EUROPEAN CONFERENCE ON COMPUTER VISION

MILANO
2024

TexGen: Text-Guided 3D Texture Generation with Multi-view Sampling and Resampling

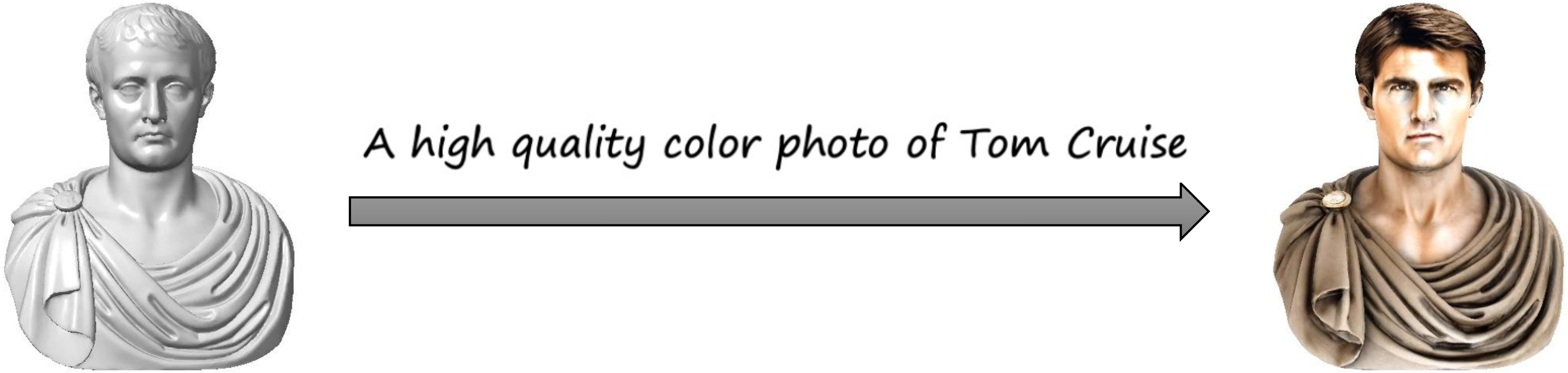
ECCV 2024

Dong Huo[1,3], Zixin Guo[2], Xinxin Zuo[3], Zhihao Shi[3], Juwei Lu[3],
Peng Dai[3], Songcen Xu[3], Li Cheng[1], Yee-Hong Yang[1]

[1]University of Alberta, Canada,
[2]University of Toronto, Canada,
[3]Huawei Noah's Ark Lab

Background

Generate 3D textures for a given mesh, guided by a text prompt



A high quality color photo of Tom Cruise

Background

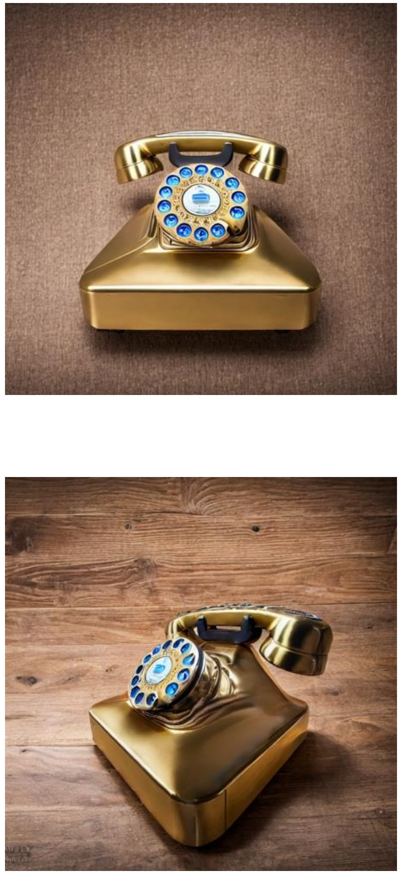
Two main solutions

1. Progressive inpainting

Background

Two main solutions

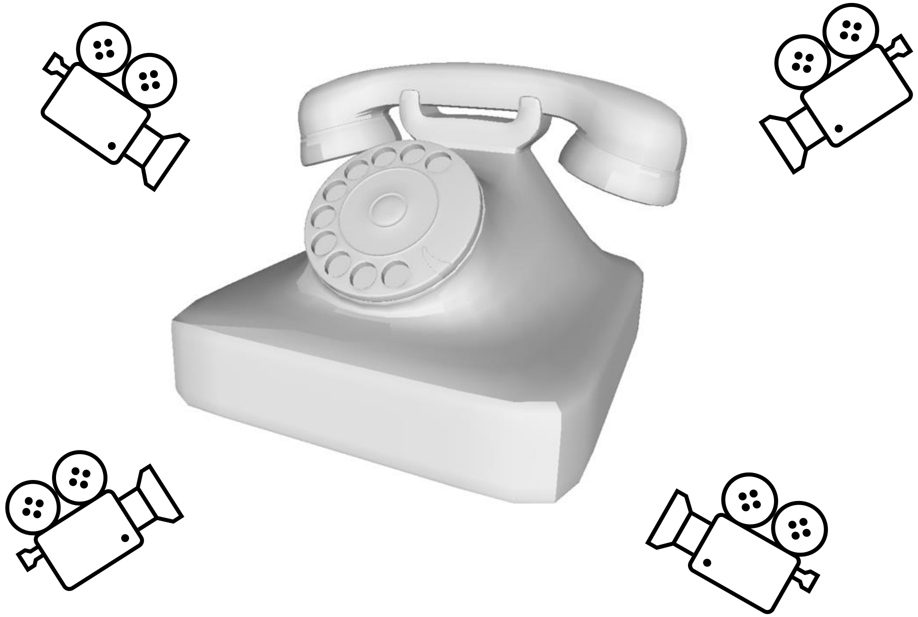
1. Progressive inpainting



Background

Two main solutions

1. Progressive inpainting

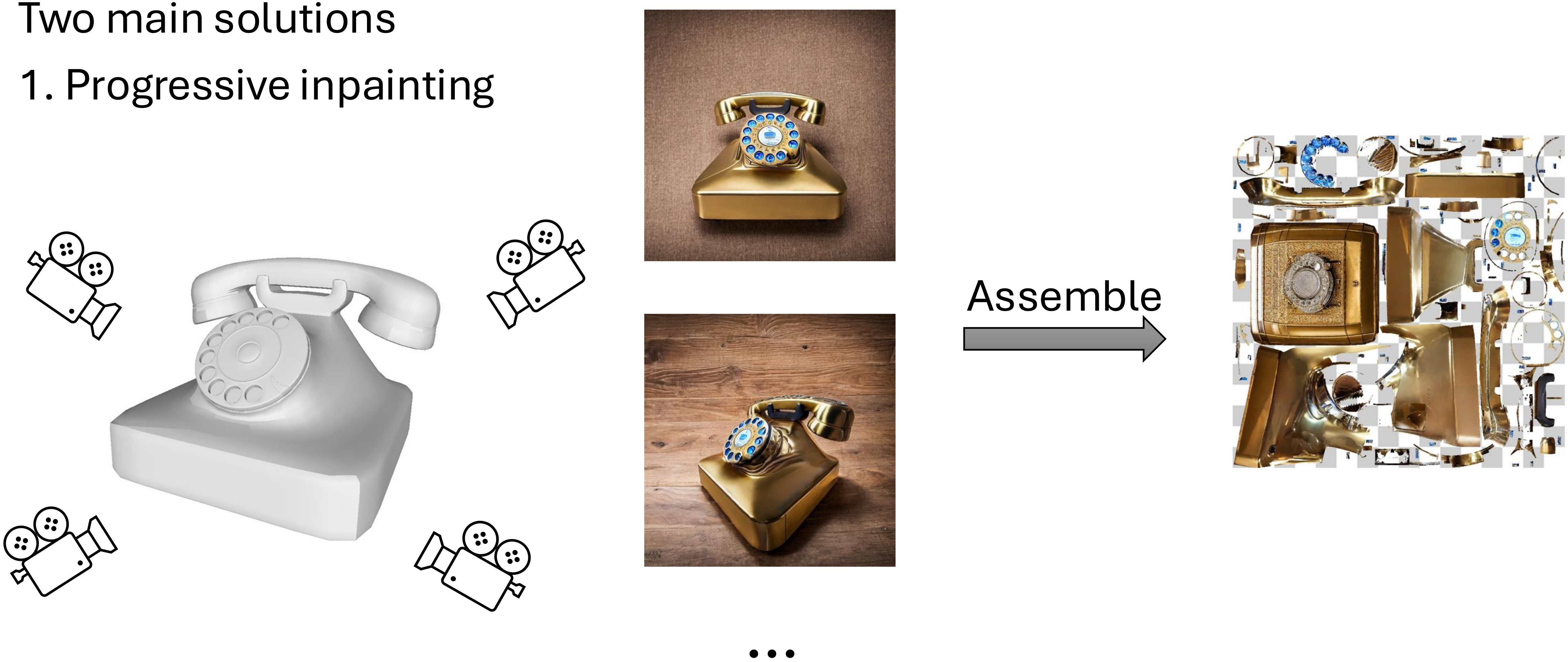


...

