Structure *from* Duplicates: Neural Inverse Graphics from a Pile of Objects

Tianhang Cheng[1] **Wei-Chiu Ma**[2] **Kaiyu Guan**[1] **Antonio Torralba**[2] **Shenlong Wang**[1]
[1]University of Illinois Urbana-Champaign [2]Massachusetts Institute of Technology
{tcheng12, kaiyug, shenlong}@illinois.edu
{weichium, torralba}@mit.edu

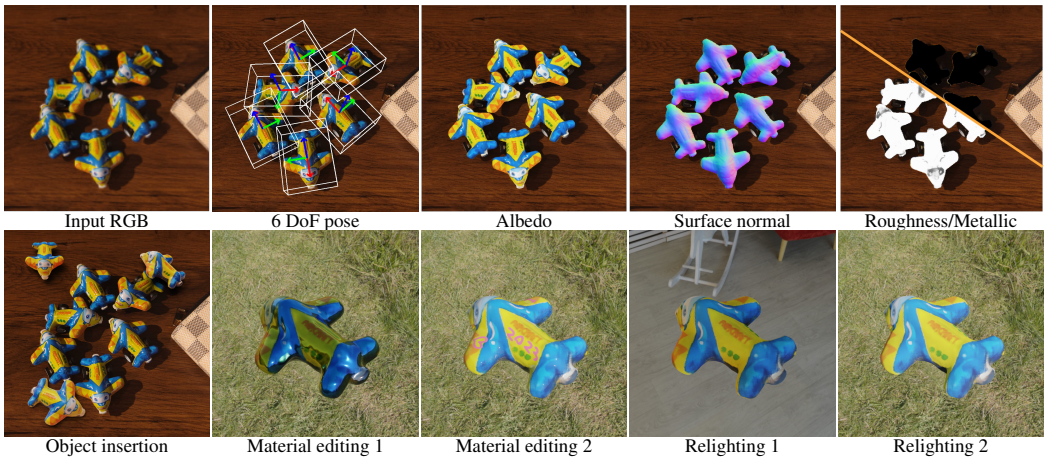


Figure 1: **Structure from duplicates (SfD)** is a novel inverse graphics framework that reconstructs geometry, material, and illumination from a single image containing multiple identical objects.

Abstract

Our world is full of identical objects (*e.g.*, cans of coke, cars of same model). These duplicates, when seen together, provide additional and strong cues for us to effectively reason about 3D. Inspired by this observation, we introduce Structure from Duplicates (SfD), a novel inverse graphics framework that reconstructs geometry, material, and illumination from a single image containing multiple identical objects. SfD begins by identifying multiple instances of an object within an image, and then jointly estimates the 6DoF pose for all instances. An inverse graphics pipeline is subsequently employed to jointly reason about the shape, material of the object, and the environment light, while adhering to the shared geometry and material constraint across instances. Our primary contributions involve utilizing object duplicates as a robust prior for single-image inverse graphics and proposing an in-plane rotation-robust Structure from Motion (SfM) formulation for joint 6-DoF object pose estimation. By leveraging multi-view cues from a single image, SfD generates more realistic and detailed 3D reconstructions, significantly outperforming existing single image reconstruction models and multi-view reconstruction approaches with a similar or greater number of observations. Code available at https://github.com/Tianhang-Cheng/SfD

1 Introduction

Given a single/set of image(s), the goal of inverse rendering is to recover the underlying geometry, material, and lighting of the scene. The task is of paramount interest to many applications in computer vision, graphics, and robotics and has drawn extensive attention across the communities over the past few years[74; 47; 21; 45].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Figure 2: **Repetitions in the visual world.** Our physical world is full of identical objects (*e.g.*, cans of coke, cars of the same model, chairs in a classroom). These duplicates, when seen together, provide additional and strong cues for us to effectively reason about 3D.

Since the problem is ill-posed, prevailing inverse rendering approaches often leverage multi-view observations to constrain the solution space. While these methods have achieved state-of-the-art performance, in practice, it is sometimes difficult, or even impossible, to obtain those densely captured images. To overcome the reliance on multi-view information, researchers have sought to incorporate various structural priors, either data-driven or handcrafted, into the models[5; 36]. By utilizing the regularizations, these approaches are able to approximate the intrinsic properties (*e.g.*, material) and extrinsic factors (*e.g.*, illumination) even from one single image. Unfortunately, the estimations may be biased due to the priors imposed. This makes one ponder: is it possible that we take the best of both worlds? Can we extract multi-view information from a single image under certain circumstances?

Fortunately the answer is yes. Our world is full of repetitive objects and structures. Repetitive patterns in single images can help us extract and utilize multi-view information. For instance, when we enter an auditorium, we often see many identical chairs facing slightly different directions. Similarly, when we go to a supermarket, we may observe multiple nearly-identical apples piled on the fruit stand. Although we may not see *the exact same object* from multiple viewpoints in just one glance, we do see many of the "identical twins" from various angles, which is equivalent to multi-view observations and even more (see Sec. 3 for more details). Therefore, the goal of this paper is to develop a computational model that can effectively infer the underlying 3D representations from a single image by harnessing the repetitive structures of the world.

With these motivations in mind, we present Structure from Duplicates (SfD), a novel inverse rendering model that is capable of recovering high-quality geometry, material, and lighting of the objects from a single image. SfD builds upon insights from structure from motion (SfM) as well as recent advances on neural fields. At its core lies two key modules: (i) a *in-plane rotation robust pose estimation module*, and (ii) a *geometric reconstruction module*. Given an image of a scene with duplicate objects, we exploit the pose estimation module to estimate the relative 6 DoF poses of the objects. Then, based on the estimated poses, we align the objects and create multiple "virtual cameras." This allows us to effectively map the problem from a single-view multi-object setup to a multi-view single-object setting (see Fig. 3). Finally, once we obtain multi-view observations, we can leverage the geometric module to recover the underlying intrinsic and extrinsic properties of the scene. Importantly, SfD can be easily extended to multi-image setup. It can also be seen as a superset of existing NeRF models, where the model will reduce to NeRF when there is only one single object in the scene.

We validate the efficacy of our model on a new dataset called **Dup**, which contains synthetic and real-world samples of duplicated objects since current multi-view datasets lack duplication samples. This allows us to benchmark inverse rendering performance under single-view or multi-view settings. Following previous work [60; 65; 47; 69; 74], we evaluate rendering, relighting and texture quality with MSE, PSNR, SSIM, LPIPS [70], geometry with Chamfer Distance (CD), and environment light with MSE.

Our contributions are as follows: 1) We proposed a novel setting called "single-view duplicate objects" (S-M), which expands the scope of the inverse graphics family with a multi-view single-instance (M-S) framework. 2) Our method produces more realistic material texture than the existing multi-view inverse rendering model when using the same number of training views. 3) Even only relying on a single-view input, our approach can still recover comparable or superior materials and geometry compared to baselines that utilize multi-view images for supervision.

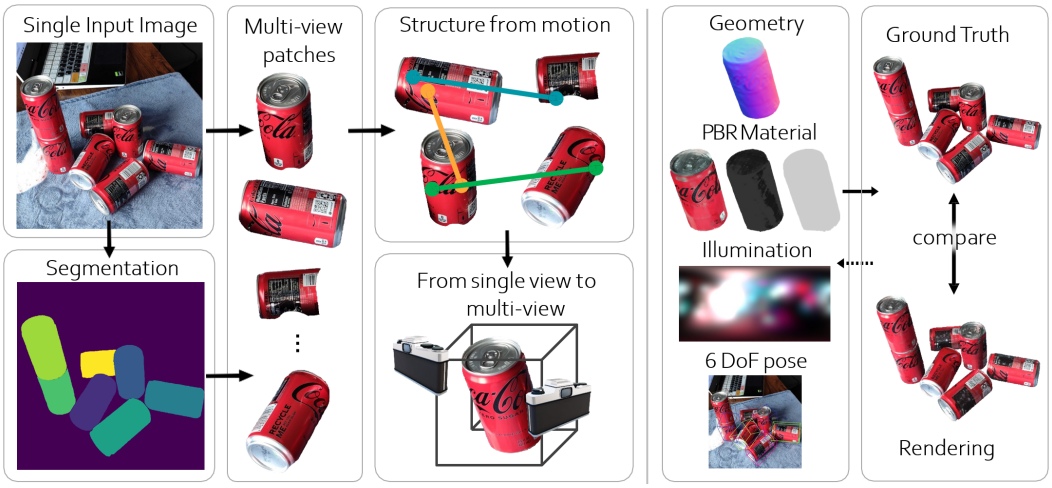


Figure 3: **Method overview: (Left)** SfD begins by identifying multiple instances of an object within an image, and then jointly estimates the 6DoF pose for all instances. **(Right)** An inverse graphics pipeline is subsequently employed to reason about the shape, material of the object, and the environment light, while adhering to the shared geometry and material constraint across instances.

