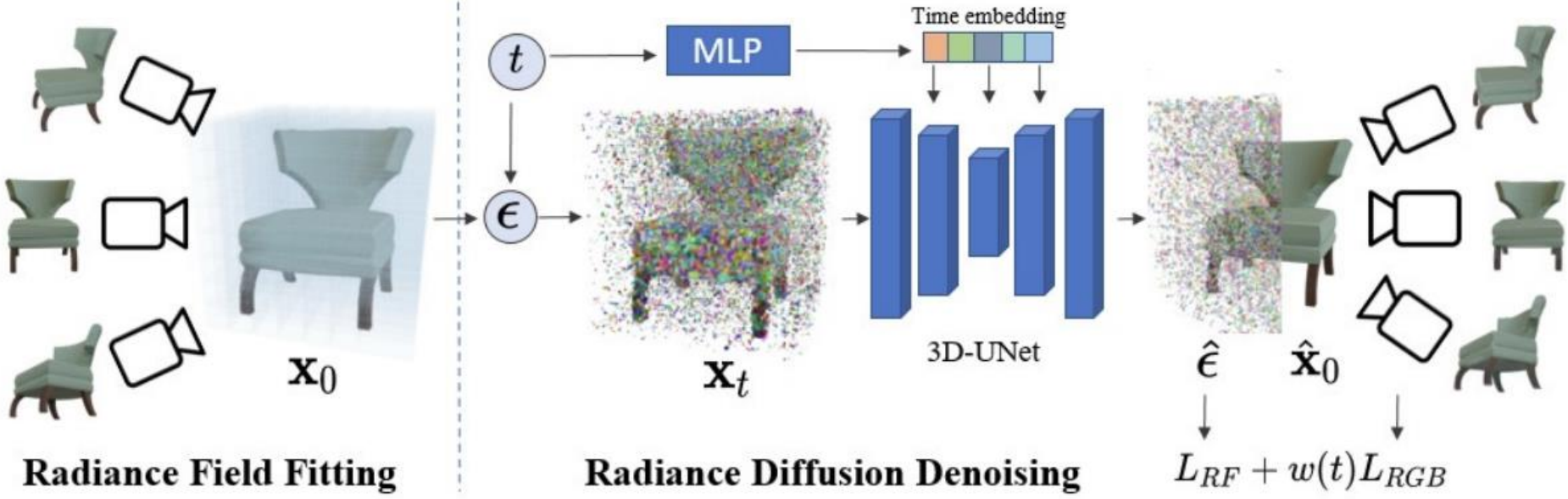
# 3D Aware Diffusion

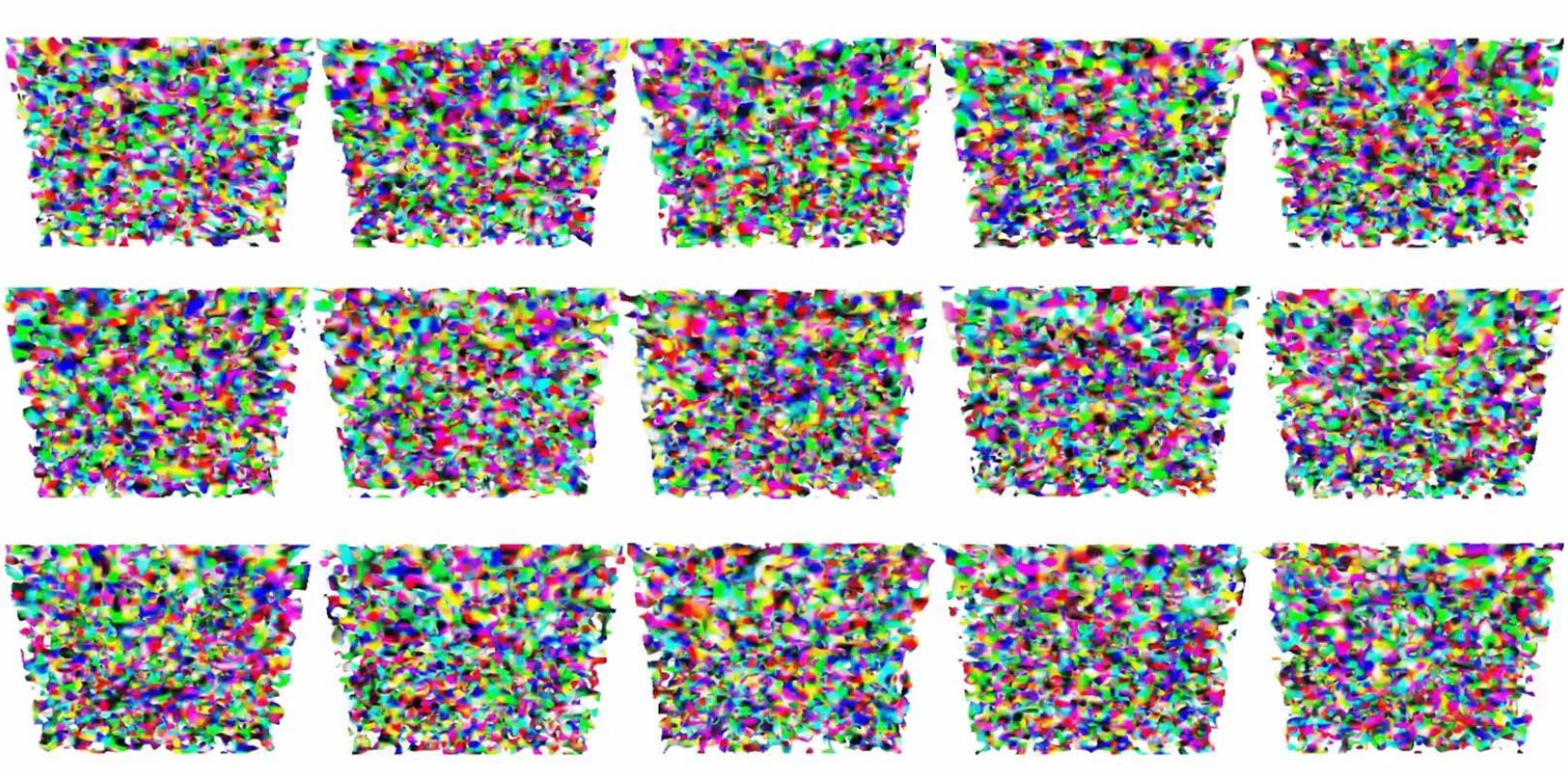
# DiffRF: Train with 3D Ground Truth






[Mueller et al. 23] DiffRF

# DiffRF: Results

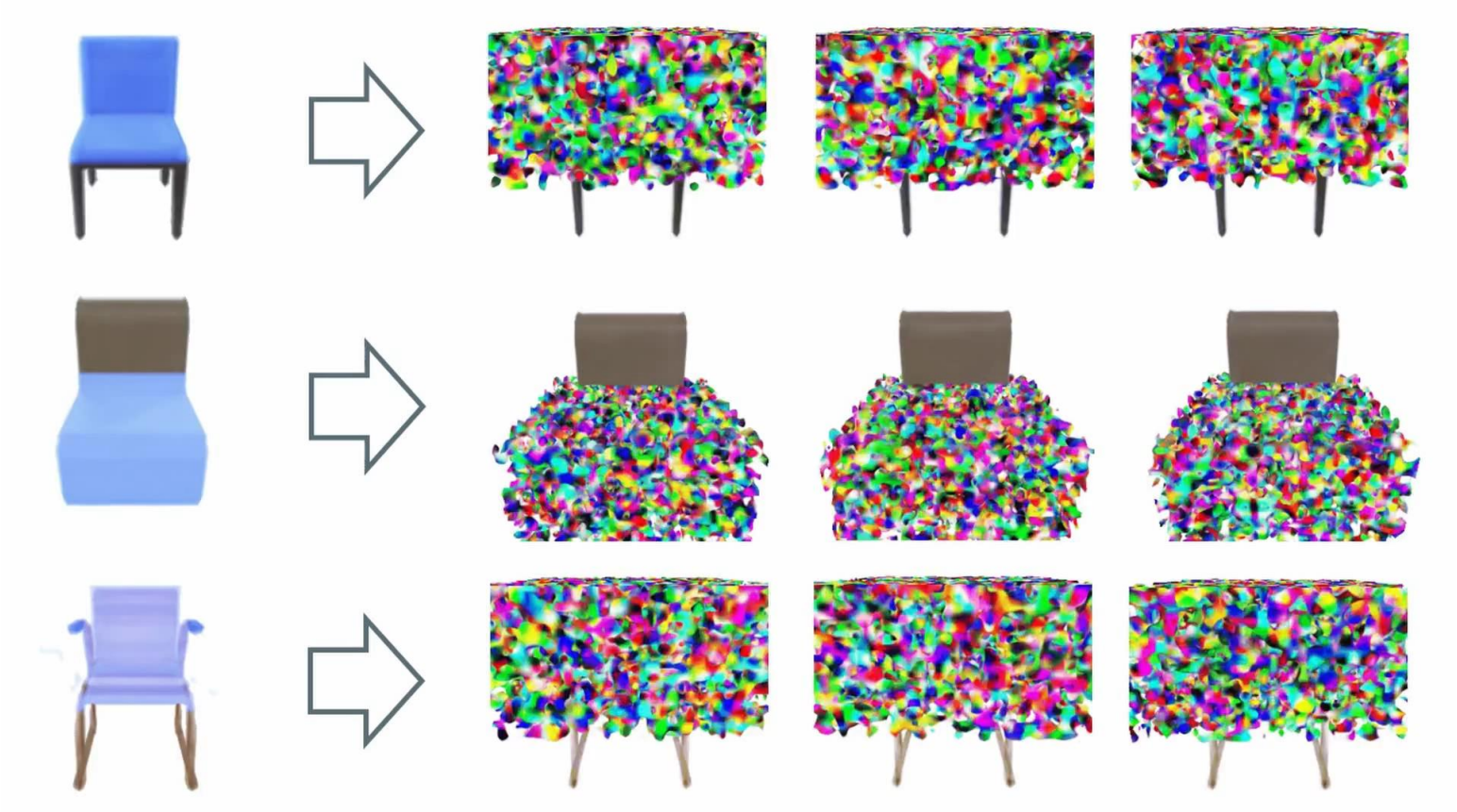


[Mueller et al. 23] DiffRF

# DiffRF: Masked Predictions

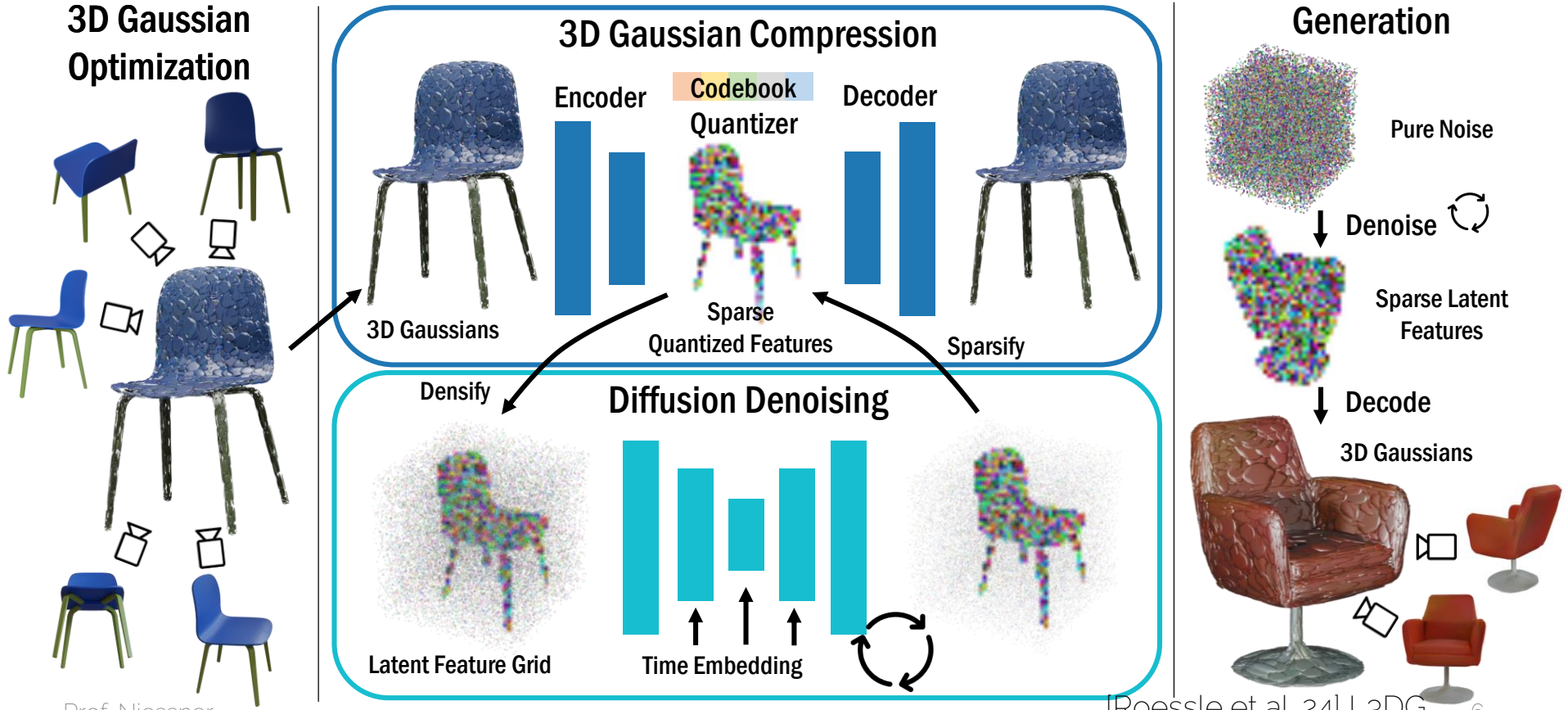

[Mueller et al. 23] DiffRF

# DiffRF: Masked Predictions



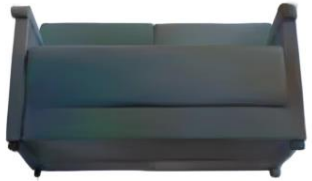
[Mueller et al. 23] DiffRF

# L3DG: Latent 3D Gaussian Diffusion



# L3DG: Latent 3D Gaussian Diffusion

# L3DG: Latent 3D Gaussian Diffusion








[Roessle et al. 24] L3DG

# L3DG: Latent 3D Gaussian Diffusion



→ Generating 3D Gaussians in latent space enables higher detail on object-level generation and scalability to room-sized scenes.



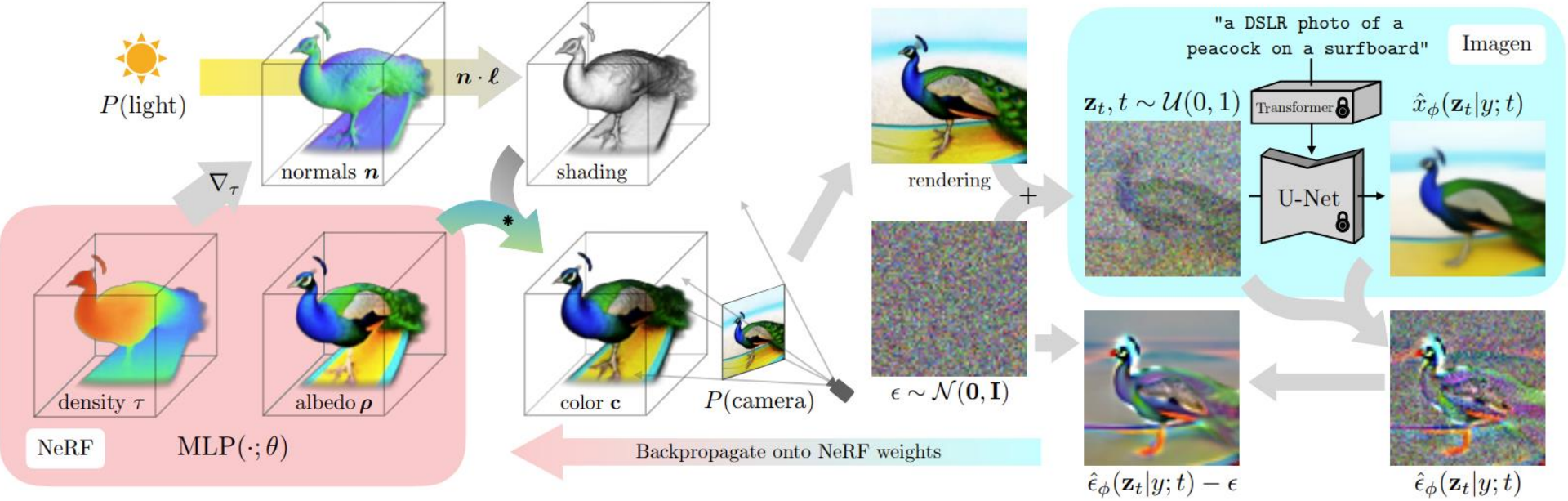
[Roessle et al. 24] L3DG

# Discussion: Diffusion vs GANs

Problem is always training data…

- Diffusion: input vs output need same dimensionality
- GANs: partial information feasible (e.g., reprojection, similar to GRAF, PiGAN, EG3D)

# DreamFusion




Score Distillation Sampling (SDS)



[Poole et al. 22] DreamFusion