Background

Two main solutions

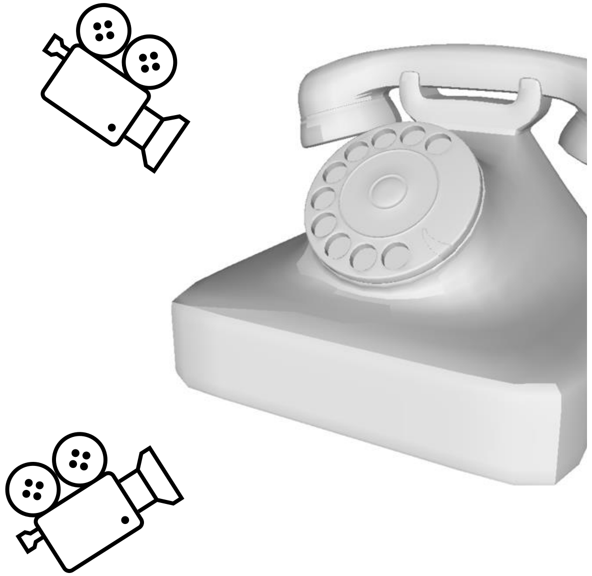
1. Progressive inpainting

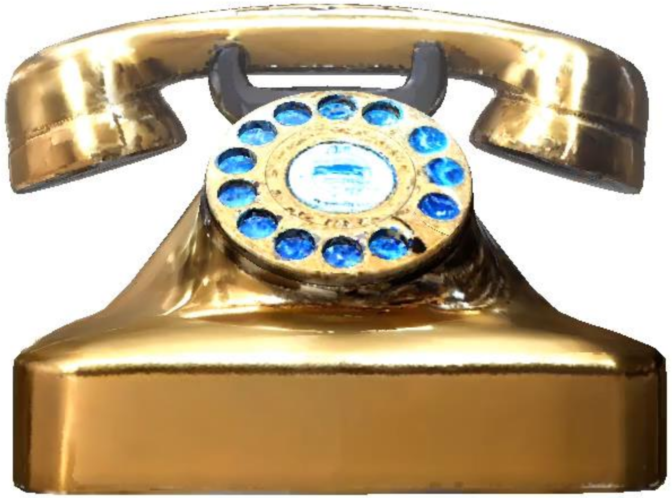


Background

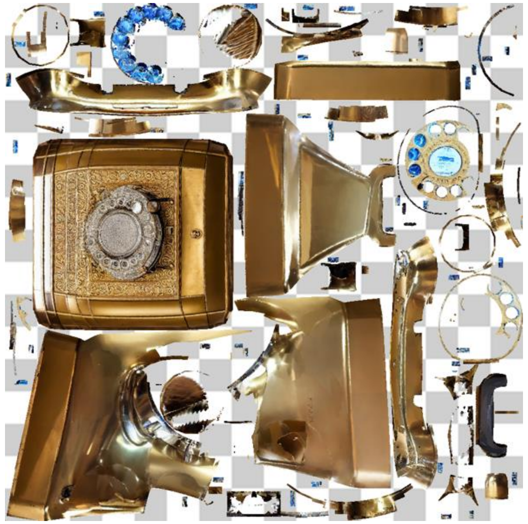
Two main solutions

1. Progressive inpainting





e

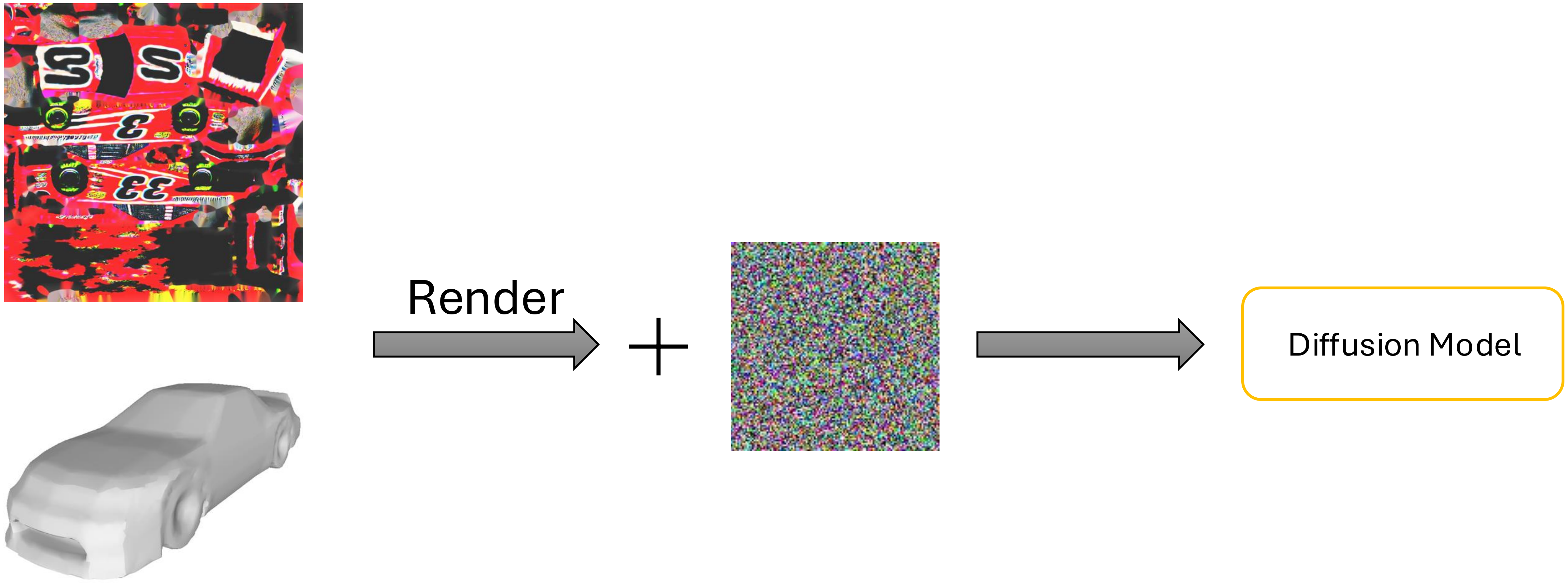


• • •

Background

Two main solutions

2. Score distillation sampling



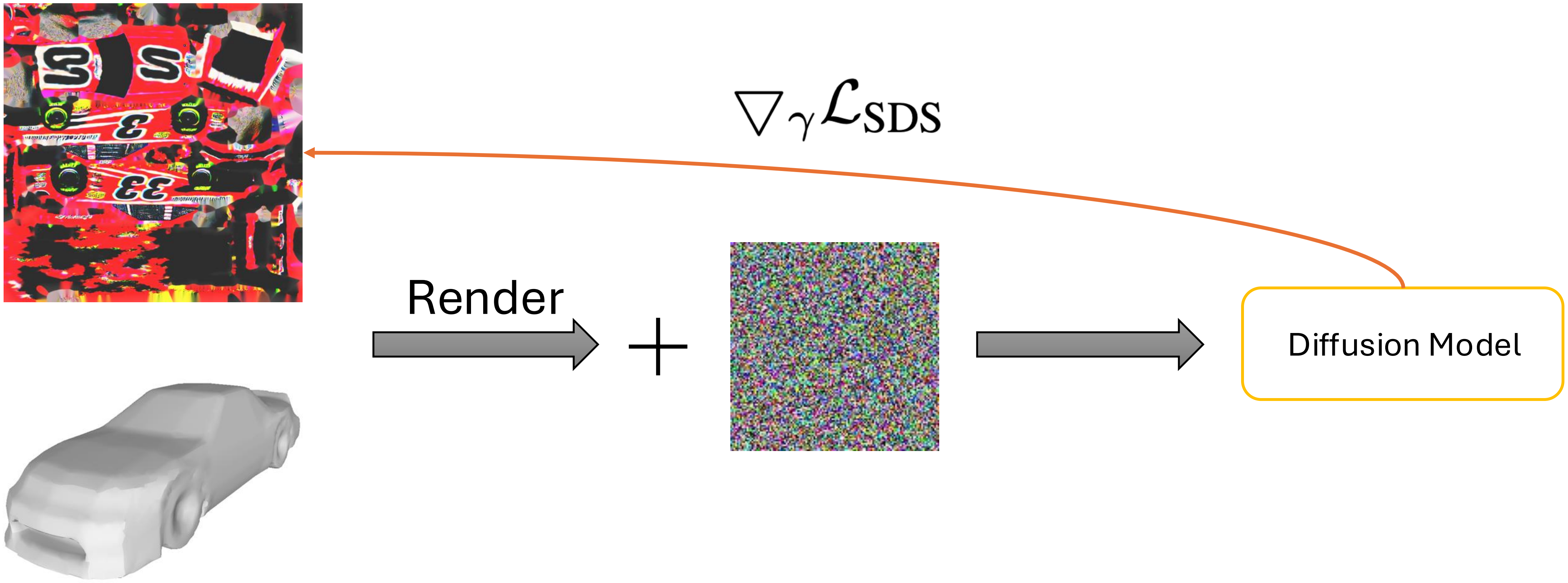
Render

Diffusion Model

Background

Two main solutions

2. Score distillation sampling



$\nabla_\gamma \mathcal{L}_{\text{SDS}}$

Render

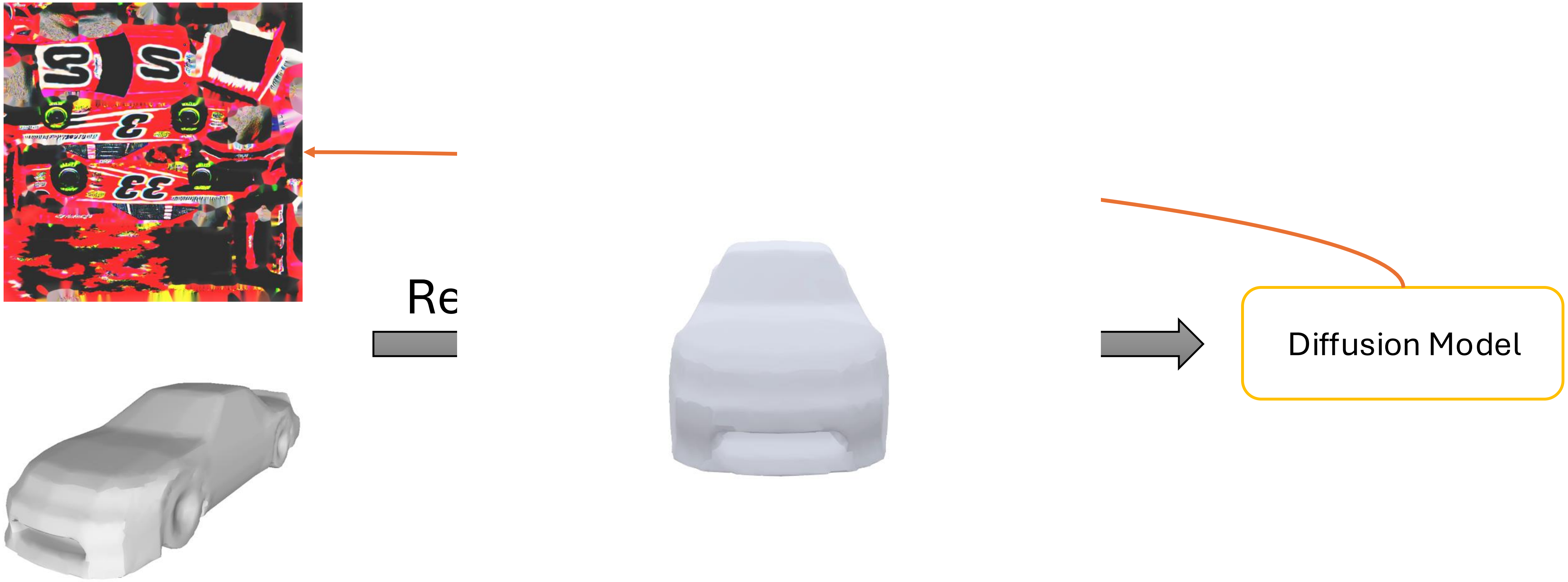
+

Diffusion Model

Background

Two main solutions

2. Score distillation sampling

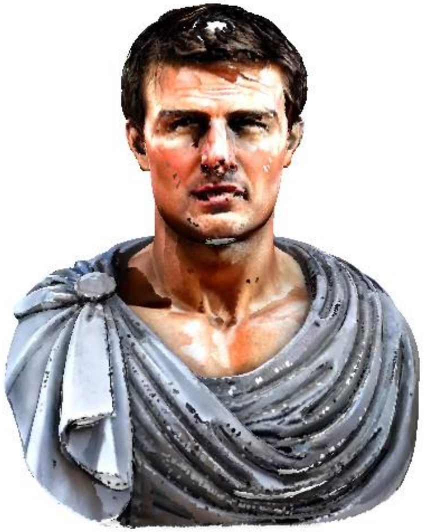


Re

Diffusion Model

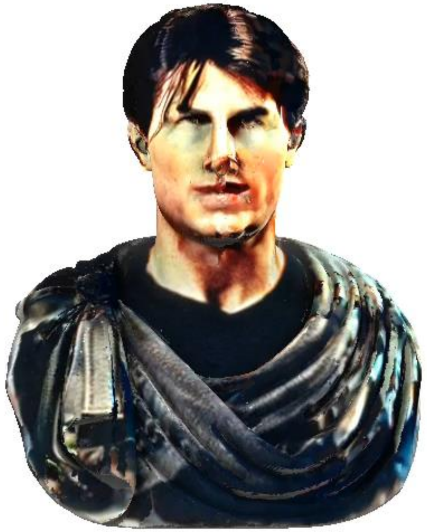
Problems

1. Progressive inpainting



Text2Tex

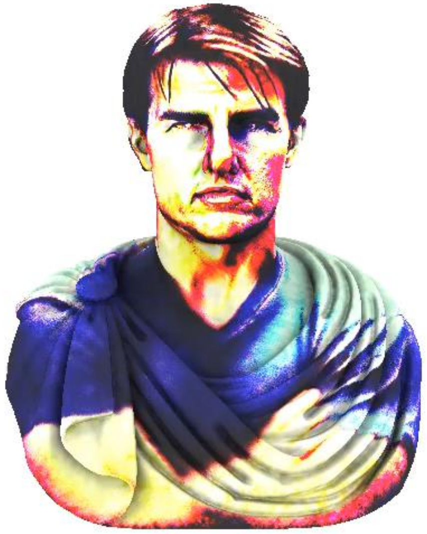
