DMV3D: Denoising Multi-view Diffusion Using **3D Large Reconstruction Model**

ICLR 2024 submission (avg. score 8)

&

LRM: **Large Reconstruction Model** for Single Image to 3D

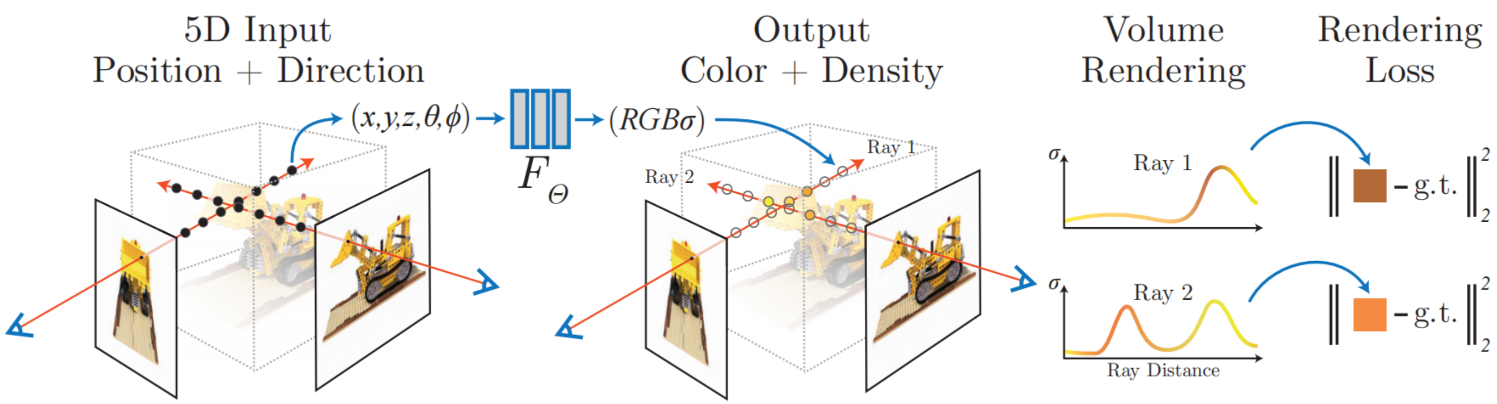
ICLR 2024 submission (avg. score 8.5)

Presenter: Rundong Luo

2023.12.10

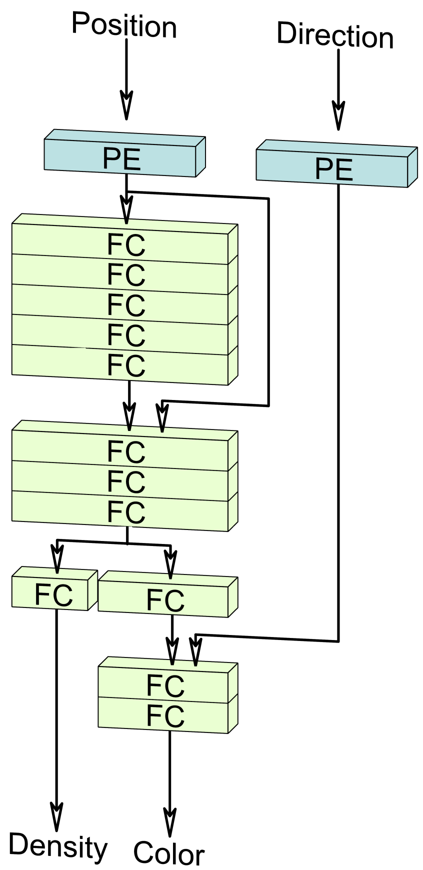
Background: NeRF

Overview



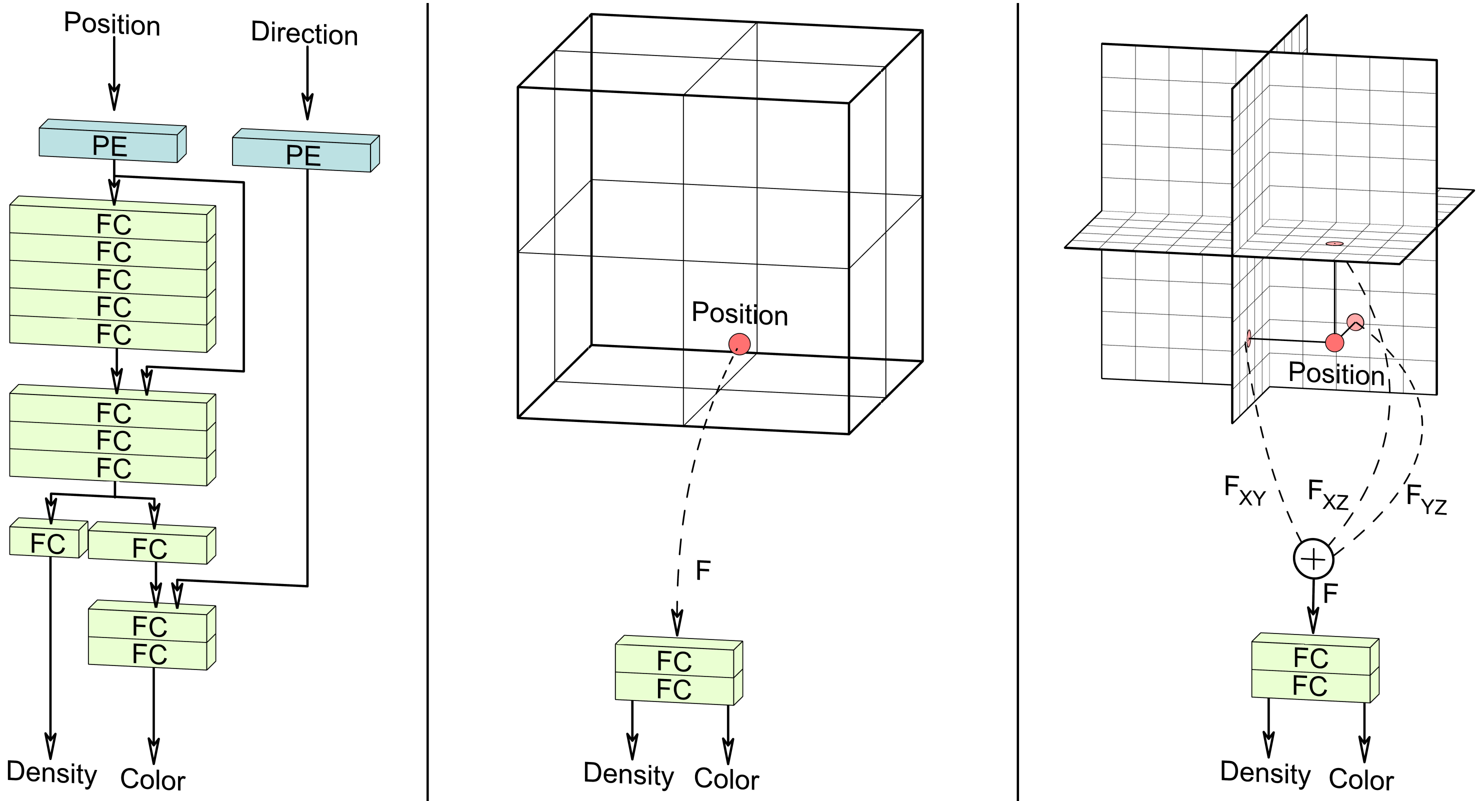
Volume rendering

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,,\ \text{where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right).$$



