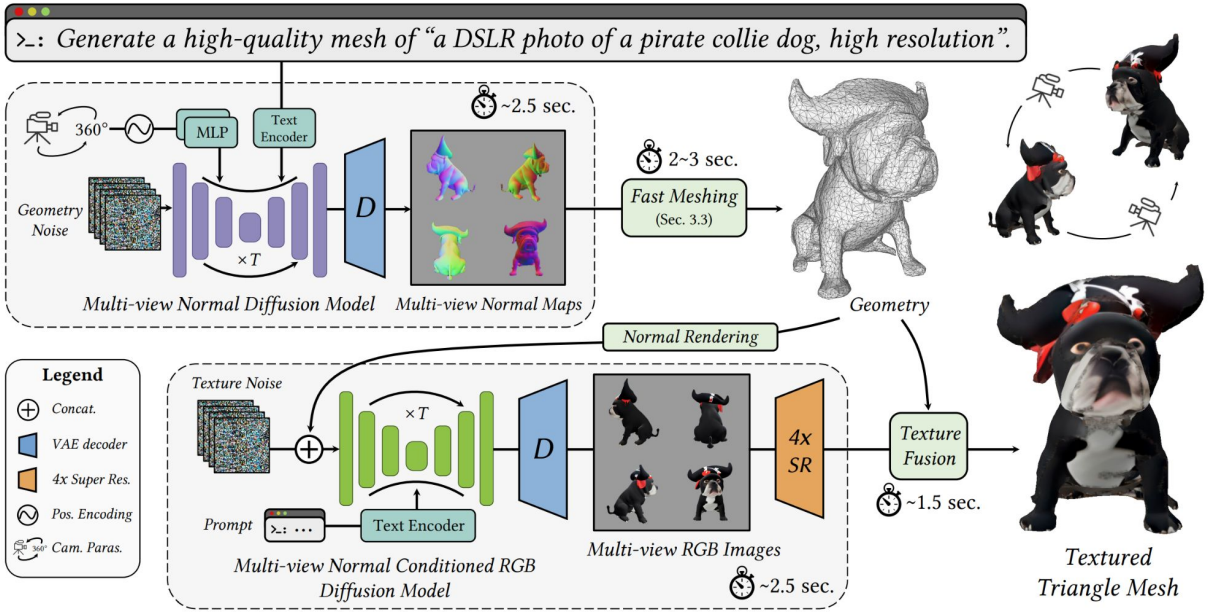


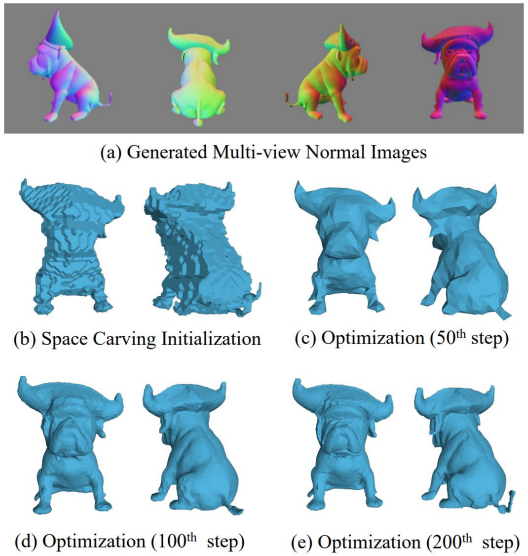
SyncDreamer: Generating Multiview-consistent Images from a Single-view Image

D.Optimization: Intermediate representation: multiview normal

finetune 2D diffusion model to get multiview normal images

TL;DR: Diverse, Janus-free, and high-fidelity 3D content generation in only 10 seconds.

