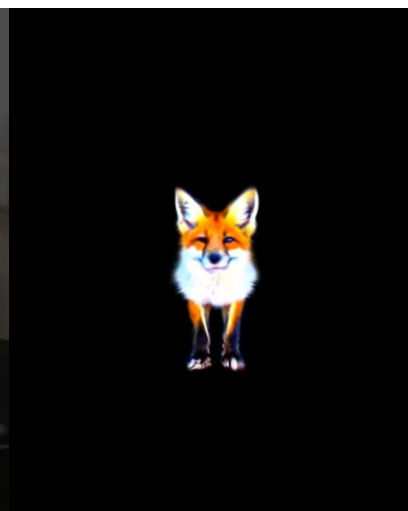


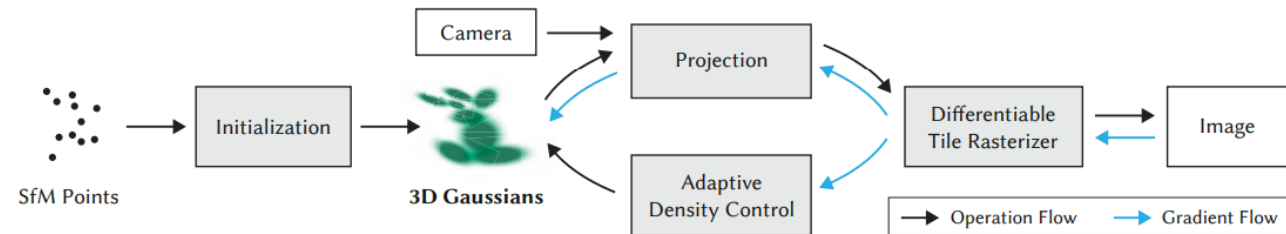


# Modeling, Understanding and Creating The 3D/4D World with Gaussian Splatting

2024.1.18  
Jiemin Fang



# Preview: 3D Gaussian Splatting



$$C = \sum_{i=1}^N T_i \alpha_i \mathbf{c}_i$$

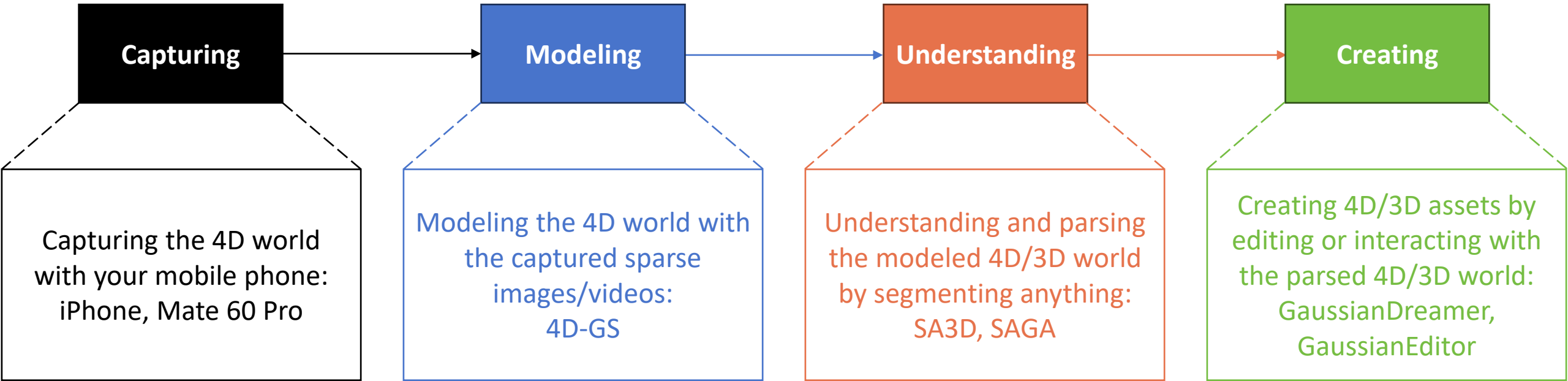
Volume rendering  
(NeRF-like ones)

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)$$

Point blending  
(3DGS)

Kerbl B, Kopanas G, Leimkühler T, et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering[J]. ACM Transactions on Graphics, 2023, 42(4).

# Playing with The 4D World Just Needs A Mobile Phone





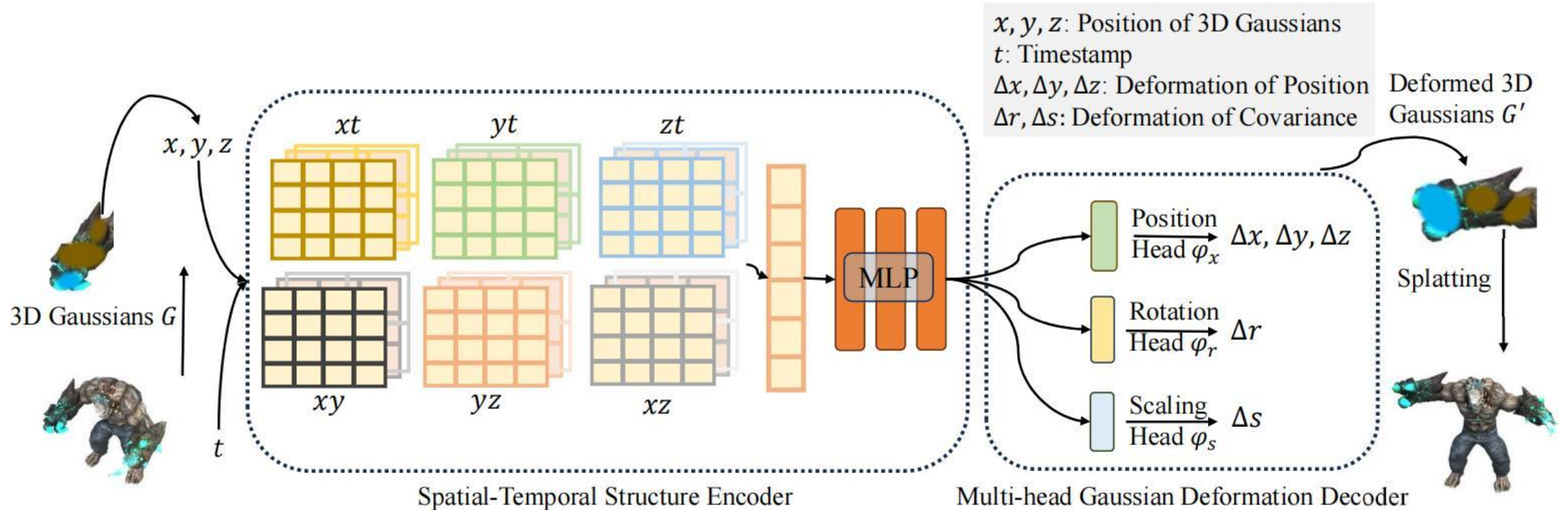
# 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering (4DGS)



4D-GS can learn a dynamic scene within **20 minutes** and reach the rendering speed over **30 fps** on one RTX 3080 GPU, **30×** faster than previous methods, with a minimum storage overhead of **18MB**