(a) NeRF (Implicit)

Background: Triplane Representation for NeRF



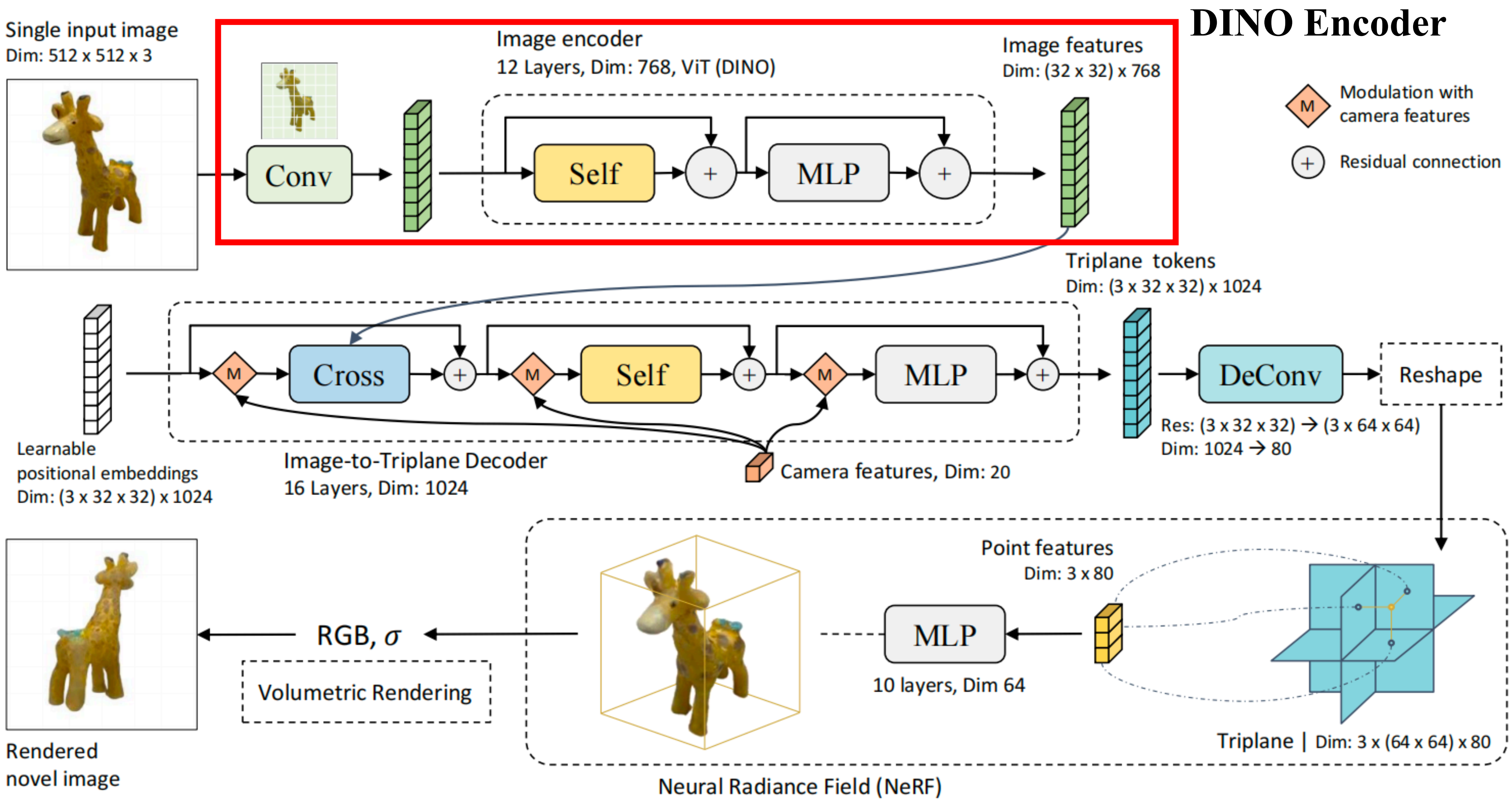
(a) NeRF (Implicit) | (b) Voxels (Explicit or Hybrid) | (c) Ours (Hybrid)

"The primary advantage of this hybrid representation is efficiency."

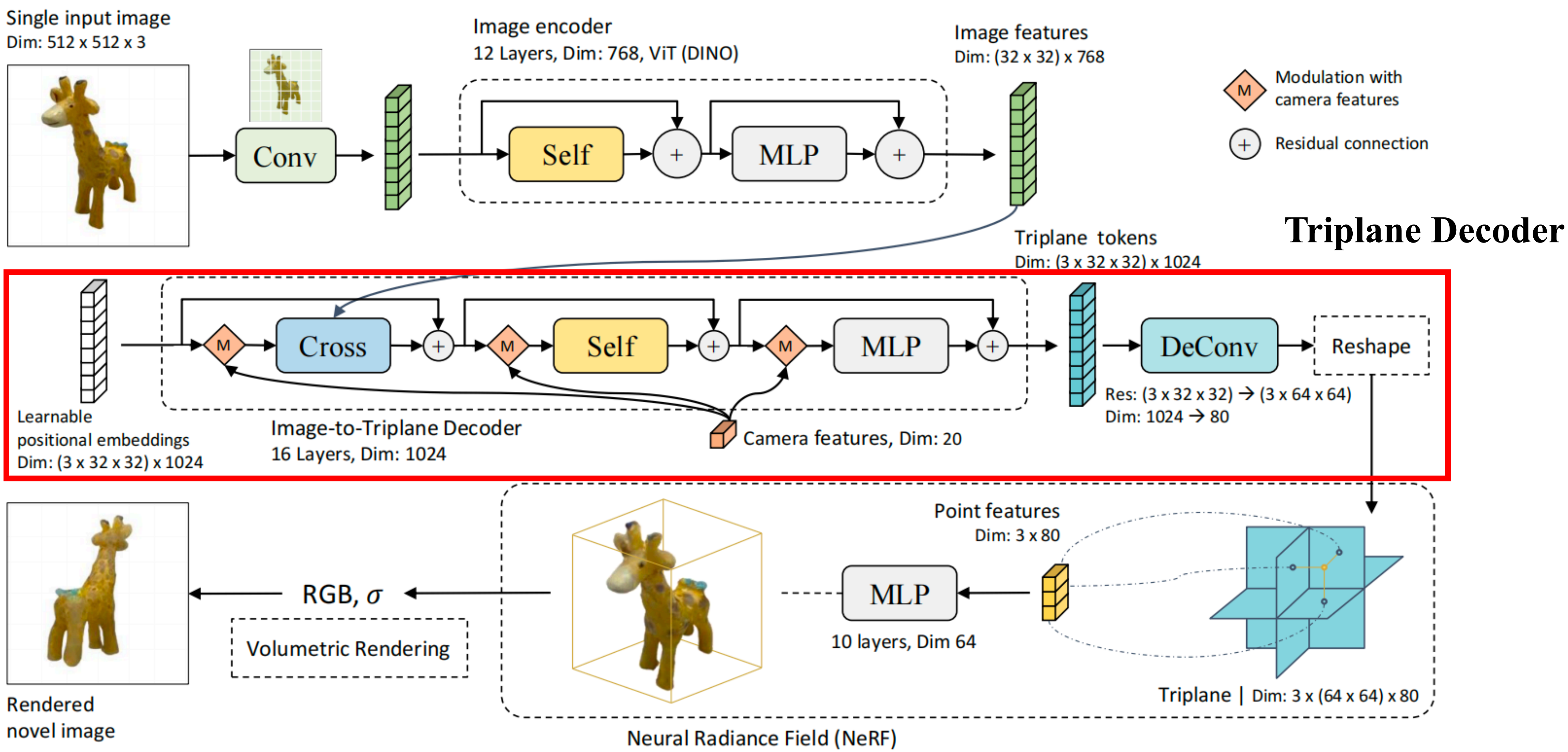
Large Reconstruction Model: Task Description and Overview