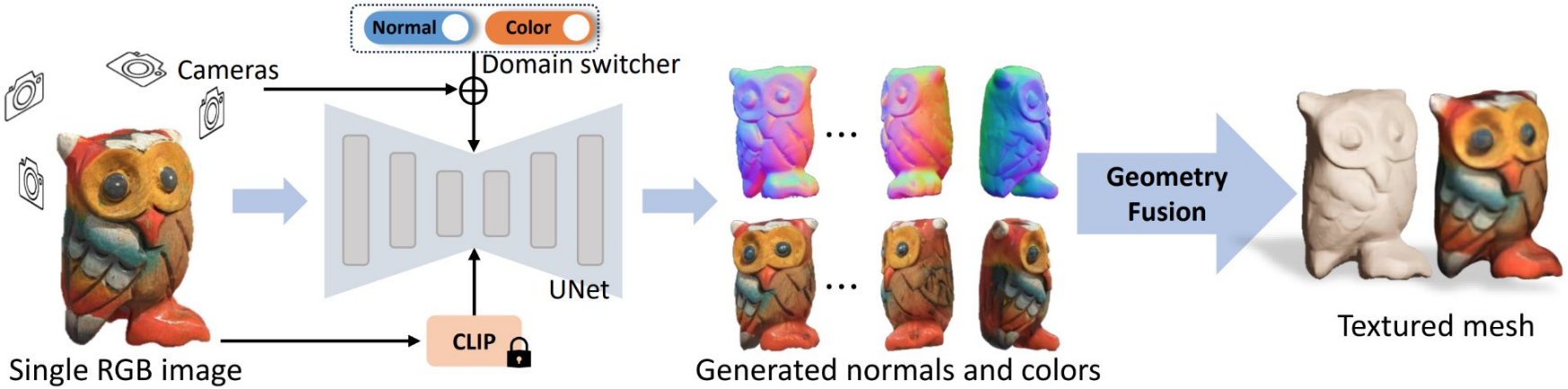




Direct2.5: Diverse Text-to-3D Generation via Multi-view 2.5D Diffusion

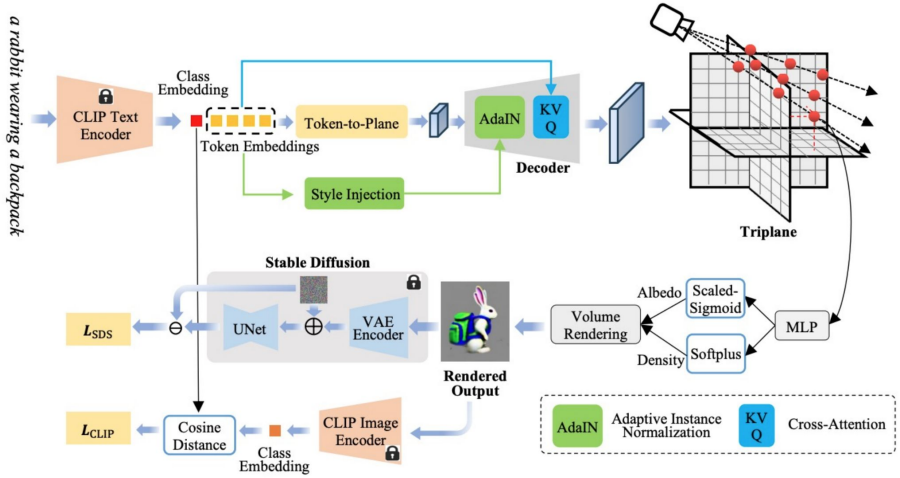
https://nju-3dv.github.io/projects/direct25/

D.Optimization: Intermediate representation: multiview image & normal



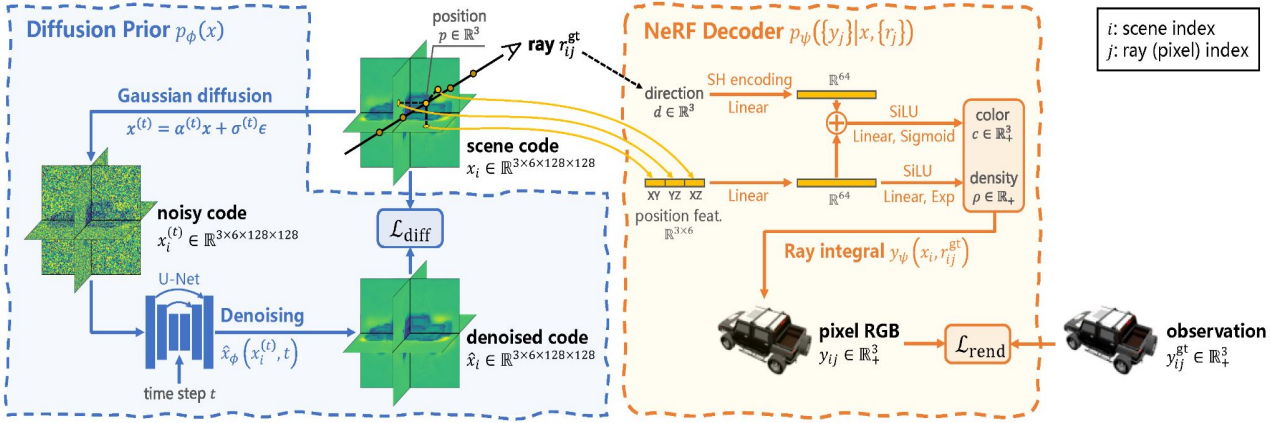
Wonder3D: Single Image to 3D using Cross-Domain Diffusion

