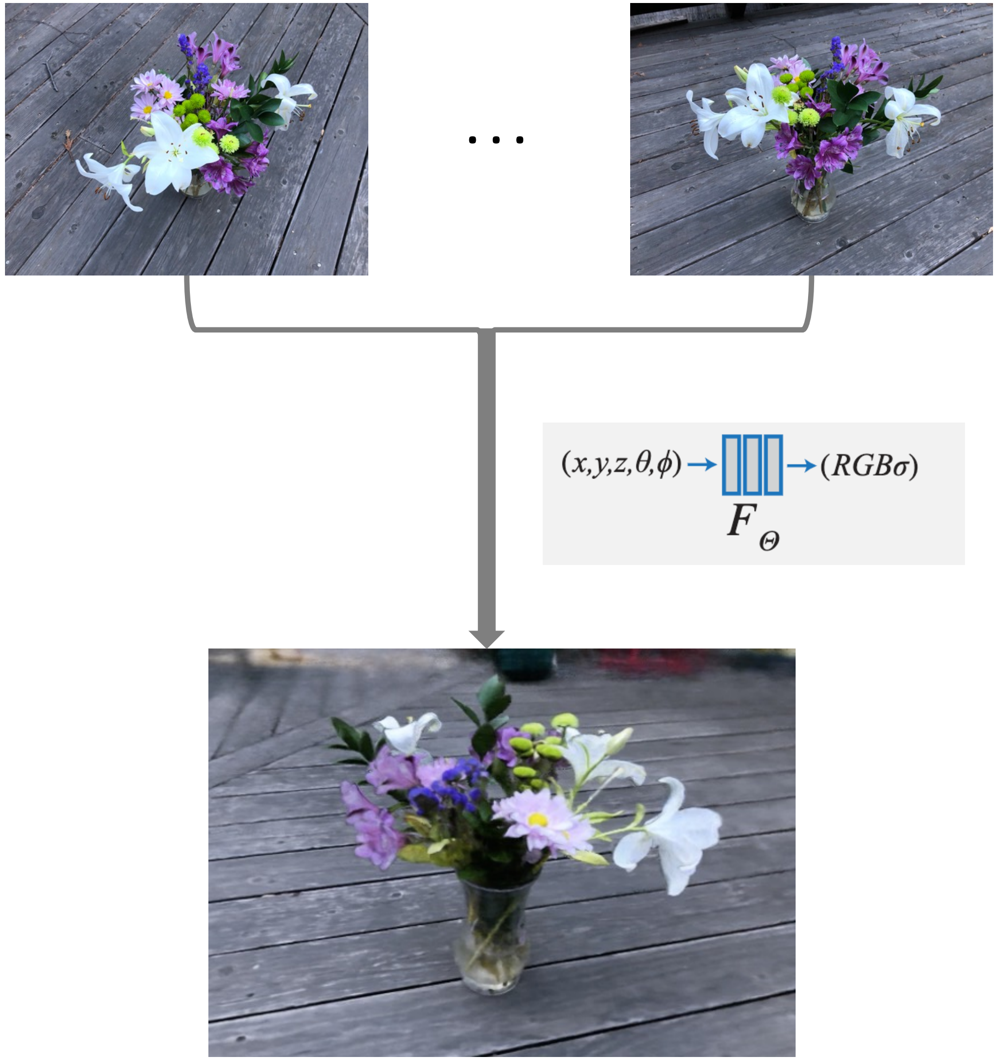
In-N-Out: Lifting 2D Diffusion Prior for 3D Object Removal via Latents Alignment

Dongting Hu[1], Huan Fu[2], Jiaxian Guo[3], Liuhua Peng[1]
Tingjin Chu[1], Feng Liu[1], Tongliang Liu[4,5], Mingming Gong[1,5]

[1] The University of Melbourne [2] Alibaba Group
[3] Google Research [4] The University of Sydney
[5] Mohamed bin Zayed University of Artificial Intelligence

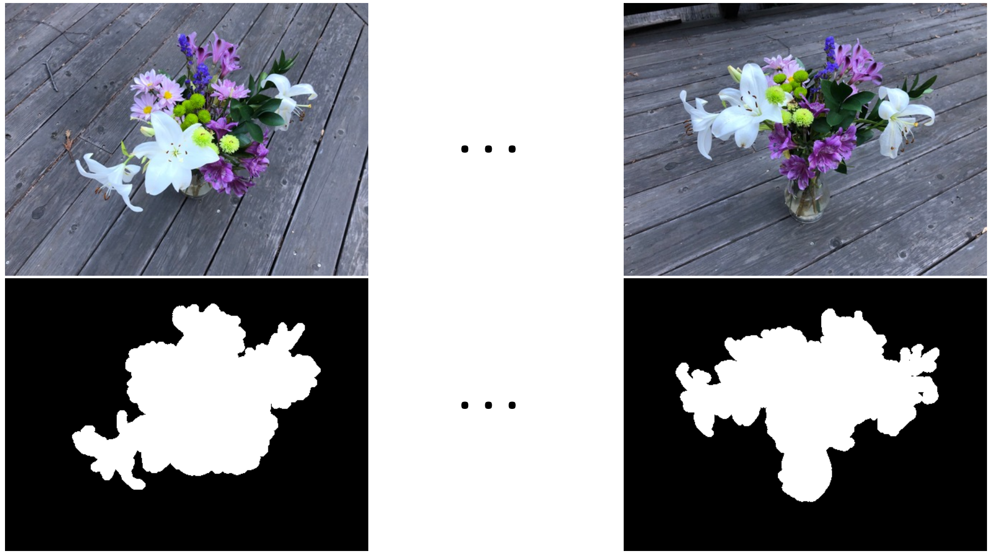
Background: 3D Object Removal



Given:

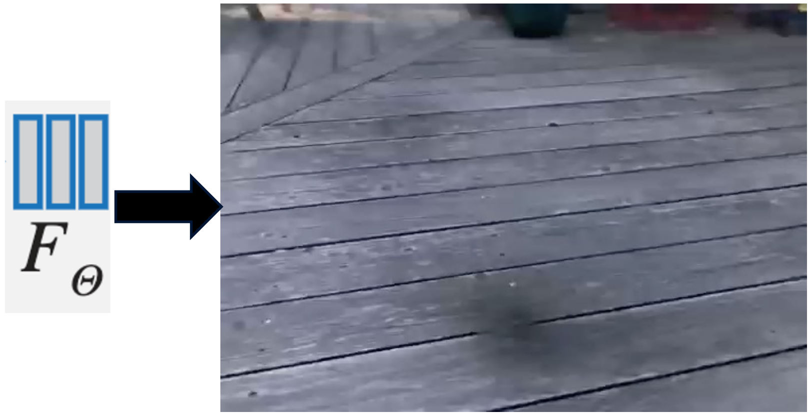
Images

Masks



Target:

Novel Views
w/o object

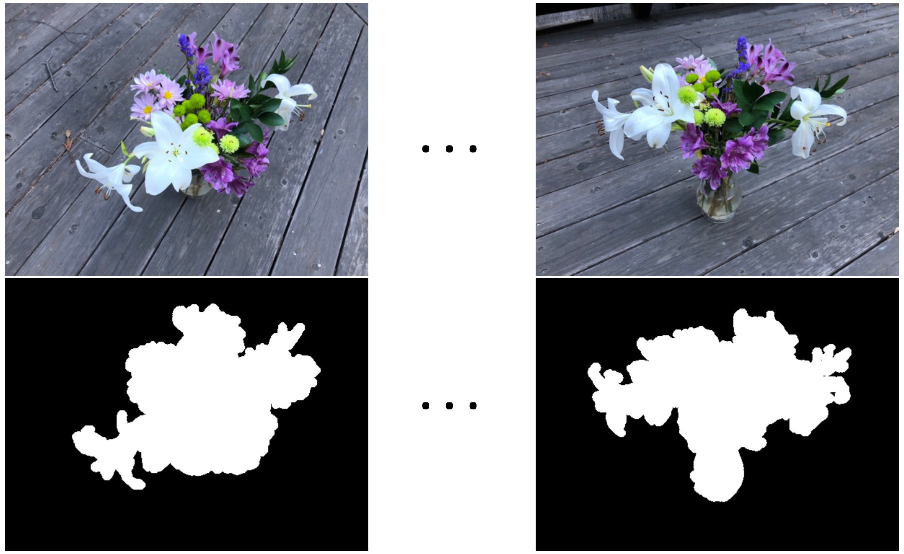


Background: 3D Object Removal

Given:

Images

Masks



Target:

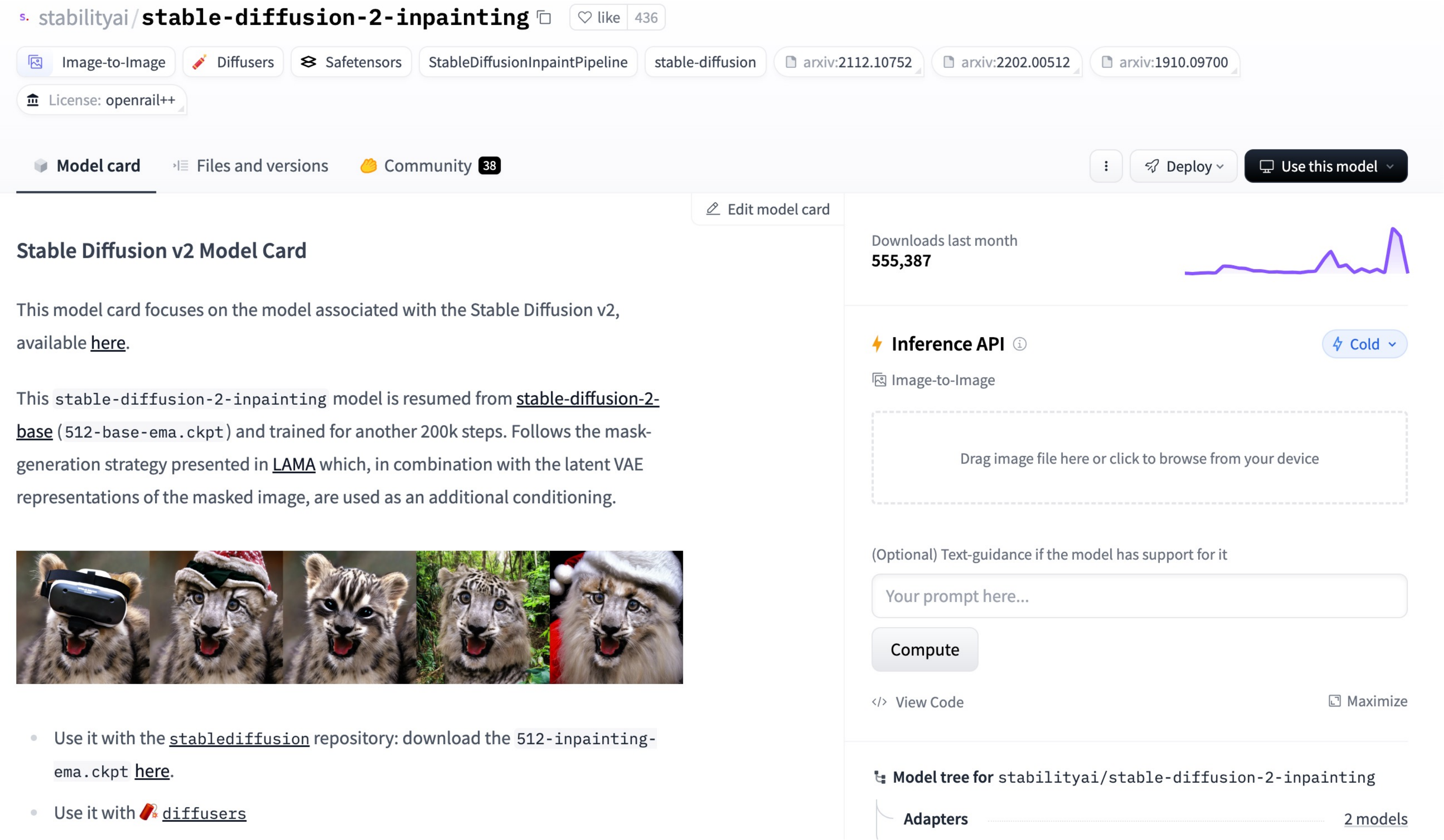
Novel Views
w/o object

Motivation:

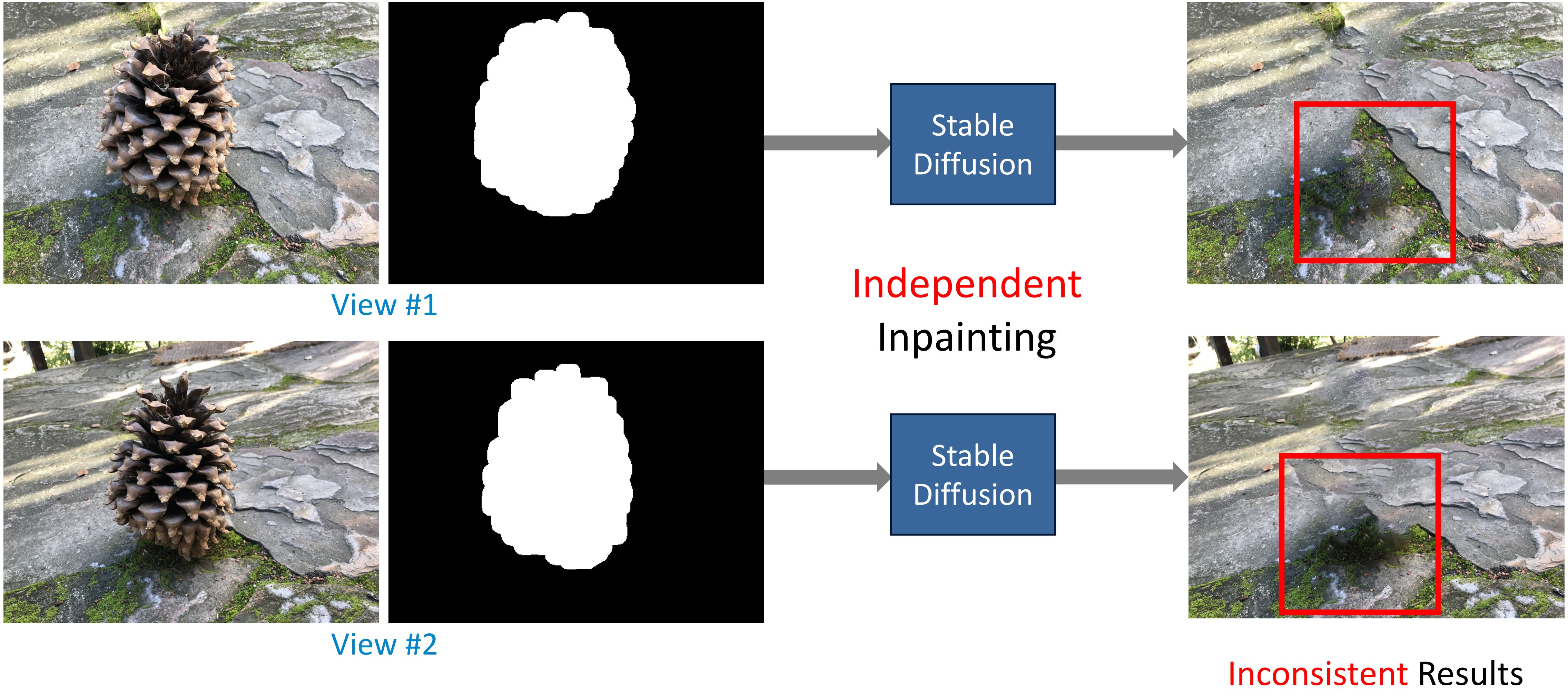
- 3D representation (e.g. NeRF) learns from pixels.
- There is no supervision in the **occluded area.**
- We need to fill in the pixels in the masked area.