- Task: Single-image to 3D (Triplane representation → Mesh)
- Overview: The first large-scale (500M params) 3D reconstruction model
  - Trained on one million 3D shapes and video data across diverse categories
  - Category-agnostic
  - Training objective: simple L2 reconstruction loss
- Performance: Can reconstruct high-fidelity 3D shapes from a wide range of images captured in the real world in five seconds

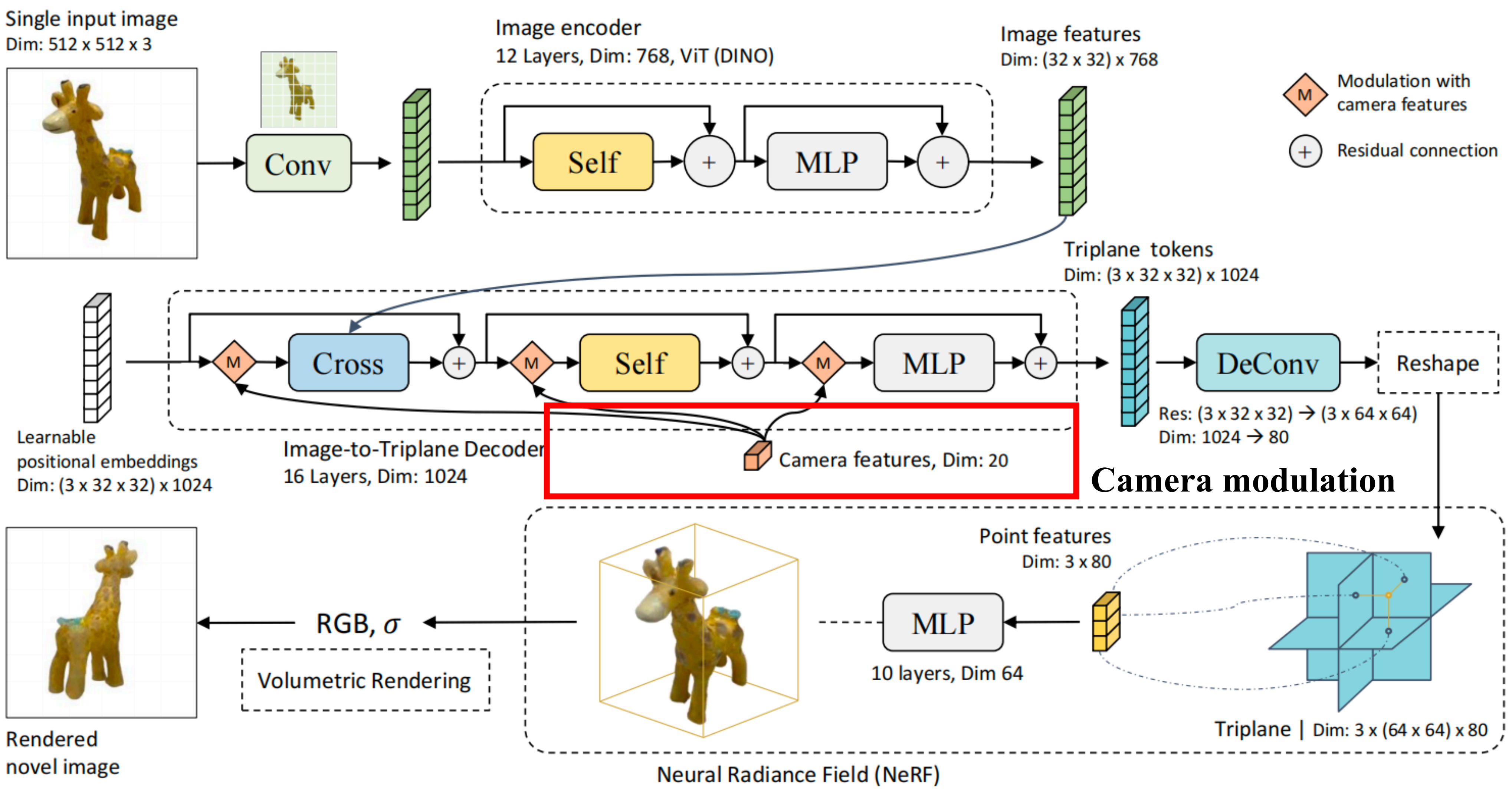
Large Reconstruction Model: Pipeline