2 Related Work

Inverse rendering: The task of inverse rendering can be dated back to more than half a century ago [31; 23; 24; 2]. The goal is to factorize the appearance of an object or a scene in the observed image(s) into underlying geometry, material properties, and lighting conditions [53; 43; 66]. Since the problem is severely under-constrained, previous work mainly focused on controlled settings or simplifications of the problem [19; 22; 1]. For instance, they either assume the reflectance of an object is spatially invariant [69], presume the lighting conditions are known [57], assume the materials are lambertian [76; 41], or presume the proxy geometry is available [52; 32; 10; 30]. More recently, with the help of machine learning, in particular deep learning, researchers have gradually moved towards more challenging in-the-wild settings [73; 47]. By pre-training on a large amount of synthetic yet realistic data [36] or baking inductive biases into the modeling pipeline [7; 72], these approaches can better tackle unconstrained real-world scenarios (*e.g.*, unknown lighting conditions) and recover the underly physical properties more effectively [47; 68]. For example, through properly modeling the indirect illumination and the visibility of direct illumination, Zhang *et al.* [74] is able to recover interreflection- and shadow-free SVBRDF materials (*e.g.*, albedo, roughness). Through disentangling complex geometry and materials from lighting effects, Wang *et al.* [62] can faithfully relight and manipulate a large outdoor urban scene. Our work builds upon recent advances in neural inverse rendering. Yet instead of grounding the underlying physical properties through multi-view observations as in prior work, we focus on the single image setup and capitalize on the duplicate objects in the scene for regularization. The repetitive structure not only allows us to ground the geometry, but also provide additional cues on higher-order lighting effects (*e.g.*, cast shadows). As we will show in the experimental section, we can recover the geometry, materials, and lighting much more effectively even when comparing to multi-view observations.

3D Reconstruction: Recovering the spatial layout of the cameras and the geometry of the scene from a single or a collection of images is a longstanding challenge in computer vision. It is also the cornerstone for various downstream applications in computer graphics and robotics such as inverse rendering [43; 60; 62], 3D editing [39; 63], navigation [42; 67], and robot manipulation [27; 38]. Prevailing 3D reconstruction systems, such as structure from motion (SfM), primarily rely on multi-view geometry to estimate the 3D structure of a scene [40; 20; 54]. While achieving significant successes, they rely on densely captured images, limiting their flexibility and practical use cases. Single image 3D reconstruction, on the other hand, aims to recover metric 3D information from a monocular image [14; 6; 50; 49]. Since the problem is inherently ill-posed and lacks the ability to leverage multi-view geometry for regularization, these methods have to resort to (learned) structural priors to resolve the ambiguities. While they offer greater flexibility, their estimations may inherit biases from the training data. In this paper, we demonstrate that, under certain conditions, it is possible to incorporate multi-view geometry into a single image reconstruction system. Specifically, we leverage repetitive objects within the scene to anchor the underlying 3D structure. By treating

each of these duplicates as an observation from different viewpoints, we can achieve highly accurate metric 3D reconstruction from a single image.

Repetitions: Repetitive structures and patterns are ubiquitous in natural images. They play important roles in addressing numerous computer vision problems. For instance, a single natural image often contains substantial redundant patches [77]. The recurrence of small image patches allows one to learn a powerful prior which can later be utilized for various tasks such as super-resolution [18; 26], image deblurring [44], image denoising [15], and texture synthesis [12]. Moving beyond patches, repetitive primitives or objects within the scene also provide informative cues about their intrinsic properties [33; 64]. By sharing or regularizing their underlying representation, one can more effectively constrain and reconstruct their 3D geometry [25; 16; 9], as well as enable various powerful image/shape manipulation operations [34; 59]. In this work, we further push the boundary and attempt to recover not just the geometry, but also the materials (*e.g.*, albedo, roughness), visibilities, and lighting conditions of the objects. Perhaps closest to our work is [75]. Developed independently and concurrently, Zhang *et al.* build a generative model that aims to capture object intrinsics from a single image with multiple similar/same instances. However, there exist several key differences: 1) we explicitly recover metric-accurate camera poses using multi-geometry, whereas Zhang et al. learn this indirectly through a GAN-loss; 2) we parameterize and reason realistic PBR material and environmental light; 3) we handle arbitrary poses, instead of needing to incorporate a prior pose distribution. Some previous work also tried to recover 6D object pose from crowded scene or densely packed objects[46; 11] from RGB-D input, but our model only require RGB input.

3 Structure from Duplicates

In this paper, we seek to devise a method that can precisely reconstruct the geometry, material properties, and lighting conditions of an object from *a single image containing duplicates of it*. We build our model based on the observation that repetitive objects in the scene often have different poses and interact with the environment (*e.g.*, illumination) differently. This allows one to extract rich *multi-view* information even from one single view and enables one to recover the underlying physical properties of the objects effectively.

We start by introducing a method for extracting the "multi-view" information from duplicate objects. Then we discuss how to exploit recent advances in neural inverse rendering to disentangle both the object intrinsics and environment extrinsics from the appearance. Finally, we describe our learning procedure and design choices.

3.1 Collaborative 6-DoF pose estimation

As we have alluded to above, a single image with multiple duplicate objects contains rich multi-view information. It can help us ground the underlying geometry and materials of the objects, and understand the lighting condition of the scene.

Our key insight is that the image can be seen as *a collection of multi-view images stitching together*. By cropping out each object, we can essentially transform the single image into a set of multi-view images of the object from various viewpoints. One can then leverage structure from motion (SfM) [55] to estimate the relative poses among the multi-view images, thereby aggregating the information needed for inverse rendering. Notably, the estimated camera poses can be inverted to recover the 6 DoF poses of the duplicate objects. As we will elaborate in Sec. 3.2, this empowers us to more effectively model the extrinsic lighting effect (which mainly depends on world coordinate) as well as to properly position objects in perspective and reconstruct the exact same scene.

To be more formal, let $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ be an image with N duplicate objects. Let $\{\mathcal{I}_i^{\text{obj}}\}_{i=1}^N \in \mathbb{R}^{w\times h\times 3}$ be the corresponding object image patches. We first leverage SfM [55; 51] to estimate the camera poses of the multi-view cropped images* $\{\boldsymbol{\xi}_i \in \mathbb{SE}(3)\}_{i=1}^N$:

$$\boldsymbol{\xi}_1^{\text{cam}}, \boldsymbol{\xi}_2^{\text{cam}}, ..., \boldsymbol{\xi}_N^{\text{cam}} = f^{\text{SfM}}(\mathcal{I}_1^{\text{obj}}, \mathcal{I}_2^{\text{obj}}, ..., \mathcal{I}_K^{\text{obj}}). \tag{1}$$

Next, since there is only one real camera in practice, we can simply align the N virtual cameras $\{\boldsymbol{\xi}_i\}_{i=1}^N$ to obtain the 6 DoF poses of the duplicate objects. Without loss of generality and for

*In practice, the cropping operation will change the intrinsic matrix of the original camera during implementation. For simplicity, we assume the intrinsics are properly handled here.

simplicity, we align all the cameras to a reference coordinate $\boldsymbol{\xi}^{\text{ref}}$. The 6 DoF poses of the duplicate objects thus become $\boldsymbol{\xi}_i^{\text{obj}} = \boldsymbol{\xi}^{\text{ref}} \circ (\boldsymbol{\xi}_i^{\text{cam}})^{-1}$, where $\circ$ is matrix multiplication for pose composition.

In practice, we first employ a state-of-the-art panoptic segmentation model [8] to segment all objects in the scene. Then we fit a predefined bounding box to each object and crop it. Lastly, we run COLMAP [55] to estimate the 6 DoF virtual camera poses, which in turn provides us with the 6 DoF object poses. Fig. 3(left) depicts our collaborative 6-DoF pose estimation process.

Caveats of random object poses: Unfortunately, naively feeding these object patches into SfM would often leads to failure, as little correspondence can be found. This is due to the fact that state-of-the-art correspondence estimators [51] are trained on Internet vision data, where objects are primarily upright. The poses of the duplicate objects in our case, however, vary significantly. Moreover, the objects are often viewed from accidental viewpoints [17]. Existing estimators thus struggle to match effectively across such extreme views.

Rotation-aware data augmentation: Fortunately, the scene contains numerous duplicate objects. While estimating correspondences reliably across arbitrary instances may not always be possible, there are certain objects whose viewpoints become significantly similar after in-plane rotation. Hence, we have developed an in-plane rotation-aware data augmentation for correspondence estimation.

Specifically, when estimating correspondences between a pair of images, we don't match the images directly. Instead, we gradually rotate one image and then perform the match. The number of correspondences at each rotation is recorded and then smoothed using a running average. We take the `argmax` to determine the optimal rotation angle. Finally, we rotate the correspondences from the best match back to the original pixel coordinates. As we will demonstrate in Sec. 4, this straightforward data augmentation strategy significantly improves the accuracy of 6 DoF pose estimation. In practice, we rotate the image by 4° per step. All the rotated images are batched together, enabling us to match the image pairs in a single forward pass.

3.2 Joint shape, material, and illumination estimation

Suppose now we have the 6 DoF poses of the objects $\{\boldsymbol{\xi}_i^{\text{obj}}\}_{i=1}^N$. The next step is to aggregate the information across duplicate objects to recover the *intrinsics properties of the objects* (*e.g.*, geometry, materials) and *the extrinsic factors of the world* (*e.g.*, illumination). We aim to reproduce these attributes as faithfully as possible, so that the resulting estimations can be utilized for downstream tasks such as relighting and material editing. Since the task is under-constrained and joint estimation often leads to suboptimal results, we follow prior art [74; 62] and adopt a stage-wise procedure.