Seams
View-inconsistency



TEXTure

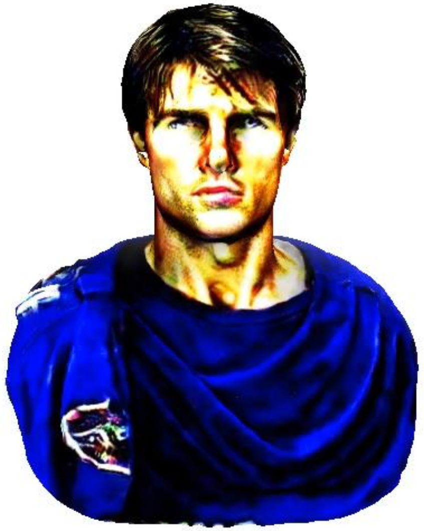
Problems

2. Score distillation sampling



Over-saturation

Blurry edges

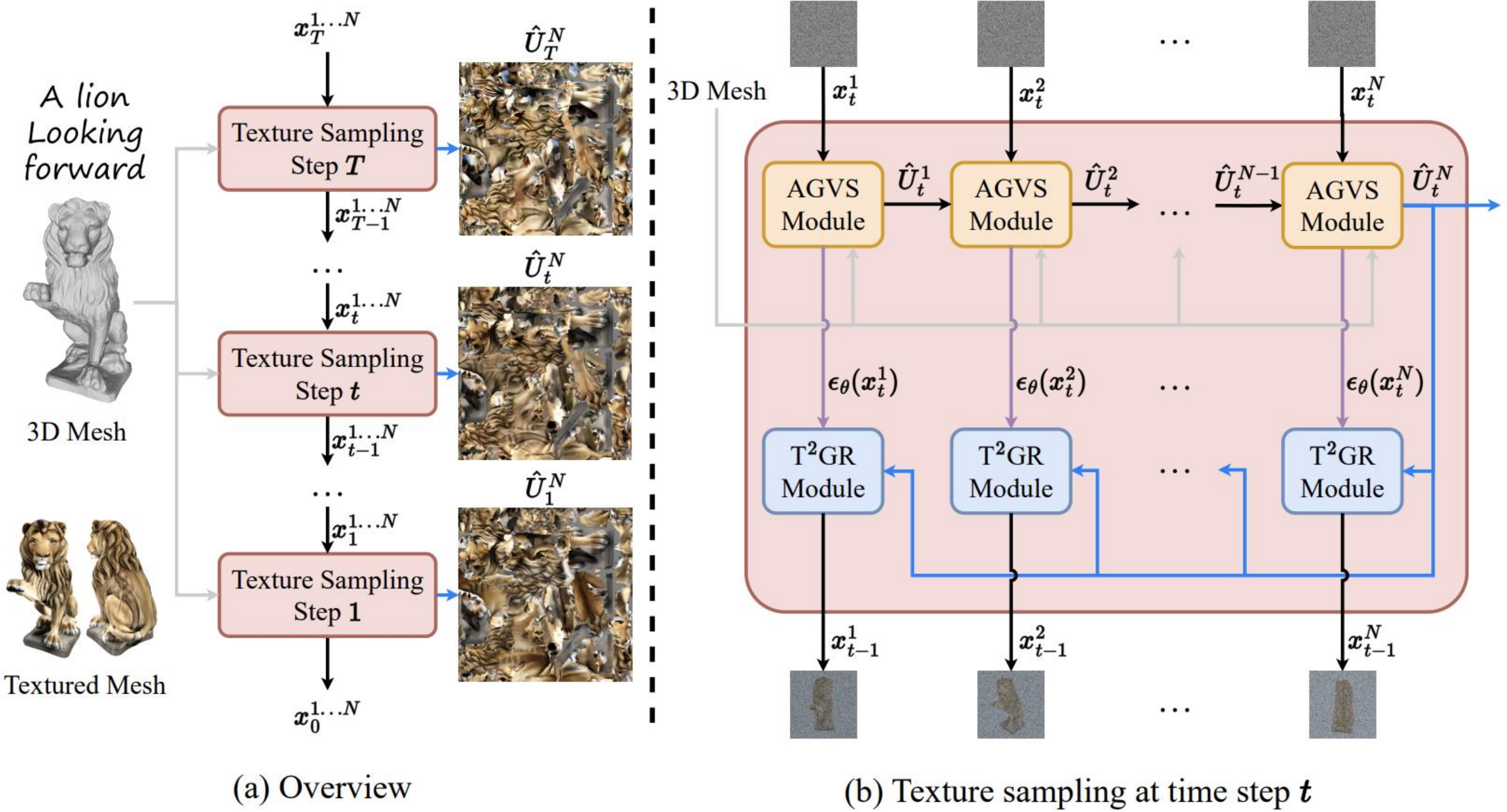


Fantasia3D

ProlificDreamer

Method

Framework

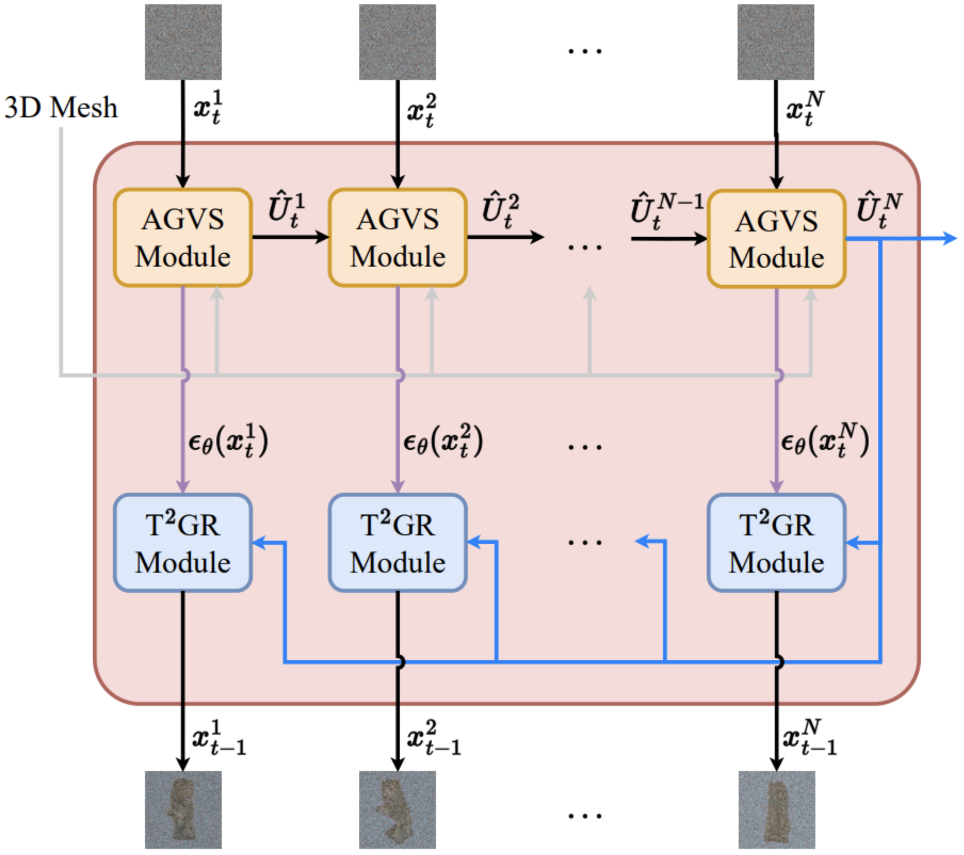


(a) Overview

(b) Texture sampling at time step t

Method

Framework

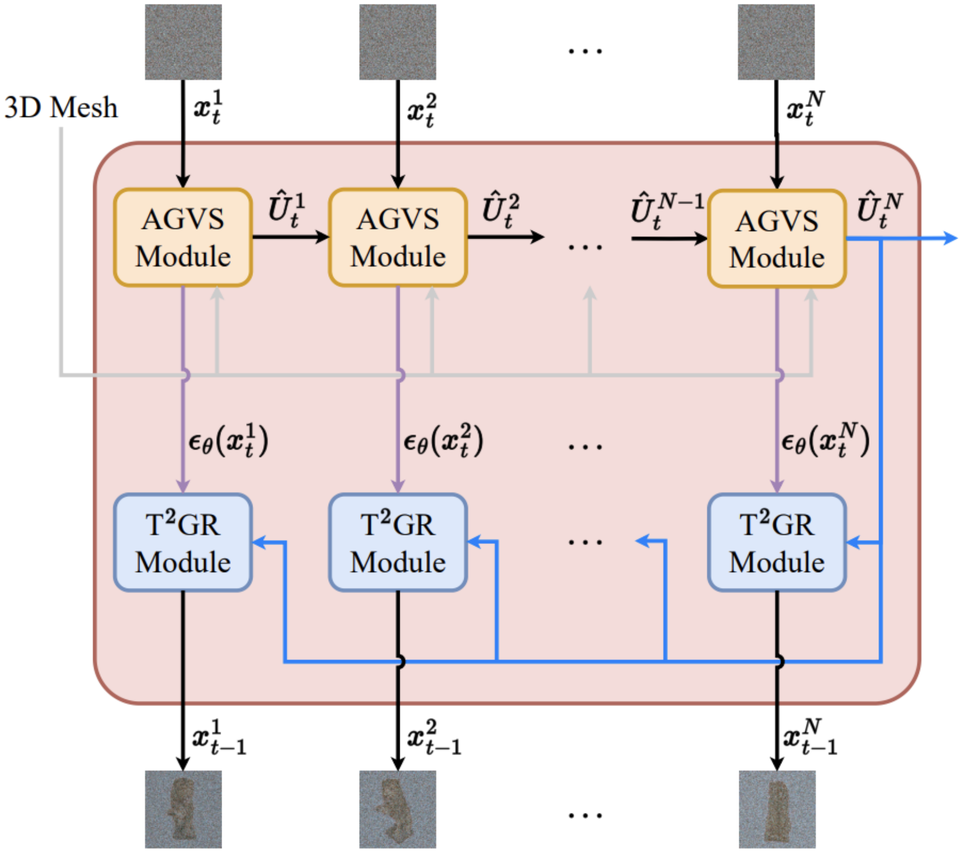


(b) Texture sampling at time step $\boldsymbol{t}$

Progressive inpainting at the end of denoising

Method

Framework

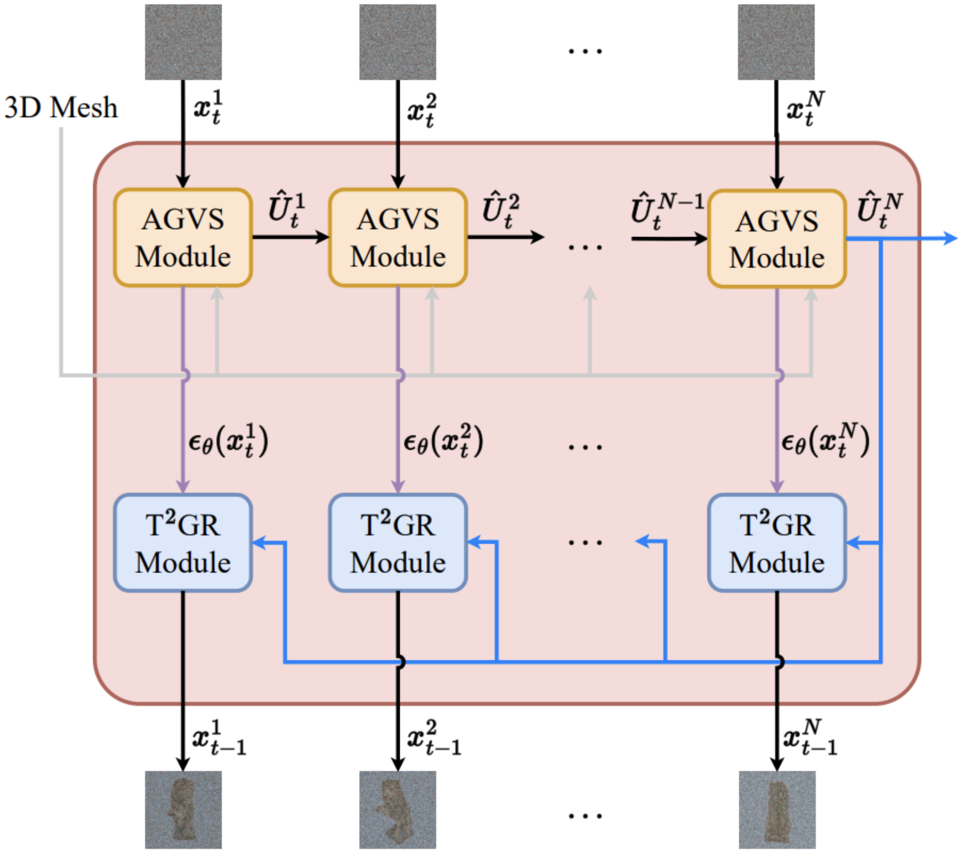


(b) Texture sampling at time step t

Progressive inpain ✗ at the end of denoising

Method

Framework



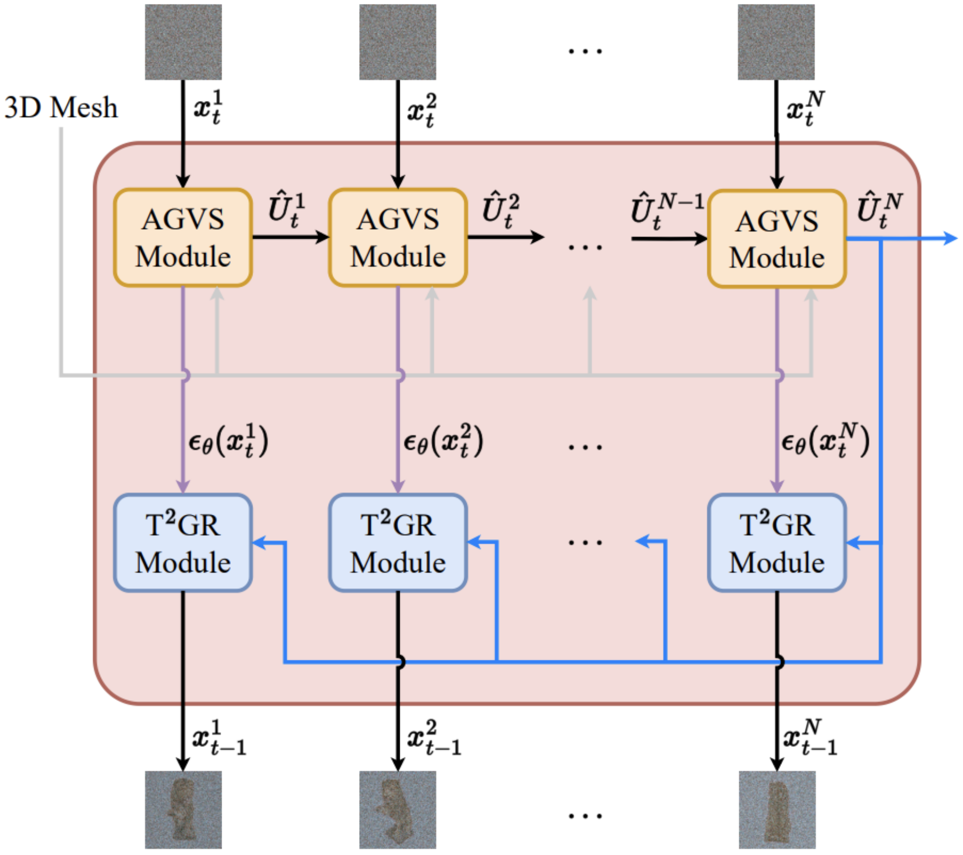
(b) Texture sampling at time step t

Progressive inpain ✗ at the end of denoising

Progressive inpainting at each denoising step , which reduces the accumulated error