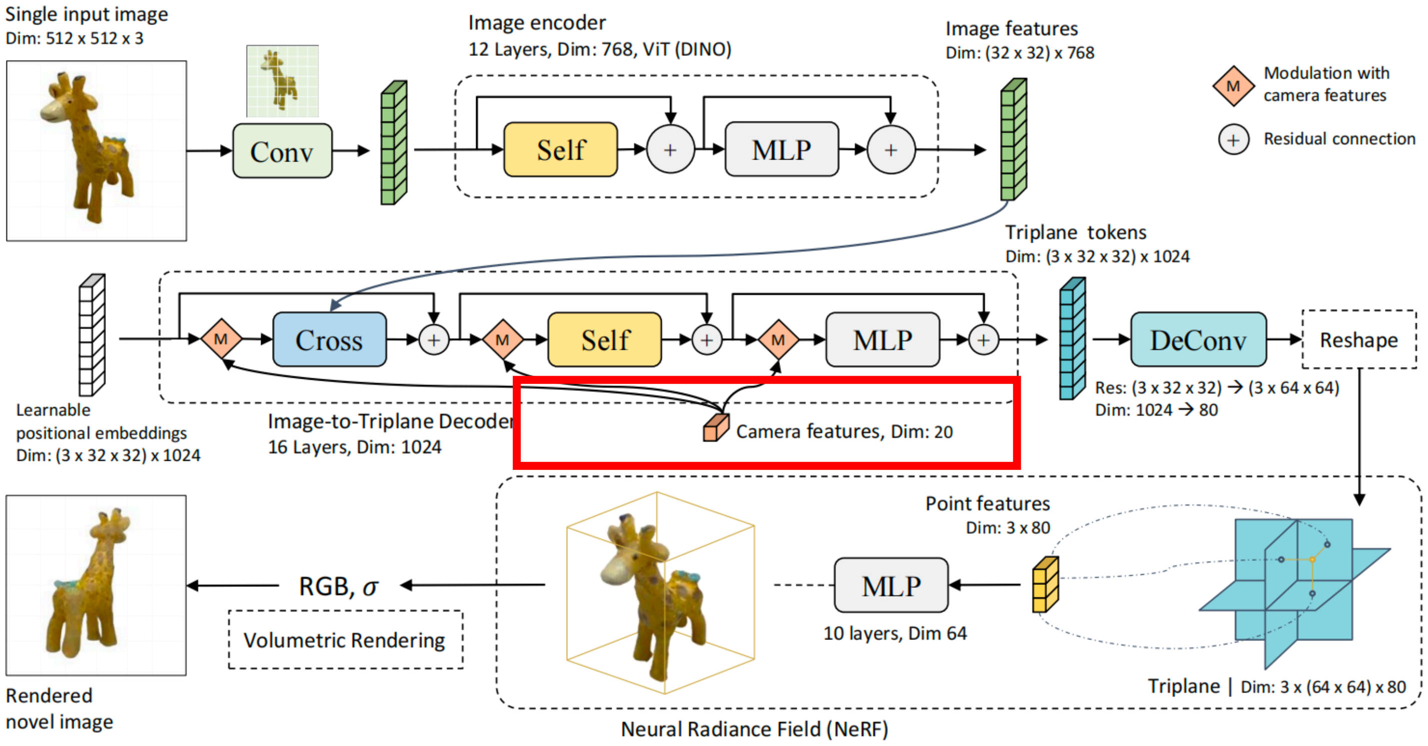
Large Reconstruction Model: Pipeline



Large Reconstruction Model: Pipeline



Large Reconstruction Model: Pipeline



Camera modulation (AdaLN)

$$\boldsymbol{c} = [\boldsymbol{E}_{1\times16}, foc_x, foc_y, pp_x, pp_y]$$

$$\gamma, \beta = \mathrm{MLP}^{\mathrm{mod}}(\tilde{\boldsymbol{c}})$$

$$\mathrm{ModLN}_c(\boldsymbol{f}_j) = \mathrm{LN}(\boldsymbol{f}_j) \cdot (1+\gamma) + \beta$$

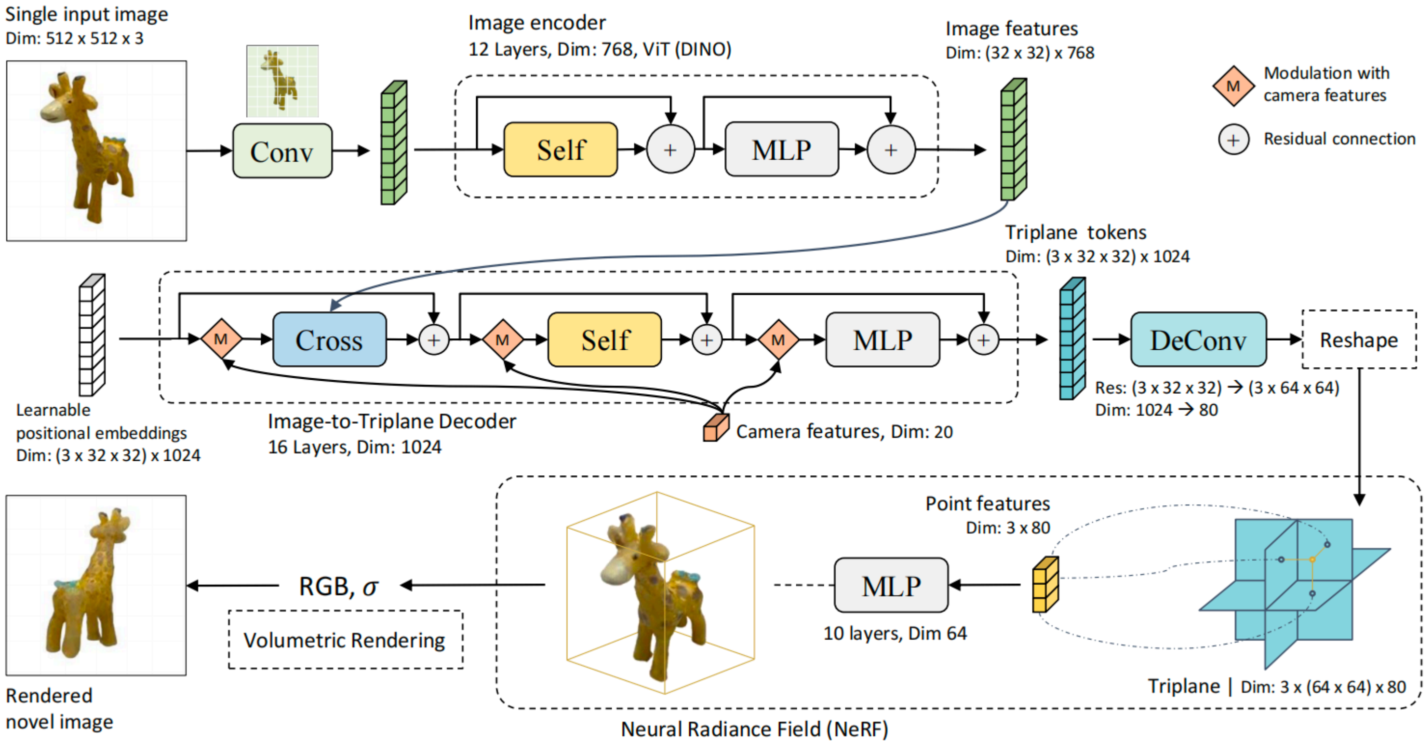
Decoder (overall)