D.Optimization: Forward model: triplane



Text-to-3D generation without per-prompt training, taking only **under a second**.
But the quality is not very realistic

Instant3D: Instant Text-to-3D Generation

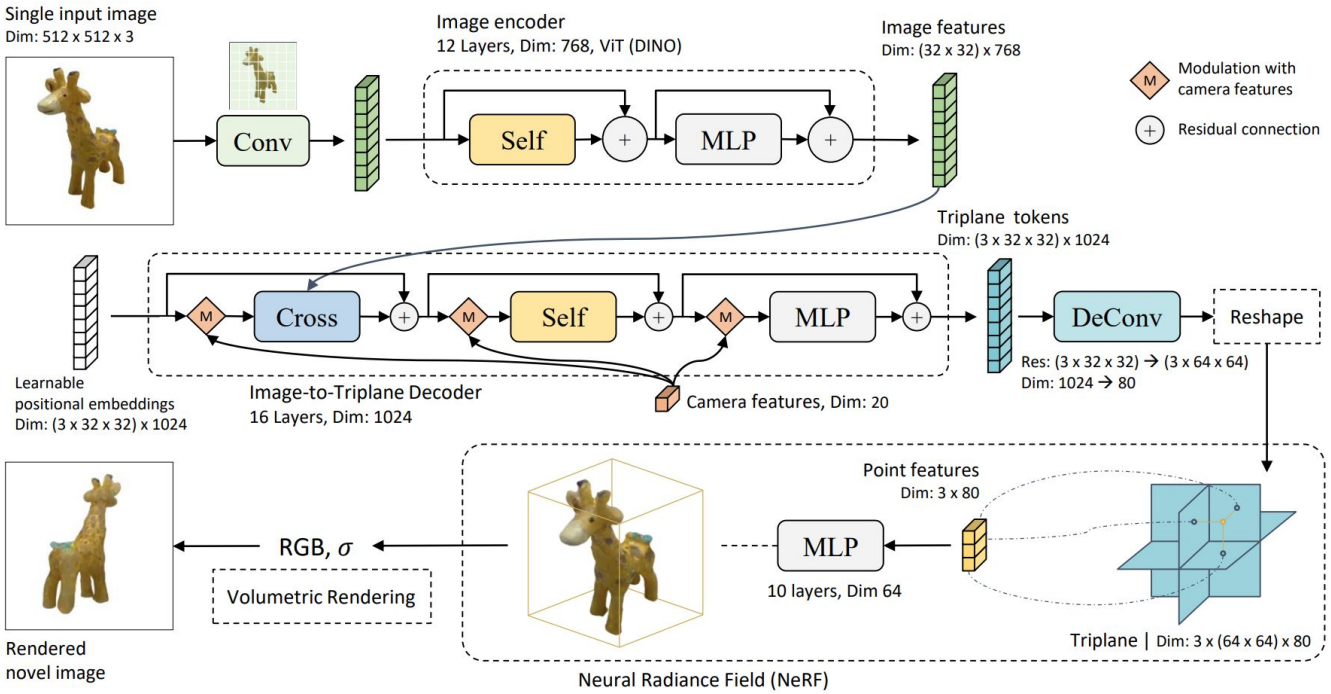


i: scene index
j: ray (pixel) index

Under the unconditional generation setting (50 DDIM steps), sampling a batch of 8 scenes takes **4.63 sec** on a single RTX 3090 GPU.

Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction

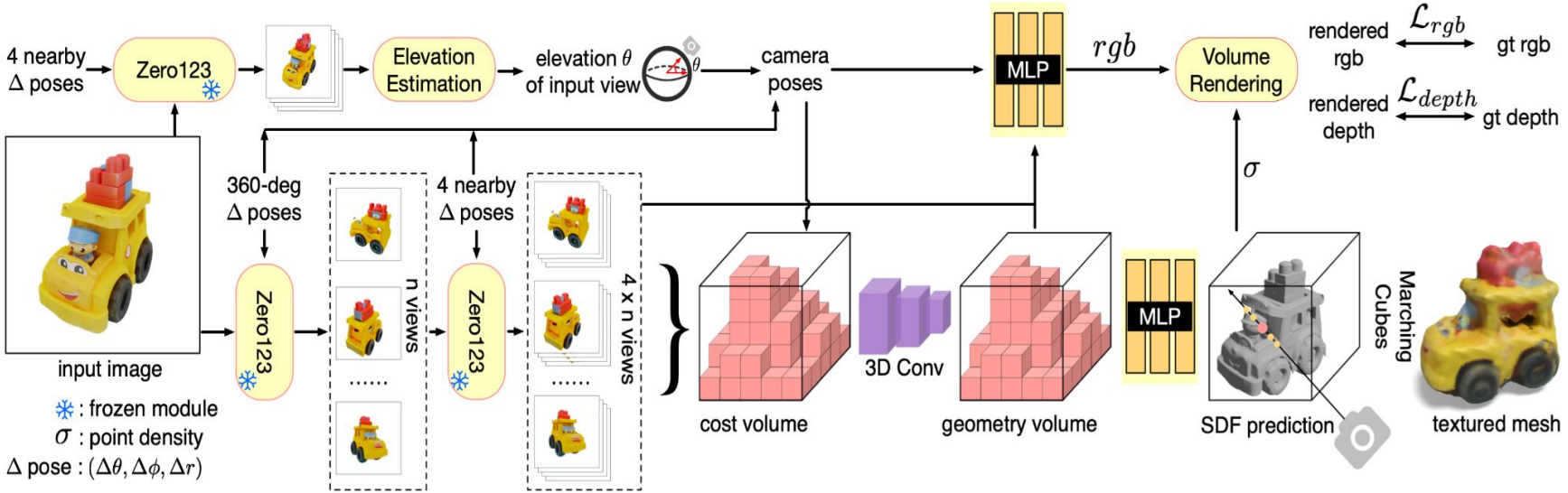
D.Optimization: Forward model: triplane



LRM: LARGE RECONSTRUCTION MODEL FOR SINGLE IMAGE TO 3D

D.Optimization: Forward model: cost-volume

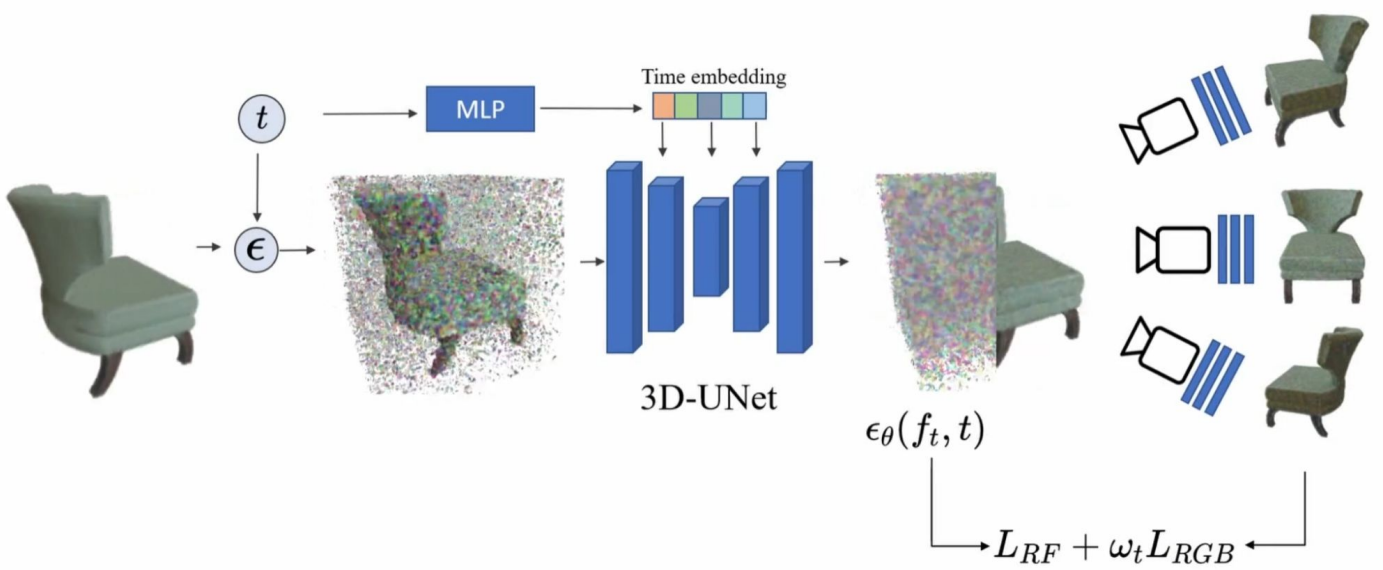
e.g. zero123



One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization

D.Optimization: Forward model: train 3D U-Net

directly generate nerf from 3D U-Net

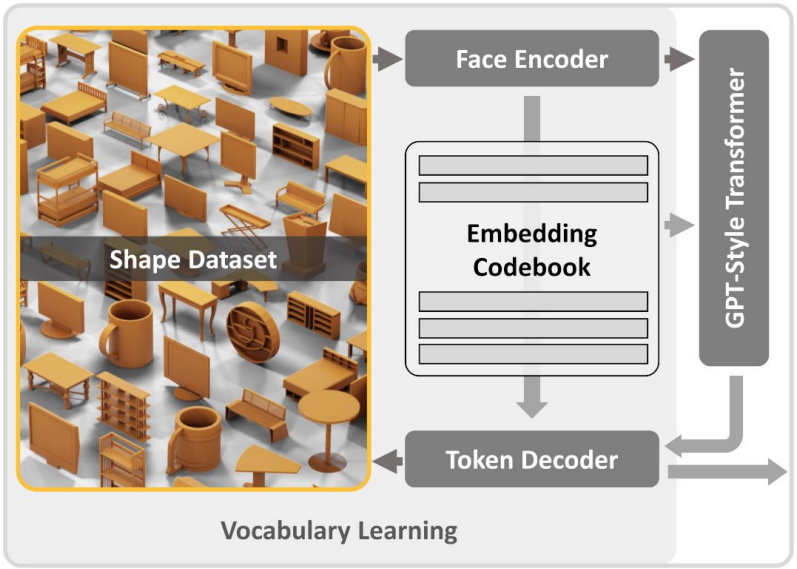


DiffRF is a denoising diffusion probabilistic model directly operating on 3D radiance fields and trained with an additional volumetric rendering loss. This enables learning strong radiance priors with high rendering quality and accurate geometry.

DiffRF: Rendering-guided 3D Radiance Field Diffusion

D.Optimization: Forward model: train a decoder

train a decoder transformer based on large 3d mesh dataset



MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers

D.Optimization: GPT-control

(a). *"The sun shines brightly in a clear blue sky. Not a cloud in sight."*

(b). *"Thick gray clouds cover the entire sky, diffusing the sunlight."*

(c). *"The sky appears hazy, with a soft, diffuse sunlight filtering through."*

(d). *"The sky is painted in shades of orange and pink as the sun sets behind the island."*

(e). *"The sky is a mixture of soft purples and pinks as the sun rises."*

(f). *"A canopy of stars fills the dark night sky, with the island bathed in moonlight."*

(a). Solution: From the description of "clear blue sky" and "not a cloud in sight", we can infer that the sun_intensity should be high, the sun_elevation should be around 90 (daytime), the sun_rotation can be set to 0. The air_density and dust_density should be low, so we can set them to 1 and 0 respectively. Since there are no clouds, the cloud_density should be 0. To make the sky appear bluer, we can increase the ozone value, so let's set it to 2. We can model the sky by calling the following function:

```python
sky_texture_node(sun_intensity='high', sun_elevation=90, sun_rotation=0, air_density=1, dust_density=0, ozone=2, cloud_density=0)
```

3D-GPT: Procedural 3D Modeling with Large Language Models