Background: Diffusion Inpainting Method

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



Background: Diffusion Inpainting Method



Background: Motivation

3D Inconsistency

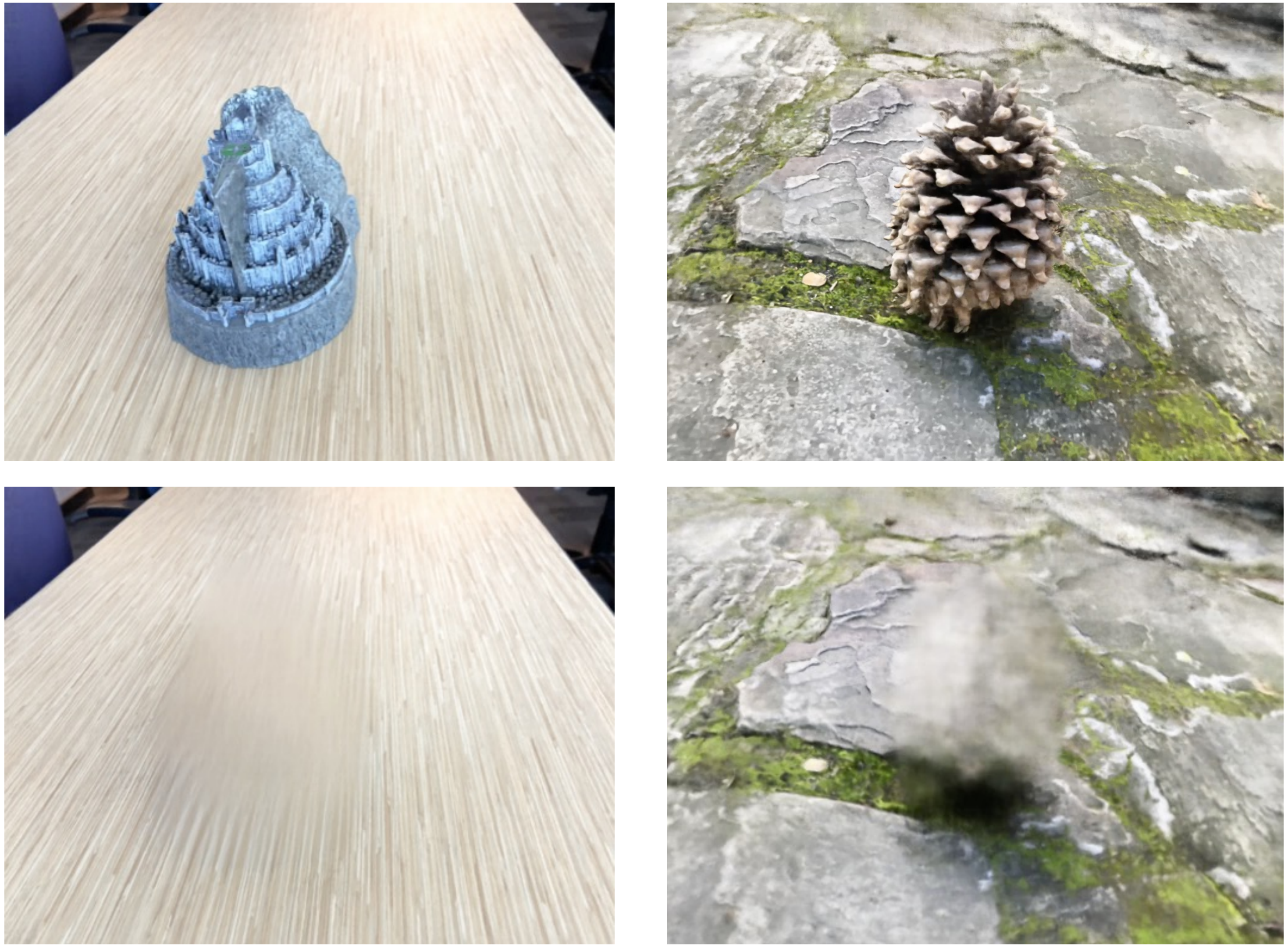
--> Geometric Mismatch

--> Blurred Results

Motivation:

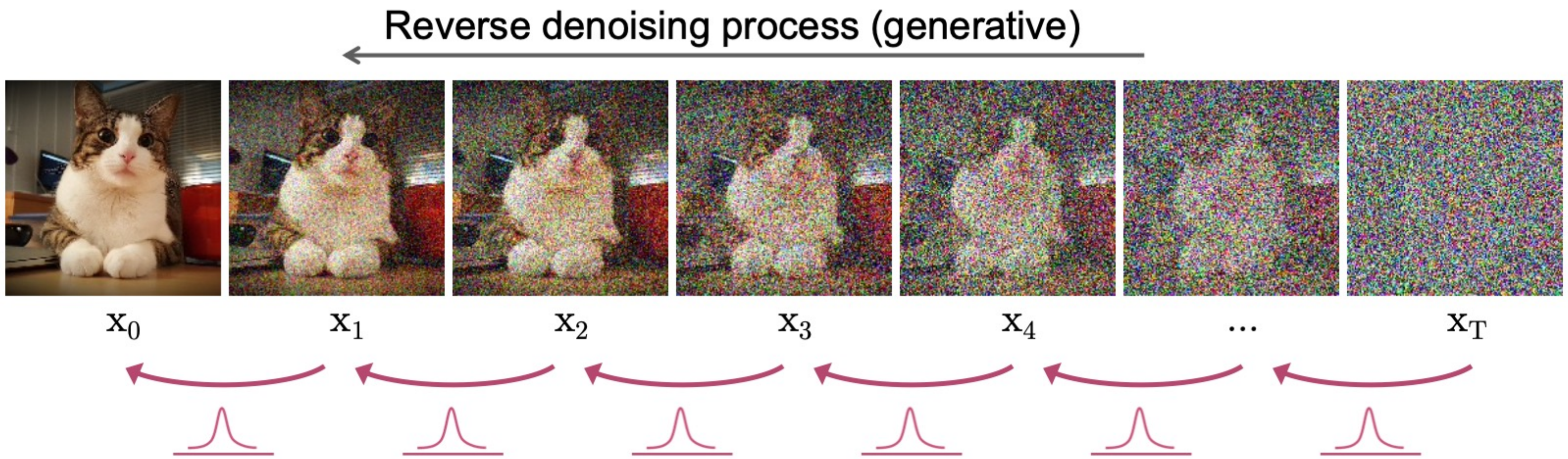
Inpaint **consistent** results

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 ~ 0).



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\text{Trainable network}}, \sigma_t^2\mathbf{I})$$

Trainable network (U-net, Denoising Autoencoder)

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

Initial Latent ($p(\mathbf{x}_T)$)

Intermediate Latents ($p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$)

