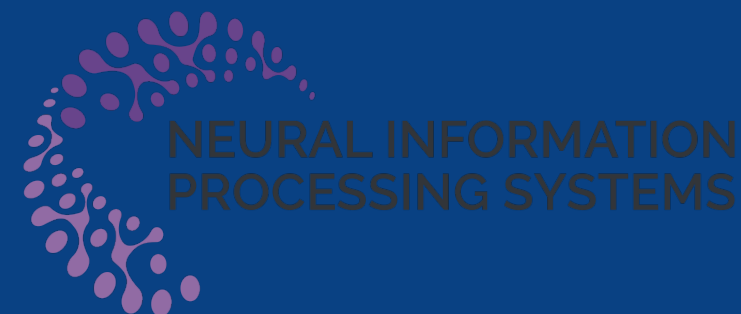




THE UNIVERSITY OF
MELBOURNE



In-N-Out: Lifting 2D Diffusion Prior for 3D Object Removal via Latents Alignment

**Dongting Hu¹, Huan Fu², Jiaxian Guo³, Lihua Peng¹
Tingjin Chu¹, Feng Liu¹, Tongliang Liu^{4,5}, Mingming Gong^{1,5}**

¹The University of Melbourne ²Alibaba Group

³Google Research ⁴The University of Sydney

⁵Mohamed bin Zayed University of Artificial Intelligence

Background: 3D Object Removal



$$(x, y, z, \theta, \phi) \rightarrow \begin{matrix} \text{||||} \\ F_{\theta} \end{matrix} \rightarrow (RGB\sigma)$$



Given:

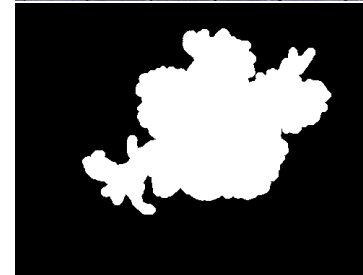
Images



...



Masks



...



Target:

Novel Views
w/o object

$$\begin{matrix} \text{||||} \\ F_{\theta} \end{matrix}$$



Background: 3D Object Removal

Given:

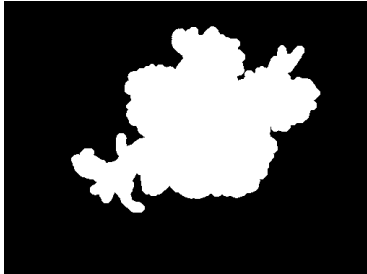
Images



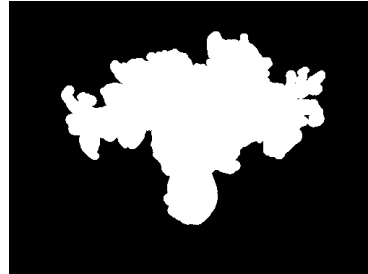
...



Masks



...



Motivation:

- 3D representation (e.g. NeRF) learns from pixels.
- There is no supervision in the **occluded area**.
- We need to fill in the pixels in the masked area.

Target:

Novel Views
w/o object



Background: Diffusion Inpainting Method

Diffusion Models: State-of-the-art 2D inpainting method

stabilityai / **stable-diffusion-2-inpainting** like 436

Image-to-Image Diffusers Safetensors StableDiffusionInpaintPipeline stable-diffusion arxiv:2112.10752 arxiv:2202.00512 arxiv:1910.09700


License: openrail++

Model card Files and versions Community 38 Deploy Use this model

Stable Diffusion v2 Model Card

This model card focuses on the model associated with the Stable Diffusion v2, available [here](#).

This stable-diffusion-2-inpainting model is resumed from [stable-diffusion-2-base](#) (512-base-ema.ckpt) and trained for another 200k steps. Follows the mask-generation strategy presented in [LAMA](#) which, in combination with the latent VAE representations of the masked image, are used as an additional conditioning.



- Use it with the [stablediffusion](#) repository: download the 512-inpainting-ema.ckpt [here](#).
- Use it with [diffusers](#)

Downloads last month **555,387**

Inference API Cold

Image-to-Image

Drag image file here or click to browse from your device

(Optional) Text-guidance if the model has support for it

Your prompt here...