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

$$\boldsymbol{f}_j^{self} = \mathrm{SelfAttn}(\mathrm{ModLN}_c(\boldsymbol{f}_j^{cross}); \{\mathrm{ModLN}_c(\boldsymbol{f}_{j'}^{cross})\}_{j'}) + \boldsymbol{f}_j^{cross}$$

$$\boldsymbol{f}_j^{out} = \mathrm{MLP}^{tfm}(\mathrm{ModLN}_c(\boldsymbol{f}_j^{self})) + \boldsymbol{f}_j^{self}$$

Large Reconstruction Model: Pipeline



Training objective

$$\mathcal{L}_{\text{recon}}(\boldsymbol{x}) = \frac{1}{V}\sum_{v=1}^{V}\left(\mathcal{L}_{\text{MSE}}(\hat{\boldsymbol{x}}_v, \boldsymbol{x}_v^{GT}) + \lambda\mathcal{L}_{\text{LPIPS}}(\hat{\boldsymbol{x}}_v, \boldsymbol{x}_v^{GT})\right)$$

Large Reconstruction Model: Experiments

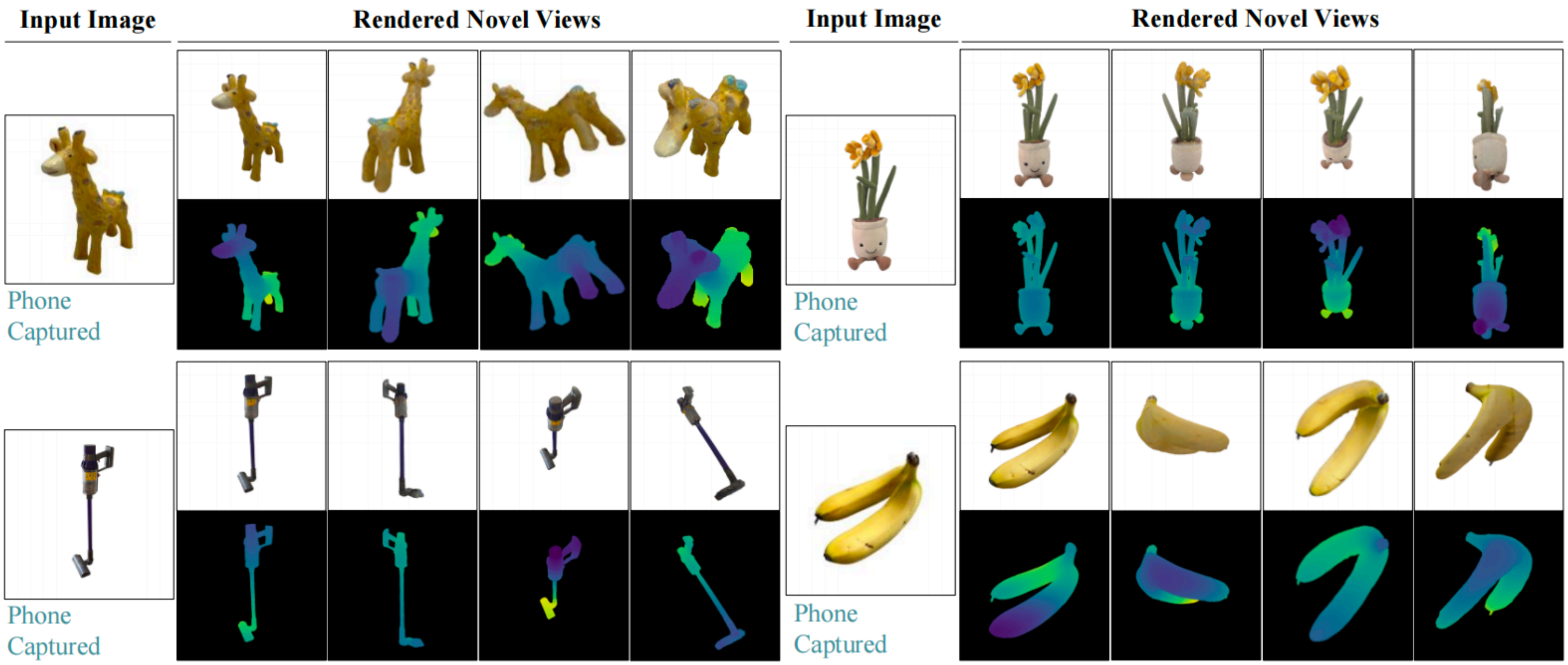
Datasets

- Objaverse (~730k object meshes)
- MVImgNet (~220k object-centric videos)