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

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

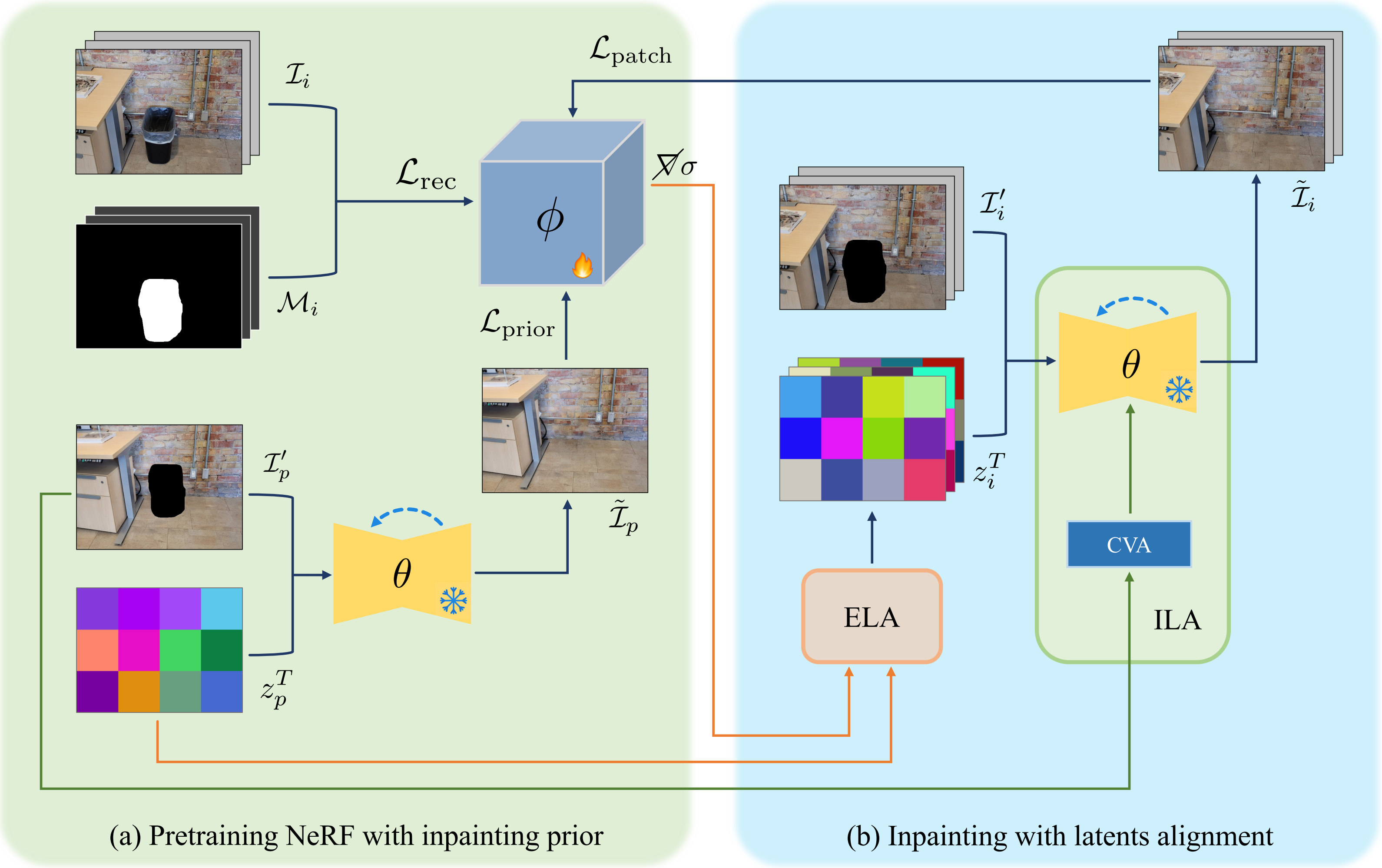
Initial Latent — Intermediate Latents

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

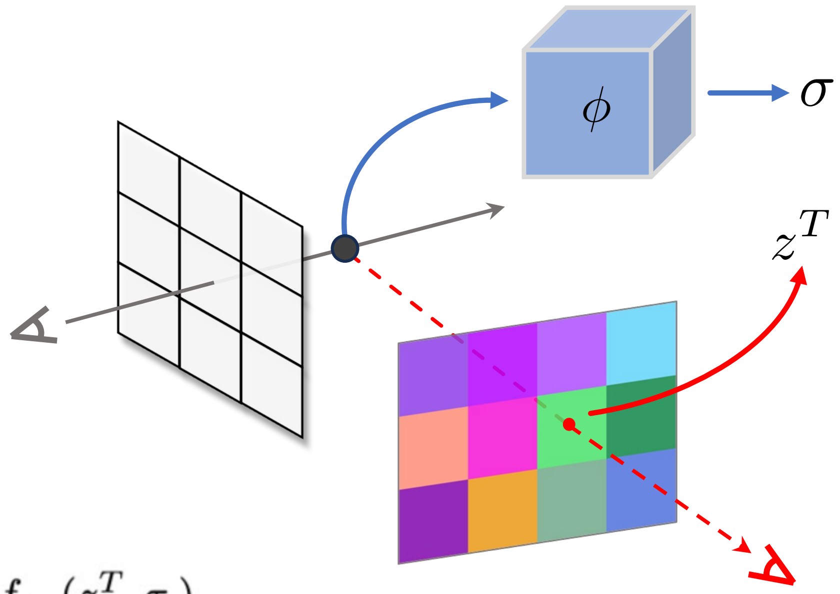


Method: Explicit Initial Latent Alignment

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

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



Method: Implicit Intermediate Latent Alignment

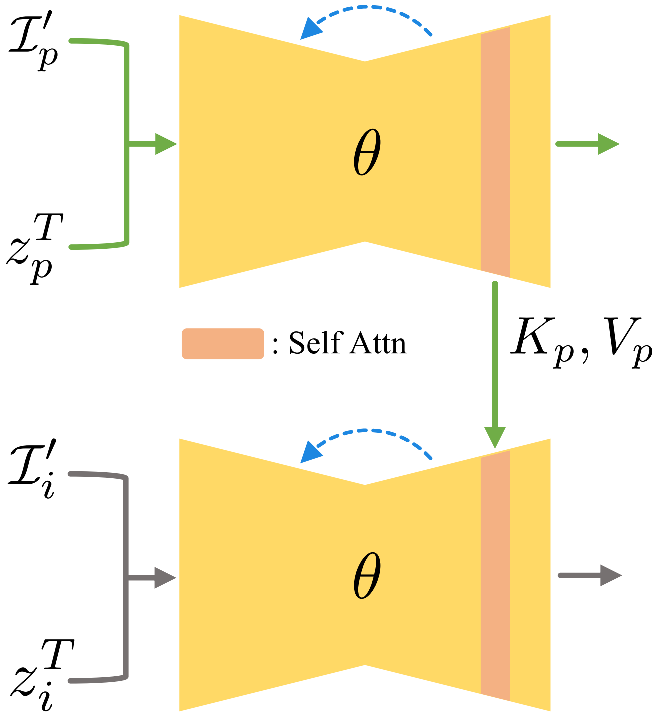
ILA:

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

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

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

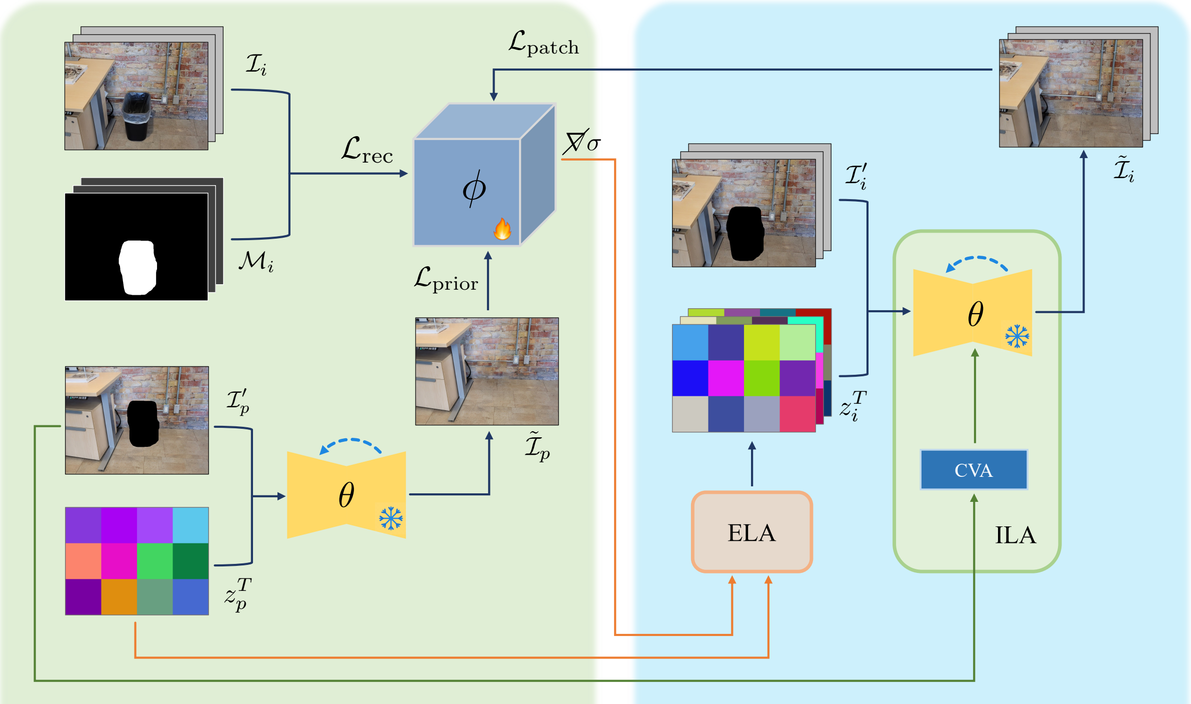
$$\mathrm{CVA}(Q_i, K_p, V_p) = \mathrm{Softmax}\left(\frac{Q_i (K_p)^T}{\sqrt{d}}\right) V_p$$