Compute

View Code Maximize

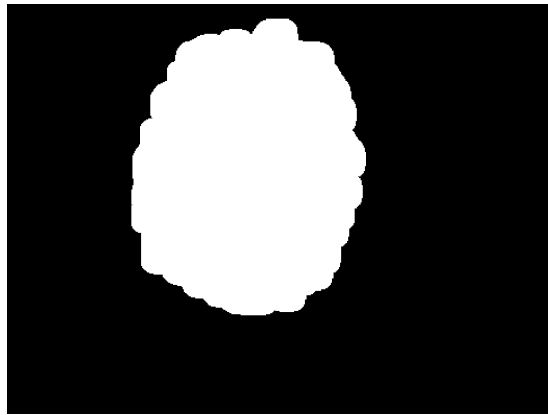
Model tree for stabilityai/stable-diffusion-2-inpainting

Adapters 2 models

Background: Diffusion Inpainting Method



View #1



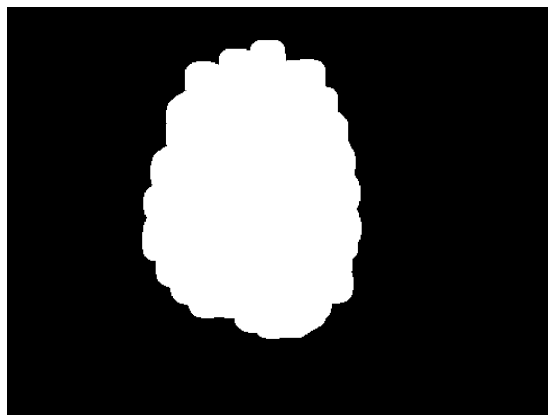
Stable
Diffusion



Independent
Inpainting



View #2



Stable
Diffusion



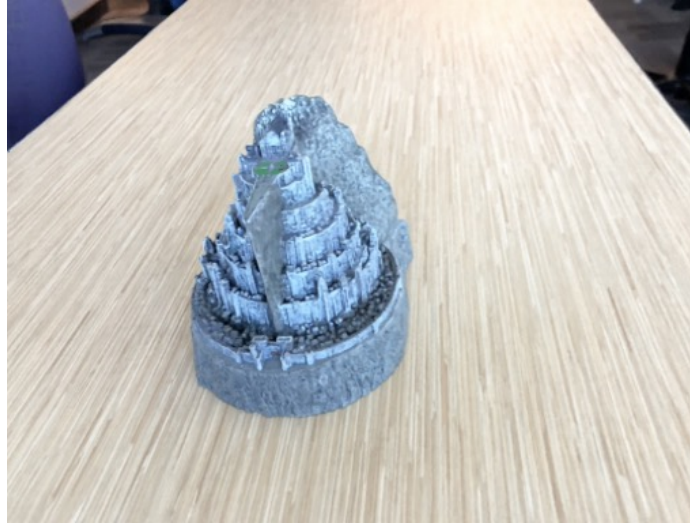
Inconsistent Results

Background: Motivation

3D Inconsistency

--> Geometric Mismatch

--> Blurred Results



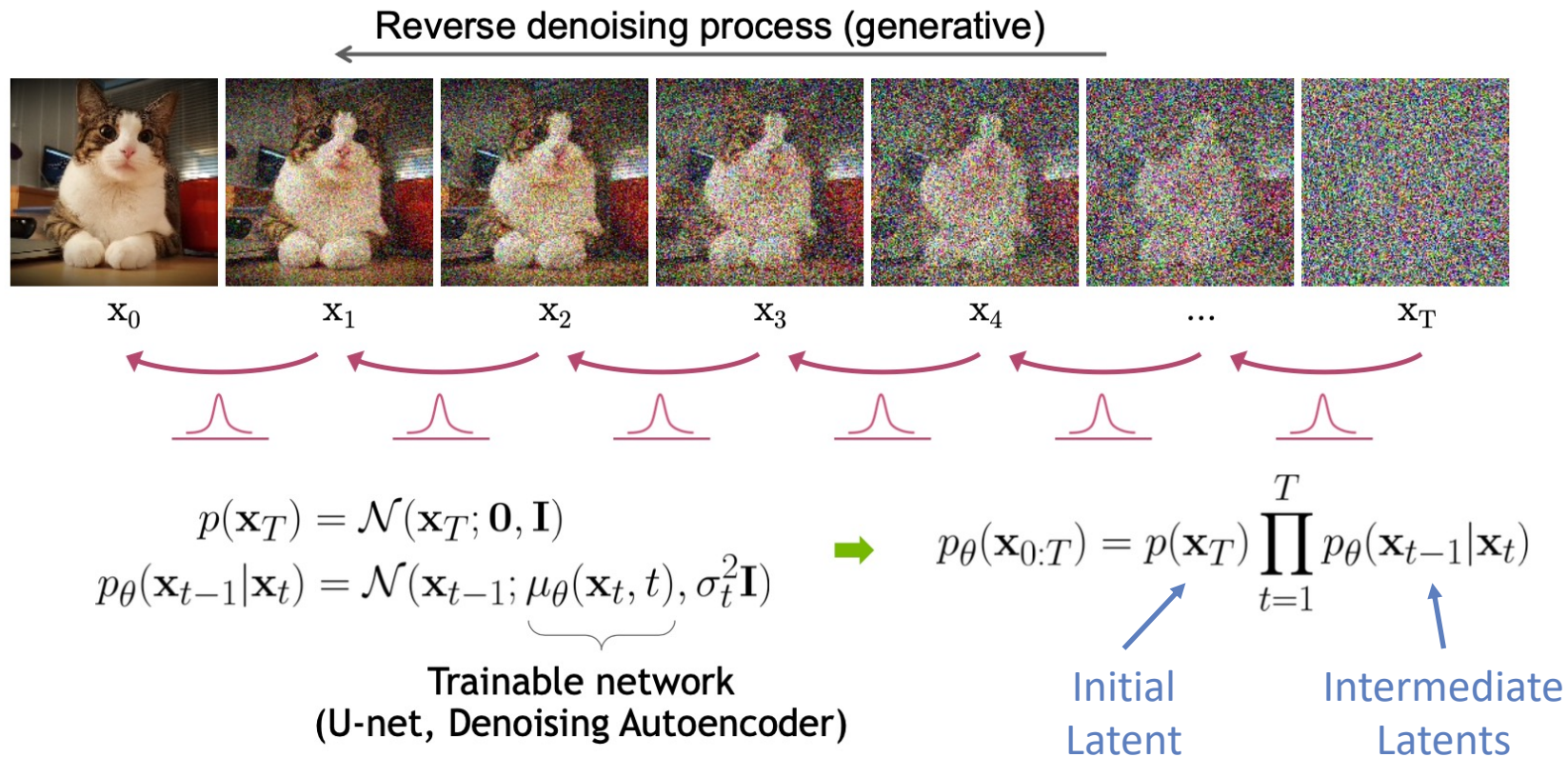
Motivation:

Inpaint **consistent** results
for multi-view image input



Background: Diffusion Models

Inpainted image is generated from a sampled (gaussian) noise, and denoised into a clean image iteratively ($T \sim 0$).



Solution: Align the latents for different input images to achieve consistent multi-view inpainted images.

Method Overview

A conditional-sampling-like approach:

Sample **one** view as the base view,
and align latents of **other views** with it