Method

Framework



(b) Texture sampling at time step t

The most straightforward solution:

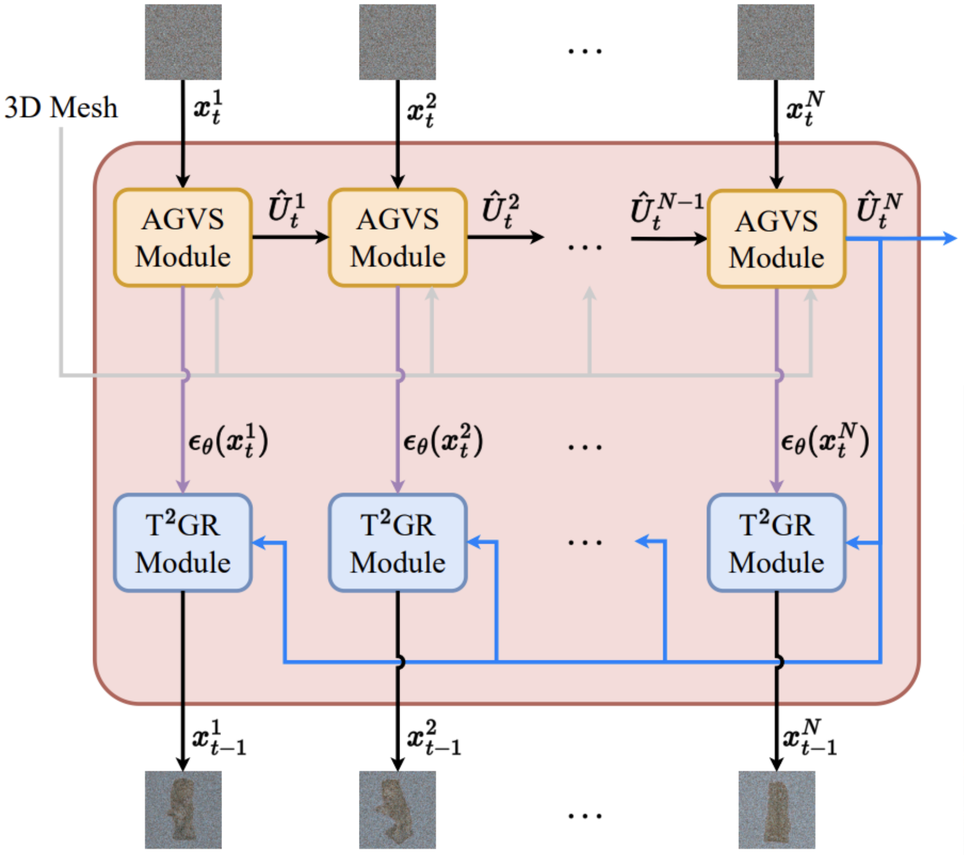
at each viewpoints, decode the predicted x0 and assemble onto the texture map

$$x_{t-1}^i = \sqrt{\alpha_{t-1}} \cdot \hat{x}_0^i(x_t^i) + \sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta(x_t^i),$$

$$\hat{x}_0^i(x_t^i) = \frac{x_t^i - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(x_t^i)}{\sqrt{\alpha_t}},$$

Method

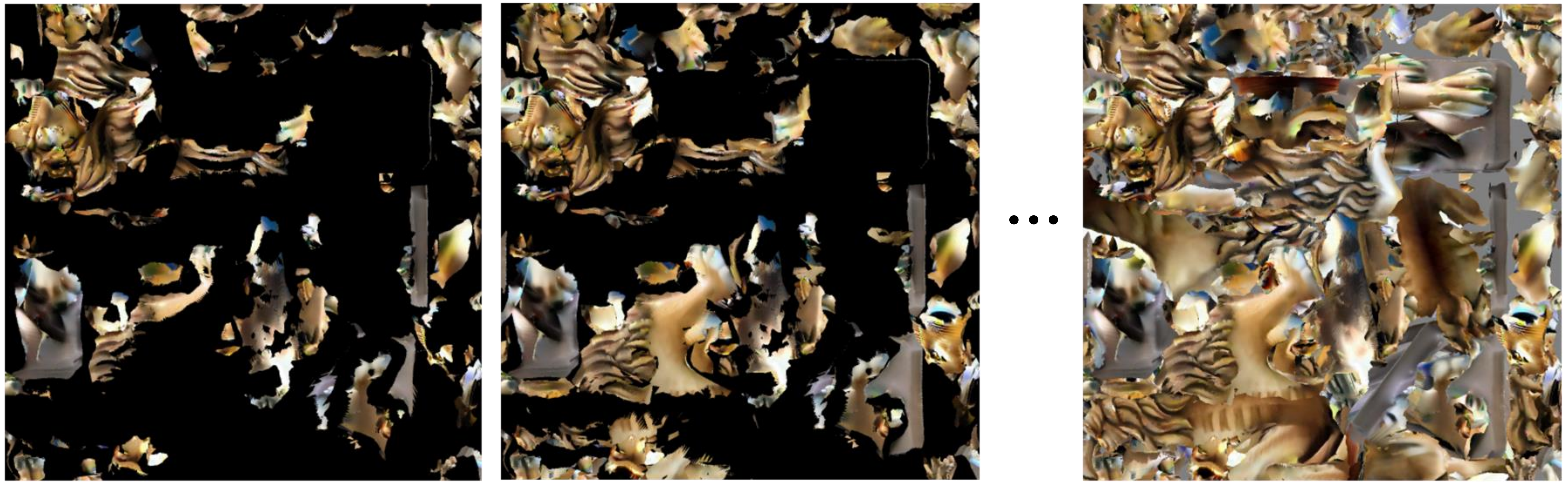
Framework



(b) Texture sampling at time step t

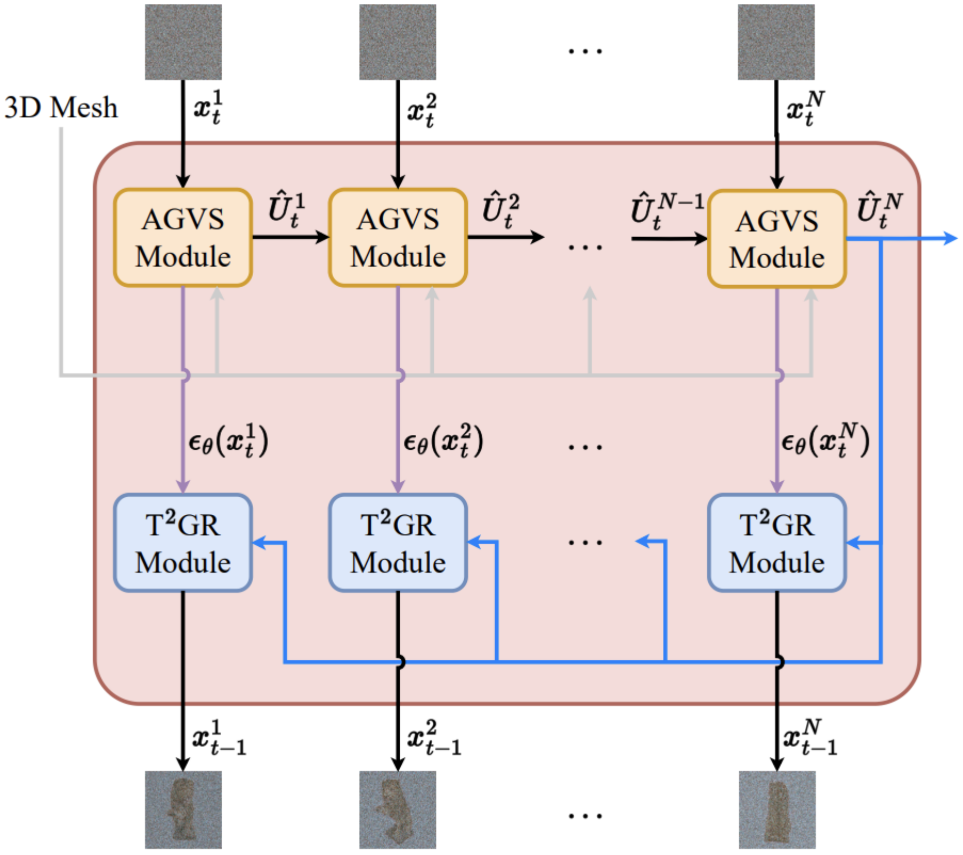
The most straightforward solution:

at each viewpoints, decode the predicted x0 and assemble onto the texture map



Method

Framework



(b) Texture sampling at time step t

The most straightforward solution:

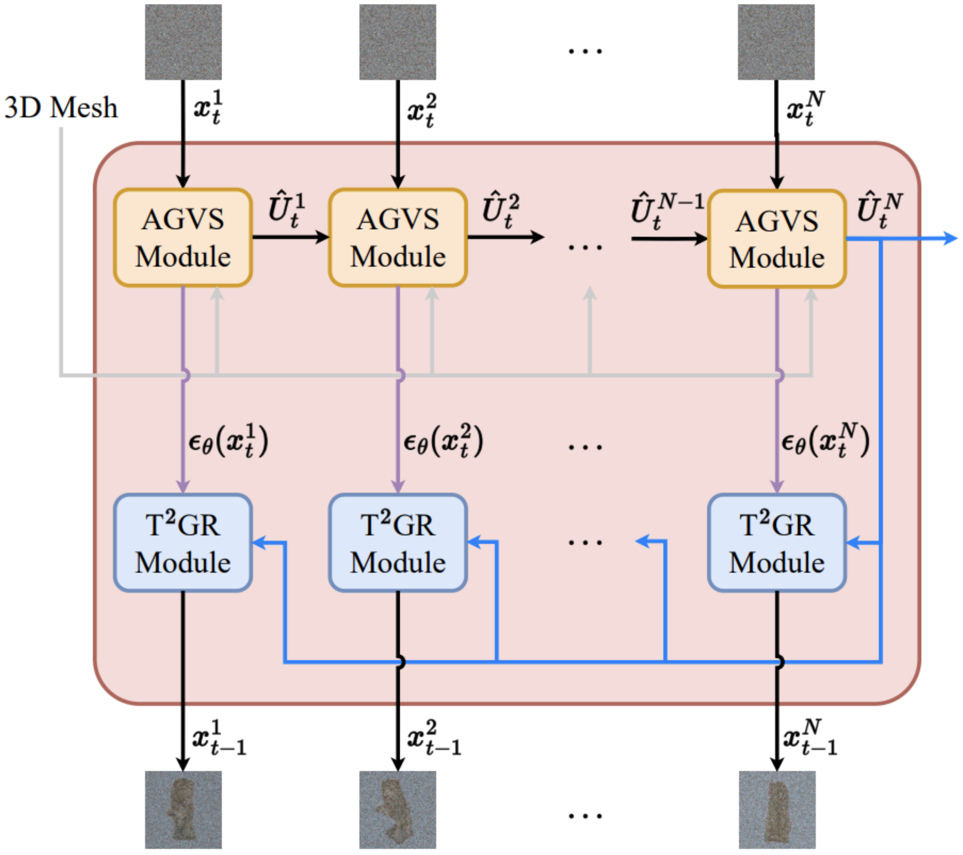
then encode the RGB texture rendering of each view as the new x_0 for the next denoising step

$$x_{t-1}^i = \sqrt{\alpha_{t-1}} \cdot \hat{x}_0^i(x_t^i) + \sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta(x_t^i),$$

$$\hat{x}_0^i(x_t^i) = \frac{x_t^i - \sqrt{1-\alpha_t} \cdot \epsilon_\theta(x_t^i)}{\sqrt{\alpha_t}},$$

Method

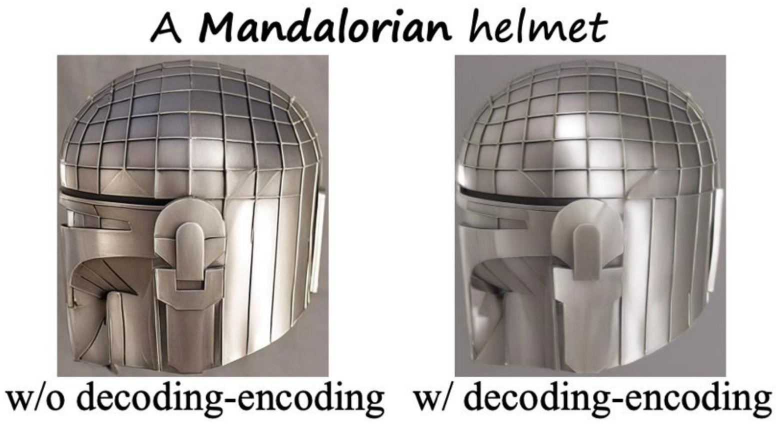
Framework



(b) Texture sampling at time step $\boldsymbol{t}$

Problem!

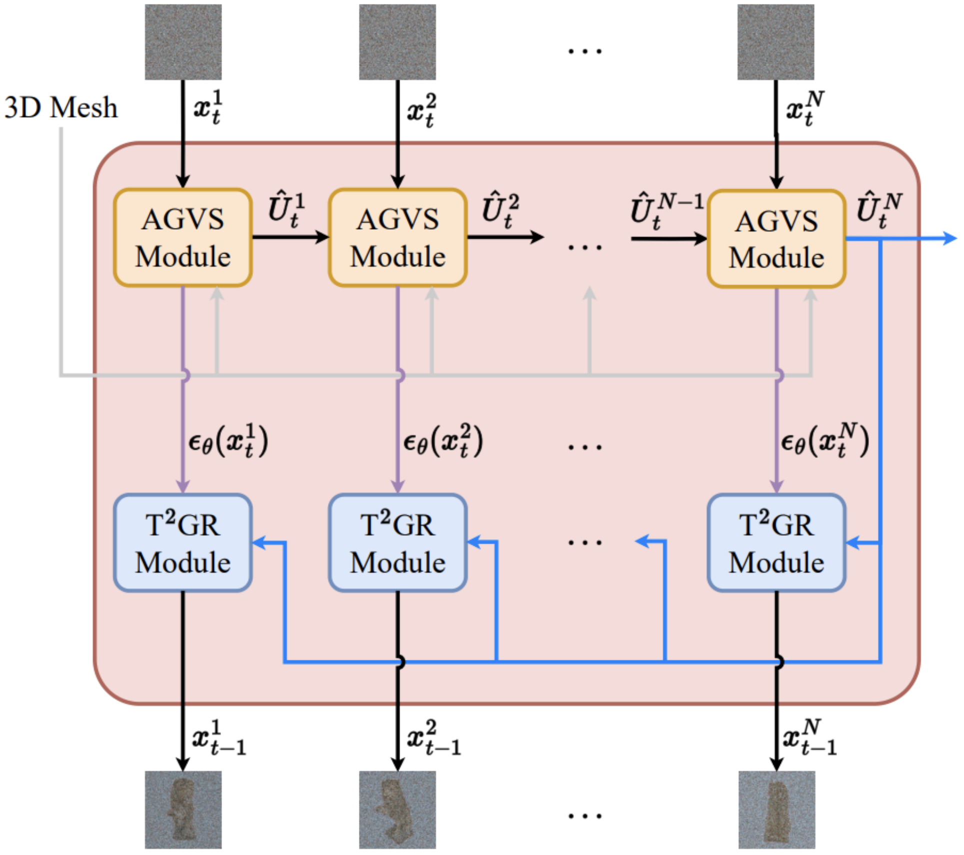
Since the VAE encoding is lossy-compression, repetitive decoding-encoding leads to blur



w/o decoding-encoding w/ decoding-encoding

Method

Framework



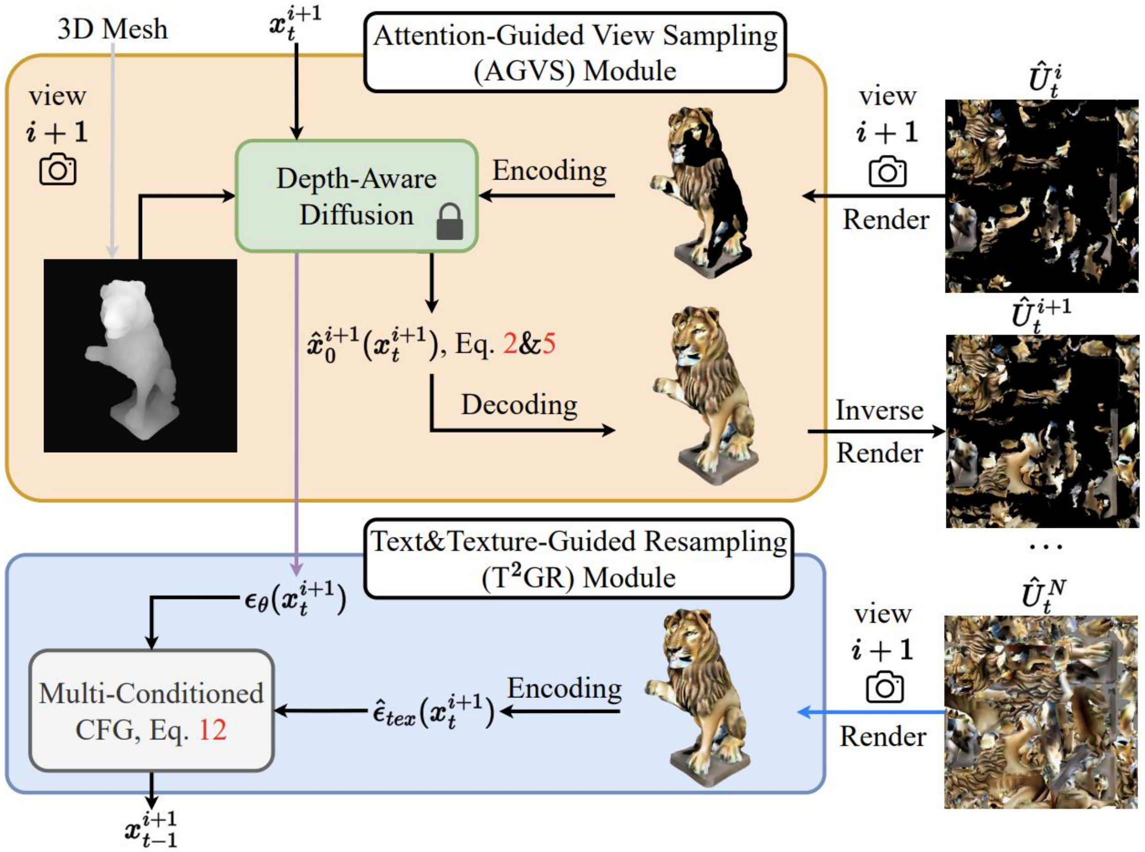
(b) Texture sampling at time step t

A better solution:

Regard the intermediate texture as an extra condition for noise estimation

Method

Framework

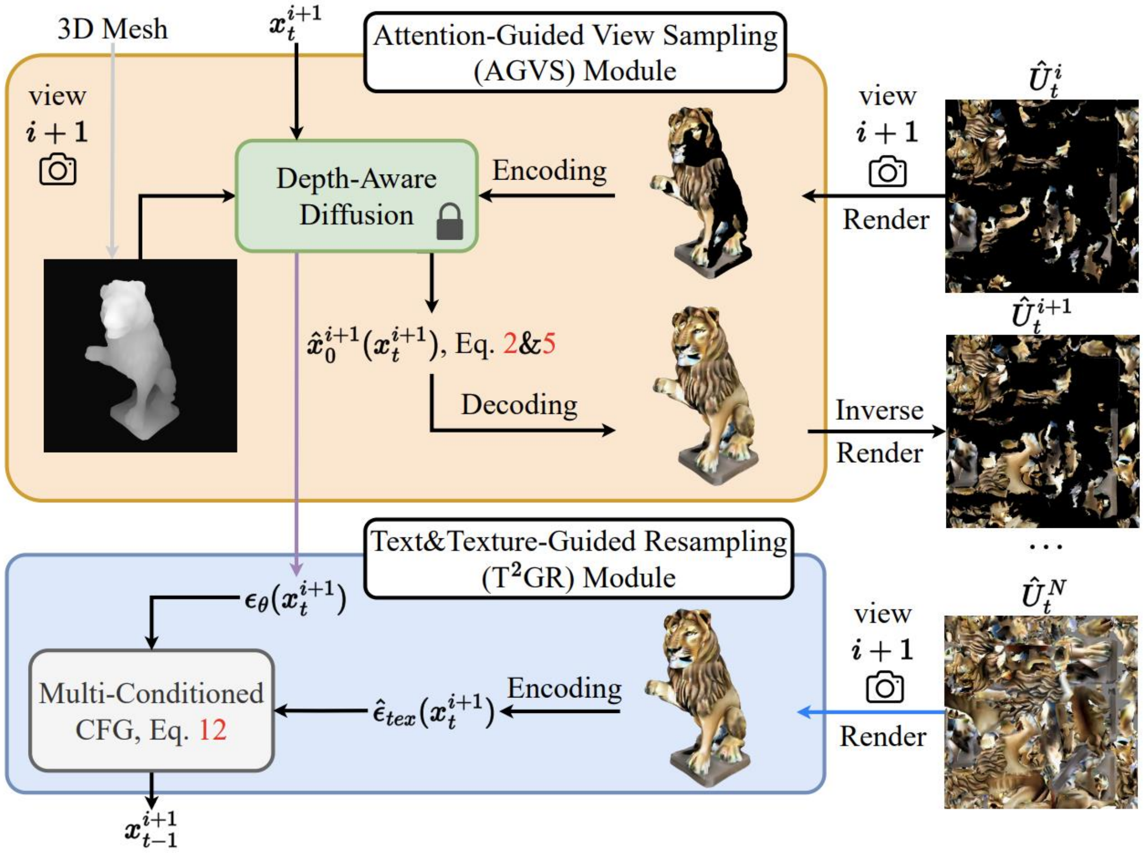

3D Mesh
x_t^{i+1}
Attention-Guided View Sampling (AGVS) Module
view $i+1$
Depth-Aware Diffusion
Encoding
view $i+1$
Render
$\hat{U}_t^i$
$\hat{x}_0^{i+1}(x_t^{i+1})$, Eq. 2&5
Decoding
Inverse Render
$\hat{U}_t^{i+1}$
...
Text&Texture-Guided Resampling (T^{2}GR) Module
$\epsilon_\theta(x_t^{i+1})$
Multi-Conditioned CFG, Eq. 12
$\hat{\epsilon}_{tex}(x_t^{i+1})$
Encoding
view $i+1$
Render
$\hat{U}_t^N$
x_{t-1}^{i+1}

A better solution:

Regard the intermediate texture as an extra condition for noise estimation

Method

Framework


3D Mesh
x_t^{i+1}
Attention-Guided View Sampling (AGVS) Module
view $i+1$
Depth-Aware Diffusion
Encoding
view $i+1$
Render
$\hat{U}_t^i$
$\hat{x}_0^{i+1}(x_t^{i+1})$, Eq. 2&5
Decoding
Inverse Render
$\hat{U}_t^{i+1}$
…
Text&Texture-Guided Resampling (T^{2}GR) Module
$\epsilon_\theta(x_t^{i+1})$
Multi-Conditioned CFG, Eq. 12
$\hat{\epsilon}_{tex}(x_t^{i+1})$
Encoding
view $i+1$
Render
$\hat{U}_t^N$
x_{t-1}^{i+1}

A better solution:

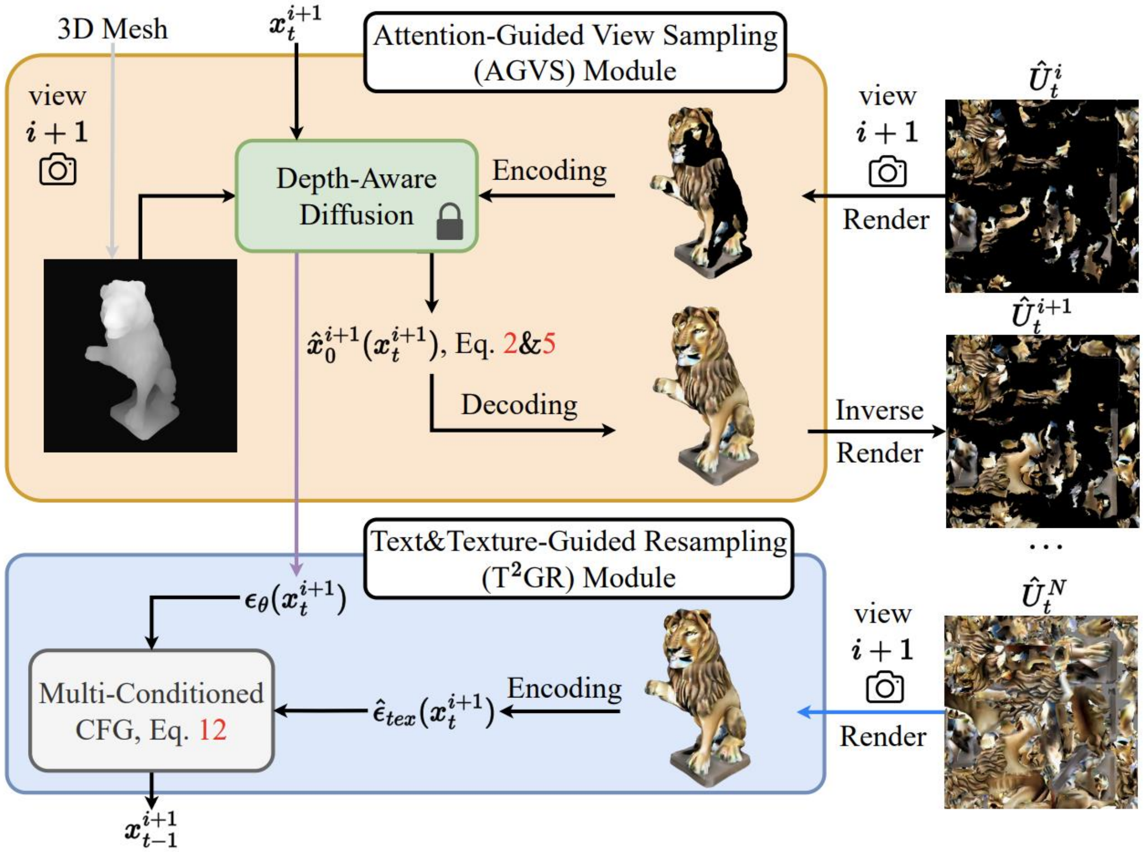
Regard the intermediate texture as an extra condition for noise estimation

$$\hat{x}_0^i(x_t^i) = \frac{x_t^i - \sqrt{1-\alpha_t}\cdot \epsilon_\theta(x_t^i)}{\sqrt{\alpha_t}}$$

$$\hat{\epsilon}_{tex}(x_t^i) = \frac{x_t^i - \sqrt{\alpha_t}\cdot \mathcal{E}(Render^i(\hat{U}_t^N))}{\sqrt{1-\alpha_t}}$$

Method

Framework



3D Mesh
x_t^{i+1}
Attention-Guided View Sampling (AGVS) Module
view $i+1$
Depth-Aware Diffusion
Encoding
view $i+1$
Render
$\hat{U}_t^i$
$\hat{x}_0^{i+1}(x_t^{i+1})$, Eq. 2&5
Decoding
Inverse Render
$\hat{U}_t^{i+1}$
...
Text&Texture-Guided Resampling (T^{2}GR) Module
$\epsilon_\theta(x_t^{i+1})$
Multi-Conditioned CFG, Eq. 12
$\hat{\epsilon}_{tex}(x_t^{i+1})$
Encoding
view $i+1$
Render
$\hat{U}_t^N$
x_{t-1}^{i+1}

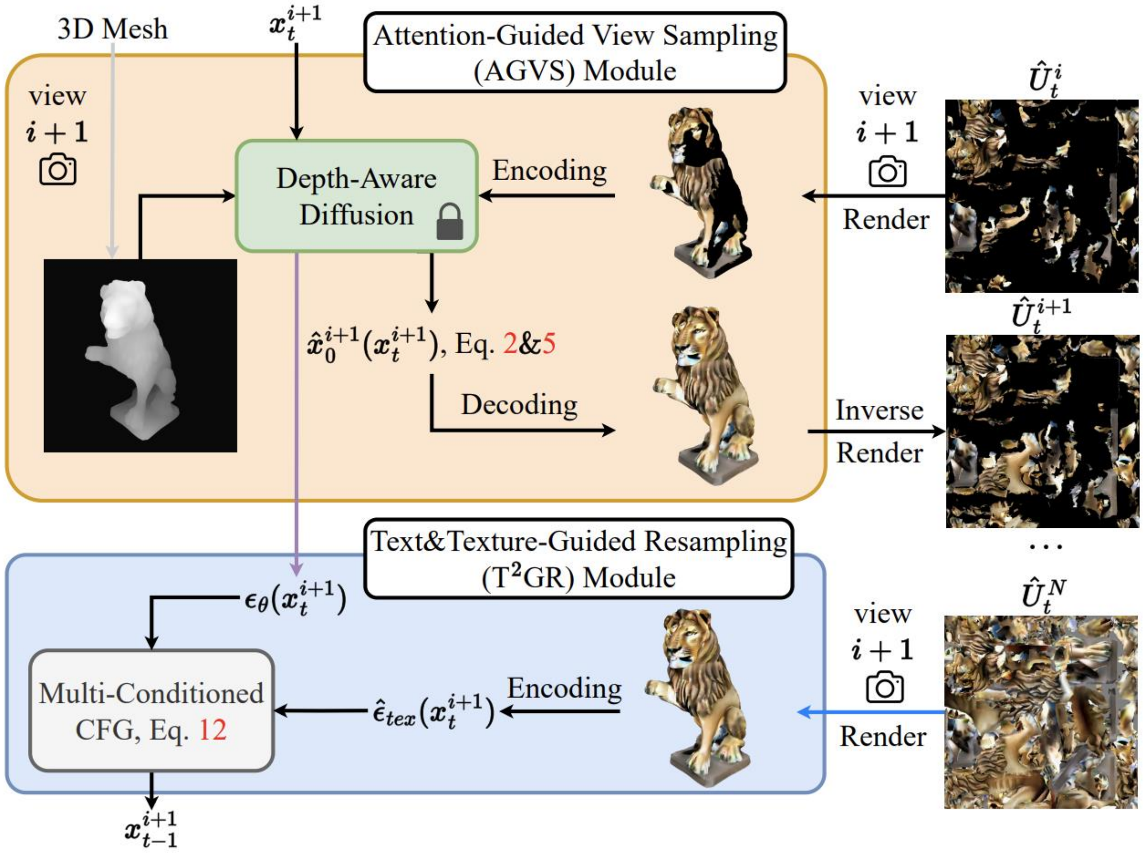
A better solution:

Regard the intermediate texture as an extra condition for noise estimation

$$\epsilon_\theta(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing))$$

Method

Framework


3D Mesh
x_t^{i+1}
Attention-Guided View Sampling (AGVS) Module
view $i+1$
Depth-Aware Diffusion
Encoding
view $i+1$
Render
$\hat{U}_t^i$
$\hat{x}_0^{i+1}(x_t^{i+1})$, Eq. 2&5
Decoding
Inverse Render
$\hat{U}_t^{i+1}$
...
Text&Texture-Guided Resampling (T^{2}GR) Module
$\epsilon_\theta(x_t^{i+1})$
Multi-Conditioned CFG, Eq. 12
$\hat{\epsilon}_{tex}(x_t^{i+1})$
Encoding
view $i+1$
Render
$\hat{U}_t^N$
x_{t-1}^{i+1}

A better solution:

Regard the intermediate texture as an extra condition for noise estimation

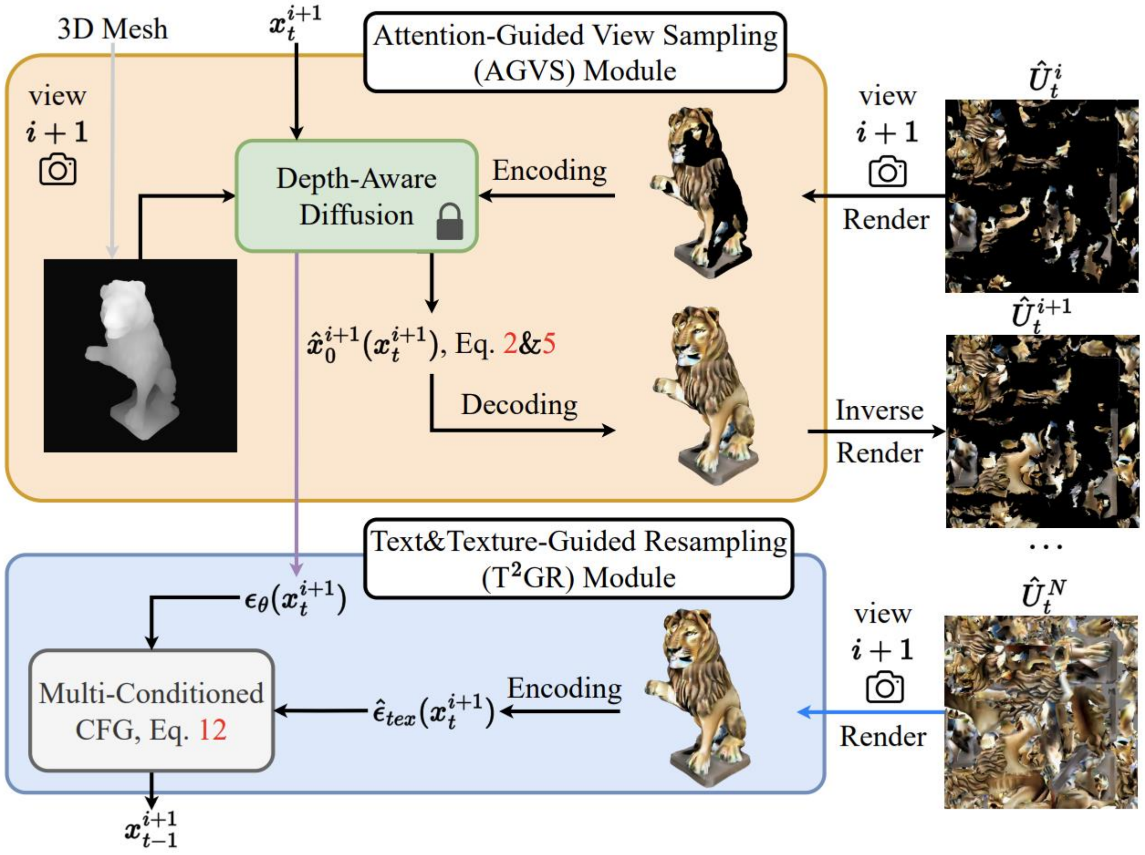
$$\epsilon_\theta(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing))$$

Analogously,

$$\epsilon_{tex}(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega(\epsilon_{tex}(x_t^i|\hat{U}_t^N) - \epsilon_\theta(x_t^i|\varnothing))$$

Method

Framework


3D Mesh
x_t^{i+1}
Attention-Guided View Sampling (AGVS) Module
view $i+1$
Depth-Aware Diffusion
Encoding
view $i+1$
Render
$\hat{U}_t^i$
$\hat{x}_0^{i+1}(x_t^{i+1})$, Eq. 2&5
Decoding
Inverse Render
$\hat{U}_t^{i+1}$
...
Text&Texture-Guided Resampling (T^{2}GR) Module
$\epsilon_\theta(x_t^{i+1})$
Multi-Conditioned CFG, Eq. 12
$\hat{\epsilon}_{tex}(x_t^{i+1})$
Encoding
view $i+1$
Render
$\hat{U}_t^N$
x_{t-1}^{i+1}

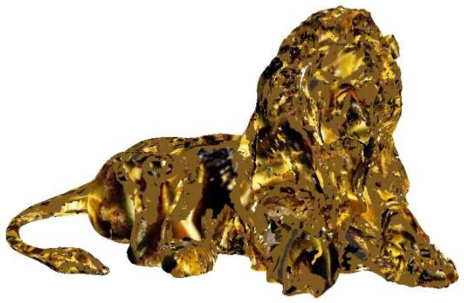
A better solution:

Multi-conditioned Classifier-free guidance

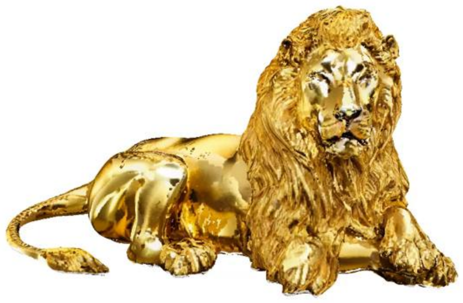
$$\epsilon_m(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega_1(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing))$$
$$+ \omega_2(\epsilon_{tex}(x_t^i|\hat{U}_t^N) - \epsilon_\theta(x_t^i|\varnothing))$$