# Score Distillation Sampling (SDS)

Loss functions for diffusion models

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

Training a diffusion model:

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

Sampling from a diffusion model?

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

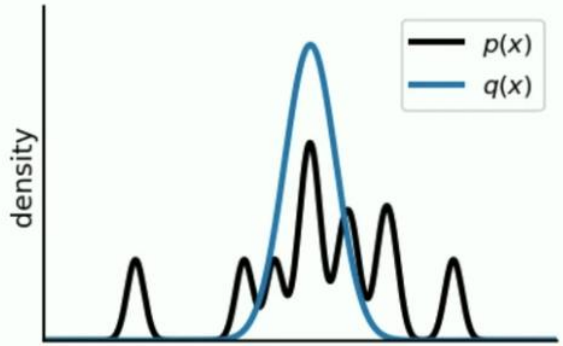
$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x} = g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t)\underbrace{(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon)}_{\text{Noise Residual}}\underbrace{\frac{\partial\hat{\epsilon}_\phi(\mathbf{z}_t; y, t)}{\mathbf{z}_t}}_{\text{U-Net Jacobian}}\underbrace{\frac{\partial\mathbf{x}}{\partial\theta}}_{\text{G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

# Score Distillation Sampling (SDS)

Score distillation sampling

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

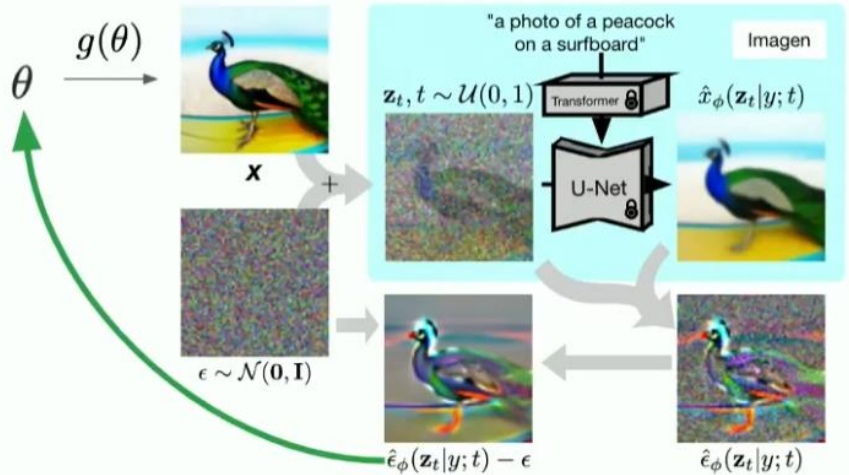
$$\mathcal{L}_{\text{SDS}}(\theta) = \mathbb{E}_t\left[w(t)\text{KL}(q(z_t; \theta, y, t)\|p_\phi(z_t; y, t))\right]$$



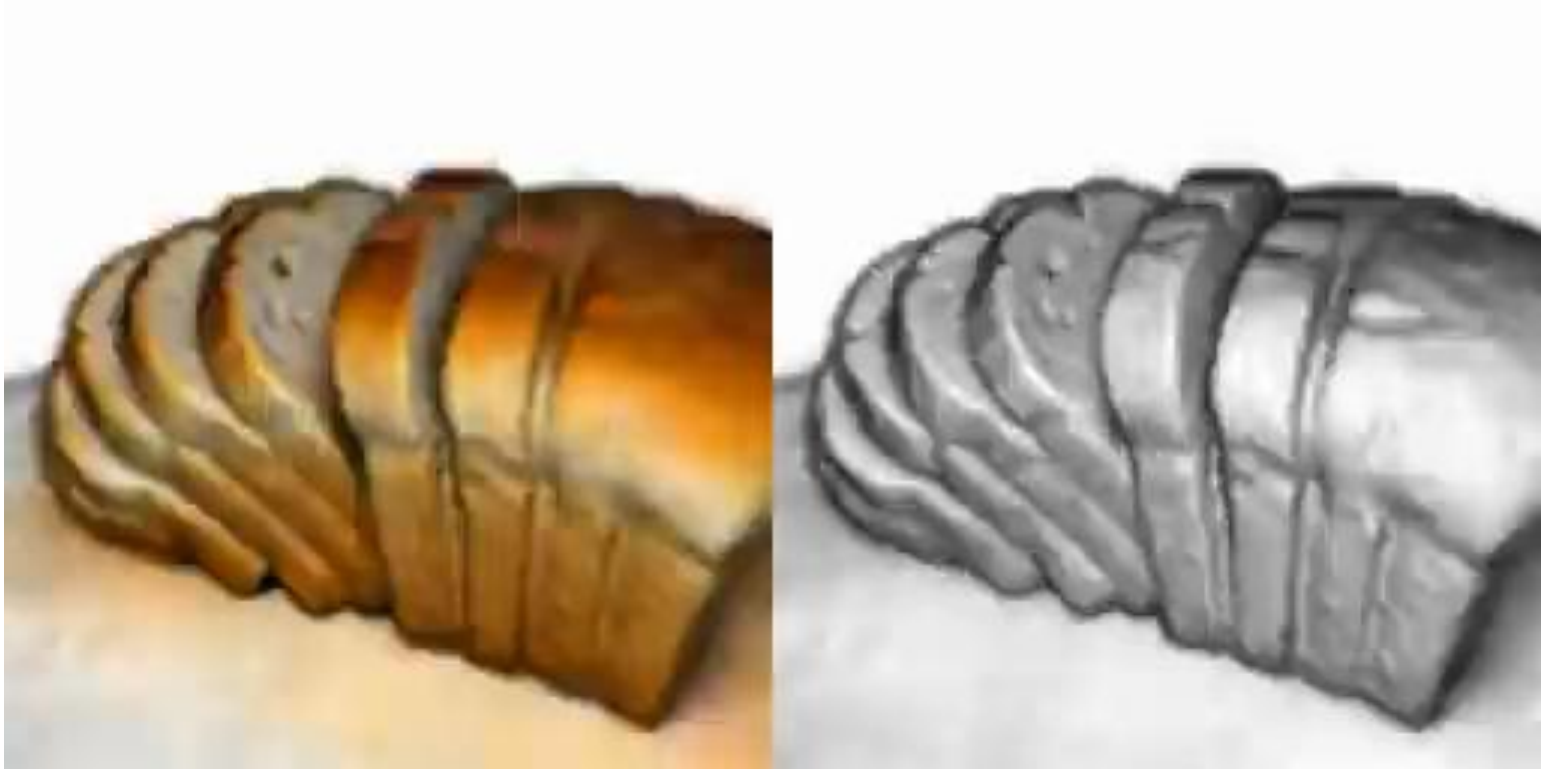
# Score Distillation Sampling (SDS)