$$p_{\theta}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)$$

Initial Latent

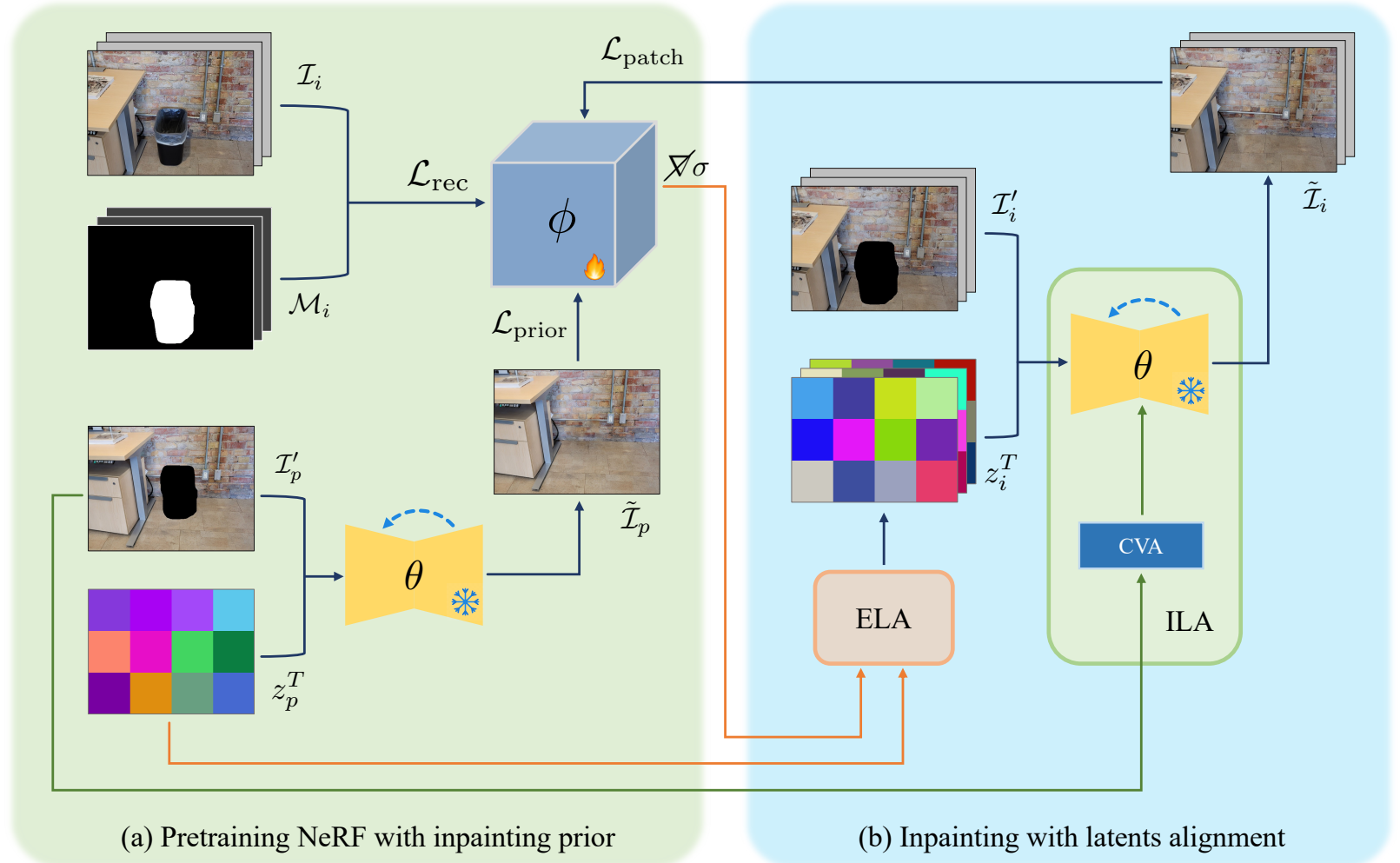
Intermediate Latents

ELA: **Explicit** Initial Latent Alignment

Sample the initial (latent) noise according to **geometry** clue of the 3D representation

ILA: **Implicit** Inter. Latents Alignment

Network **predicts** latent using **shared** key and value **attention elements**

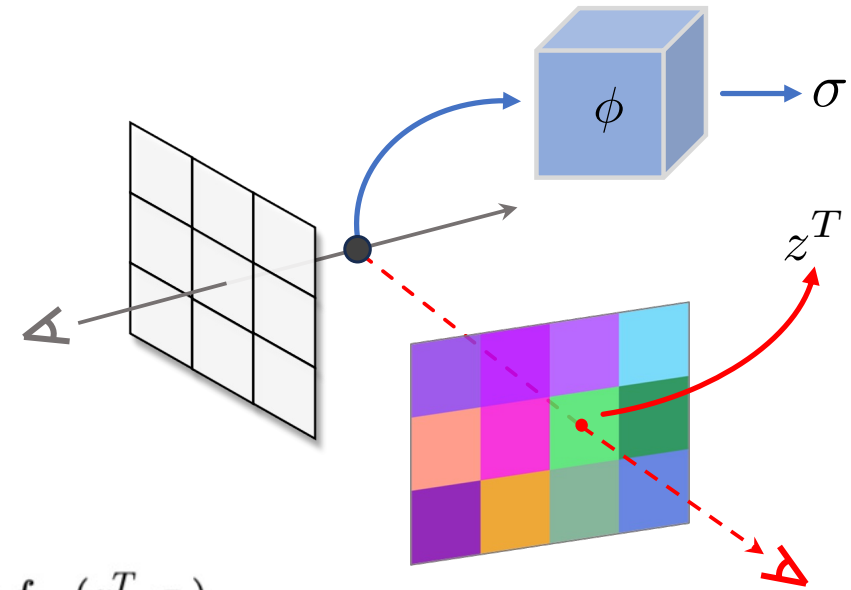


Method: Explicit Initial Latent Alignment

ELA:

- Leverage geometric information to explicitly align the initial latent in 3D space
- Volume rendering using density σ
- Pixel-wise initial noise is sampled by

$$z^T(r) = \sum \Gamma_i \left(1 - \exp(-\sigma(\tau_i)\delta(\tau_i)) \right) z^T(\tau_i), \quad \text{with } z^T(\tau_i) = f_{p,i}(z_p^T, \tau_i).$$



Method: Implicit Intermediate Latent Alignment

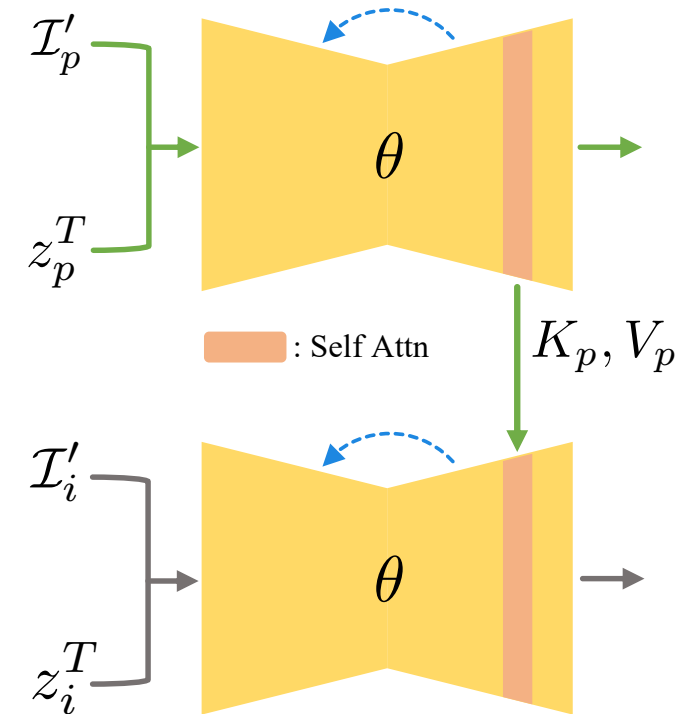
ILA:

- Intermediate latents are predicted by inpainting network, which could only be aligned implicitly
- Inpainting network use self-attention (SA) to infer noise prediction

$$SA(Q_i, K_i, V_i) = \text{Softmax} \left(\frac{Q_i K_i^T}{\sqrt{d}} \right) V_i$$

- We can inject the key (K) and value (V) from base view into current denoising function