Geometry reconstruction: We start by reconstructing the geometry of the objects. In line with NeuS [60], we represent the object surfaces as the zero level set of a signed distance function (SDF).

We parameterize the SDF with a multi-layer perceptron (MLP) $S : \mathbf{x}^{\text{obj}} \mapsto s$ that maps a 3D point under object coordinate $\mathbf{x}^{\text{obj}} \in \mathbb{R}^3$ to a signed distance value $s \in \mathbb{R}$. Different from NeuS [60], we model the geometry of objects in local object space. This allows us to guarantee shape consistency across instances by design.

We can also obtain the surface normal by taking the gradient of the SDF: $\mathbf{n}(\mathbf{x}^{\text{obj}}) = \nabla_{\mathbf{x}^{\text{obj}}} S$. To learn the geometry from multi-view images, we additionally adopt an *auxiliary* appearance MLP $C : \{\mathbf{x}, \mathbf{x}^{\text{obj}}, \mathbf{n}, \mathbf{n}^{\text{obj}}, \mathbf{d}, \mathbf{d}^{\text{obj}}\} \mapsto \mathbf{c}$ that takes as input a 3D point $\mathbf{x}, \mathbf{x}^{\text{obj}}$, surface normal $\mathbf{n}, \mathbf{n}^{\text{obj}}$, and view direction $\mathbf{d}, \mathbf{d}^{\text{obj}}$ under both coordinate systems and outputs the color $\mathbf{c} \in \mathbb{R}^3$. The input from world coordinate system helps the MLP to handle the appearance inconsistencies across instances caused by lighting or occlusion. We tied the weights of the early layers of C to those of S so that the gradient from color can be propagated to geometry. We determine which object coordinate to use based on the object the ray hits. This information can be derived either from the instance segmentation or another allocation MLP $A : \{\mathbf{x}, \mathbf{d}\} \mapsto q$ (distilled from SDF MLP), where $q \in \mathbb{N}$ is the instance index. After we obtain the geometry MLP S, we discard the auxiliary appearance MLP C. As we will discuss in the next paragraph, we model the object appearance using physics-based rendering (PBR) materials so that it can handle complex real-world lighting scenarios.

Material and illumination model: Now we have recovered the geometry of the objects, the next step is to estimate the environment light of the scene as well as the material of the object. We assume

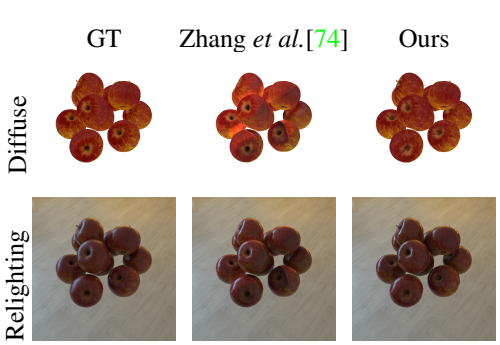


Figure 4: **Multi-view inverse rendering.**

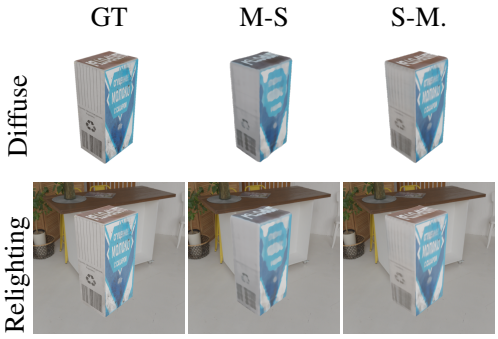


Figure 5: **Multi-view single object (M-S) vs single-view multi-objects (S-M).**

all lights come from an infinitely faraway sphere and only consider direct illumination. Therefore, the illumination emitted to a 3D point in a certain direction is determined solely by the incident light direction $\boldsymbol{w}_i$ and is independent of the point's position. Similar to [69; 74], we approximate the environment light with $M = 128$ Spherical Gaussians (SGs):

$$L_i\left(\omega_i\right) = \sum\nolimits_{k=1}^{M} \boldsymbol{\mu}_k e^{\lambda_k(\boldsymbol{w}_i \cdot \boldsymbol{\phi}_k - 1)}, \tag{2}$$

where $\lambda \in \mathbb{R}^+$ is the lobe sharpness, $\boldsymbol{\mu}$ is the lobe amplitude, and $\boldsymbol{\phi}$ is the lobe axis. This allows us to effectively represent the illumination and compute the rendering equation (Eq. 3) in closed-form.

As for object material, we adopt the simplified Disney BRDF formulation [3; 29] and parameterized it as a MLP $M : \mathbf{x}^{\text{obj}} \mapsto \{\mathbf{a}, r, m\}$. Here, $\mathbf{a} \in \mathbb{R}^3$ denotes albedo, $r \in [0, 1]$ corresponds to roughness, and $m \in [0, 1]$ signifies metallic. Additionally, inspired by InvRender [74], we incorporate a visibility MLP $V : (\mathbf{x}, \boldsymbol{w}_i) \mapsto v \in [0, 1]$ to approximate visibility for each world coordinate faster reference. The difference is that we use the sine activation function[56] rather than the ReLU activation for finer detail. Since the visibility field is in world space, we can model both inter-object self-casted shadows and inter-object occlusions for multiple instances in our setup. We query only surface points, which can be derived from the geometry MLP S using volume rendering. The material MLP M also operates in object coordinates like S, which ensure material consistency across all instances. Moreover, the variations in lighting conditions between instances help us better disentangle the effects of lighting from the materials. We set the dielectrics Fresnel term to $F_0 = 0.02$ and the general Fresnel term to $\mathbf{F} = (1 - m)F_0 + m\mathbf{a}$ to be compatible with both metals and dielectrics.

Combining all of these components, we can generate high-quality images by integrating the visible incident lights from hemisphere and modeling the effects of BRDF [28]:

$$L_o(\boldsymbol{w}_o; \mathbf{x}) = \int_{\Omega} L_i(\boldsymbol{w}_i) f_r(\boldsymbol{w}_i, \boldsymbol{w}_o; \mathbf{x})(\boldsymbol{w}_i \cdot \mathbf{n}) d\boldsymbol{w}_i. \tag{3}$$

Here, $\boldsymbol{w}_i$ is the incident light direction, while $\boldsymbol{w}_o$ is the viewing direction. The BRDF function f_r can be derived from our PBR materials. We determine the visibility of an incident light either through sphere tracing or following Zhang *et al.* [74] to approximate it with the visibility MLP V. We refer the readers to the supplementary materials for more details.

3.3 Optimization

Optimizing shape, illumination, and material jointly from scratch is challenging. Taking inspiration from previous successful approaches [74; 69], we implement a multi-stage optimization pipeline. We progressively optimize the geometry first, then visibility, and finally, material and illumination.

Geometry optimization: We optimize the geometry model by minimizing the difference between rendered cues and observed cues.

$$\min_{S,C} E_{\text{color}} + \lambda_1 E_{\text{reg}} + \lambda_2 E_{\text{mask}} + \lambda_3 E_{\text{normal}}, \tag{4}$$

where $\lambda_1 = 0.1$, $\lambda_2 = \lambda_3 = 0.5$. And each term is defined as follows:

Multi-view	Albedo	Roughness	Relighting	Env Light	Geometry
	PSNR ↑	MSE ↓	PSNR ↑	MSE ↓	CD ↓
PhySG	16.233	0.087	21.323	0.054	0.024
Nv-DiffRec	16.123	0.116	17.418	0.168	0.268
InvRender	16.984	0.084	22.224	0.067	0.024
Ours	**21.961**	**0.026**	**25.486**	**0.029**	**0.011**

Single-view	Albedo	Roughness	Relighting	Env Light	Geometry
	PSNR ↑	MSE ↓	PSNR ↑	MSE ↓	CD ↓
PhySG*	14.977	0.255	18.504	0.082	**0.033**
Nv-DiffRec*	14.021	0.165	17.214	0.067	0.050
InvRender*	14.724	0.247	17.998	0.082	0.033
Ours	**17.629**	**0.062**	**21.374**	**0.052**	0.034

Table 1: **(Left) Multi-view inverse rendering on synthetic data**. Both our model and the baseline are trained on multi-view images. Our model is significantly better than baseline in terms of geometry and PBR texture. **(Right) Single-view inverse rendering on synthetic data.** While our model is trained on a *single-view image*, the baselines $*$ are trained on 10 *multi-view images* of the same scene.