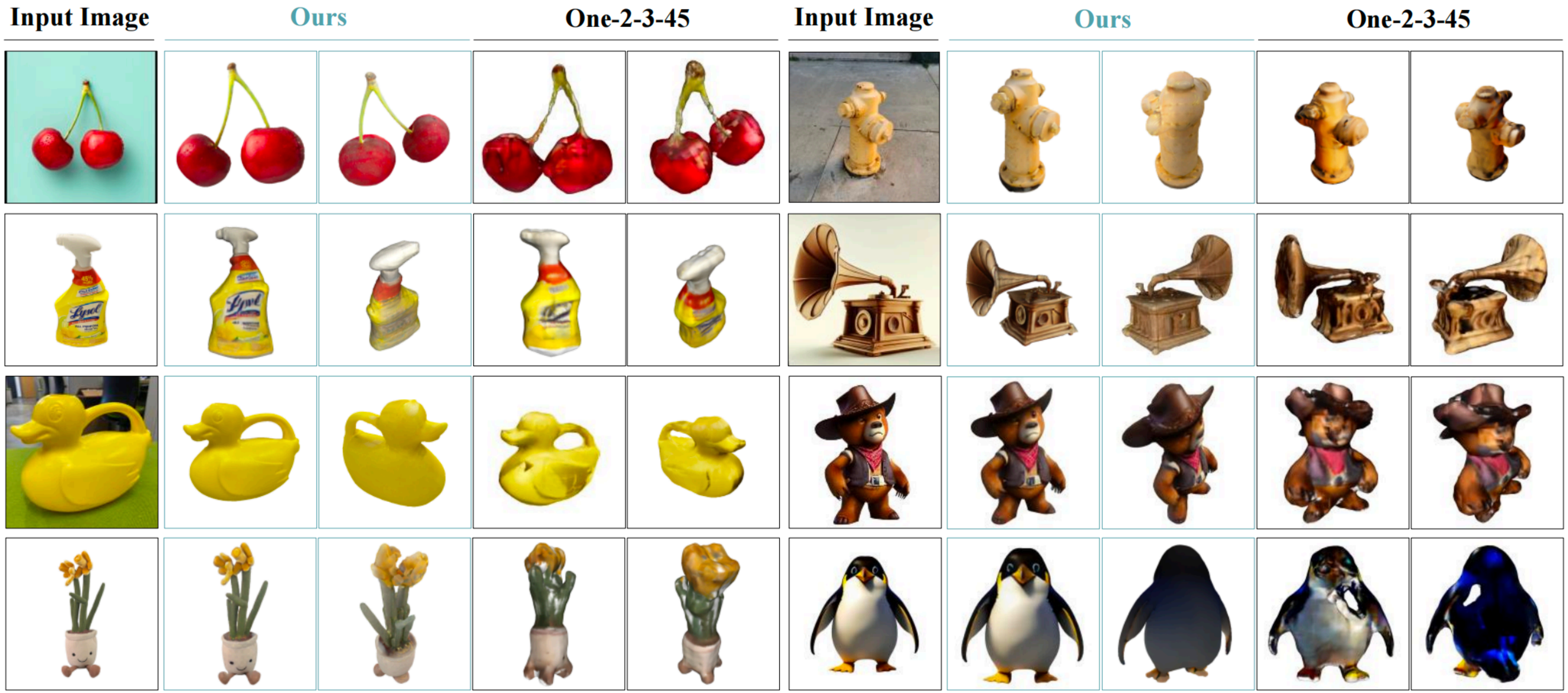
Pre-processing: remove background

GPU: 128 A100, 3 days 

Large Reconstruction Model: Experiments



Large Reconstruction Model: Experiments



Denoising Multi-view Diffusion: Task Description

- Task: Single-image/Text to 3D (Triplane representation → Mesh)
- Overview: Single-stage framework that leverages multi-view 2D image diffusion model to achieve 3D generation;
  - Trained on one million 3D shapes and video data across diverse categories
  - Training objective: simple L2 reconstruction loss
  - Probabilistic approach, *i.e.*, multiple reasonable 3D outputs given the same input
- Performance: High-quality text-to-3D generation and single-image reconstruction through direct model inference within 30 seconds on an A100 GPU.

Denoising Multi-view Diffusion: Task Description

Input (posed multi-view images): $\mathcal{I} = \{\mathbf{I}_1, ..., \mathbf{I}_N\}$ $\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_N\}$

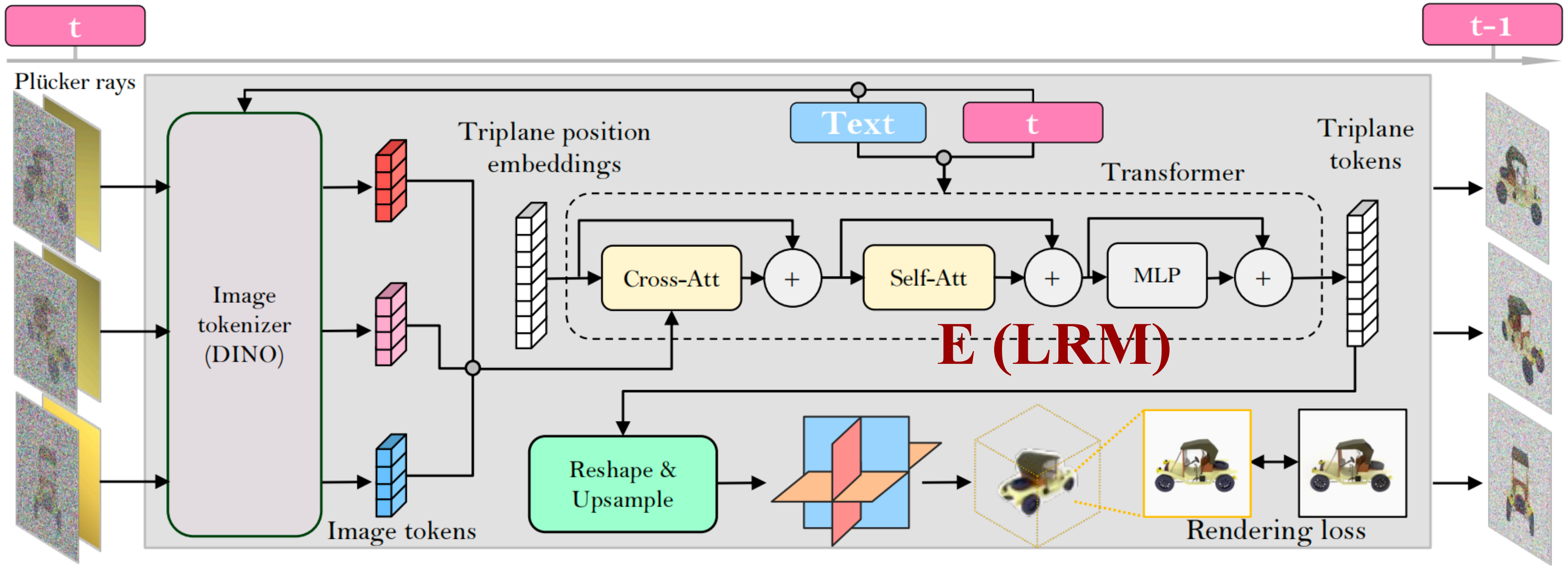
Diffusion process (forward): $\mathcal{I}_t = \{\sqrt{\bar{\alpha}_t}\mathbf{I} + \sqrt{1-\bar{\alpha}_t}\epsilon_{\mathbf{I}} | \mathbf{I} \in \mathcal{I}\}$

Denoising process: $\mathbf{I}_{r,t} = \mathrm{R}(\mathrm{S}_t, \boldsymbol{c}), \quad \mathrm{S}_t = \mathrm{E}(\mathcal{I}_t, t, \mathcal{C})$