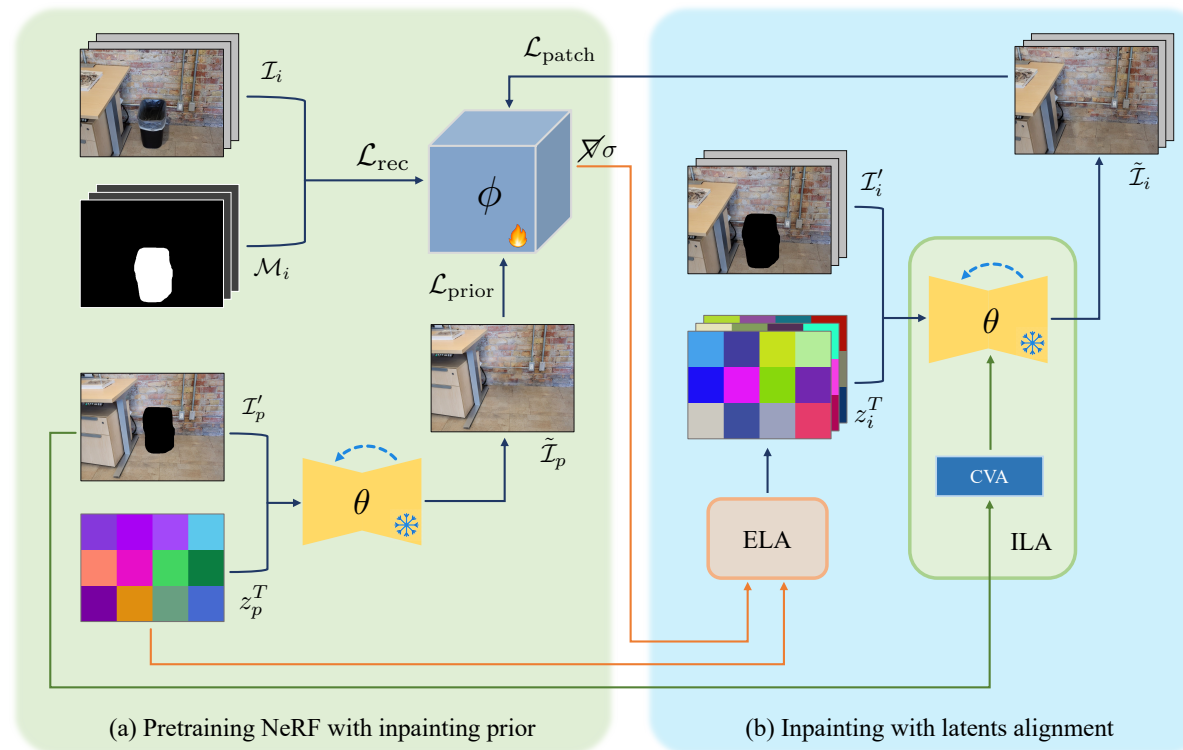
$$CVA(Q_i, K_p, V_p) = \text{Softmax} \left(\frac{Q_i (K_p)^T}{\sqrt{d}} \right) V_p$$



Method: Optimization

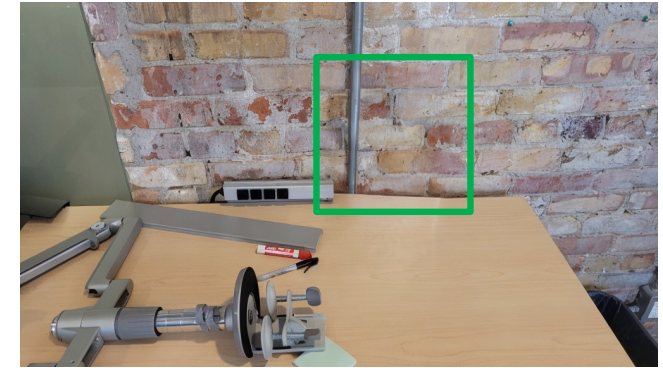
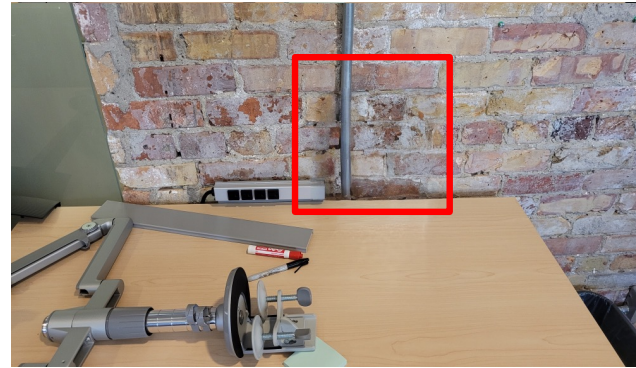
Loss function: L1 loss + perceptual loss + adversarial loss

$$\mathcal{L}_{\text{patch}}(\phi) = \sum_{\rho \in \mathcal{P}_{\text{sub}}} \left\| \hat{I}_{\phi}(\rho) - \tilde{\mathcal{I}}(\rho) \right\|_1 + \mathcal{L}_{\text{lpips}}(\hat{I}_{\phi}(\rho), \tilde{\mathcal{I}}(\rho)) + \mathcal{L}_{\text{adv}}(\hat{I}_{\phi}(\rho), \tilde{\mathcal{I}}(\rho)),$$