Using the score distillation loss

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



# DreamFusion



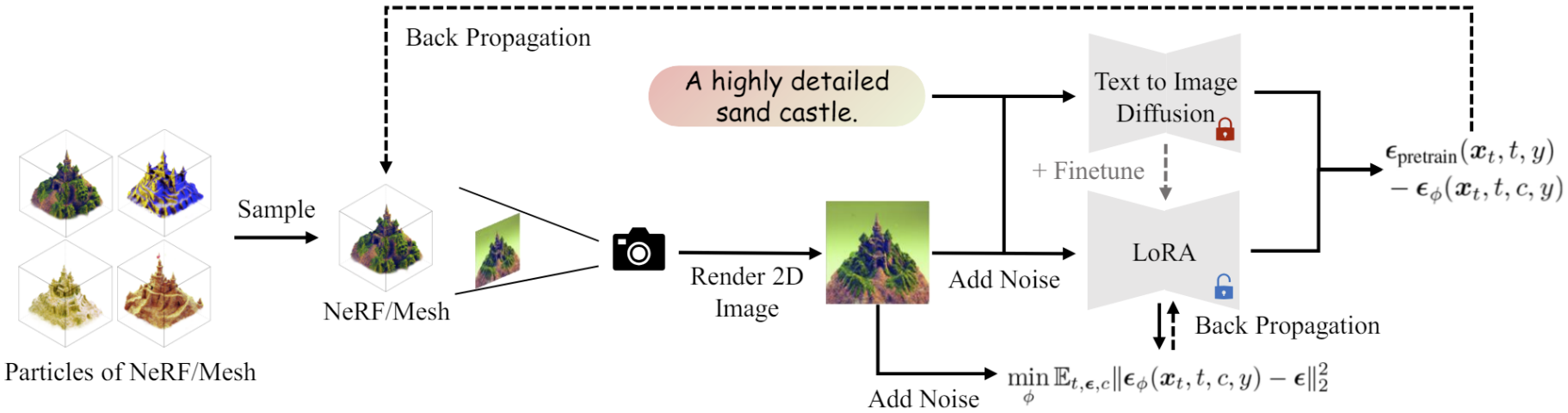
[Poole et al. 22] DreamFusion

# DreamFusion

[Poole et al. 22] DreamFusion

# SDS Follow Ups

- ProlificDreamer (variational SDS)



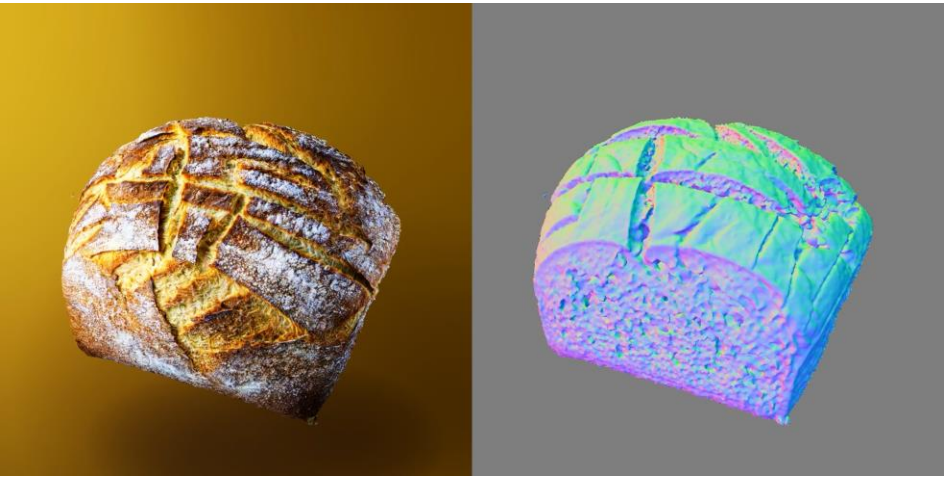

$$\nabla_\theta \mathcal{L}_{\text{VSD}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)\left(\boldsymbol{\epsilon}_{\text{pretrain}}(\boldsymbol{x}_t, t, y^c) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, c, y)\right)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right],$$

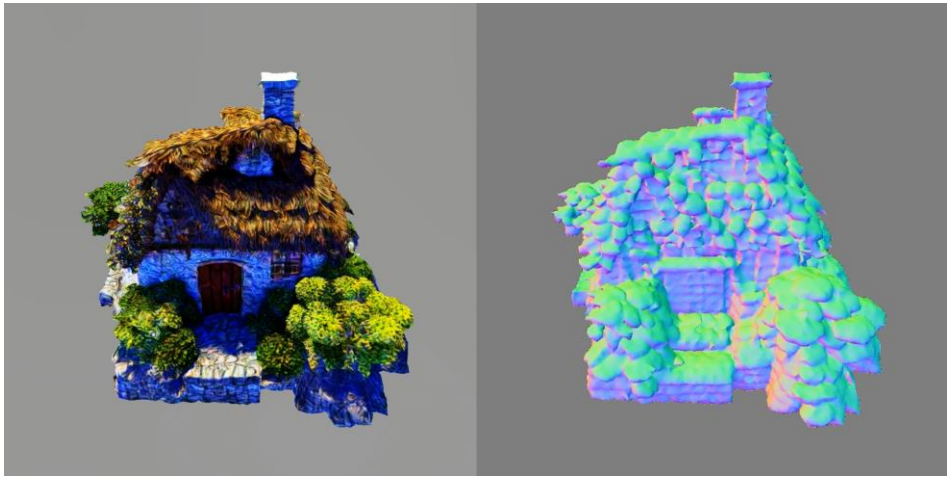
where $\boldsymbol{x}_t = \alpha_t \boldsymbol{g}(\theta, c) + \sigma_t \boldsymbol{\epsilon}$.

[Wang et al. 23] ProlificDreamer

# SDS Follow Ups

- ProlificDreamer (variational SDS)

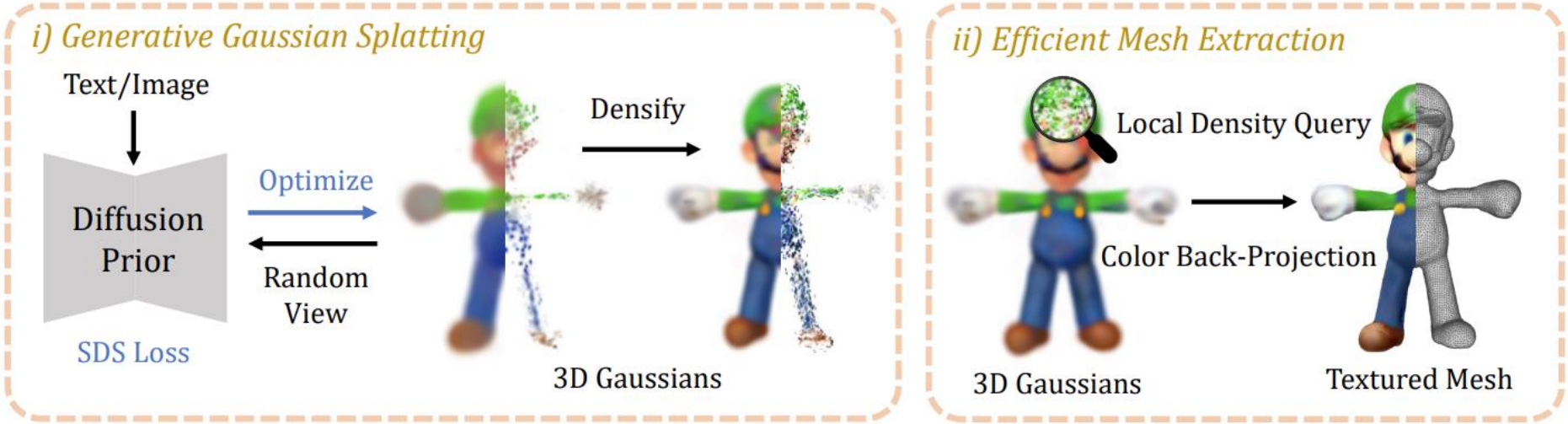






[Wang et al. 23] ProlificDreamer

# DreamGaussian

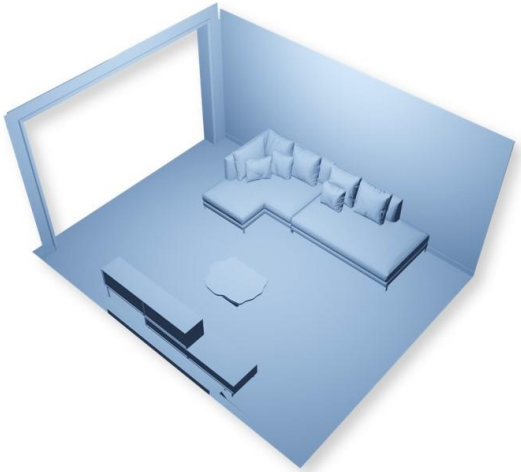



[Tang et al.. 24] Dream Gaussian

# DreamGaussian


[Tang et al.. 24] Dream Gaussian

# SceneTex



Scene geometry

*“A Bohemian style living room”*

*“A country style living room”*

Scene with generated texture

[Chen et al., 24] SceneTex


Baroque

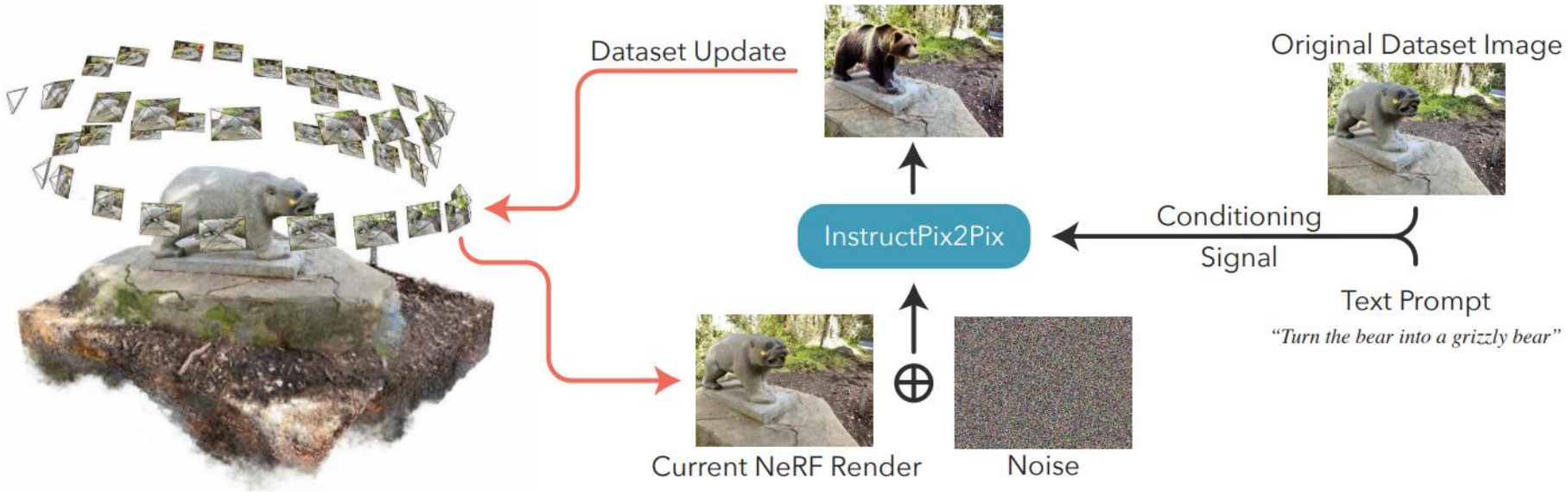

Bohemian


French


Japanese

# InstructNerf2Nerf

- Scene Editing: iteratively replace dataset with image editing model + keep optimizing NeRF