- The *color consistency term* E_{color} is a L2 color consistency loss between the rendered color $\mathbf{c}$ and the observed color $\hat{\mathbf{c}}$ for all pixel rays: $E_{\text{color}} = \sum_{\mathbf{r}} \|\mathbf{c_r} - \mathbf{c_r}\|_2$.
- The *normal consistency term* E_{normal} measures the rendered normal $\hat{\mathbf{n}}$ and a predicted normal $\hat{\mathbf{n}}$: $E_{\text{normal}} = \sum_{\mathbf{r}} \|1 - \hat{\mathbf{n}}_{\mathbf{r}}^{\mathrm{T}} \mathbf{n_r}\|_1 + \|\hat{\mathbf{n}}_{\mathbf{r}} - \mathbf{n_r}\|_1$. Our monocular predicted normal $\hat{\mathbf{n}}$ is obtained using a pretrained Omnidata model [13].
- The *mask consistency term* E_{mask} measures the discrepancy between the rendered mask $\mathbf{m_r}$ and the observed mask $\hat{\mathbf{m}}_{\mathbf{r}}$, in terms of binary cross-entropy (BCE): $L_{\text{mask}} = \sum_{\mathbf{r}} \text{BCE}(\mathbf{m_r}, \hat{\mathbf{m}}_{\mathbf{r}})$.
- Finally, inspired by NeuS [60], we incorporate an *Eikonal regularization* to ensure the neural field is a valid signed distance field: $L_{\text{eikonal}} = \sum_{\mathbf{x}} (\|\nabla_{\mathbf{x}} S\|_2 - 1)^2$,

Visibility optimization: Ambient occlusion and self-cast shadows pose challenges to the accuracy of inverse rendering, as it's difficult to separate them from albedo when optimizing photometric loss. However, with an estimated geometry, we can already obtain a strong visibility cue. Consequently, we utilize ambient occlusion mapping to prebake the visibility map onto the object surfaces obtained from the previous stage. We then minimize the visibility consistency term to ensure the rendered visibility $v_{\mathbf{r}}$ from MLP V aligns with the derived visibility $\hat{v}_r$: $\min_V \sum_{\mathbf{r}} \text{BCE}(v_{\mathbf{r}}, \hat{v}_{\mathbf{r}})$.

Material and illumination optimization: In the final stage, given the obtained surface geometry and the visibility network, we jointly optimize the environmental light and the PBR material network. The overall objective is as follows:

$$\min_{M,\boldsymbol{\omega},\boldsymbol{\phi}} E_{\text{color}} + \lambda_4 E_{\text{sparse}} + \lambda_5 E_{\text{smooth}} + \lambda_6 E_{\text{metal}}, \tag{5}$$

where $\lambda_4 = 0.01$, $\lambda_5 = 0.1$, $\lambda_6 = 0.01$. The four terms are defined as follows:

- The *color consistency term* E_{color} minimizes the discrepancy between the rendered color and observed color, akin to the geometry optimization stage. However, we use PBR-shaded color in place of the color queried from the auxiliary radiance field.
- The *sparse regularization* E_{sparse} constrains the latent code $\boldsymbol{\rho}$ of the material network to closely align with a constant target vector $\boldsymbol{\rho}'$: $E_{\text{latent}} = \text{KL}(\boldsymbol{\rho} || \boldsymbol{\rho}')$. We set $\boldsymbol{\rho}' = 0.05$.
- The *smooth regularization* E_{smooth} force the BRDF decoder to yield similar value for close latent code $\boldsymbol{z}$: $E_{smooth} = \|D(\boldsymbol{z}) - D(\boldsymbol{z} + \boldsymbol{dz})\|_1$, where $\boldsymbol{dz}$ is randomly sampled from normal distribution $N(0; 0.01)$.
- Lastly, inspired by the fact that most common objects are either metallic or not, we introduce a *metallic regularization* L_{metal} to encourage the predicted metallic value to be close to either 0 or 1: $L_{\text{metal}} = \sum_{\mathbf{r}} m_{\mathbf{r}}(1 - m_{\mathbf{r}})$.

4 Experiment

In this section, we evaluate the effectiveness of our model on synthetic and real-world datasets, analyze its characteristics, and showcase its applications.

4.1 Experiment setups

Data: Since existing multi-view datasets do not contain duplicate objects, we collect ***Dup***, a novel inverse rendering dataset featuring various duplicate objects. *Dup* consists of 13 synthetic and 6

	Albedo	Roughness	Env Light	Mesh	Rotation	Translation
Num	**PSNR ↑**	**MSE ↓**	**MSE ↓**	**CD ↓**	**degree ↓**	**length ↓**
6	18.48	0.13	0.116	0.025	2.515	0.070
8	18.26	0.09	0.172	0.017	0.600	0.003
10	18.99	0.09	0.204	0.029	0.186	0.003
20	19.22	0.09	0.063	0.021	0.299	0.005
30	19.69	0.07	0.119	0.020	0.448	0.006
50	18.56	0.08	0.153	0.038	0.464	0.008
60	16.45	0.14	0.150	0.091	51.512	0.644

	Albedo	Roughness	Metallic	Relighting
Model	**PSNR ↑**	**MSE ↓**	**MSE ↓**	**PSNR ↑**
full model	**17.27**	0.30	0.02	**19.95**
w/o clean seg	16.68	0.23	0.13	19.12
w/o metal loss	17.09	**0.19**	0.08	19.02
w/o latent smooth	17.00	0.23	**0.02**	19.71
w/o normal	16.51	0.42	0.97	18.81
w/o eik loss	16.79	0.51	0.97	19.31
w/o mask	15.14	0.29	0.02	18.11

Table 2: **(Left) Performance vs. number of duplicates for "color box" dataset.** We highlight the best, second and third values. **(Right) Ablation study for the contribution of each loss term.**

	Rendering		
	PSNR ↑	**SSIM ↑**	**LPIPS ↓**
PhySG*	**20.624**	0.641	0.263
Nv-DiffRec*	18.818	0.569	0.282
InvRender*	20.665	0.639	0.262
Ours	20.326	**0.660**	**0.192**

	Albedo	Roughness	Relighting	Env Light	Geometry
	PSNR ↑	**MSE ↓**	**PSNR ↑**	**MSE ↓**	**CD ↓**
M-S	20.229	0.096	21.328	**0.045**	0.010
S-M	**23.448**	**0.050**	**24.254**	0.052	**0.007**

Table 3: **(Left) Single-view inverse rendering on real-world data**. $*$ indicates that the baselines are trained on multi-view observations. **(Right) Multi-view single-object (M-S) vs. single-view multi-object (S-M).**

real-world scenes, each comprising 5-10 duplicate objects such as cans, bottles, fire hydrants, etc. For synthetic data, we acquire 3D assets from PolyHaven[†] and utilize Blender Cycles for physics-based rendering. We generate 10-300 images per scene. As for the real-world data, we place the objects in different environments and capture 10-15 images using a mobile phone. The data allows for a comprehensive evaluation of the benefits of including duplicate objects in the scene for inverse rendering. We refer the readers to the supp. materials for more details.

Metrics: Following prior art [74; 69], we employ Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and LPIPS [71] to assess the quality of rendered and relit images. For materials, we utilize PSNR to evaluate albedo, and mean-squared error (MSE) to quantify roughness and environmental lighting. And following[60; 65; 47], we leverage the Chamfer Distance to measure the distance between our estimated geometry and the ground truth.

Baselines: We compare against three state-of-the-art multi-view inverse rendering approaches: Physg [69], InvRender [74], and NVdiffrec [47]. Physg and InvRender employ implicit representations to describe geometric and material properties, while NVdiffrec utilizes a differentiable mesh representation along with UV textures. Additionally, we enhance Physg by equipping it with a spatially-varying roughness.

Implementation details: We use the Adam optimizer with an initial learning rate 1e-4. All experiments are conducted on a single Nvidia A40 GPU. The first stage takes 20 hours, and the 2nd and 3rd stage takes about 2 hours. Please refer to the supp. material for more details.

4.2 Experimental results

Single-view inverse rendering: We first evaluate our approach on the single-image multi-object setup. Since the baselines are not designed for this particular setup, we randomly select another 9 views, resulting in a total of 10 multi-view images, to train them. As shown in Tab. 1(right), we are able to leverage duplicate objects to constrain the underlying geometry, achieving comparable performance to the multi-view baselines. The variations in lighting conditions across instances also aid us in better disentangling the effects of lighting from the materials.

Multi-view inverse rendering: Our method can be naturally extended to the multi-view setup, allowing us to validate its effectiveness in traditional inverse rendering scenarios. We utilize synthetic data to verify its performance. For each scene, we train our model and the baselines using 100 different views and evaluate the quality of the reconstruction results. Similar to previous work, we

[†]https://polyhaven.com/models

Figure 6: **The surface normal and rendering result on real-world cola image.** Our model has the smoothest surface normal compared with other baselines.

assume the ground truth poses are provided. As shown in Tab. 1(left), we outperform the baselines on all aspects. We conjecture that this improvement stems from our approach explicitly incorporating a material- and geometry-sharing mechanism during the modeling process. As a result, we have access to a significantly larger number of "effective views" during training compared to the baselines. We show some qualitative results in Fig. 4(left).

Real-world single-view inverse rendering: Up to this point, we have showcased the effectiveness of our approach in various setups using synthetic data. Next, we evaluate our model on real-world data. Due to the challenge of obtaining highly accurate ground truth for materials and geometry, we focus our comparison solely on the rendering results. As indicated in Tab. 3(Left), our method achieves comparable performance to the multi-view baselines, even when trained using only a single view. We visualize some results in Fig. 1, Fig. 6 and Fig. 7.

Ablation study for each loss term: As shown in Tab. 2(right), since the metallicness of natural materials are usually binary, incorporating the metallic loss can properly regularize the underlying metallic component and prevent overfitting; Eikinol loss and mask loss help us better constrain the surface and the boundary of the objects, making them more accurate and smooth. Removing either term will significantly affect the reconstructed geometry and hence affect the relighting performance; The pre-trained surface normal provides a strong geomeotry prior, allowing us to reduce the ambiguity of sparse view inverse rendering. Removing it degrades the performance on all aspects.