Method: Optimization

Loss function: L1 loss + perceptual loss + adversarial loss

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



(a) Pretraining NeRF with inpainting prior (b) Inpainting with latents alignment

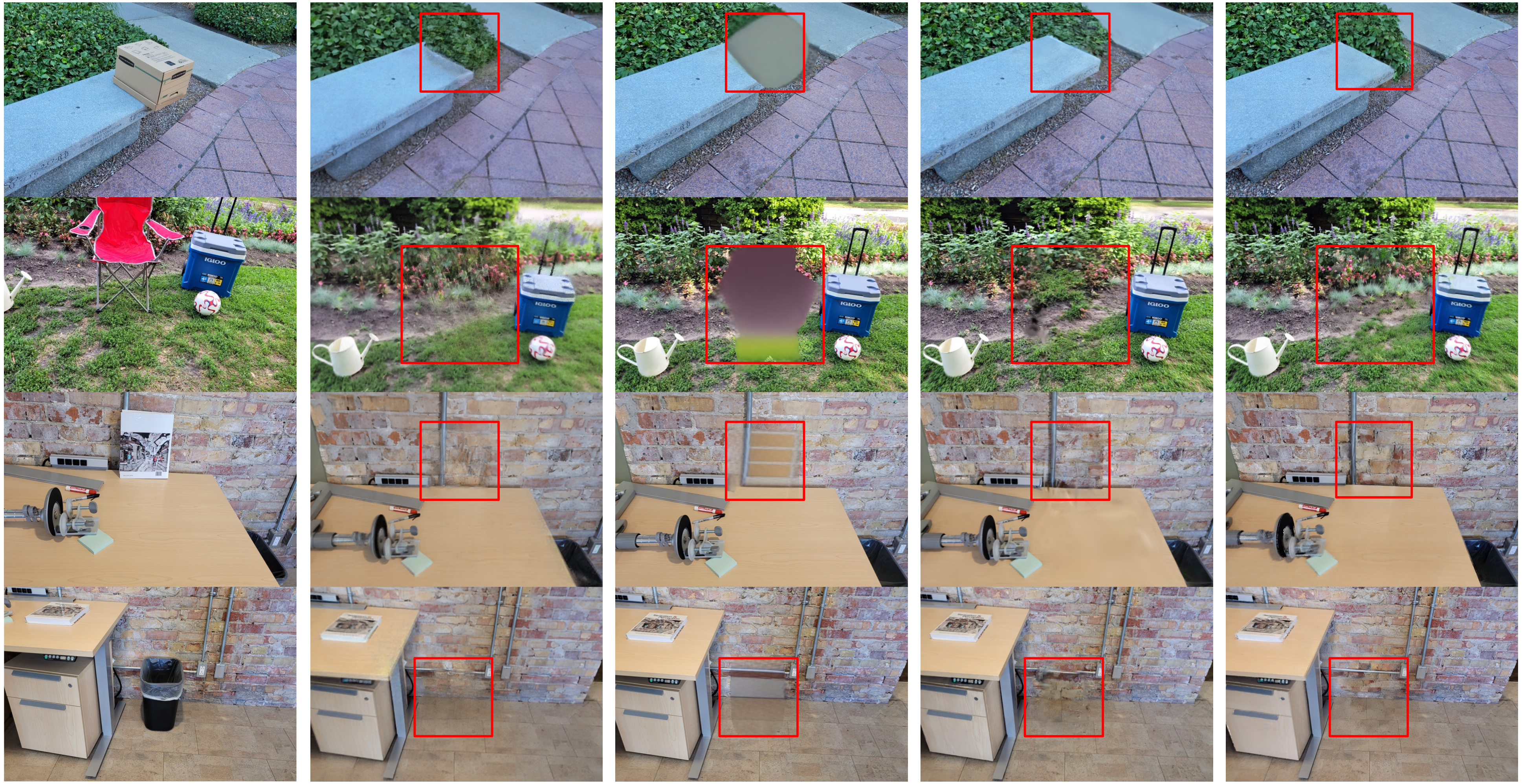
Method: 2D inpainting example

View #1

View #2

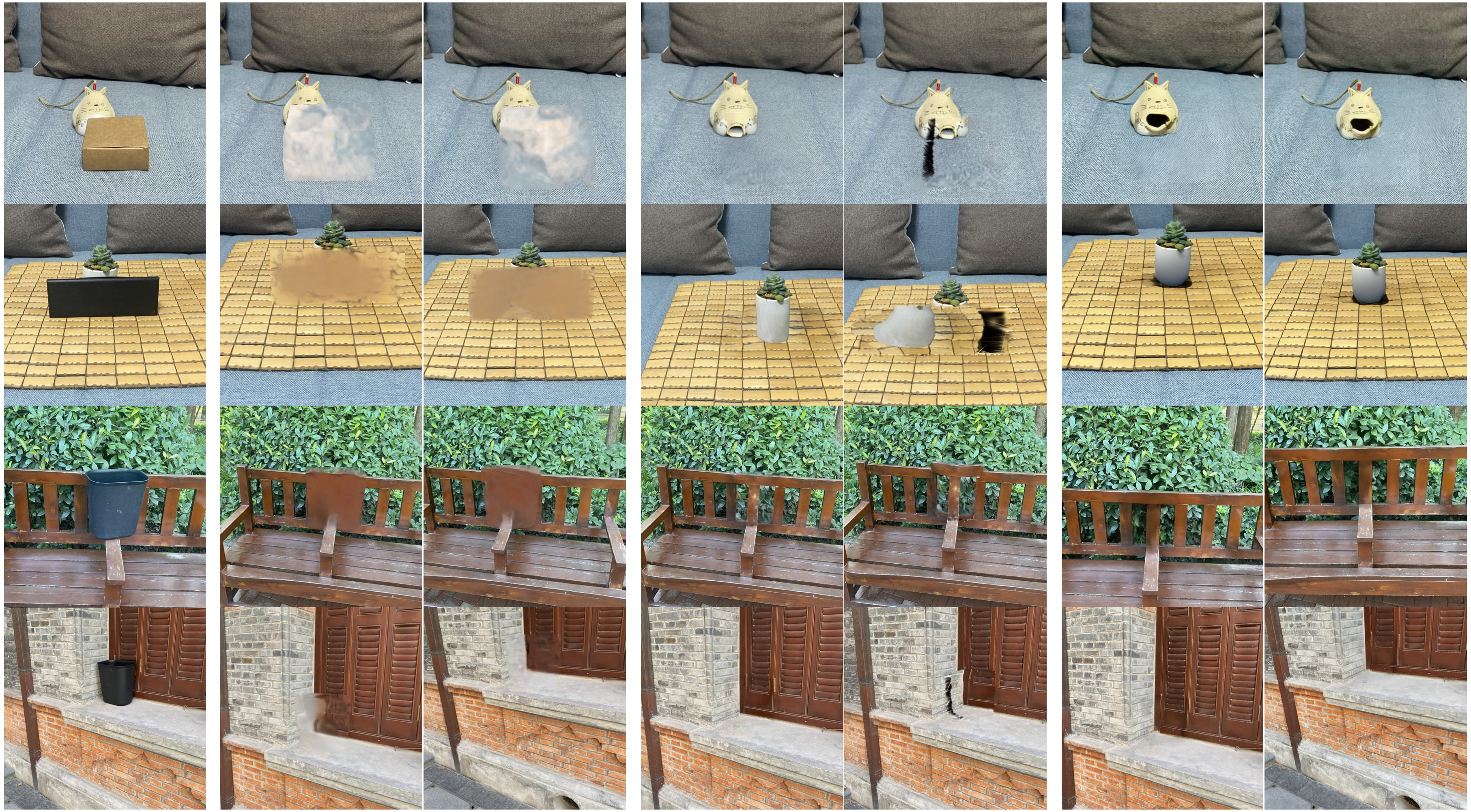
(a) Original Image (b) Inpainted by Stable Diffusion (c) Inpainted by Ours

Results: SPIn-NeRF Dataset



Training Inputs | SPIn-NeRF | NeRFiller | InFusion | Ours

Results: Self-Collected Dataset



Training Input NeRFiller InFusion Ours

Thank you