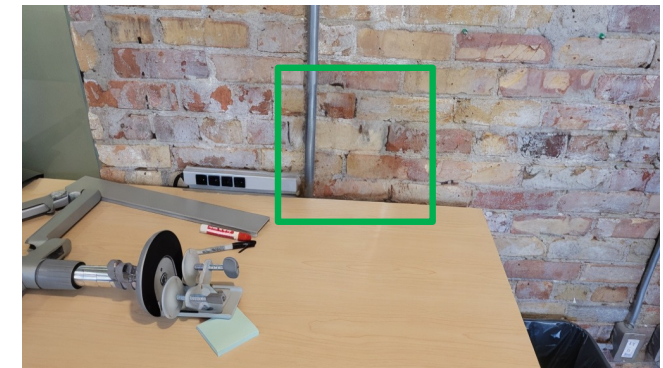


Method: 2D inpainting example

View
#1



View
#2

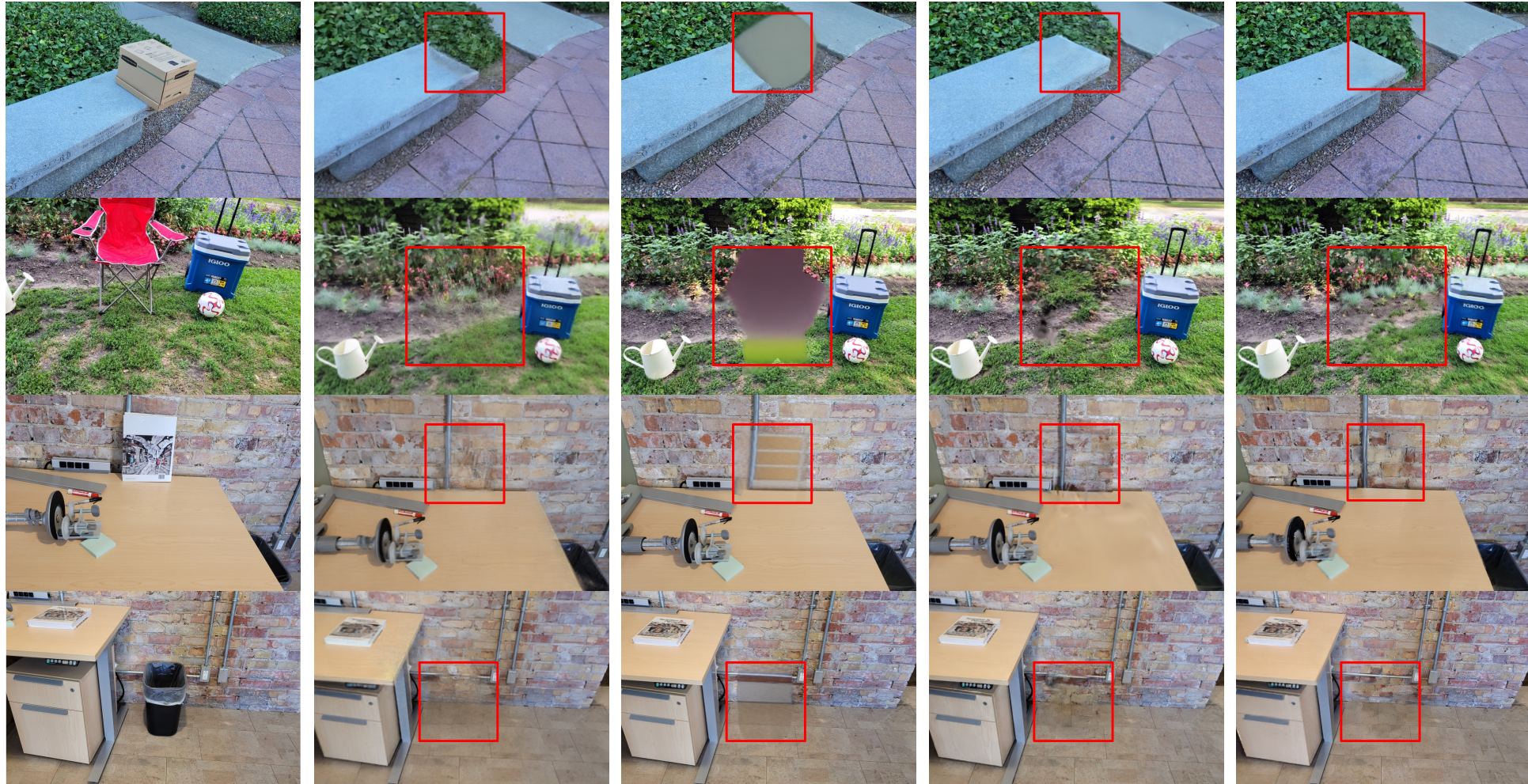


(a) Original Image

(b) Inpainted by Stable Diffusion

(c) Inpainted by Ours

Results: SPIn-NeRF Dataset