[Haque et al., 23]

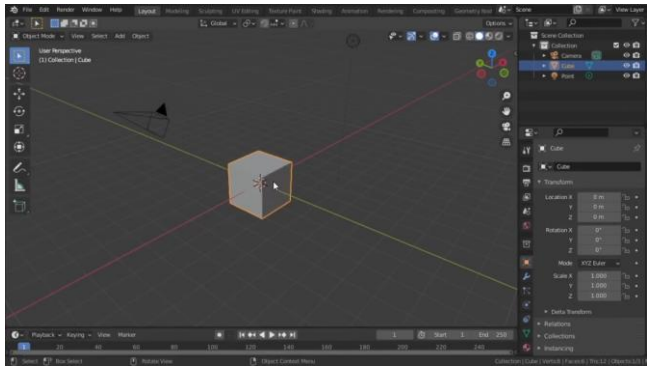
# InstructNerf2Nerf

[Haque et al., 23]

# World Building

# Exploit powerful 2D priors

- Manual creation of scenes takes tremendous expertise and time:

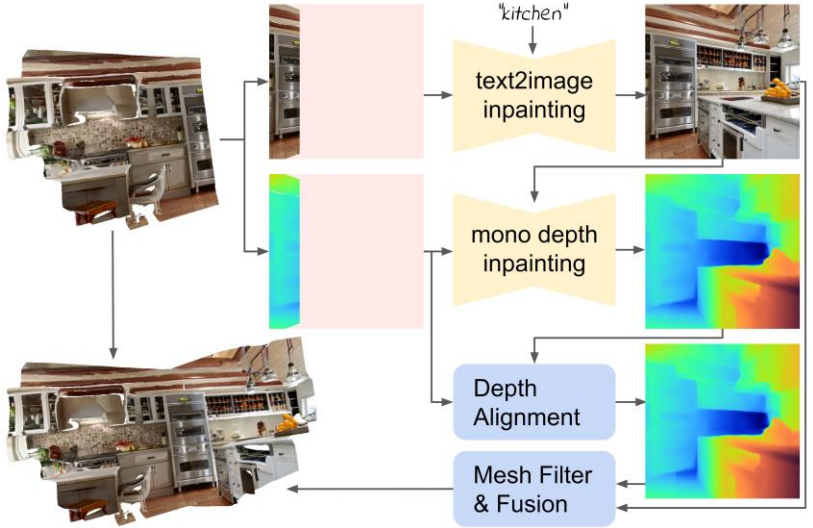


Source: https://youtu.be/yCHT23A6aJA

- How can we leverage 2D diffusion models to generate entire 3D worlds?

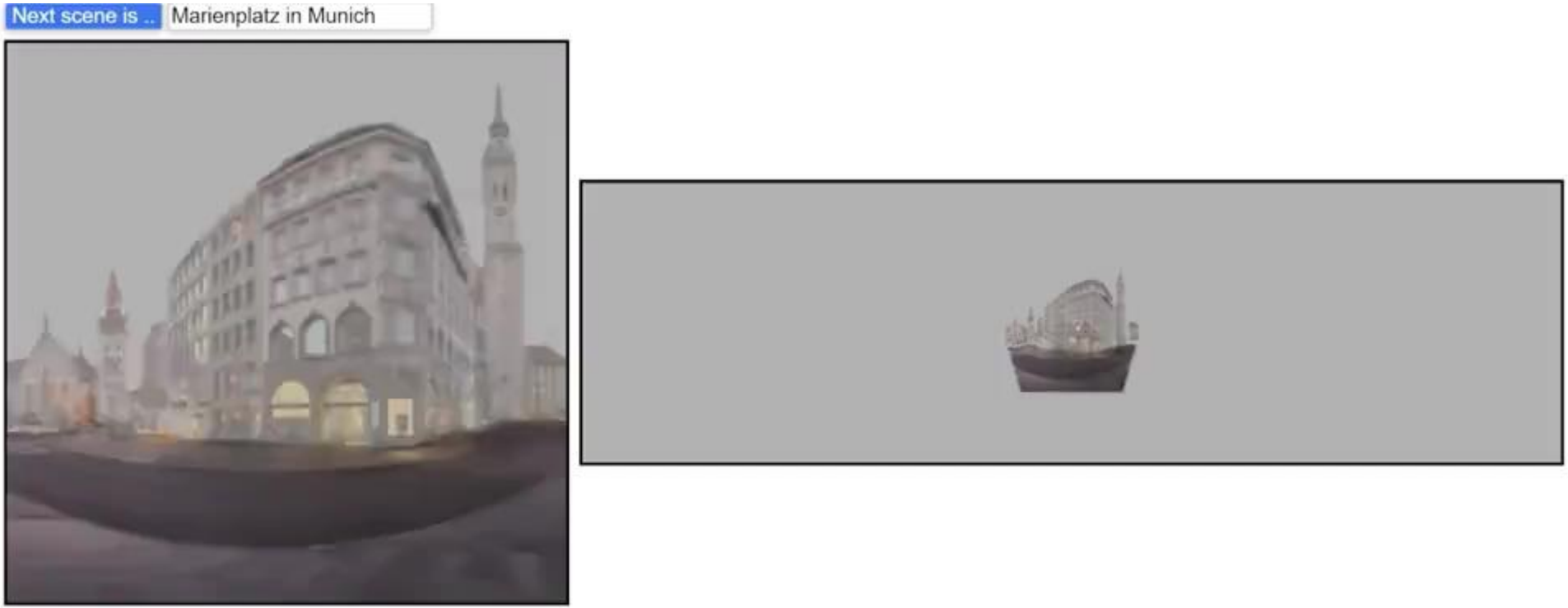
# Text2Room

- Scene Generation: iteratively lift generated images into textured mesh via render-refine-repeat pattern




[Höllein, Cao et al., 23]

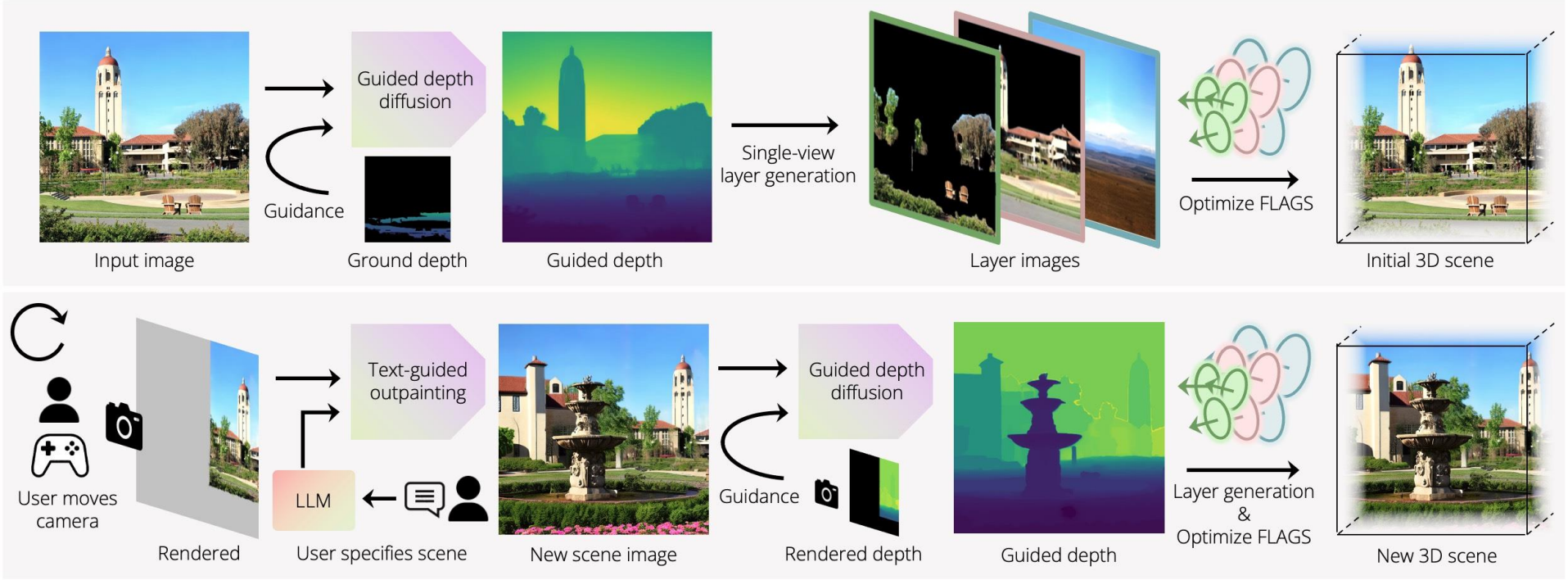
# WonderWorld: Interactive 3D Scene Generation from a Single Image