E: 3D reconstruction module

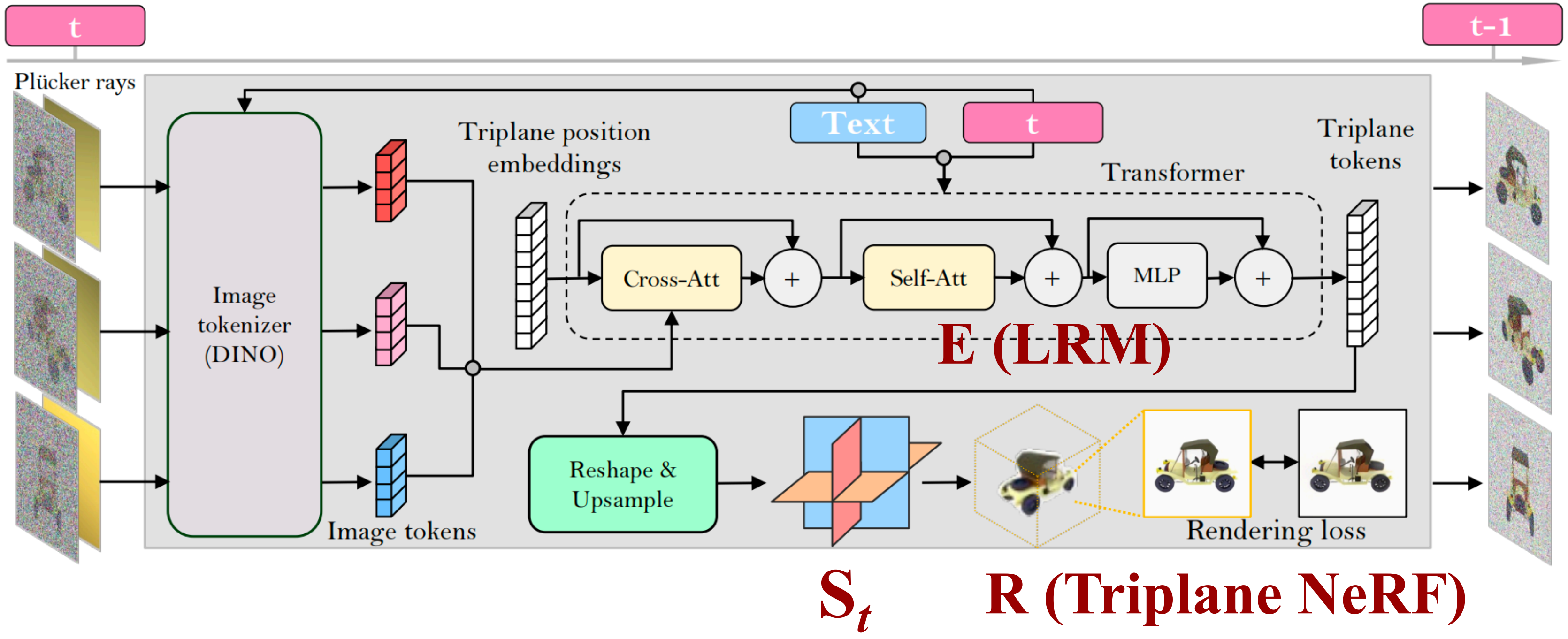
S_t: 3D representations

R: Rendering module



$\mathbf{S}_t$ R (Triplane NeRF)

Denoising Multi-view Diffusion: Method



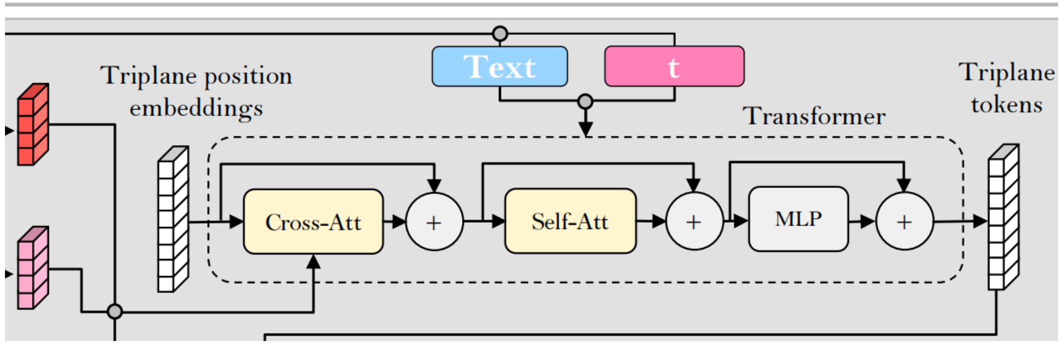
- Time conditioning: AdaLN block
- Camera conditioning: concatenate Plucker Coordinates with input pixels

$$\boldsymbol{r} = (\boldsymbol{o} \times \boldsymbol{d}, \boldsymbol{d})$$

Denoising Multi-view Diffusion: Method

Conditional generation

- Single-image condition
  - Keep the condition image noise-free
  - Align the Triplane coordinates with the condition view's coordinates
  - Normalize input camera view as LRM does
- Text condition
  - Use the CLIP text encoder to obtain text embeddings.
  - Inject them into the denoiser using cross-attention.



Denoising Multi-view Diffusion: Method

Conditional generation

- Single-image condition
  - Keep the condition image noise-free
  - Align the Triplane coordinates with the condition view's coordinates
  - Normalize input camera view as LRM does
- Text condition
  - Use the CLIP text encoder to obtain text embeddings.
  - Inject them into the denoiser using cross-attention.
- Training objective:

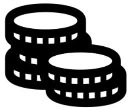
$$\mathrm{L} = \mathbb{E}_{t \sim U[1,T], (\mathbf{I}, \mathbf{c}) \sim (\mathcal{I}_{full}, \mathcal{C}_{full})} \ \ell\big(\mathbf{I}, \mathrm{R}(\mathrm{E}(\mathcal{I}_t, t, \mathcal{D}, y), \mathbf{c})\big)$$

Denoising Multi-view Diffusion : Experiments

Datasets

- Objaverse (~730k object meshes)
- MVImgNet (~220k object-centric videos)
- Cap3D (~660k image & caption pairs)