4.3 Analysis

Importance of the number of duplicate objects: Since our model utilize duplicate objects as a prior for 3D reasoning, one natural question to ask is: how many duplicate objects do we need? To investigate this, we render 9 synthetic scenes with 4$\sim$60 duplicates with the same object. We train our model under each setup and report the performance in Tab. 2. As expected, increasing the number of duplicates from 6 to 30 improves the accuracy of both material and geometry, since it provides more constraints on the shared object intrinsics. However, the performance decrease after 30 instances due to the accumulated 6 DoF pose error brought by heavier occlusion, the limited capacity of the visibility field, and limited image resolution.

The influence of different geometry representation: To demonstrate that our method does not rely on a specific neural network architecture, we replace the vanilla MLP using Fourier position encoding to triplane representation from PET-NeuS[61] and hash position encoding from neuralangelo[35] respectively. The result (please refer to supplementary material) shows that the triplane (28.0MB) can recover better geometry and texture than our original model (15.8MB) because the explicit planes provide higher resolution features and can better capture local detail[4]. However, the model of hash positional encoding (1.37GB) and produces high frequency noisy in geometry, indicating that it overfits the training view.

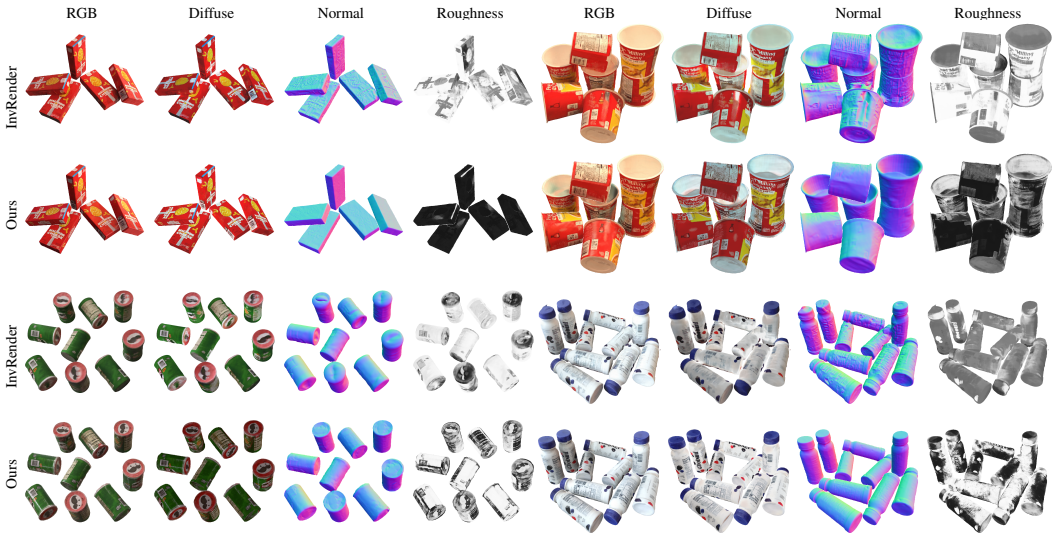


Figure 7: **Qualitative comparison on real-world data.** InvRender [74] takes as input 10 images, while we only consider one single view. Yet our approach is able to recover the underlying geometry and materials more effectively.

Multi-view single object (M-S) vs. single-view multi-objects (S-M): Is observing an object from multiple views equivalent to observing multiple objects from a single view? Which scenario provides more informative data? To address this question, we first construct a scene containing 10 duplicate objects. Then, we place the same object into the same scene and capture 10 multi-view images. We train our model under both setups. Remarkably, the single-view setting outperforms the multi-view setting in all aspects (see Tab. 3(right)). We conjecture this discrepancy arises from the fact that different instances of the object experience environmental lighting from various angles. Consequently, we are better able to disentangle the lighting effects from the material properties in the single-view setup. Fig. 5(right) shows some qualitative results.

Applications: Our approach supports various scene edits. Once we recover the material and geometry of the objects, as well as the illumination of the scene, we can faithfully relight existing objects, edit their materials, and seamlessly insert new objects into the environment as if they were originally present during the image capturing process (see Fig. 1).

Limitations: One major limitation of our approach is that we require the instances in each image to be nearly identical. Our method struggles when there are substantial deformations between different objects, as we adopt a geometry/material-sharing strategy. One potential way to address this is to loosen the sharing constraints and model instance-wise variations. We explore this possibility in the supplementary material. Furthermore, our model requires accurate instance segmentation masks. Additionally, our approach currently requires decent 6 DoF poses as input and keeps the poses fixed. We found that jointly optimizing the 6 DoF pose and SDF field with BARF [37] will cause the pose error to increase. This is because recovering camera pose from sparse views and changing light is a difficult non-convex problem and prone to local minima. We believe the more recent works, like CamP[48] and SPARF[58] could potentially further refine our estimations. Moreover, our model cannot effectively model/constrain the geometry of unseen regions similar to existing neural fields methods.

5 Conclusion

We introduce a novel inverse rendering approach for single images with duplicate objects. We exploit the repetitive structure to estimate the 6 DoF poses of the objects, and incorporate a geometry and material-sharing mechanism to enhance the performance of inverse rendering. Experiments show that our method outperforms baselines, achieving highly detailed and precise reconstructions.

References

[1] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2014.

[2] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. vis. syst*, 1978.

[3] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Siggraph*, 2012.

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021.

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023.

[6] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *NeurIPS*, 2016.

[7] Wenzheng Chen, Joey Litalien, Jun Gao, Zian Wang, Clement Fuji Tsang, Sameh Khamis, Or Litany, and Sanja Fidler. Dib-r++: learning to predict lighting and material with a hybrid differentiable renderer. *NeurIPS*, 2021.

[8] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv*, 2021.

[9] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 465–474. 2023.

[10] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG)*, 33(6):1–12, 2014.

[11] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3583–3592, 2016.

[12] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

[13] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021.

[14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.

[15] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 2006.

[16] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12814–12823, 2021.

[17] William T Freeman. The generic viewpoint assumption in a framework for visual perception. *Nature*, 1994.

[18] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV*, 2009.

[19] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022.

[22] Daniel Hauagge, Scott Wehrwein, Kavita Bala, and Noah Snavely. Photometric ambient occlusion. In *CVPR*, 2013.

[23] Berthold KP Horn. Determining lightness from an image. *Computer graphics and image processing*, 1974.

[24] Berthold KP Horn. Obtaining shape from shading information. *The psychology of computer vision*, 1975.

[25] Bo Hu, Christopher Brown, and Randal Nelson. Multiple-view 3-d reconstruction using a mirror. 2005.

[26] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015.

[27] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-nerf: Using a neural radiance field to grasp transparent objects. *arXiv*, 2021.

[28] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986.

[29] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 2013.

[30] Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE transactions on visualization and computer graphics*, 2012.

[31] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 1971.

[32] Hendrik PA Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *TOG*, 2003.

[33] Yikai Li, Jiayuan Mao, Xiuming Zhang, Bill Freeman, Josh Tenenbaum, Noah Snavely, and Jiajun Wu. Multi-plane program induction with 3d box priors. *NeurIPS*, 2020.

[34] Yikai Li, Jiayuan Mao, Xiuming Zhang, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Perspective plane program induction from a single image. In *CVPR*, 2020.

[35] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[36] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6123–6133, 2021.

[37] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[38] Yen-Chen Lin, Pete Florence, Andy Zeng, Jonathan T Barron, Yilun Du, Wei-Chiu Ma, Anthony Simeonov, Alberto Rodriguez Garcia, and Phillip Isola. Mira: Mental imagery for robotic affordances. In *CoRL*, 2023.

[39] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *ICCV*, 2021.

[40] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981.

[41] Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba. Single image intrinsic decomposition without a single intrinsic image. In *ECCV*, 2018.

[42] Wei-Chiu Ma, Ignacio Tartavull, Ioan Andrei Bârsan, Shenlong Wang, Min Bai, Gellert Mattyus, Namdar Homayounfar, Shrinidhi Kowshika Lakshmikanth, Andrei Pokrovsky, and Raquel Urtasun. Exploiting sparse semantic hd maps for self-driving vehicle localization. In *IROS*, 2019.

[43] Stephen Robert Marschner. *Inverse rendering for computer graphics*. Cornell University, 1998.

[44] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014.

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020.

[46] Chaitanya Mitash, Bowen Wen, Kostas Bekris, and Abdeslam Boularias. Scene-level pose estimation for multiple instances of densely packed objects. In *Conference on Robot Learning*, pages 1133–1145. PMLR, 2020.

[47] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022.

[48] Philipp; Mildenhall Ben; Barron Jonathan T.; Martin-Brualla Ricardo Park, Keunhong; Henzler. Camp: Camera preconditioning for neural radiance fields. *ACM Trans. Graph.*, 2023.

[49] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[50] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020.

[51] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[52] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[53] Yoichi Sato, Mark D Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH*, 1997.

[54] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[57] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[58] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023.

[59] Jingkang Wang, Sivabalan Manivasagam, Yun Chen, Ze Yang, Ioan Andrei Bârsan, Anqi Joyce Yang, Wei-Chiu Ma, and Raquel Urtasun. Cadsim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation. In *6th Annual Conference on Robot Learning*.