[Yu et al.. 25] WonderWorld: Interactive 3D Scene Generation from a Single Image

# WonderWorld: Interactive 3D Scene Generation from a Single Image

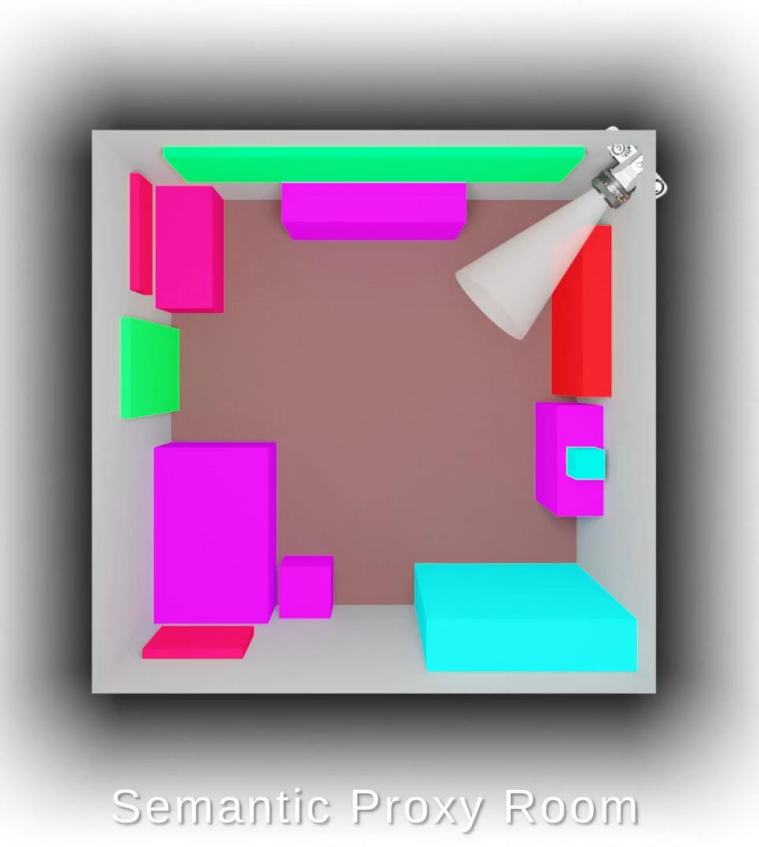



FLAGS = Fast LAyered Gaussian Surfels



[Yu et al.. 25] WonderWorld: Interactive 3D Scene Generation from a Single Image

# ControlRoom3D

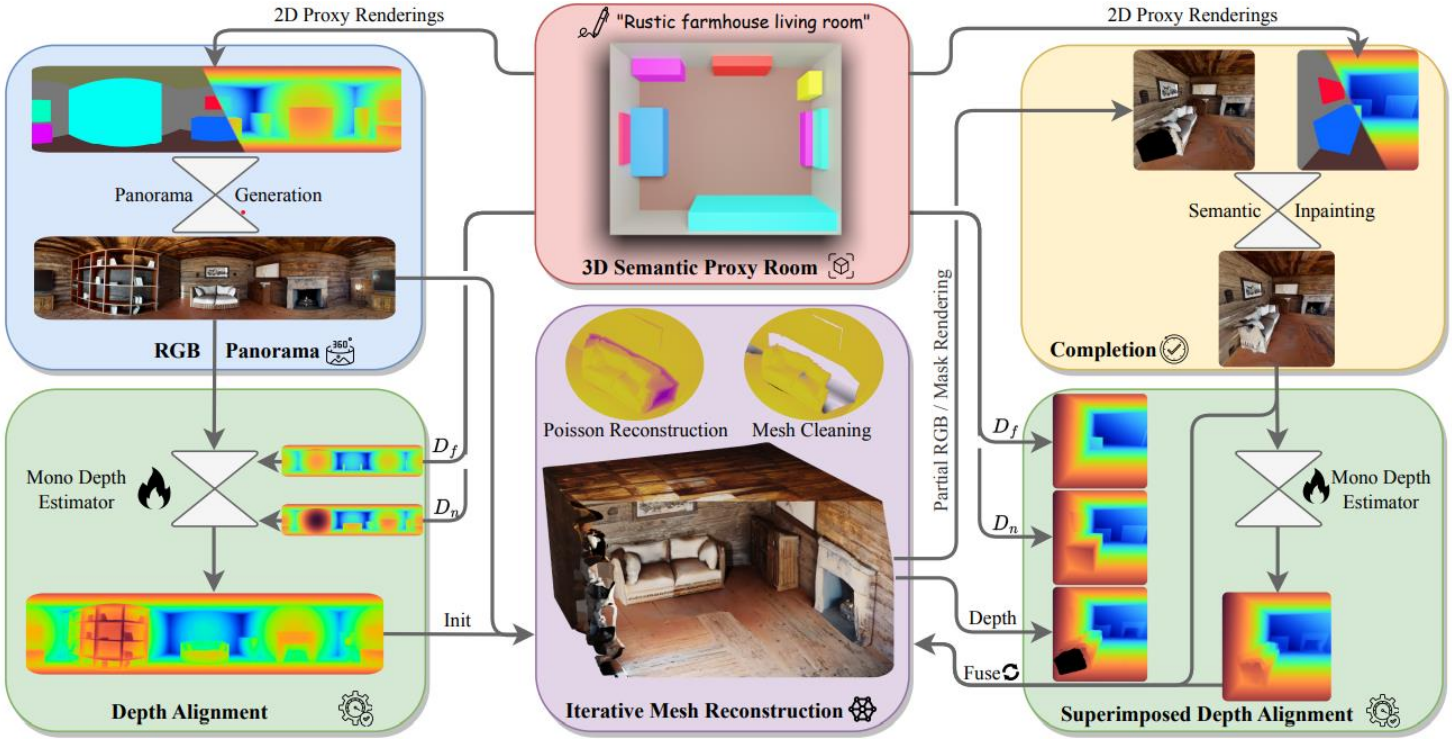





[Schult et al., 24]

# ControlRoom3D

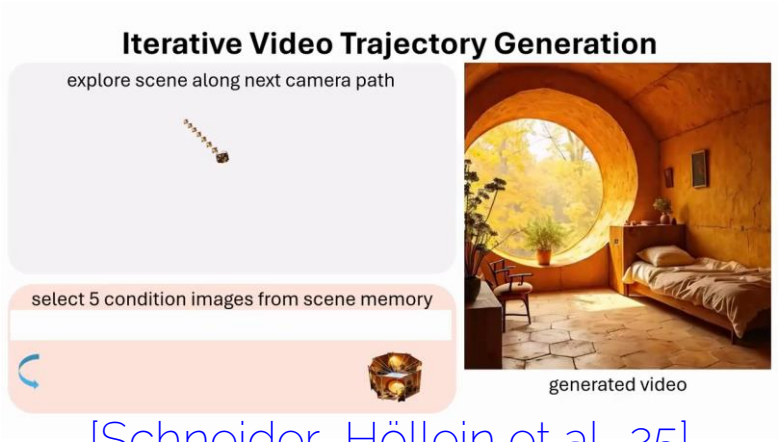
- Scene Generation with semantic control




[Schult et al., 24]

# Video Models For 3D Generation

1. generate videos along camera trajectories (cond on prev data)
2. reconstruct 3D scene from all frames (e.g., 3DGS / NeRF)




[Schneider, Höllein et al., 25]

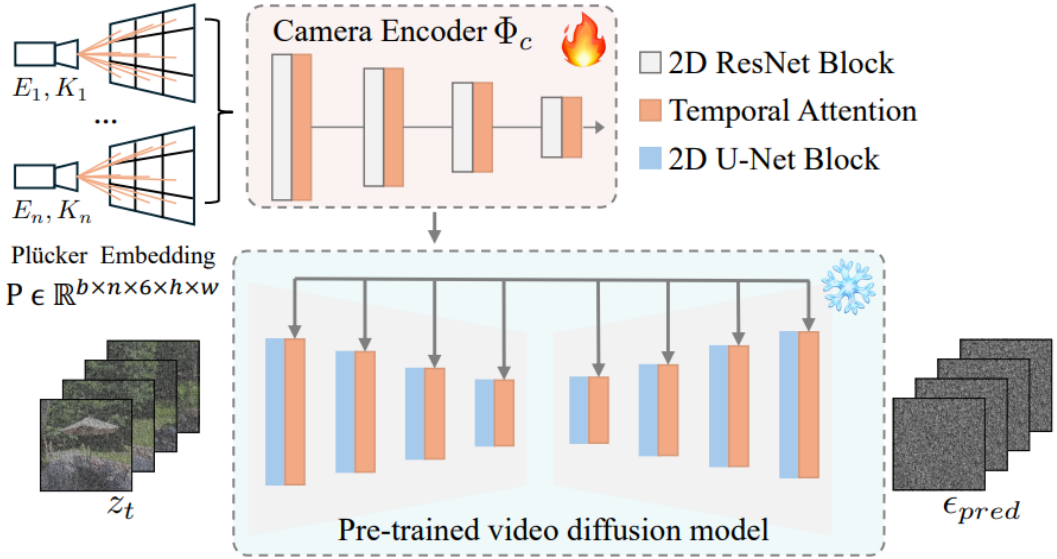
Needed: explicit camera control for video generation

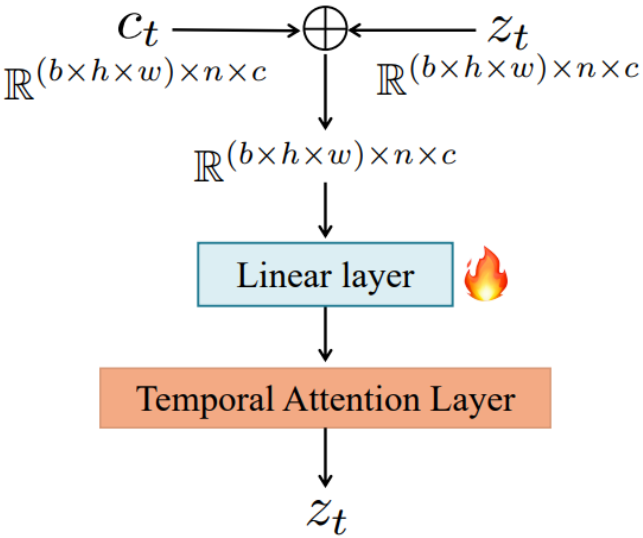
→ Via pixel-wise “raymaps”

→ Via reprojection + inpainting

# CameraCtrl

- Plucker (pixel-wise "raymaps") specifies motion
- Train only a few mapping layers + camera encoder

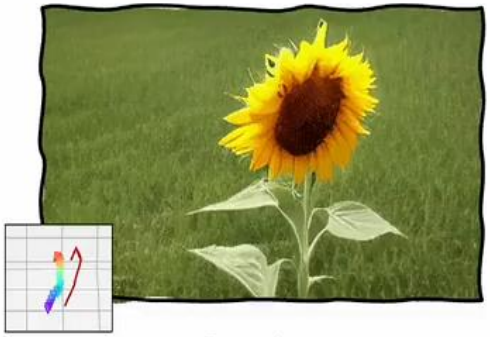



[He et al., 24]

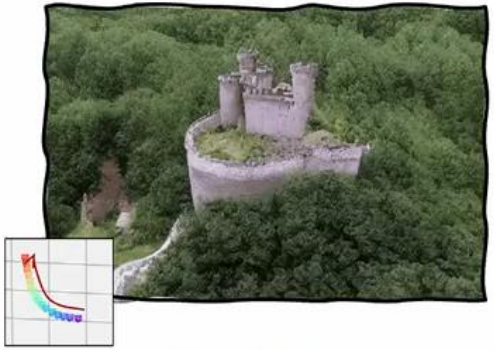
A squirrel is eating pine nuts.

A still life of vintage objects on a wooden table.

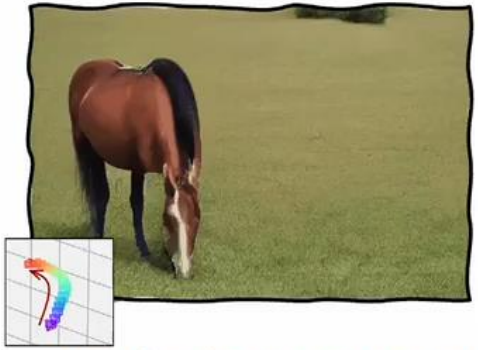


The sunflower in the sun.

A pair of worn leather boots on a porch.



A castle in the forest.

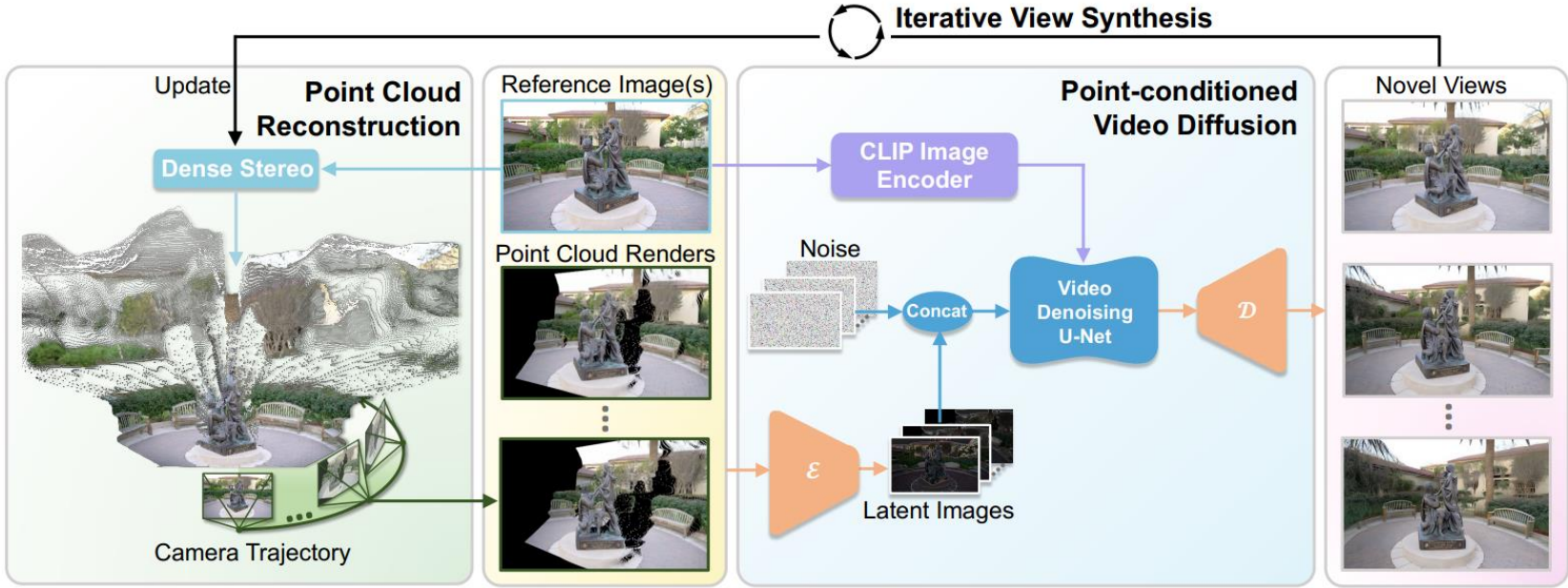


A horse is eating grass on the grassland.