# Core Method

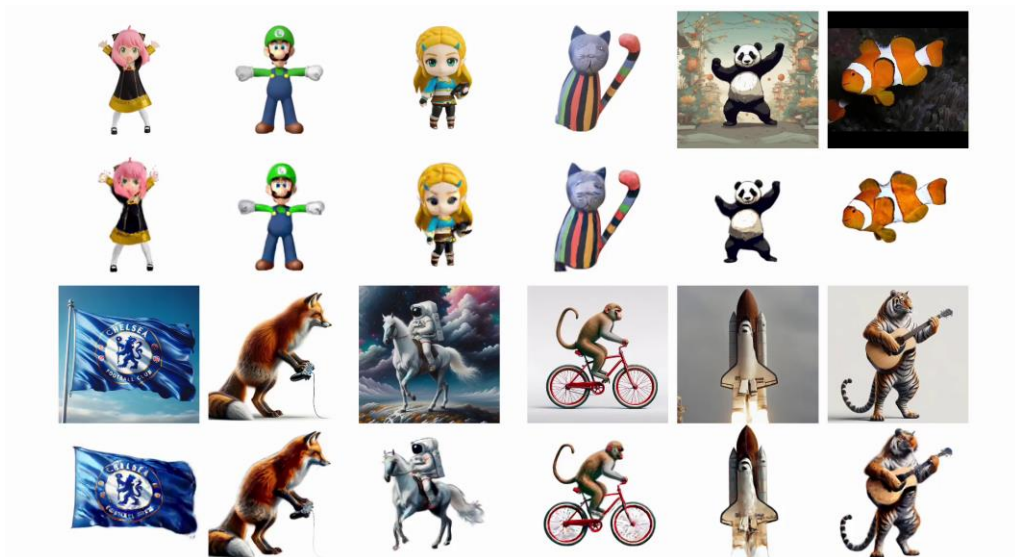
Construct an efficient spatio-temporal encoder based on neural voxel grids to model the motion of a scene, lifting 3D Gaussians to 4D



# 4D Scene Composition

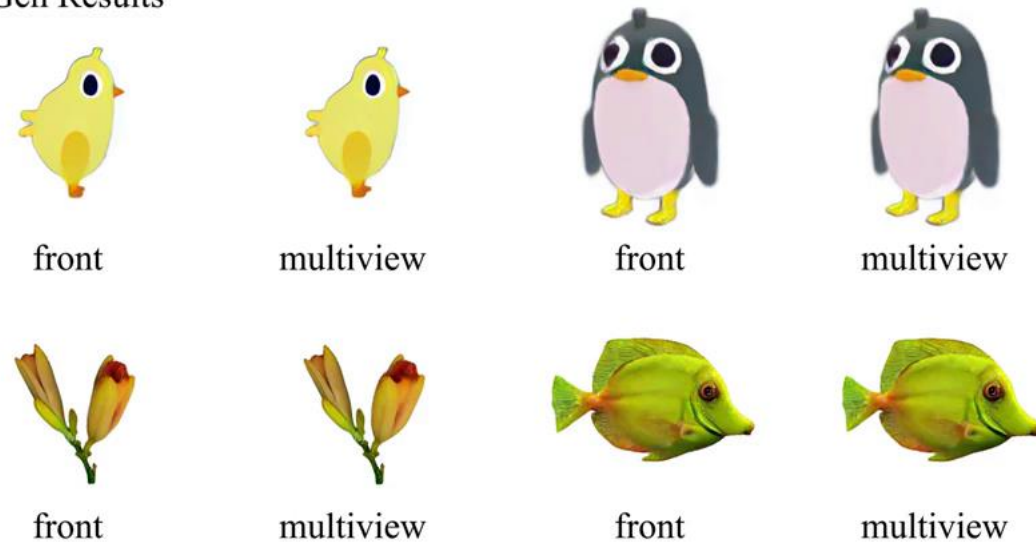


## 4D Generation with 4DGS



Ren J, et al. DreamGaussian4D, 2023.

## 4DGen Results

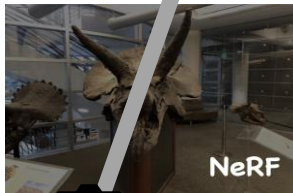


Yin Y, et al. 4DGen, 2023



# Segment Anything in 3D with NeRFs (SA3D)

First,  
Render a  
view



Second,  
Input the  
prompt



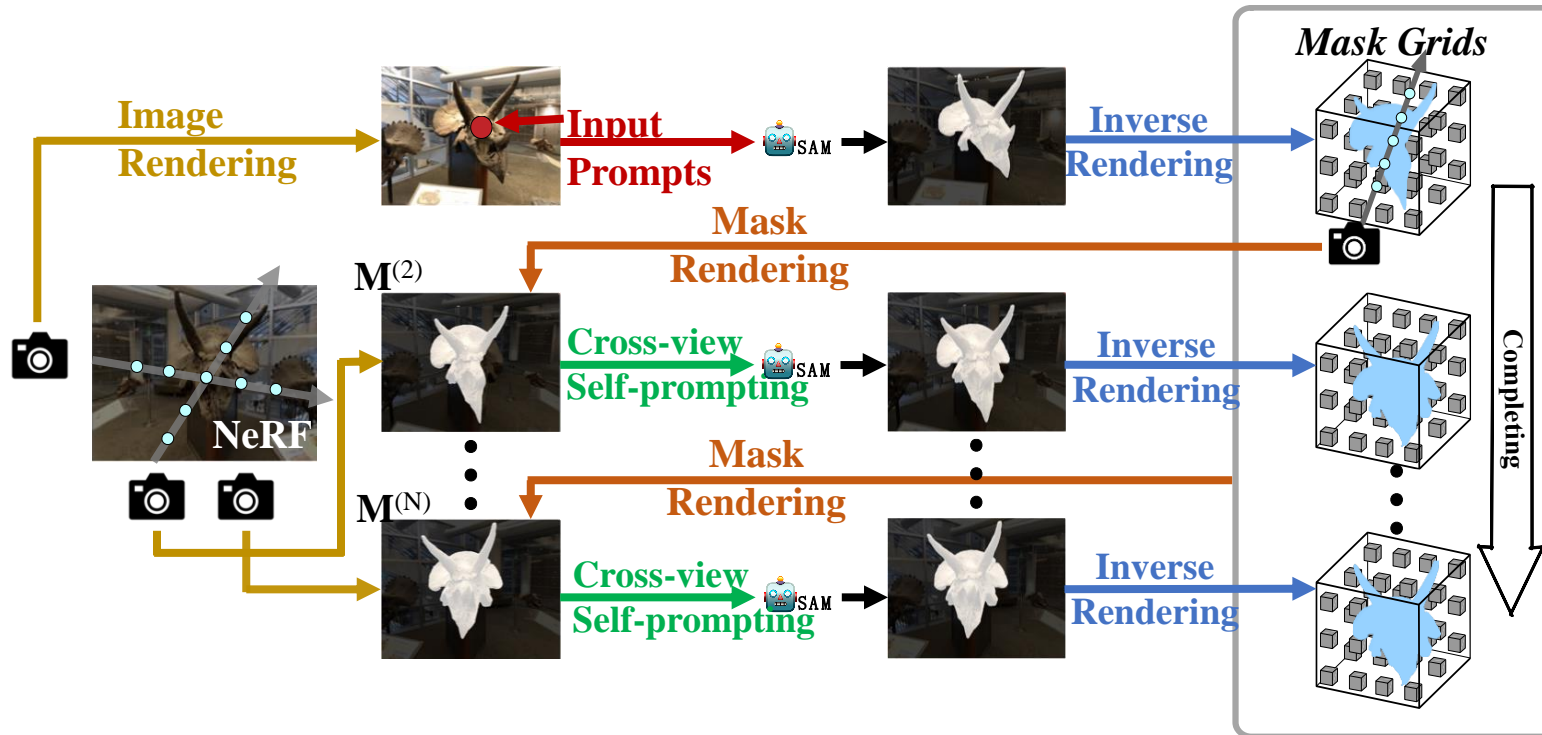
Third,  
Get the 3D  
object



SA3D extends the segmentation capabilities of the 2D segmentation foundation model (SAM) to 3D, enabling 3D target segmentation based on input prompts from a single 2D view.

*Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In NeurIPS, 2023.*

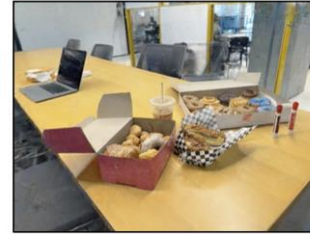
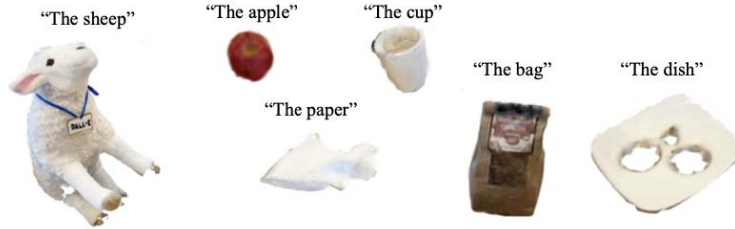
# Pipeline: Mask inverse rendering – Cross-view self-prompting



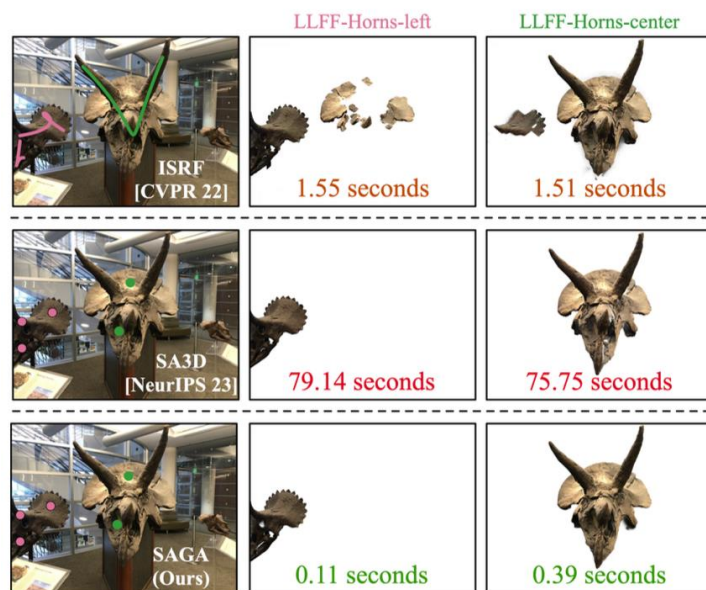
- Mask inverse rendering: project 2D segmentation results to 3D with the depth prior
- Cross-view self-prompting: generate new prompts to drive the 2D foundation model according to the temporarily inaccurate 3D segmentation result
- Repeat the above steps iteratively to refine the 3D segmentation result