[60] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[61] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Pet-neus: Positional encoding triplanes for neural surfaces. 2023.

[62] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representation for inverse rendering of urban scenes. *arXiv*, 2023.

[63] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. *NeurIPS*, 2021.

[64] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. *CVPR*, 2023.
[65] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020.
[66] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.
[67] Albert J Zhai and Shenlong Wang. Peanut: Predicting and navigating to unseen targets. *arXiv*, 2022.
[68] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *CVPR*, 2022.
[69] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.
[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
[71] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *IEEE*, 2018.
[72] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021.
[73] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv*, 2020.
[74] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022.
[75] Yunzhi Zhang, Shangzhe Wu, Noah Snavely, and Jiajun Wu. Seeing a rose in five thousand ways. *arXiv*, 2022.
[76] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*, 2015.
[77] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*. IEEE, 2011.

Structure *from* Duplicates: Neural Inverse Graphics from a Pile of Objects —Supplementary Material—

Tianhang Cheng[1] **Wei-Chiu Ma**[2] **Kaiyu Guan**[1] **Antonio Torralba**[2] **Shenlong Wang**[1]
[1]University of Illinois Urbana-Champaign [2]Massachusetts Institute of Technology
{tcheng12, kaiyug, shenlong}@illinois.edu
{weichium, torralba}@mit.edu

1 Novelty and contributions

How to make inverse graphics/3D reconstruction more robust and work under more extreme scenarios is a challenging and longstanding problem in computer vision. In this work, we take a step forward by exploring the potential of performing structure from motion and recovering object intrinsics and environmental extrinsics from a single image without pre-trained priors. Specifically, we focus on the scenarios where there are multiple (near-)identical objects within the scene. By carefully formulating a duality between multiple copies of an object in a single image and multiple views of a single object, we are able to resolve the ambiguities in 3D and effectively recover the properties of interest.

Over the years, the community has been actively investigating how to harness multi-view information from videos or sparse, extreme-view images, and push forward the frontier of 3D reconstruction and inverse graphics. Our work can be seen as an attempt in such a stride. To our knowledge, this is the first effort to conduct structure from motion from a single image. Furthermore, based on our preliminary experiments, our approach also has the potential to deal with slight variations, as shown in Sec. 6.1. Specifically, we test our approach on the crane image that [14] provided, where each instance is slightly different. By augmenting our geometry backbone with a instance-specific deformation field, we are able to reconstruct reasonable poses and recover sensible shape and material. We hope it can shed light on future research along similar directions, such as handling articulate objects or objects with large deformation.

2 Additional details of proposed pipeline

2.1 The In-plane augmentation of pose estimation for the single image

Vanilla COLMAP often fails to reconstruct the duplicated instances of a single image. The reason for the failure can probably be explained by Fig. 7: when the lighting effect and occlusion are disregarded, a single-view image containing duplicated objects can be treated as observing a single object from multiple viewpoints using a multi-view camera setup. This conversion results in numerous accidental and non-uniform multi-view images with varying orientations that may not align uniformly with the upward axis. Therefore, vanilla COLMAP cannot find stable matching points from these extreme camera distribution. By contrast, with the help of in-plane rotation augmentation, we can greatly improve the performance (e.g., from failure to success). This demonstrate that the incorporation of in-plane rotation augmentation becomes essential to facilitate robust point matching, as demonstrated in Figure 8. However, there are still limitations of this pose estimation module. First, for low-texture objects or low-resolution scenes where pixels are not distinct, our method may still suffer, like the 60 boxes in Fig. 3. Second, the time complexity of the algorithm scales with the square of the number of objects, so it will be slow when there are many instances. We believe that combining some transformation-invariant feature extractor (such as GIFT[4]) can solve this problem faster.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

For synthetic dataset we use 3200×3200 resolution for pose estimation and resize to 800×800 resolution for training. For real-world dataset we use the 3072×3072 resolution for pose estimation and resize to 800×800 resolution for training.

For n instances in the scene, we first find relative rotation angles for all $\frac{n(n-1)}{2}$ instance pairs. For each pair, we fix one of them and gradually rotate the other, increasing the angle by 4 degrees each time. We record the number of matching points n at different rotation angles θ, i.e.$n(\theta)$, and mark those rotation angles with a greater number of matching points than the average as "good angles", i.e. $\boldsymbol{\theta_{good}} \in \left\{\theta | \theta > \sum_{i=0}^{N} \frac{n(\theta_i)}{N}\right\}$, where $N = \frac{360}{d\theta} = 90$.

However, the relative rotation angles $\boldsymbol{\theta}_{good}$ of one pair is usually conflict with other pairs. To resolve this issue, We use a customized procedure to transform multiple relative pairwise poses into an initialized global rotation angle for each instance. Specifically, we use Scipy's BFGS optimizer to find a global rotation angle that minimizes the loss. Transferring relative rotation angles into global ones ensures the integration into the standard BA pipeline. After we correct each instance with the global rotation, we use Superglue[7] and Superpoint[1] to extract and match key points. Then we apply a standard bundle adjustment algorithm to solve the 6Dof pose of each instances. The algorithm is as follows:

Require: $\boldsymbol{\theta}^i_{good}$, where $i \in [0, \frac{n(n-1)}{2} - 1]$ (good relative rotation angles of each pair)
  $L \leftarrow \inf$ (Initialize loss)
  $\boldsymbol{\theta}^j_{global} \leftarrow U[0, 2\pi]$, where $j \in [0, n-1]$ (uniform initialized global rotation angle for each pair)
  while L not converge **do**
    $\theta^{p,q}_{rel} \leftarrow |\theta^p_{global} - \theta^q_{global}| \bmod 2\pi$, where $p, q \in [0, n-1]$
    $L_{p,q} \leftarrow \min\left(\left|\theta^{p,q}_{rel} - \boldsymbol{\theta}^p_{good}\right|\right) + \min\left(\left|\theta^{p,q}_{rel} - \boldsymbol{\theta}^q_{good}\right|\right)$
    $L \leftarrow \sum_{p,q} L_{p,q}$
  end while
  Return θ^j_{global} (resulting global rotations)

2.2 Training

We train 100000 iterations for geometry stage with 1e-4 learning rate. The visibility stage takes 3000 iterations with 2e-5 learning rate. The material stage takes 10000 iterations with 2e-4 learning rate. Please check our code for more detail. This supplementary material provides further details on our method and presents an extended set of experimental results.

3 Comparison with recent works

3.1 "Seeing a Rose in Five Thousand Ways"[14]

The setup of the two papers are similar, but they are different in the following aspects:

- Assumptions: While [14] is able to model the variations among the instances, they impose other strong assumptions such as knowing the camera distribution in advance. The strong camera assumption allows them to sidestep the pose estimation step (i.e., SfM) and focus on modeling the variation. In contrast, we assume no knowledge about the poses and attempt to solve for the full inverse rendering pipeline from the beginning. We thus resort to the (near-)identical instances to recover the exact 6 DoF poses.
- Approaches: [14] tackle the task through generative modeling. Since they need to train a generative model per scene, their approach is very data-hungry. In contrast, our approach mainly exploits multi-view geometry to recover the underlying intrinsic and extrinsic properties. By explicitly baking the constraints into the modeling procedure, our approach becomes much more data-efficient. To validate our conjecture, we train [14] on three randomly selected scenes from our dataset, each of which has 10 identical instances. As shown in the pdf, the generative model failed to recover either of them. For comparison, we also test our approach on the crane image that [14] provided (the only publicly available data), where each instance is slightly different.

By augmenting our geomtry backbone with a instance-specific deformation field, we are able to reconstruct resonable poses and recover sensible shape and material.

- Extrinsics: [14] assume a simple phong shading model and assume a dominant directional light, whereas we parameterize our materials with PBR materials and the lighting with enironmental map, allowing us to model complex real world scenarios more effectively. Finally, it is unclear how to extend [14] to multi-view setup. In contrast, our method is naturally compatible with multi-view observations.

3.2 "Modeling Indirect Illumination for Inverse Rendering"[13]

It is important to note that our model does not rely on a specific geometric model and it can be replaced with more advanced neural representations. In this paper, we build our inverse rendering pipeline from Invrender[13]. But there are some differences in execution:

- Backbone: In our approach, we utilize NeuS[9] as our neural surface model instead of IDR[11]; and for the visibility field, we opt for Siren[8] instead of ReLU.
- Metallic: Our model can reconstruct metallic object besides pure diffuse object.
- MLP distillation: We distill the geometry MLP into a smaller one for fast classification.
- Self-occlusion and Inter-occlusion: Since we have multiple instances in our setup, it is essential to model both inter-object self-casted shadows and inter-object occlusions. Our model goes beyond simple object-centric representation.

4 Additional details of the dataset

4.1 Synthetic dataset

Our new dataset **Dup** consists of 13 synthetic scenes. "Apple", "Medicine box", "Can" and "Driller" consists of 100 training views and 200 testing views for multi-view experiments. "Color box", "Cash machine", "Cleaner", "Clock", "Coffee machine", "Fire extinguisher", "Wood guitar", "Warning sign" and "Food tin" consists of 7-10 multi-view images to test our model on single-view and baselines for multi-view. The resolution of the raw images are 3200×3200.