[He et al., 24]

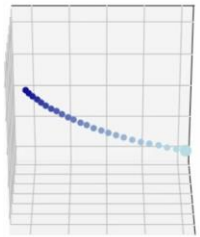
# ViewCrafter

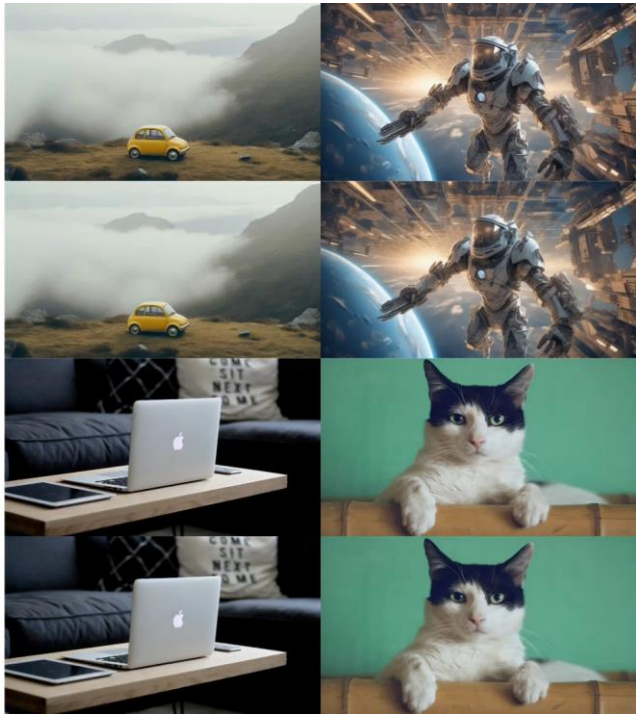
- Pointcloud renders specify camera motion
- Finetune video model on this inpainting task

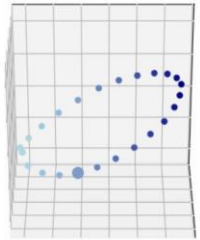


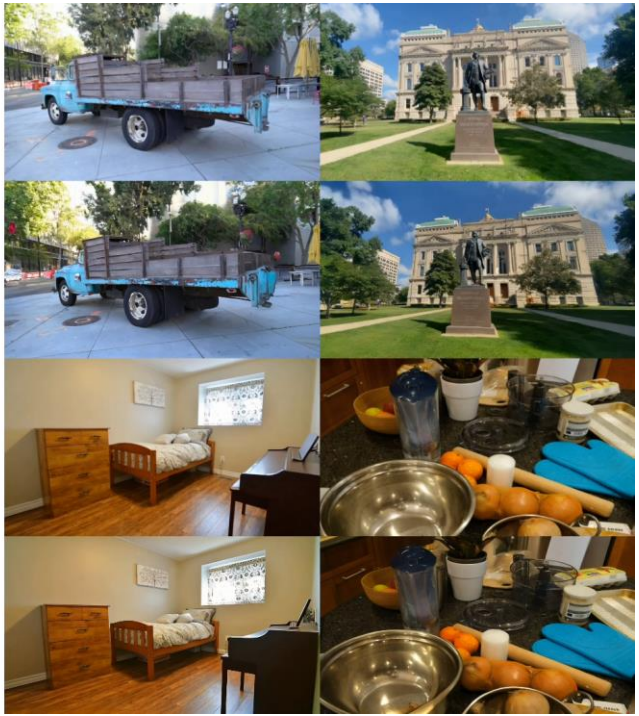

[Yu et al., 24]

# ViewCrafter









[Yu et al., 24]

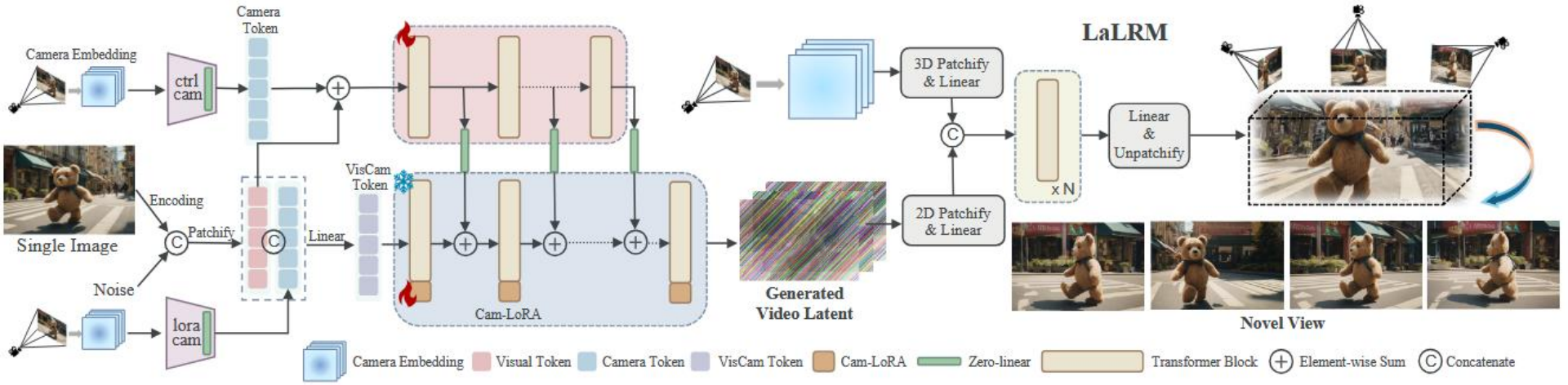
# Wonderland: Navigating 3D Scenes from a Single Image



[Liang et al.. 24] Wonderland: Navigating 3D Scenes from a Single Image

# Wonderland: Navigating 3D Scenes from a Single Image

Dual-branch camera conditioning




Camera-guided video diffusion model

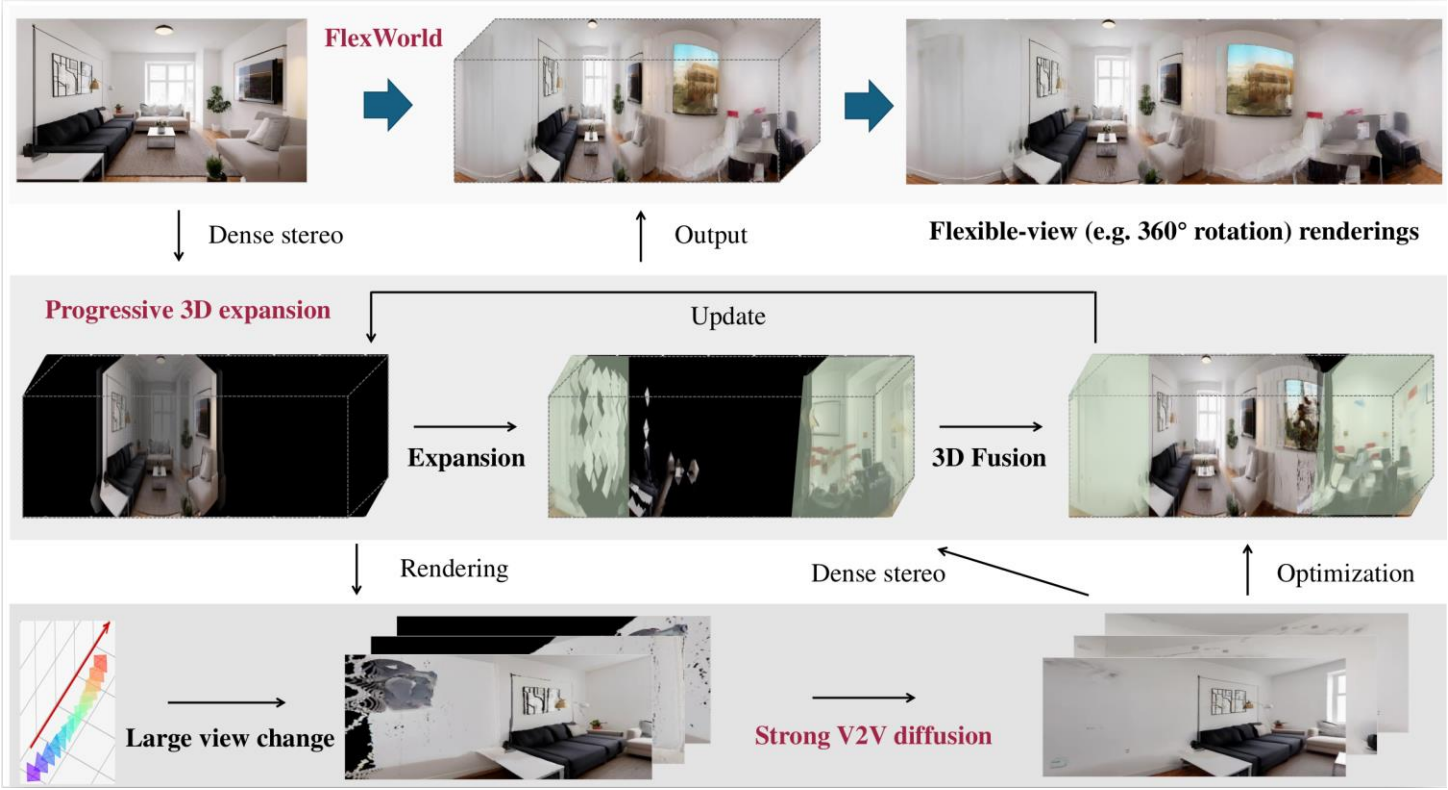
Feed-forward reconstruction with latent large reconstruction model (LaLRM)