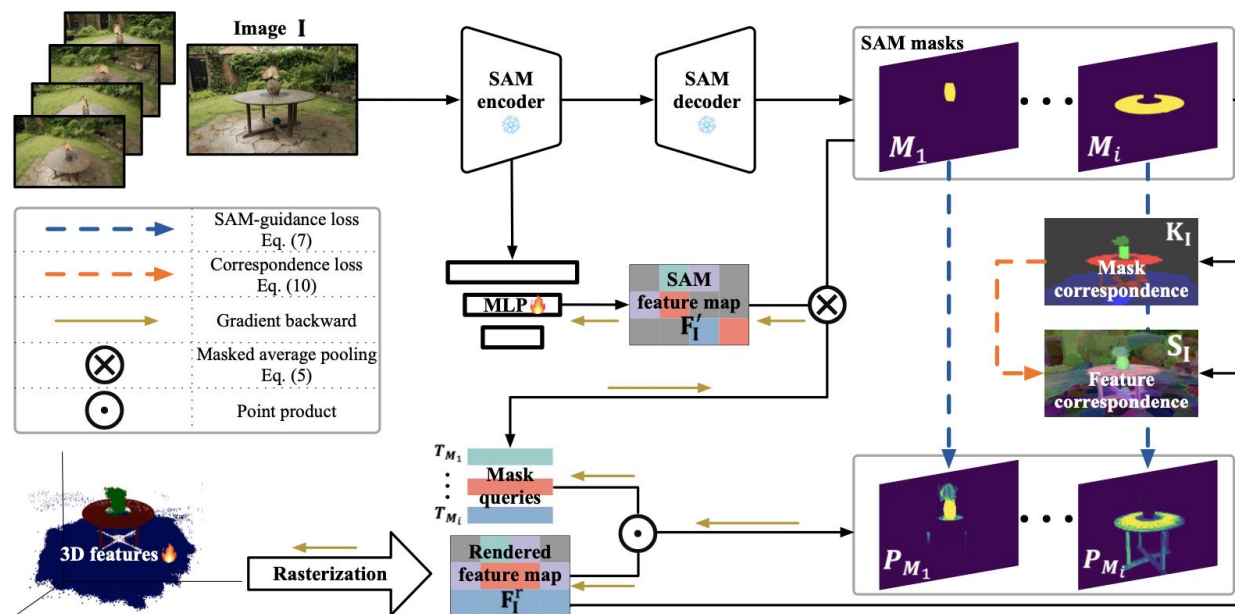
# Performance



# Segment Any 3D Gaussians (SAGA)



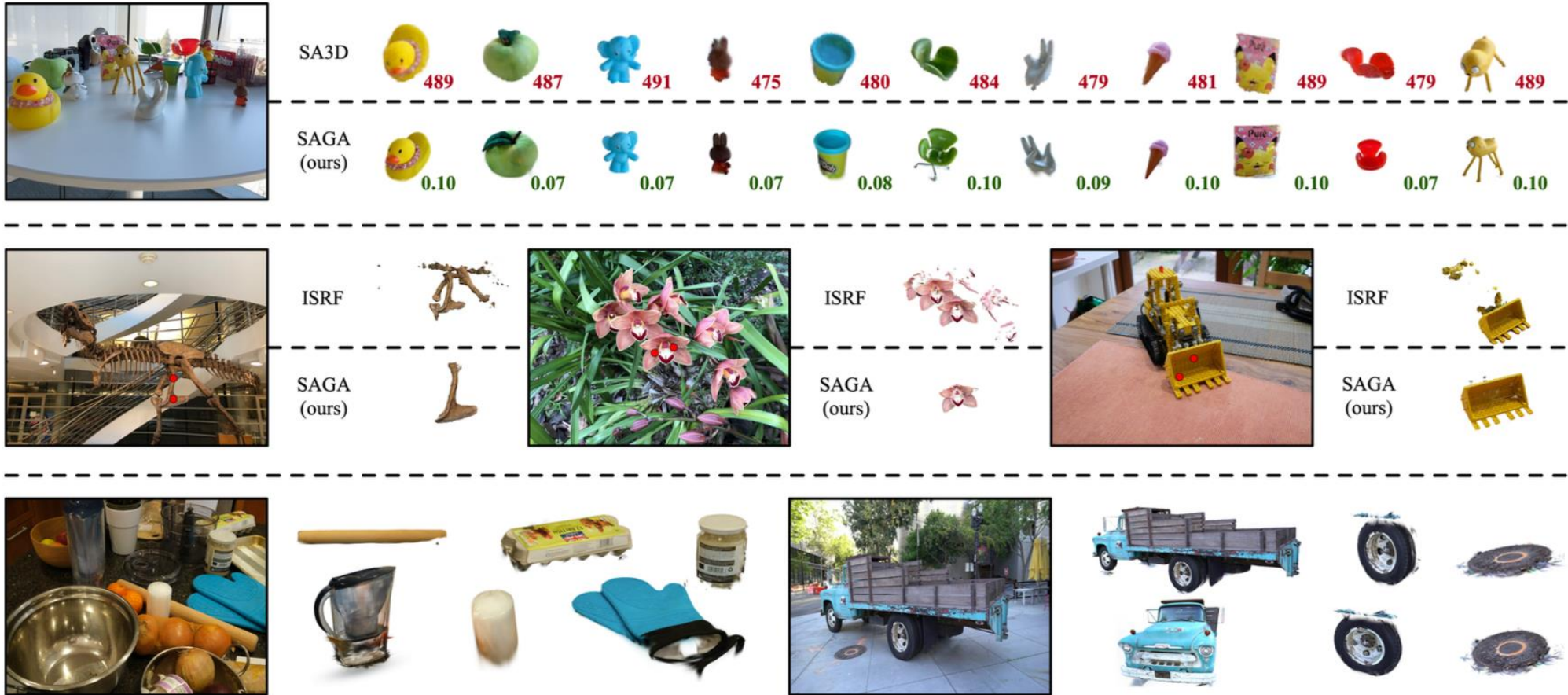
SAGA can realize 3D interactive segmentation within milliseconds, which is about one thousandth that of SA3D



SAGA distills the segmentation ability of SAM into the 3D Gaussians  
In inference stage, the segmentation can be achieved by simple  
feature matching



# Performance

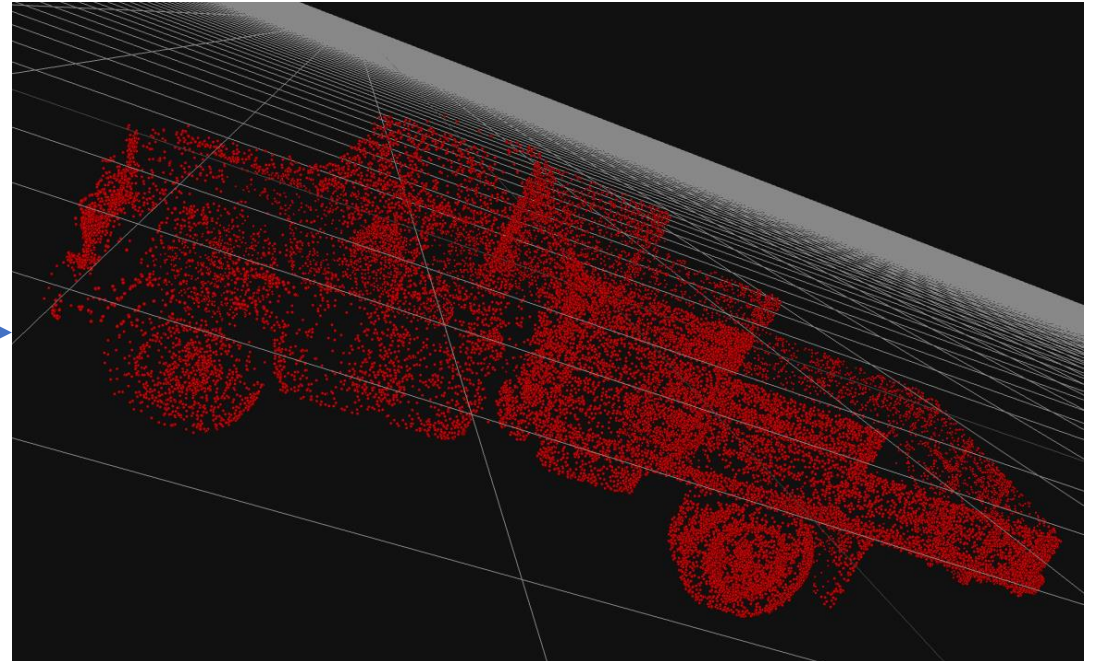


- Compared with SA3D, SAGA is much more efficient
- Compared with previous NeRF-based 3D interactive method, SAGA retains the multi-granularity ability of SAM