4.2 Real-world dataset

We scatter object on the table and use mobile phone to gather several scenes, named "Toy airplane", "Cake box", "Cheese box", "Cola", "Potato chips" and "Yogurt". The number of objects in the scene ranges from 5 to 10. The resolution of the raw images are 3072×3072.

5 Additional details or analysis of experiments

5.1 The influence of different number of instances

We conducted experiment on the image of "box". The training image are visualized in Fig. 3. Please refer to the paper for quantitative result. The results show that there is a "sweet spot" for the box dataset that achieves the best trade-off between image resolution and number of views. We believe that this "sweet spot" exists for other data sets as well. For objects with simple textures and complex geometric shapes, a smaller number of instances should be processed, otherwise there will be large errors in pose estimation. On the contrary, for objects with simple shapes and complex textures, the number of instances can be increased to reduce the ambiguity of material recovery.

5.2 The influence of different neural representation

We conducted experiment on the image of "Cash machine". The triplane representation is adapted from PET-NeuS[10] and hash representation is adapted from Neuralangelo[3].

- Triplane: The triplane representation is consist of three planes, each plane is of $512\times512\times32$ resolution. The triplane will passed to a self-attention convolution module to produce features with

different frequency bands. Then a 3D point will sample space features from these triplane and decode into SDF value and color with a small MLP.

- Hash position encoding: The point is encoded by hash function with the default setting in Neuralangelo[3]. Then the hash feature is passed to MLP layers and decode to SDF and color values. Here We use analytical gradient instead of numerical gradient in the default setting because we found the former will produce less high frequency noise in our tasks.

We show the visual image in Fig.11 and it demonstrate that the triplane has the best performance. Please refer to the paper for the quantitative result.

5.3 Multi-view single-object vs single-view multi-object

To ensure a fair comparison, we maintain the multi-view single-object (M-S) setting, while adjusting the single-view multi-object (S-M) setting to have a similar number of non-empty pixels. The training images in the M-S setting (see Fig. 12, first two rows) contain 244,335 non-empty pixels, whereas the training image in the S-M setting (see Fig. 12, bottom row) consists of 263,910 non-empty pixels. We provide a qualitative comparison in Fig. 13 and present the corresponding quantitative analysis in Table 4.

5.4 The contribution of each loss term

We experiment the contribution of each loss term on the image of "fire-extinguisher". We train 50000 iterations for geometry reconstruction, 2000 for visibility fields and 13000 iterations for material recover (less iterations than the main paper) for faster verification. The results show that our full model has the best albedo and relighting results. Its rendering performance is only surpassed by the ablation model without the metallic binary loss term.

The result table indicates that our loss function is sensible. First, The metallic of natural materials is mostly binary. However the "w/o metal loss" does not limit the metallic proprieties so it may has a stronger fitting ability, but there is also a risk of over-fitting. Second, the same as [13], the latent smooth loss term in texture-MLP can reduce the possibility of over fitting because the materials of real world objects are limited. Third, the eik loss and mask loss proposed by NeuS[9] can constrain the surface and boundary of the geometry, making the surface of the object more accurate and smooth. Fourth, the pre-trained surface normal can provide a strong geomeotry prior, reduce the ambiguity of sparse view inverse rendering.

5.5 The influence of noisy instance segmentation image

In the main paper, we use ground-truth segmentation mask for synthetic dataset and use an use interactive pre-trained segmentation models for real-world dataset. To assess the influence of segmentation map noise, we conducted a comparative analysis of the model's performance on the "fire extinguisher" image. We compared results obtained using ground truth segmentation maps(left image of Fig. 9) with those generated by pre-trained models through segmentation maps generated with 2 to 4 clicks per instance, without subsequent post-processing (right image of Fig. 9).

The result in second row of Fig. 10 shows that the our model has certain robustness to the noise of segmentation.

6 Additional experiments

6.1 Single-view reconstruction on "paper crane" dataset from Zhang *et al*.[14]

To demonstrate the difference between our method and [14], we test their on our single-image dataset and versa versa. We run their method on the image of "airplane", "cake box" and "cola", which contains 6, 7, 7 instances respectively. The result in Fig. 4 shows that their methods fail to reconstruct a good geometry and texture and generate over-smooth output. This is because they suppose a pose prior rather than accurate camera pose and don't support zero-variant scenes, thus cannot accurately capture complex materials and geometry. In addition, it will produce large errors when objects have mutual occlusion.

We also run our method on crane dataset proposed by [14] in their Github repository. We apply SfM with in-plane rotation, which gives us better registration (Fig. 8). To model the geometry variances of instances, we also adopt an deformable-aware pipeline by inserting a geometry latent into the middle of geometry MLP. Our method can reconstruct the 6DoF pose, geometry and PBR texture of these cranes. We visualize the recoverd texture in Fig. 1 and the surface normal of different instances in Fig 2. The difference from [14] is that our latent vector can correspond to the instances in the original picture one by one, but what they learn is a geometry distribution and need "latent inversion"(similar to GAN inversion) to calculate the latent for a specific instance.

However, as shown in 1, our model does not recover a fully consistent texture at corresponding points across different instances. This is because we model the variance of instances by implicit instance vector rather than the explicit displacement field. The displacement field based neural representation, such as Nvdiffrec-MC[2] and D-NeRF[6], can achieve strict consistency between different time or instances. Nevertheless, the displacement field usually has a greater number of parameters than implicit instance vector, which may lead to overfitting. We leave this for future study.

6.2 The influence of inaccurate pose

Since our method adopts a stage-wise inference procedure, errors in pose estimation can propagate and impact the quality of the inverse rendering reconstructions. To verify the extent of this impact, we conduct an oracle experiment where we replace the estimated 6 DoF object poses with ground truth. The results, presented in Table 5, demonstrate that our model achieves performance similar to that of oracles, primarily due to its precise estimation of small object poses. However, when faced with samples involving significant pose estimation errors for objects, the performance of the oracle outperforms our model, as shown in Table 6.

We jointly optimize the geometry and camera pose, achieving similar results(**??**) to the original pipeline. Optimizing camera poses under sparse-views and varying lighting conditions presents a notably ambiguous challenge. Despite the alterations in the pose of each instance, the average rotation and translation errors of the final 6Dof pose for the model have shown no reduction.

7 Additional visualizations

7.1 Synthetic multi-view experiment

The qualitative results are shown in Fig. 15, 16, 17, 18. The quantitative results are shown in table. 1.

7.2 Synthetic single-view experiment

The qualitative results are shown in Fig. 19, 20, 21, 22, 23, 24, 25, 26, 27. The quantitative results are shown in table. 2. The recovered bounding boxes are shown in Fig 6.

7.3 Real-world single-view experiment

The qualitative results are shown in Fig. 28, 29, 30, 31, 32, 33. The quantitative results are shown in table. 3. The recovered bounding boxes are shown in Fig 5.

References

[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[2] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022.

[3] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022.
[6] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.
[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.
[8] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.
[9] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
[10] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Pet-neus: Positional encoding triplanes for neural surfaces. 2023.
[11] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020.
[12] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.
[13] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022.
[14] Yunzhi Zhang, Shangzhe Wu, Noah Snavely, and Jiajun Wu. Seeing a rose in five thousand ways. *arXiv*, 2022.

	Rendering			Albedo			Roughness	Relighting			Env Light	Geometry
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑	LIPIPS ↓	MSE ↓	CD ↓
PhySG	25.985	0.809	0.199	16.233	0.620	0.363	0.087	21.323	0.748	0.270	0.054	0.024
Nv-DiffRec	**27.840**	**0.886**	**0.089**	16.123	0.533	0.412	0.116	17.418	0.459	0.388	0.168	0.268
InvRender	26.452	0.809	0.206	16.984	0.637	0.370	0.084	22.224	0.757	0.267	0.067	0.024
Ours	23.213	0.781	0.222	**21.961**	**0.655**	**0.260**	**0.026**	**25.486**	**0.830**	**0.183**	**0.029**	**0.011**

Table 1: **Synthetic multi-view result**. All models are trained with 100 multi-view images. Our model has the best texture recovery performance. Nv-DiffRec reaches the best rendering result in training image, but has the worst texture recovery due to overfitting.

	Rendering			Albedo			Roughness	Relighting			Env Light	Geometry
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑	LIPIPS ↓	MSE ↓	CD ↓
PhySG	20.047	0.584	0.323	14.977	0.460	0.405	0.255	18.504	0.554	0.356	0.082	0.033
Nv-DiffRec	20.513	0.638	0.248	14.021	0.416	0.431	0.165	17.214	0.427	0.391	0.067	0.050
InvRender	19.489	0.557	0.351	14.724	0.438	0.431	0.247	17.998	0.527	0.381	0.082	**0.033**
Ours	**24.307**	**0.752**	**0.152**	**17.629**	**0.594**	**0.229**	**0.062**	**21.374**	**0.695**	**0.189**	**0.052**	0.034

Table 2: **Synthetic single view result**. The baseline models are trained with 10 multiview images and our model is trained in single image. Our model has the best texture recovery performance.

	Rendering		
	PSNR ↑	SSIM ↑	LPIPS ↓
PhySG	**20.624**	0.641	0.263
Nv-DiffRec	18.818	0.569	0.282
InvRender	20.665	0.639	0.262
Ours	20.326	**0.660**	**0.192**

Table 3: **Experiment result on real-world single-view dataset**. Our model has a comparable quality even with a single-view.