[Liang et al., 24] Wonderland: Navigating 3D Scenes from a Single Image

# FlexWorld: Progressively Expanding 3D Scenes for Flexiable-View Synthesis

# FlexWorld: Progressively Expanding 3D Scenes for Flexiable-View Synthesis



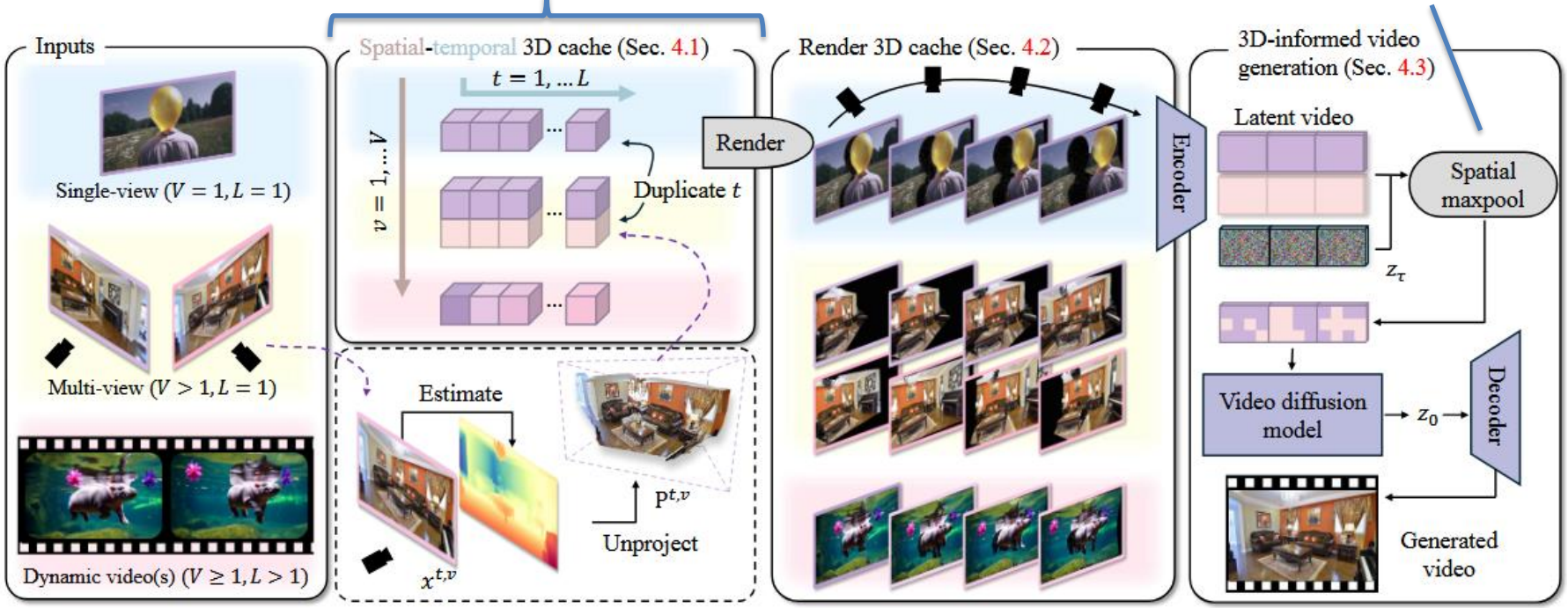
# GEN3C: 3D-Informed World-Consistent Video Generation with Precise Camera Control

# GEN3C: 3D-Informed World-Consistent Video Generation with Precise Camera Control

Spatial-temporal cache: one colored 3D point cloud per input view

Fuse renderings from different point clouds, by max pooling in latent space

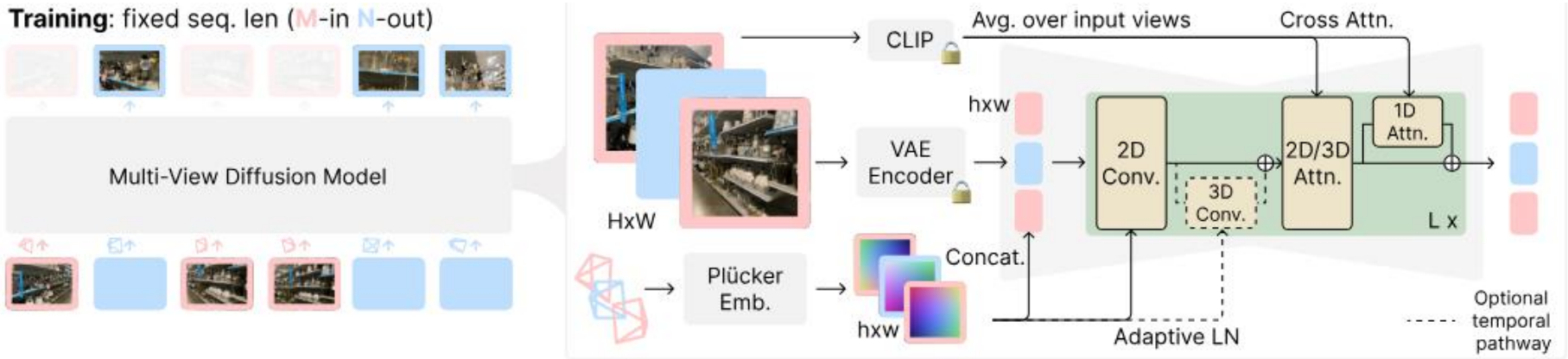
Multiple tasks supported



# Stable Virtual Camera: Generative View Synthesis with Diffusion Models

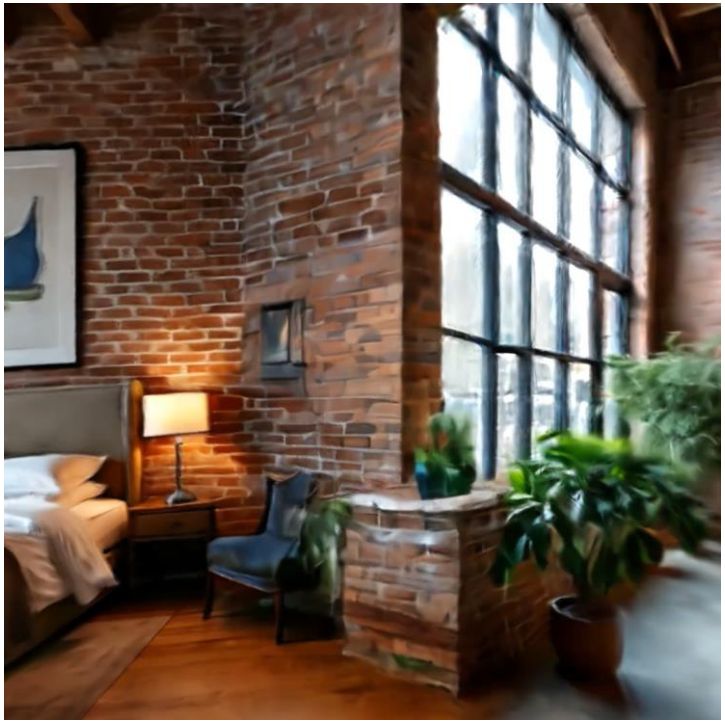
Input: single image

# Stable Virtual Camera: Generative View Synthesis with Diffusion Models



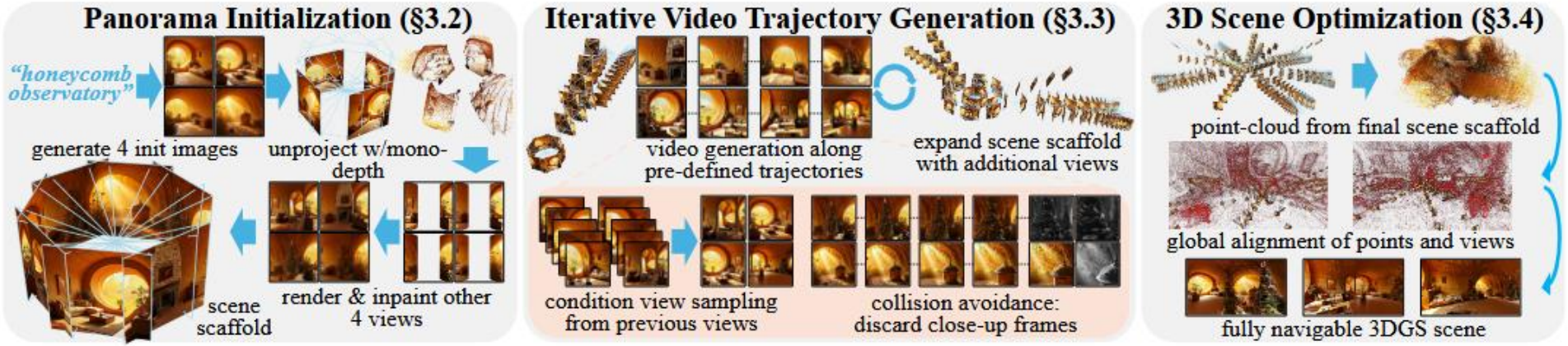
# WorldExplorer: Towards Generating Fully Navigable 3D Scenes

apartment of a bioluminescent, gravity-defying, telepathic cosmic jellyfish hive



apartment of an urban industrial loft with exposed brick and ductwork

# WorldExplorer: Towards Generating Fully Navigable 3D Scenes



# Panoramas For World Generation

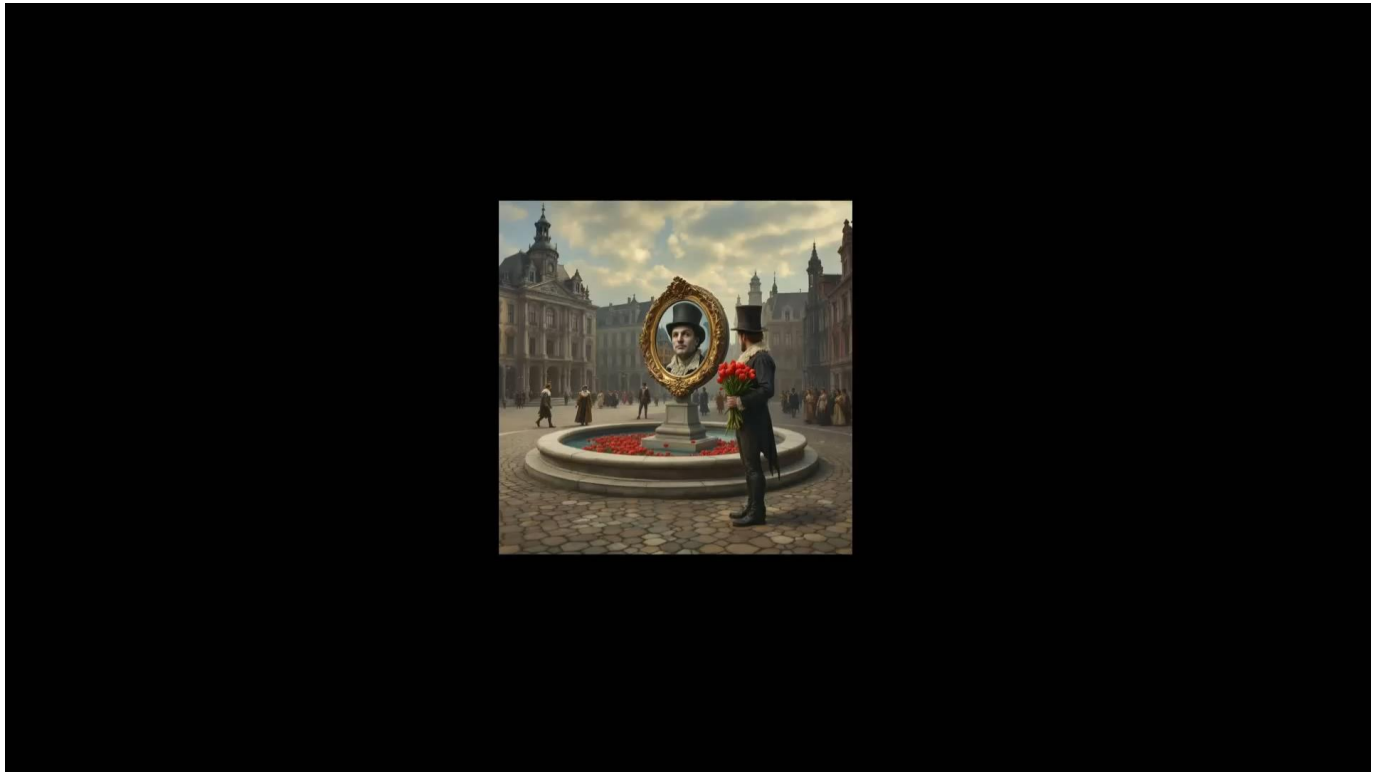
Panoramas capture significantly wider field of view

→ Useful for world generation: less iterations necessary

Two flavours:

1. Finetune diffusion model (MVDiffusion, DreamScene360, LayerPano3D, Matrix-3D)
2. Iterative Inpainting (Diffusion360, SimpleRecipe, WorldExplorer)