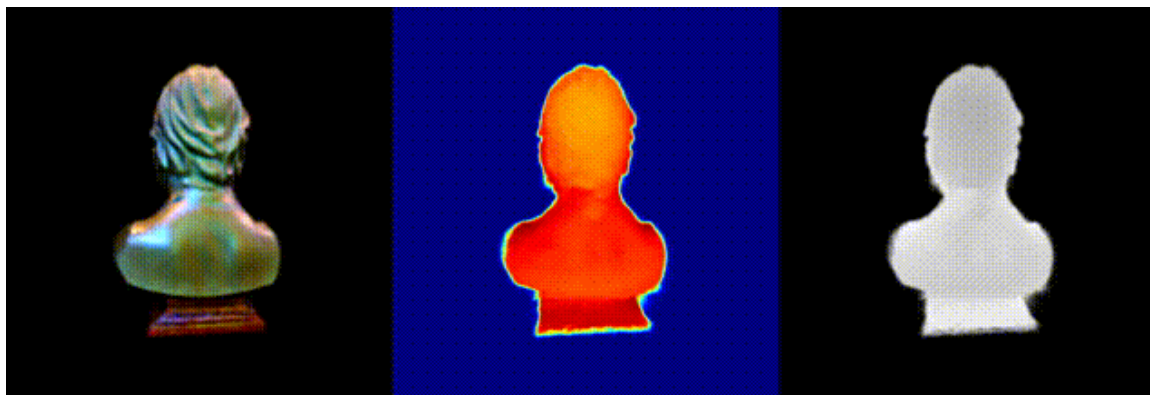
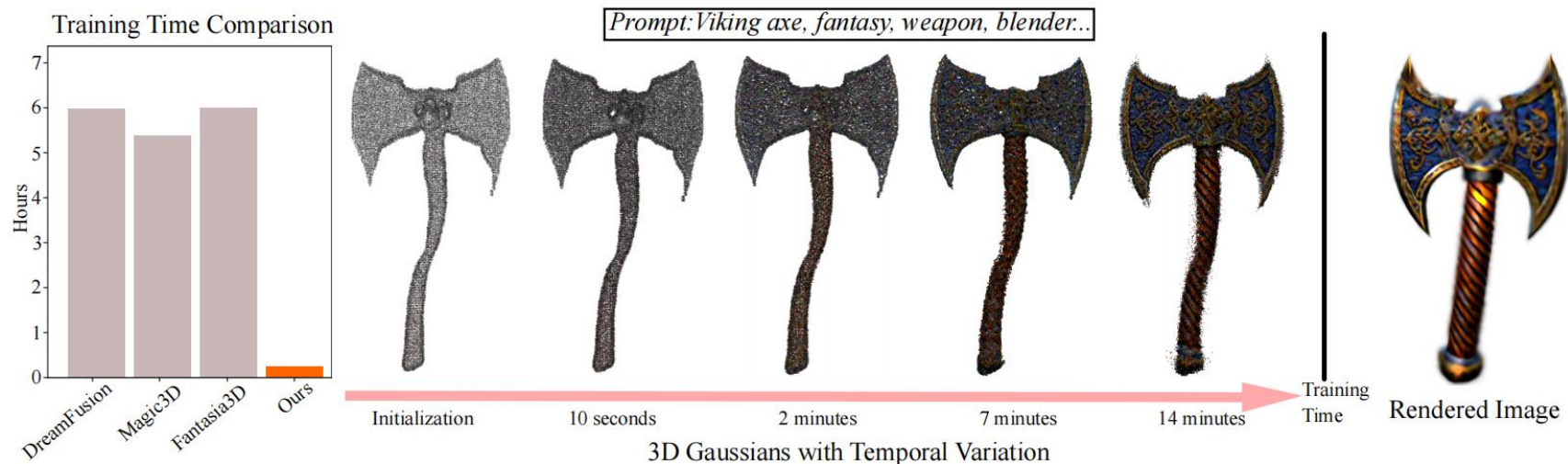


# Performance

Since 3D Gaussians can be regarded as a kind of point cloud, the segmented target object can be transferred to point cloud seamlessly, which is suitable for traditional industrial pipeline.



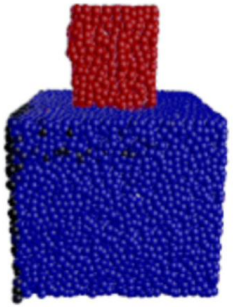
# GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models



Yi T, Fang J, Wang J, et al. *Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors*[J]. *arXiv preprint arXiv:2310.08529*, 2023.

## 3D Pretrained Diffusion Models (3D Diffusion Models)

- Speed: 🐇
- 3D Data Requirement: ★ ★ ★ ★ ★
- Typical methods: Point-E, Shap-E, 3DGen, Get3D ...



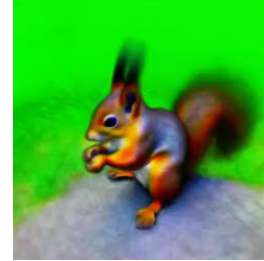
Point-E  
4 Seconds



Shap-E  
7 Seconds



GaussianDreamer (Ours)  
15 Minutes



DreamFusion  
6 Hours



Fantasia3D  
6 Hours

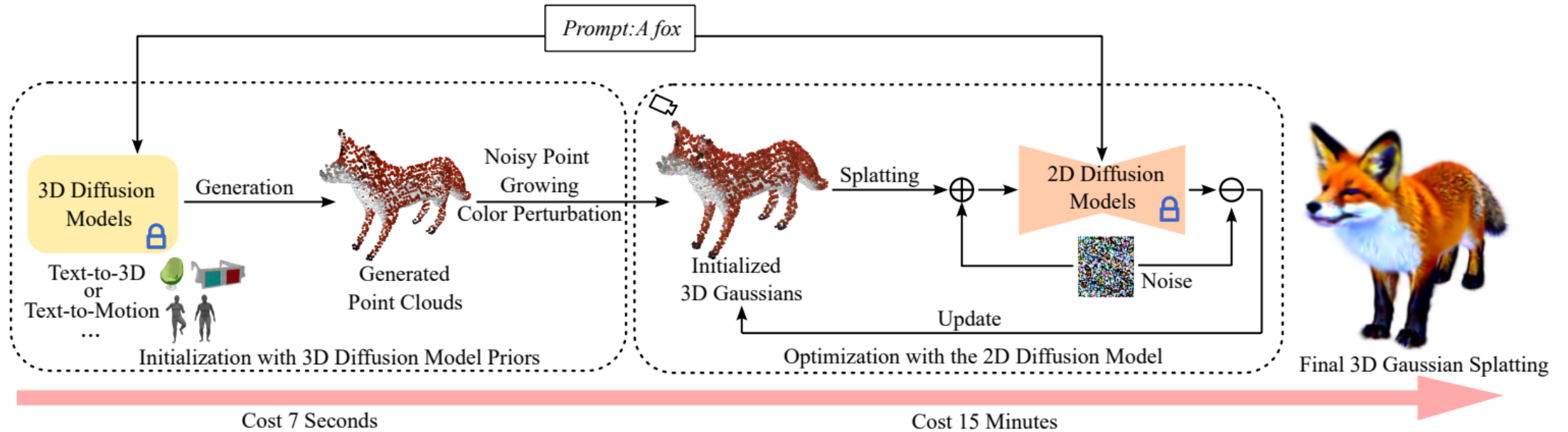
## Lifting 2D Diffusion Models to 3D (2D Diffusion Models)

- Speed: 🐢
- 3D Data Requirement: ☆
- Typical methods: DreamFusion, Fantasia3D, ProlificDreamer ...

Bridging 2D and 3D Diffusion Models



# Framework



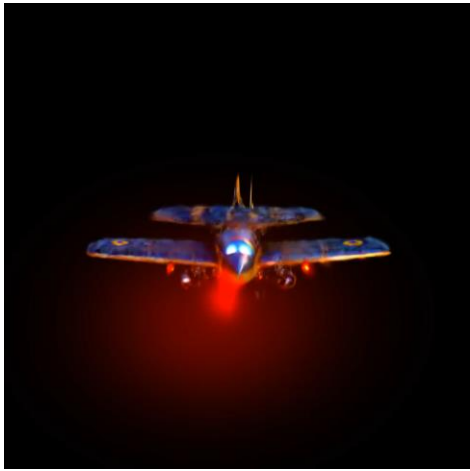
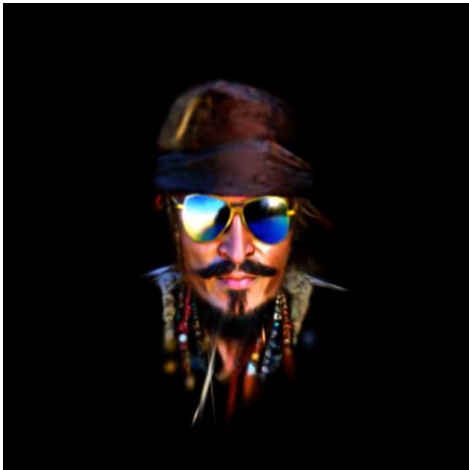
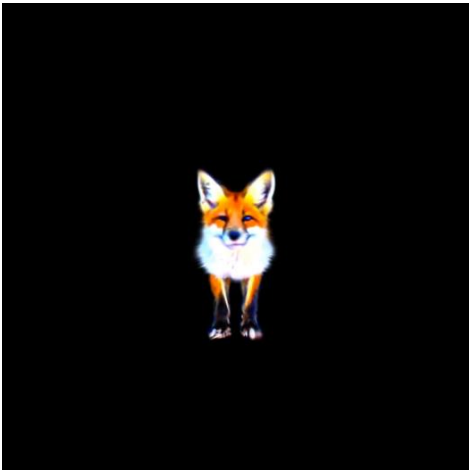
Our overall framework consists of two parts.