Experiments

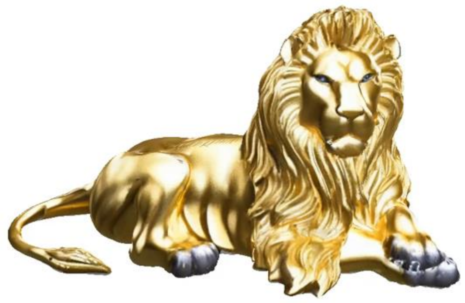
A golden lion



TEXTure



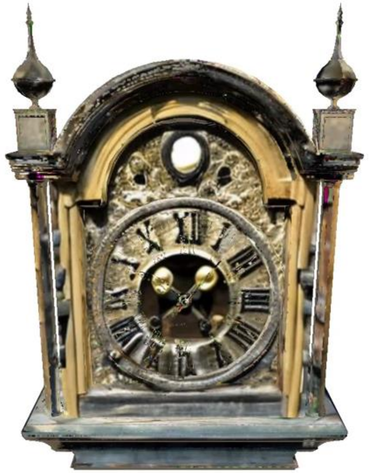
Text2Tex



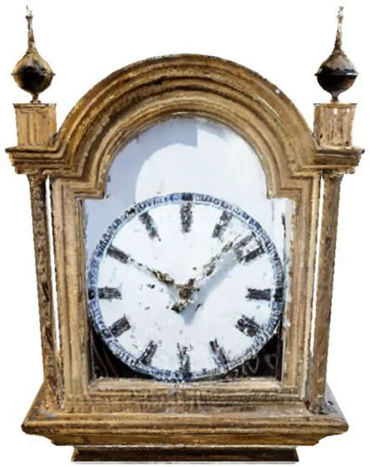
Ours

Experiments

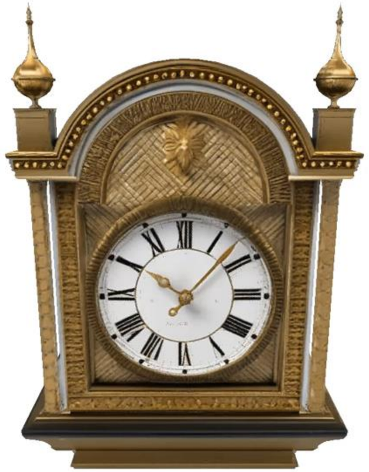
A medieval clock



TEXTure



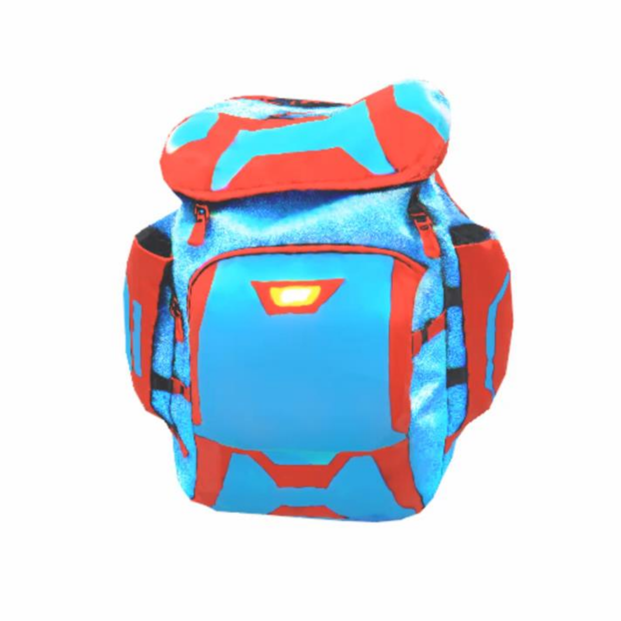
Text2Tex



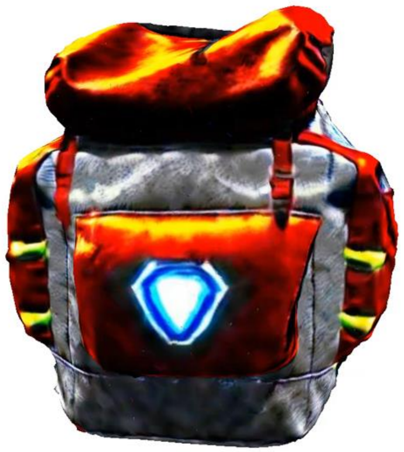
Ours

Experiments

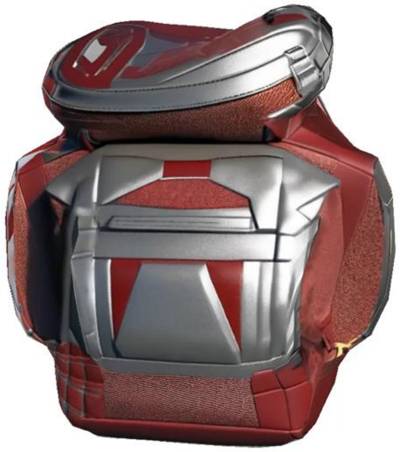
A backpack in ironman style



Fantasia3D



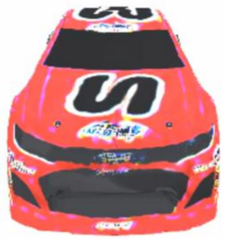
ProlificDreamer



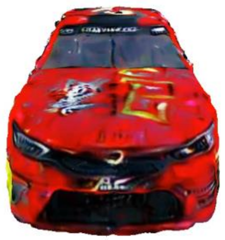
Ours

Experiments

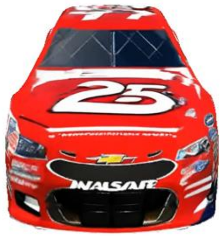
A next gen nascar in red



Fantasia3D



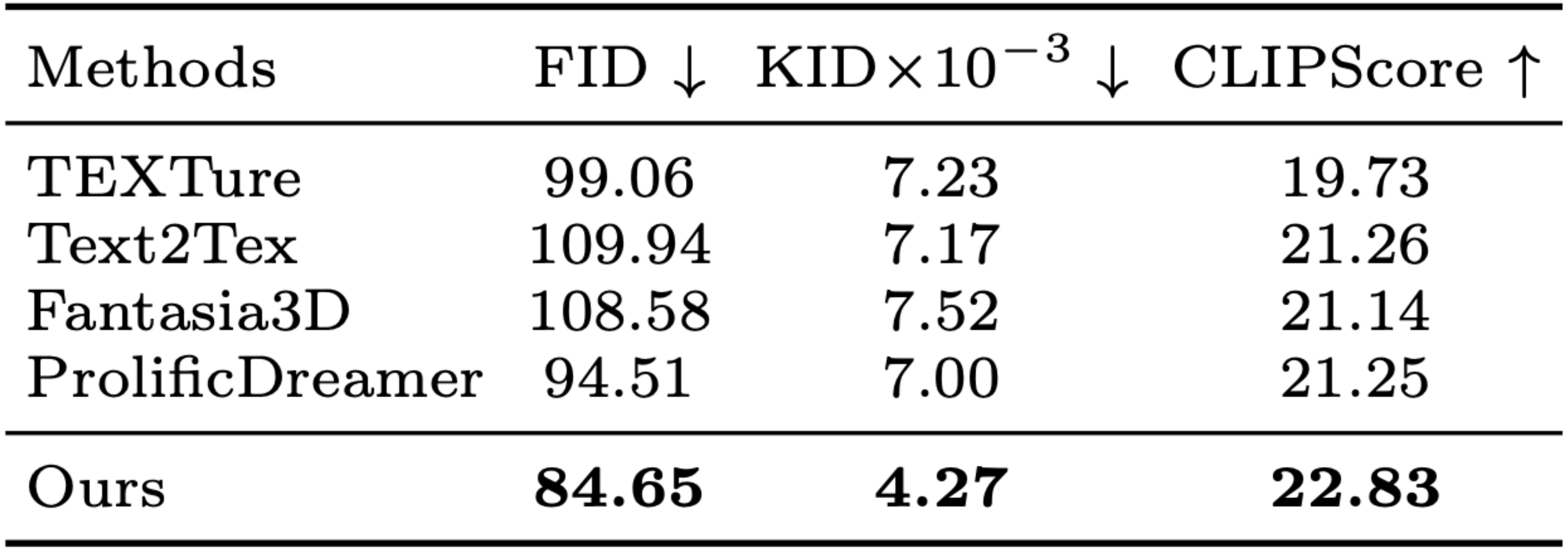
ProlificDreamer



Ours

Experiments

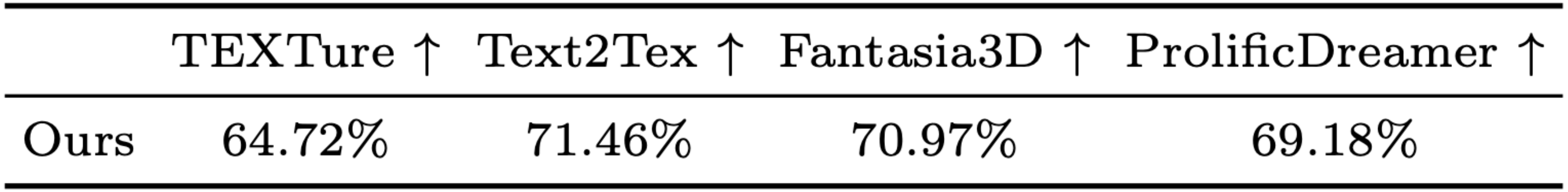
Evaluation metrics



Methods	FID ↓	KID$\times 10^{-3}$ ↓	CLIPScore ↑
TEXTure	99.06	7.23	19.73
Text2Tex	109.94	7.17	21.26
Fantasia3D	108.58	7.52	21.14
ProlificDreamer	94.51	7.00	21.25
Ours	**84.65**	**4.27**	**22.83**

Experiments

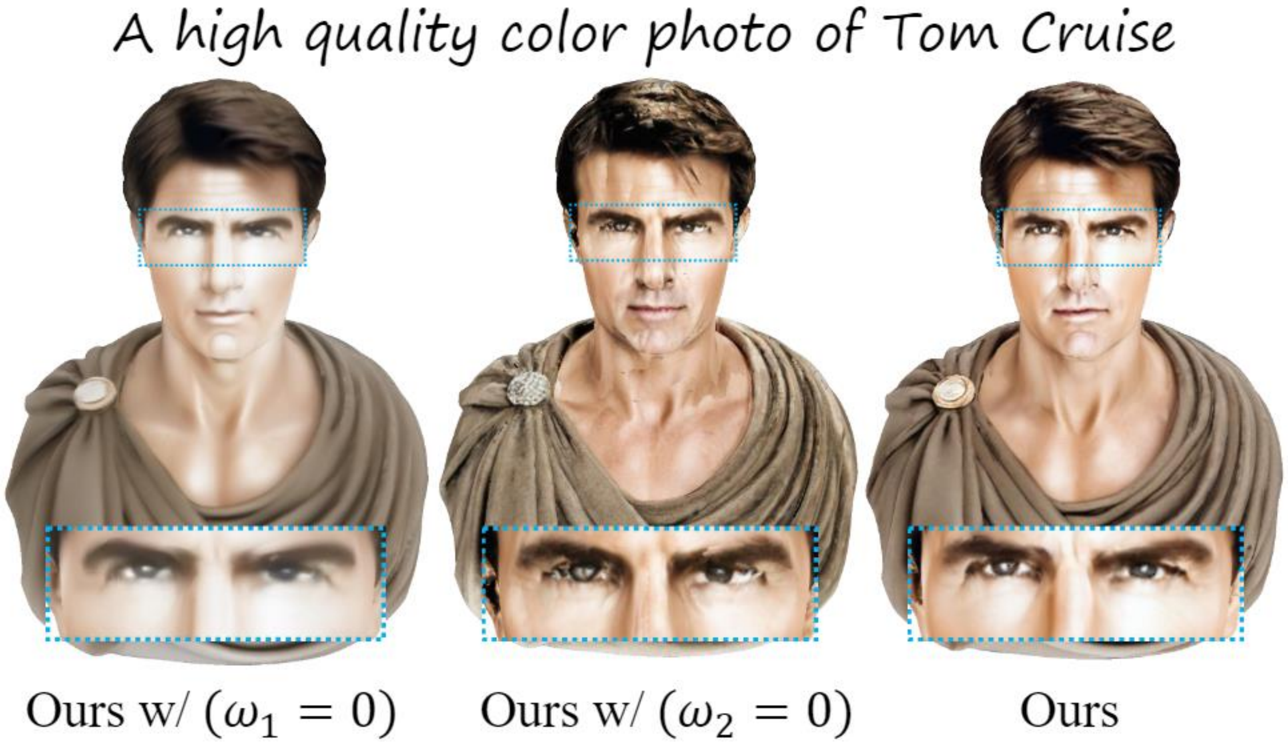
User preference



	TEXTure ↑	Text2Tex ↑	Fantasia3D ↑	ProlificDreamer ↑
Ours	64.72%	71.46%	70.97%	69.18%

Experiments

Ablation study



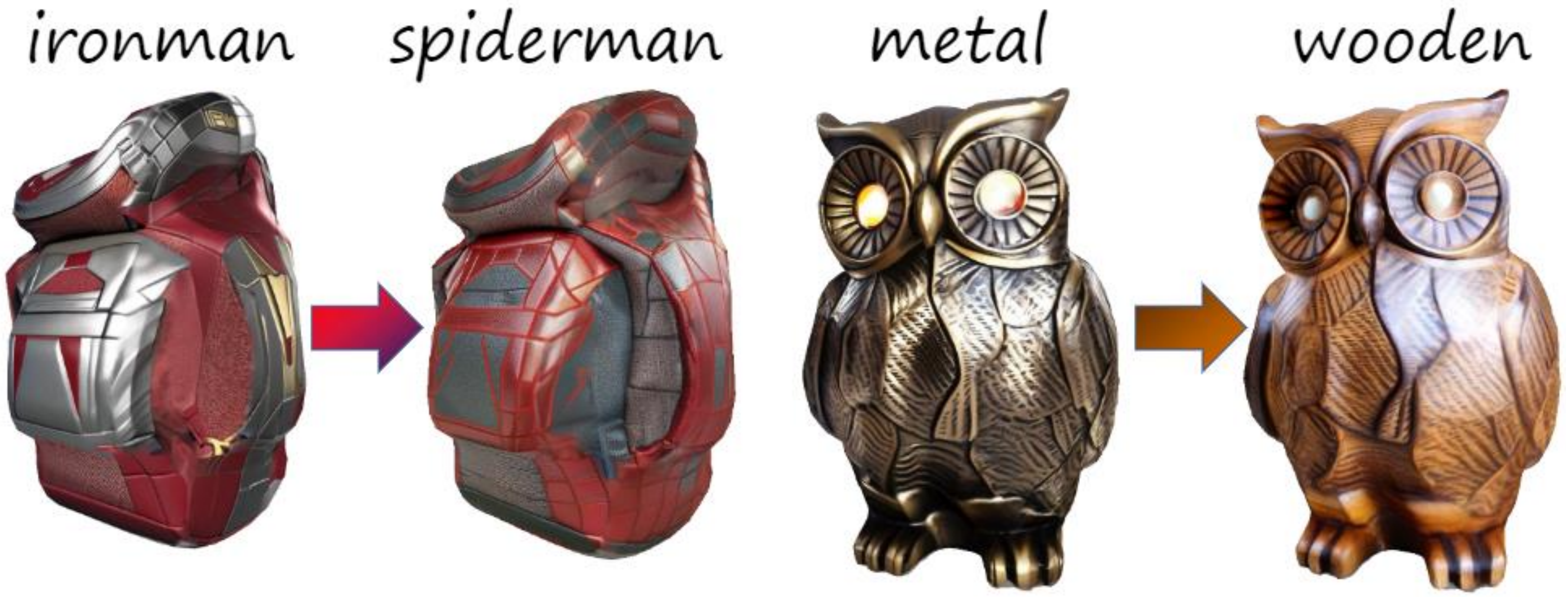
A high quality color photo of Tom Cruise

Ours w/ $(\omega_1 = 0)$ Ours w/ $(\omega_2 = 0)$ Ours

$$\epsilon_m(x_t^i) = \epsilon_\theta(x_t^i|\varnothing) + \omega_1(\epsilon_\theta(x_t^i|c) - \epsilon_\theta(x_t^i|\varnothing)) + \omega_2(\epsilon_{tex}(x_t^i|\hat{U}_t^N) - \epsilon_\theta(x_t^i|\varnothing))$$

Experiments

Texture editing



ECCV

EUROPEAN CONFERENCE ON COMPUTER VISION

MILANO
2024

Thank you for watching!