# World Generation From A Single Panorama





[Schwarz et al.. 25] A Recipe for Generating 3D Worlds From a Single Image

# World Generation From Multiple Panoramas

Input Image | Panoramic Video | 3D Scene



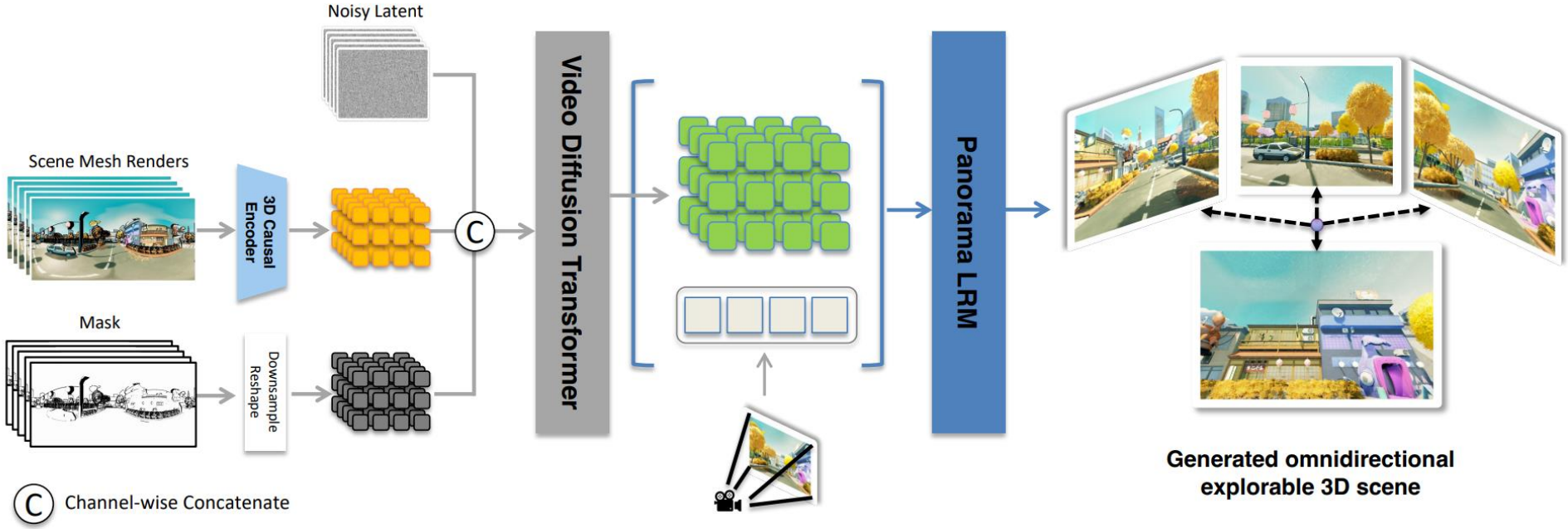





[Yang et al.. 25] Matrix-3D: Omnidirectional Explorable 3D World Generation

# World Generation From Multiple Panoramas

- Camera Control via mesh renders
- Finetune video diffusion for panorama generation
- LRM to convert to 3DGS scene






[Yang et al.. 25] Matrix-3D: Omnidirectional Explorable 3D World Generation

# Minute-Long Consistent Video Generation

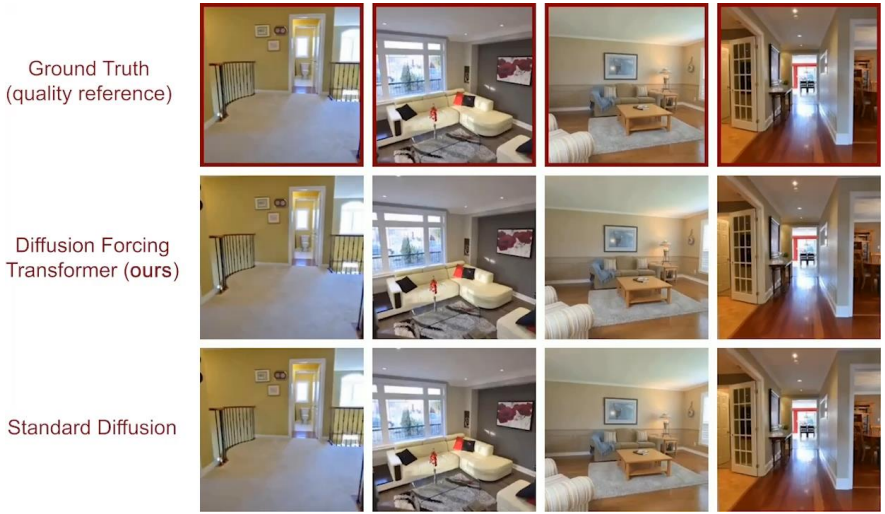
Generation purely with a video model.

Do we still need a 3D representation?

[Google DeepMind, 2025] Genie 3: A new frontier for world models

# History-Guided Video Diffusion

Autoregressive generation with video diffusion diverges at some point (error accumulation)




862-frame video generation

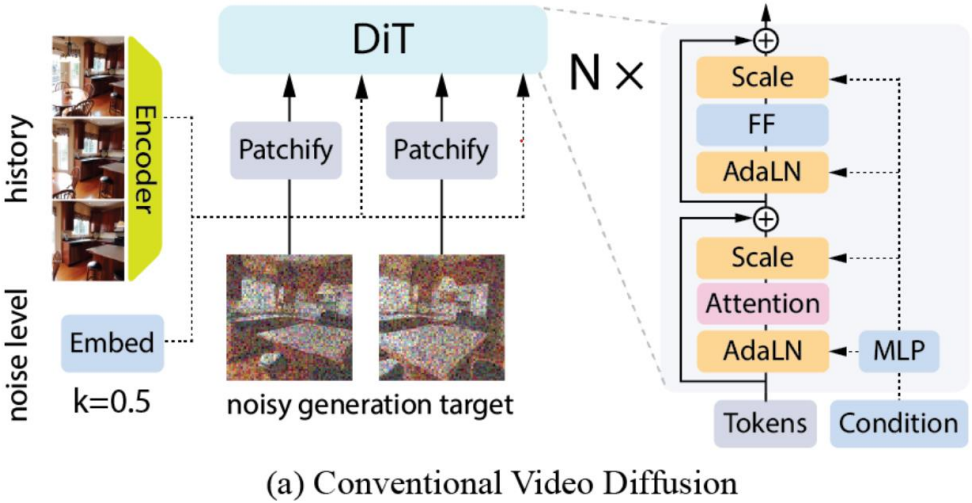
[Chen et al.. 24] Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion
[Song et al.. 25] History-Guided Video Diffusion

# History-Guided Video Diffusion

During Training: history are perfect camera images

Inference out-of-distribution: history are prev. generations (imperfect)

Diffusion Forcing: train with random noise levels per-frame

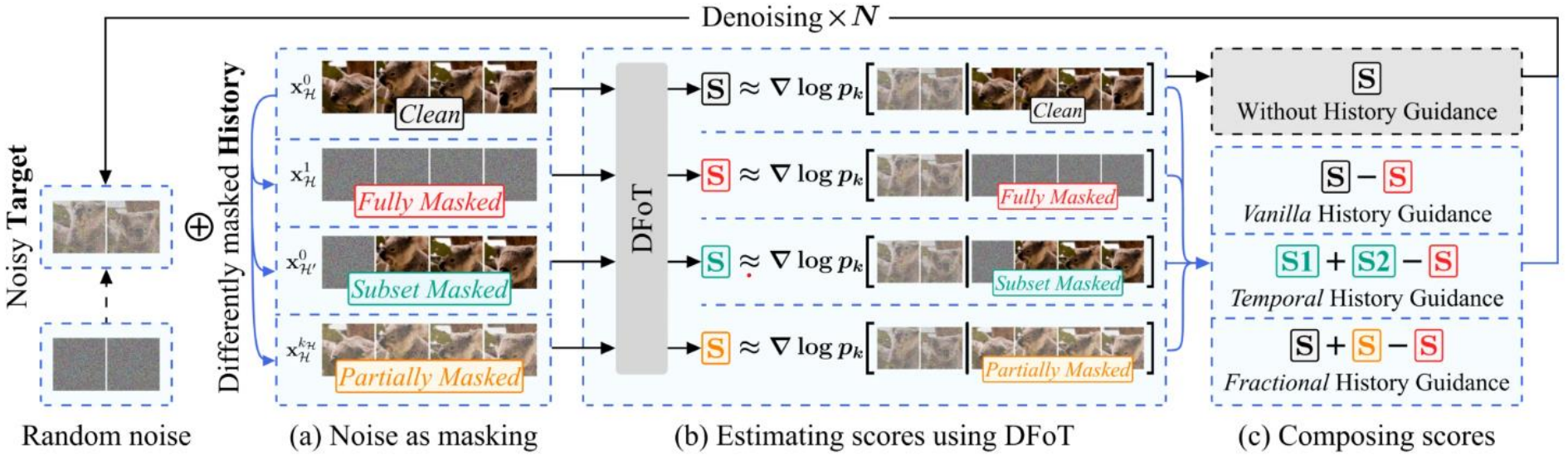



[Chen et al., 24] Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion

[Song et al., 25] History-Guided Video Diffusion

# History-Guided Video Diffusion

Inference fixed: history guidance with different amounts of noise added to previous frames (*hides frame artifacts in the noise*)





[Chen et al.. 24] Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion

[Song et al.. 25] History-Guided Video Diffusion

# 4D Generation

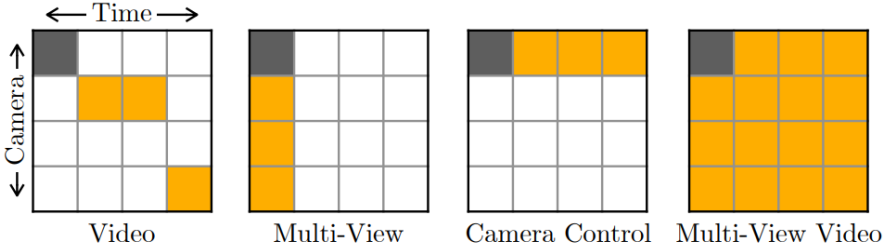



[Wu et al.. 24] CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models

# Cat4D

Curate multi-view and video datasets and train a joint diffusion model


| Input views | | Target views | | Real data | Synthetic data |
|---|---|---|---|---|---|
| Camera motion | Scene motion | Camera motion | Scene motion | | |
| ✔ | ✘ | ✔ | ✘ | CO3D [55], MVImgNet [84] Re10K [95], MC4K [36] | Kubric [19], Objaverse [12] |
| ✘ | ✔ | ✘ | ✔ | static-view videos | |
| ✔ | ✔ | ✔ | ✔ | - | |
| ✔ | ✔ | ✔ | ✘ | CO3D augmented with Lumiere [7] | |
| ✔ | ✔ | ✘ | ✔ | static-view videos augmented with CAT3D [17] | |
| ✔ | ✔ | ✘ | ✘ | single image | |






[Wu et al., 24] CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models

# Administratives

# Current "DL Curriculum"

- MA Semester 1: I2DL (+ various intro lectures)
- MA Semester 2: ADL4CV
- MA Semester 3: Practical and/or Guided Research
- MA Semester 4: Master Thesis

Clearly should be Bachelor…

# Future Project Questions

- Check out our research & see what interests you https://niessnerlab.org/publications.html
- Check out practical courses (e.g., DLinVC, etc.) https://niessnerlab.org/teaching.html
- Directly apply for GR/IDP/MA topics: https://application.vc.in.tum.de/master-application
- Check out theses topics: https://docs.google.com/document/d/1lA8WjCYemQ9Cluwgi2fqmviK4pcBul8N_MPxFhIVoSU/edit?usp=sharing
- Reach out to PhD students / me!

# Thanks for watching!