1) Using 3D Gaussian splatting to combine 3D diffusion model and 2D diffusion model, we initialize with point cloud priors.

2) Optimizing 3D Gaussians with SDS.

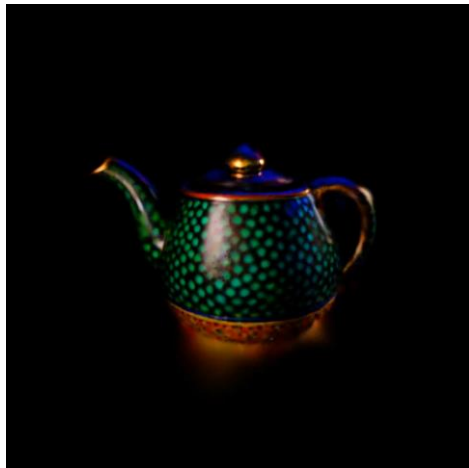
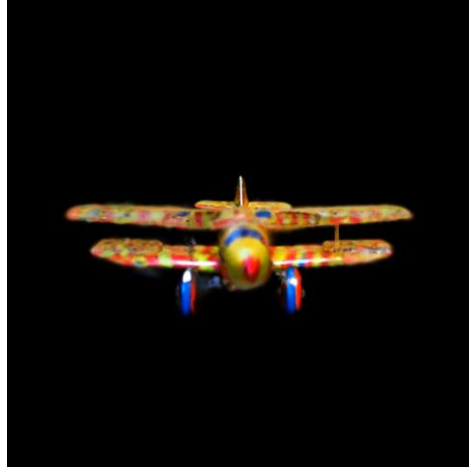
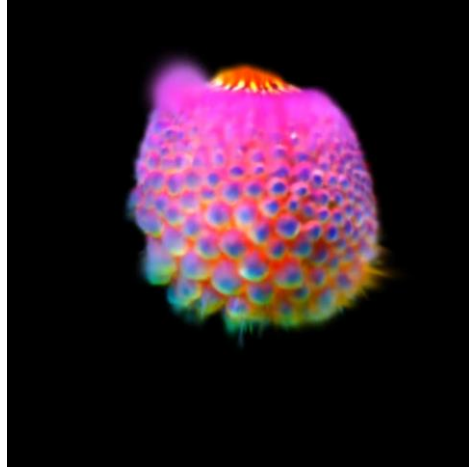
The entire process can be finished within 15 minutes. The generated 3D assets can be rendered in real time.

# 3D Asset Generation



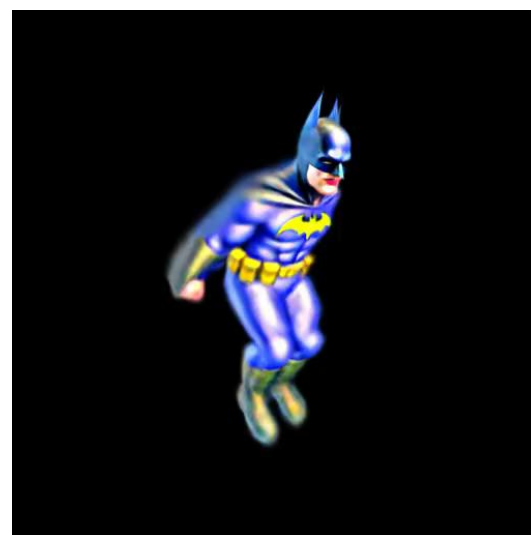
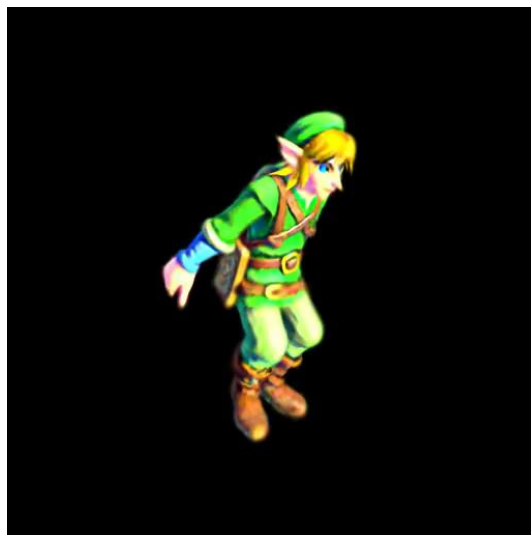
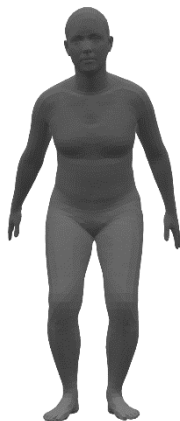
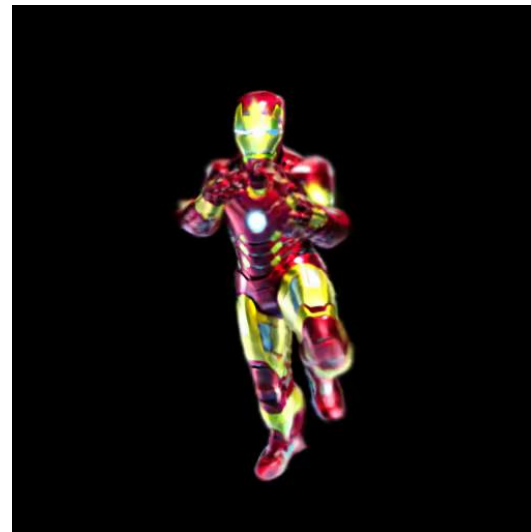
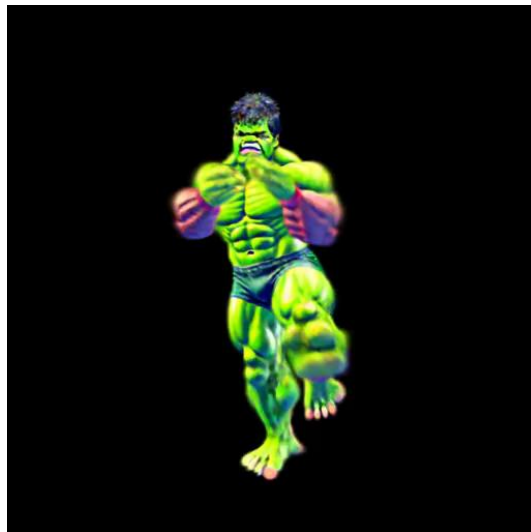
DreamFusion, ICLR 2023	6.0 Hours
Magic3D, CVPR 2023	5.3 Hours
Fantasia3D, ICCV 2023	6.0 Hours
ProlificDreamer, NeurIPS 2023	~ Hours
GSGEN, arXiv 2023	2 Hours
GaussianDreamer (Ours)	15 Minutes