GPU: 128 A100, 7 days



Denoising Multi-view Diffusion : Experiments

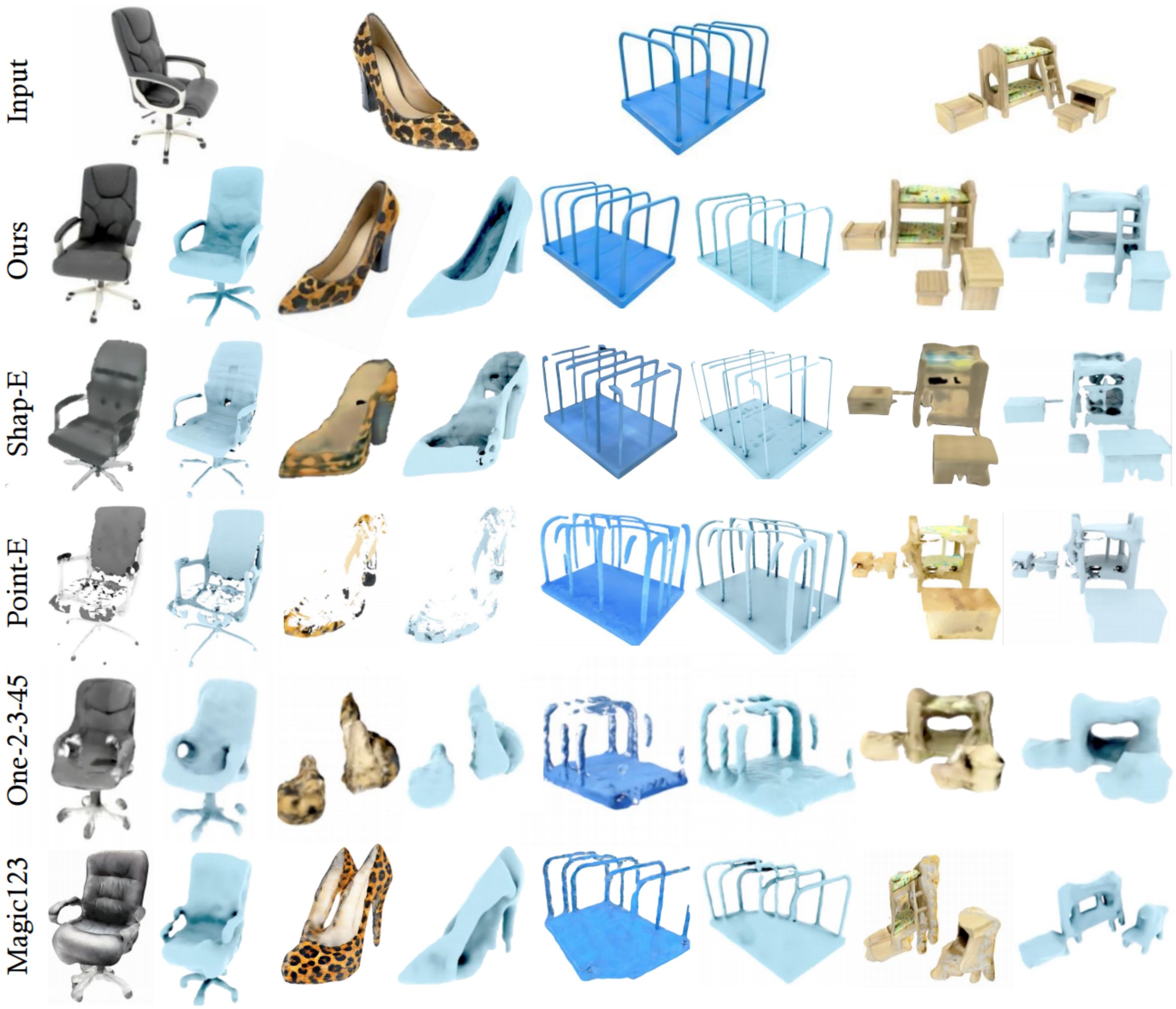


Figure 4: **Qualitative comparisons on single-image reconstruction.**

Denoising Multi-view Diffusion : Experiments

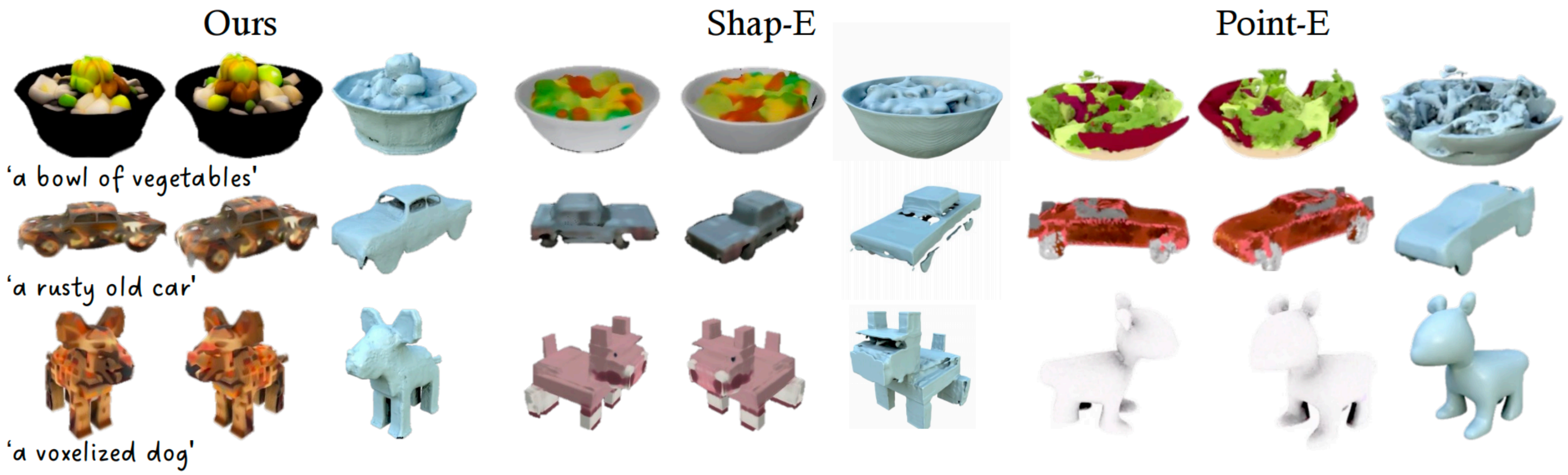


Figure 5: **Qualitative comparisons on Text-to-3D .**

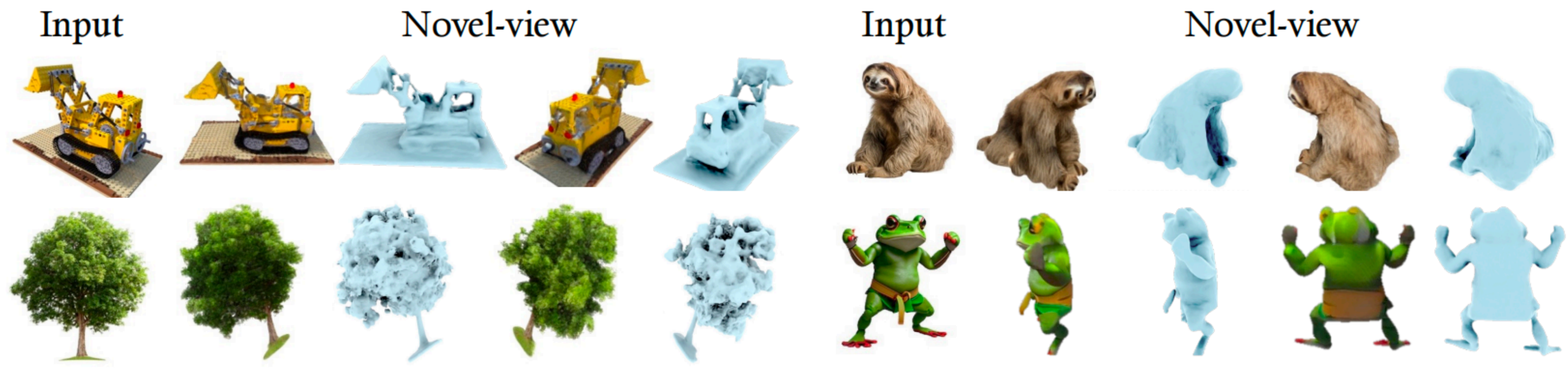


Figure 6: **Robustness to out-of-domain inputs**: synthetic (top left), real (bottom left, top right), and generated images (bottom right).

Thanks for listening!

Presenter: Rundong Luo
2023.12.10