Training Inputs

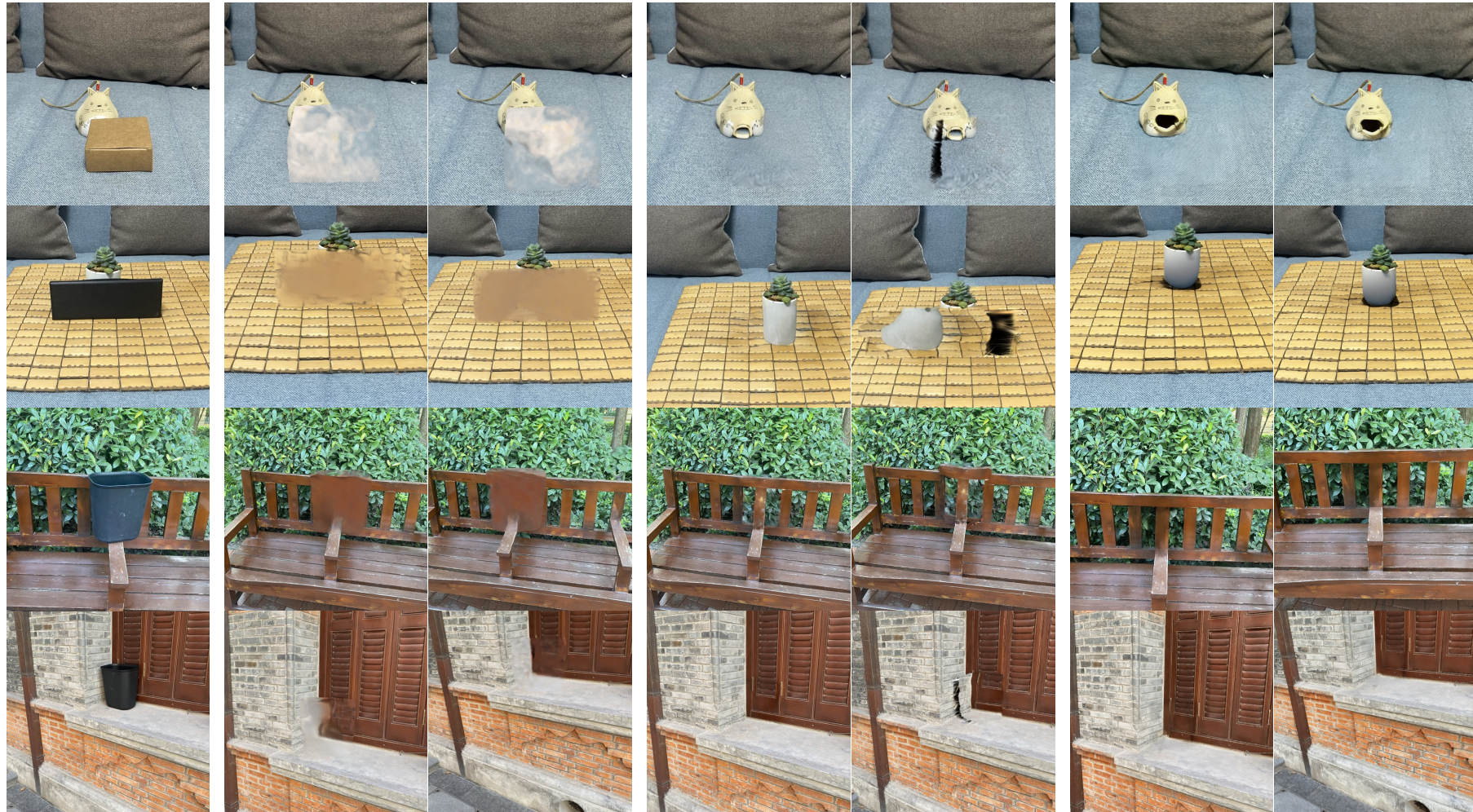
SPIn-NeRF

NeRFiller

InFusion

Ours

Results: Self-Collected Dataset



Training Input

NeRFiller

InFusion

Ours



THE UNIVERSITY OF
MELBOURNE



NEURAL INFORMATION
PROCESSING SYSTEMS

Thank you