# 3D Asset Generation



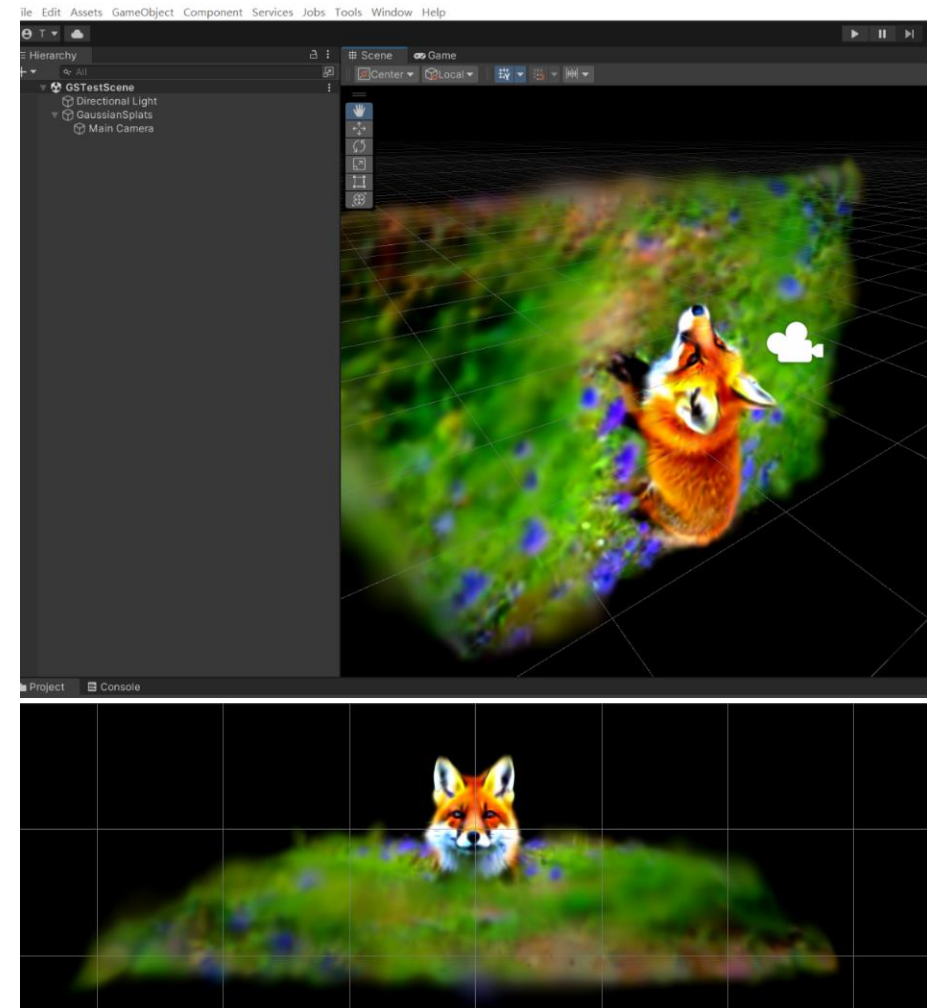


# 3D Avatar Generation



# Application

Integrate the generated 3D assets into the Unity game engine for game applications



# GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions

General 3D scene editing method,  
enabling text-based delicate editing of 3D assets in 20 minutes

First edit



**Original**



*“Make the road look like the meadow”*



*“Turn the road into river”*

Second edit



*“Turn the bench red”*



*“Turn the bench into wood”*



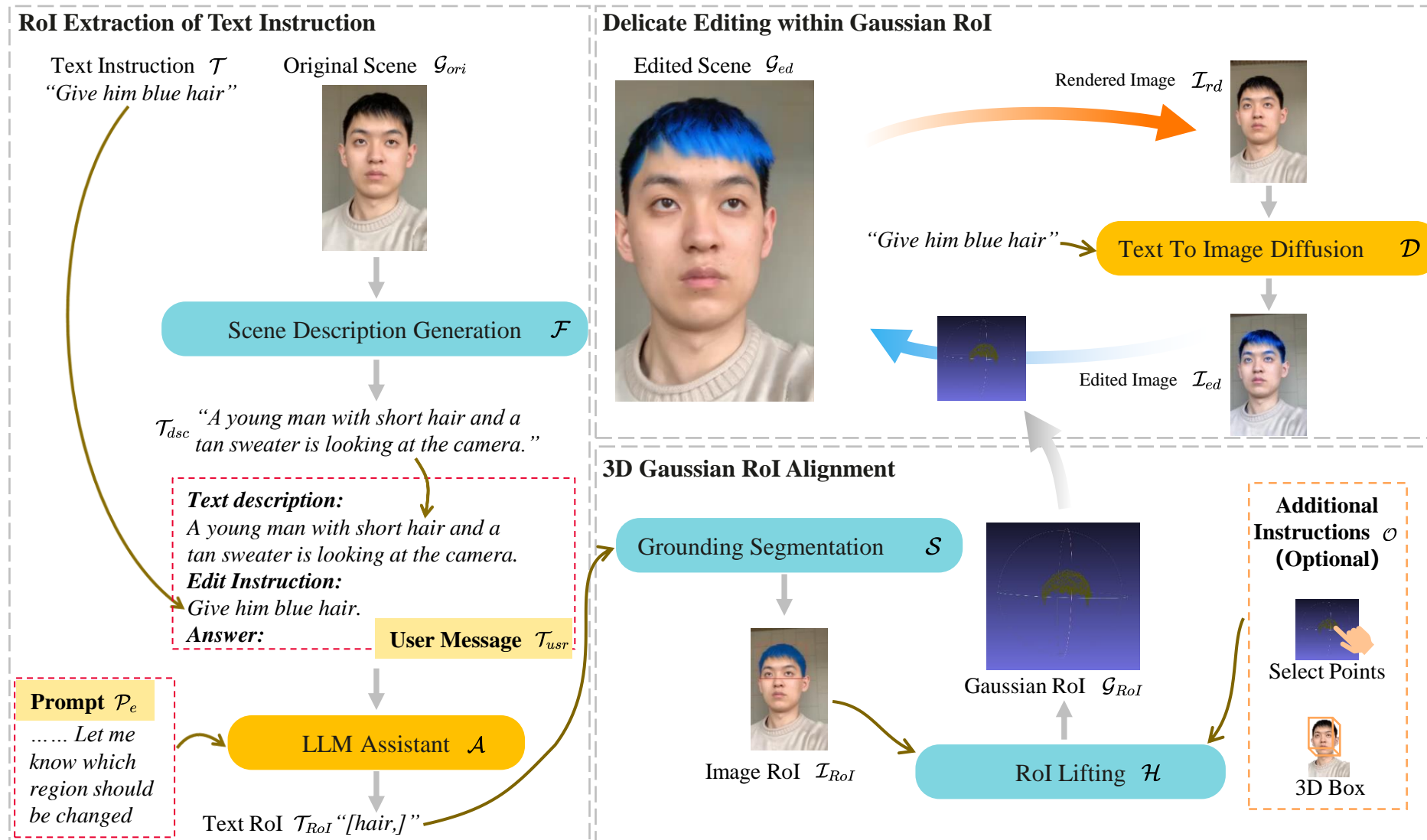
*“Turn the bench into Turquoise”*

Fang J, Wang J, Zhang X, et al. Gaussianeditor: Editing 3d gaussians delicately with text instructions[J]. arXiv preprint arXiv:2311.16037, 2023.



# Framework

LLM and multimodal model are used to achieve text RoI extraction and alignment of Rols in different spaces, enabling delicate 3D asset editing.

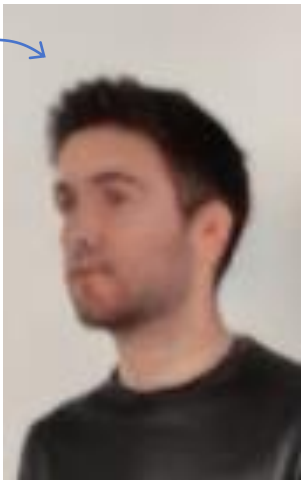
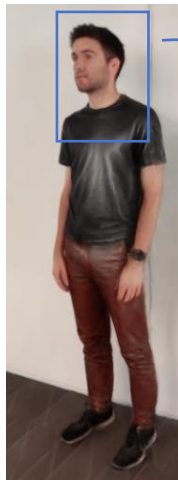
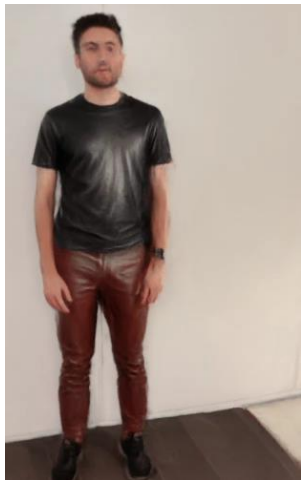
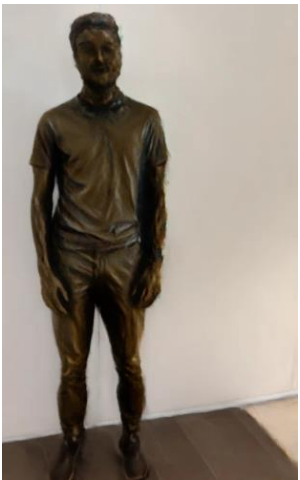


# Comparison

GaussianEditor accurately locates the editing area and preserve the facial appearance.



IN2N



Ours



Original

“Turn his *pants* into yellow”

“Turn his *pants* into a bronze statue”

“Turn his *t-shirt* into white”

“Make his *t-shirt* out of leather”

# Application: 3D Scene Editing



**Original**



*“Turn the bear into a polar bear”*

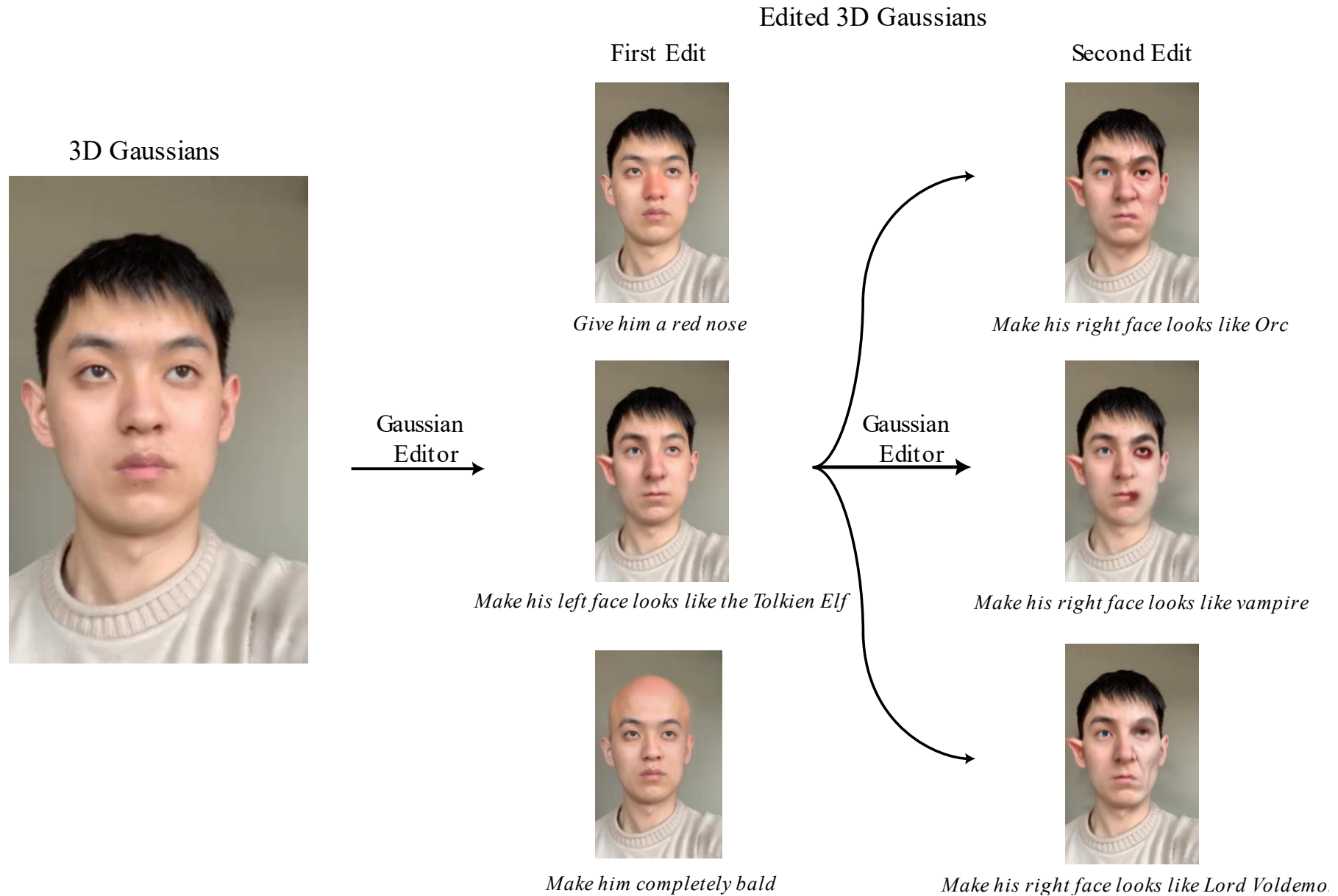


*“Turn the bear into a panda”*



*“Turn the bear into a grizzly bear”*

# Application: 3D Character Editing





# Conclusions

- 3D Gaussian Splatting (3D-GS) enjoy high efficiency and quality
- The explicit property of 3D-GS brings many benefits
  - *Easy to learn*
  - *Easy to manipulate*
  - *Free of sampling*
- 3D-GS has potential to change more things



# Thanks!

2024.1.18  
Jiemin Fang