	Rendering			Albedo			Roughness	Relighting			Env Light	Geometry
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑	LIPIPS ↓	MSE ↓	CD ↓
M-S	21.347	0.591	0.511	20.229	0.594	0.514	0.096	21.328	0.600	0.494	**0.045**	0.010
S-M (ours)	**23.994**	**0.657**	**0.375**	**23.448**	**0.666**	**0.365**	**0.050**	**24.254**	**0.667**	**0.359**	0.0519954	**0.007**

Table 4: **Quantitative results of M-S setting and S-M setting**. When the #instance $\times$ #views is a constant and with good pose estimation, our model has better performance than the traditional multi-view single object setting.

	Rendering			Albedo			Roughness	Relighting			Env Light	Geometry
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑	LIPIPS ↓	MSE ↓	CD ↓
Oracle	**24.570**	**0.782**	**0.128**	**17.858**	**0.597**	**0.223**	0.105	21.132	**0.709**	**0.174**	0.063	**0.031**
Ours	24.307	0.752	0.152	17.629	0.594	0.229	**0.062**	**21.374**	0.695	0.189	**0.051**	0.034

Table 5: **Ablation result for ground-truth 6Dof pose (oracle model)**.

				Rendering			Albedo	Roughness	Env Light	Geometry			
Sample	Model	dR (°) ↓	dT (°) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	MSE ↓	MSE ↓	CD ↓	Precision ↑	Recall ↑	F1 ↑
Cleaner	Oracle	-	-	**26.600**	**0.903**	**0.055**	**22.210**	**0.033**	**0.028**	**0.008**	**1.000**	**0.984**	**0.992**
	Ours	1.344	3.067	25.059	0.844	0.105	21.647	0.042	0.033	0.012	0.986	0.967	0.976
Gitar	Oracle	-	-	**25.966**	**0.809**	**0.132**	**20.608**	**0.057**	0.049	**0.018**	**0.954**	**0.923**	**0.938**
	Ours	1.076	1.653	24.599	0.736	0.172	19.516	0.063	**0.034**	0.046	0.984	0.513	0.675
Coffee	Oracle	-	-	22.266	**0.747**	**0.141**	**13.419**	0.286	0.128	0.040	0.649	0.587	0.617
	Ours	0.589	1.015	**22.477**	0.711	0.166	13.347	**0.064**	**0.057**	**0.039**	**0.732**	**0.610**	**0.665**

Table 6: **The influence of inaccurate pose**. Oracle model perform much better than our model when we have a large pose estimation error.

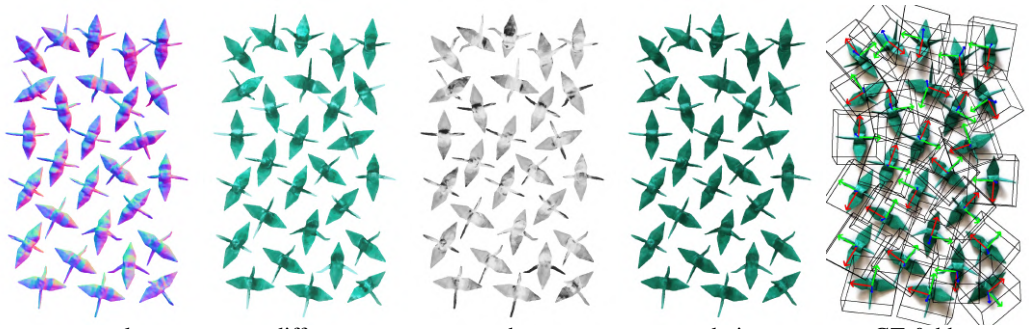


Figure 1: **Qualitative result of our method on the *Crane* image.** We manually flipped the incorrect global rotation for some cranes before extract the final matching points to reduce the impact of symmetry. The result shows that our model can recover the geometry, texture, and bounding box from a single image, even with objects with variations in shape and appearance. However, the PBR texture is not fully consistent across different instances since we use instant vector rather than learning a displacement field. We believe that mesh based representation, like Nvdiffrec-MC [2] can achieve better consistency since it modeling the scene by displacement field.

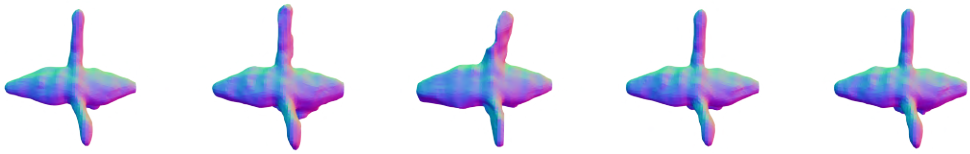


Figure 2: **Different instances of our method on paper-crane dataset.** We randomly visualize the surface normal of 5 instances.

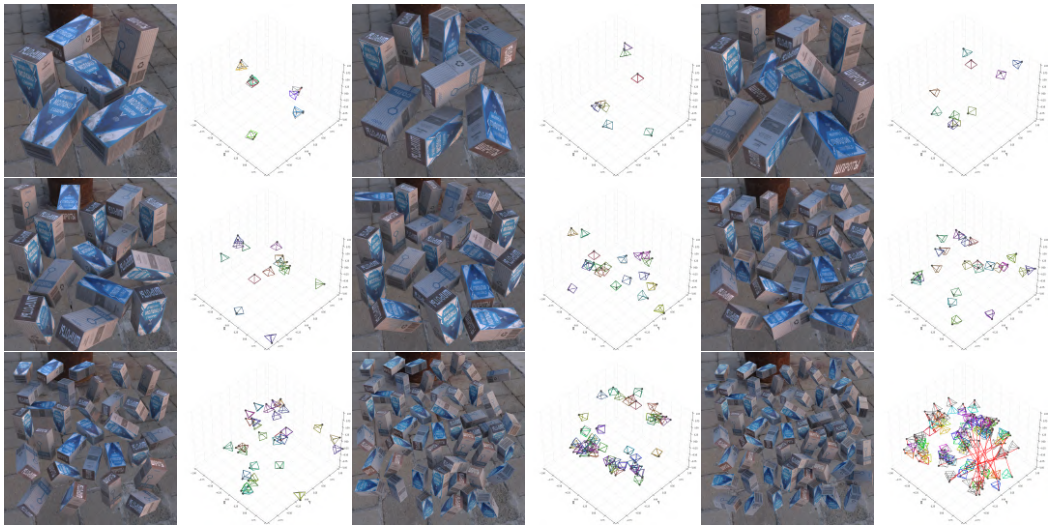


Figure 3: **Training images for different number of duplicated objects and the corresponding 6 DoF pose error.** The black camera represent the ground-truth and the colorful cameras are estimation. There are 6,8,10,15,20,25,30,50,60 instances in the scene. Our method has a large pose estimation errors for 60 boxes in 3200×3200 resolution due to the lower resolution of each instance.

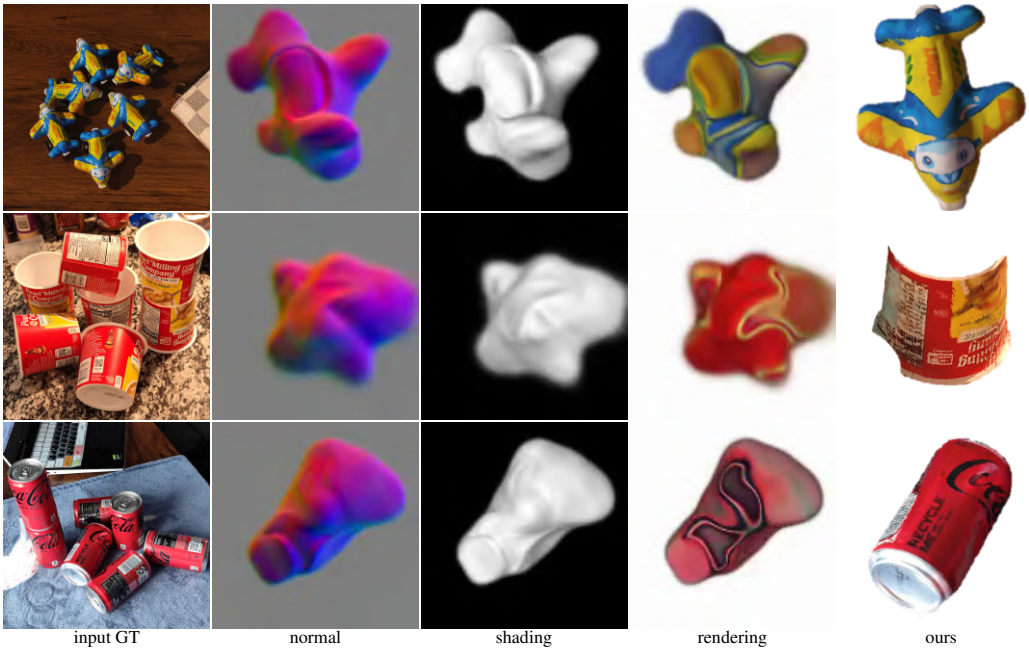


Figure 4: **Qualitative results of [13] on our single-image dataset.** When there are fewer instances, their generative approach produces only a blurred texture and imprecise geometry on our single-image datasets. In contrast, our method (as shown in the 5th columns) accurately reconstructs the objects.

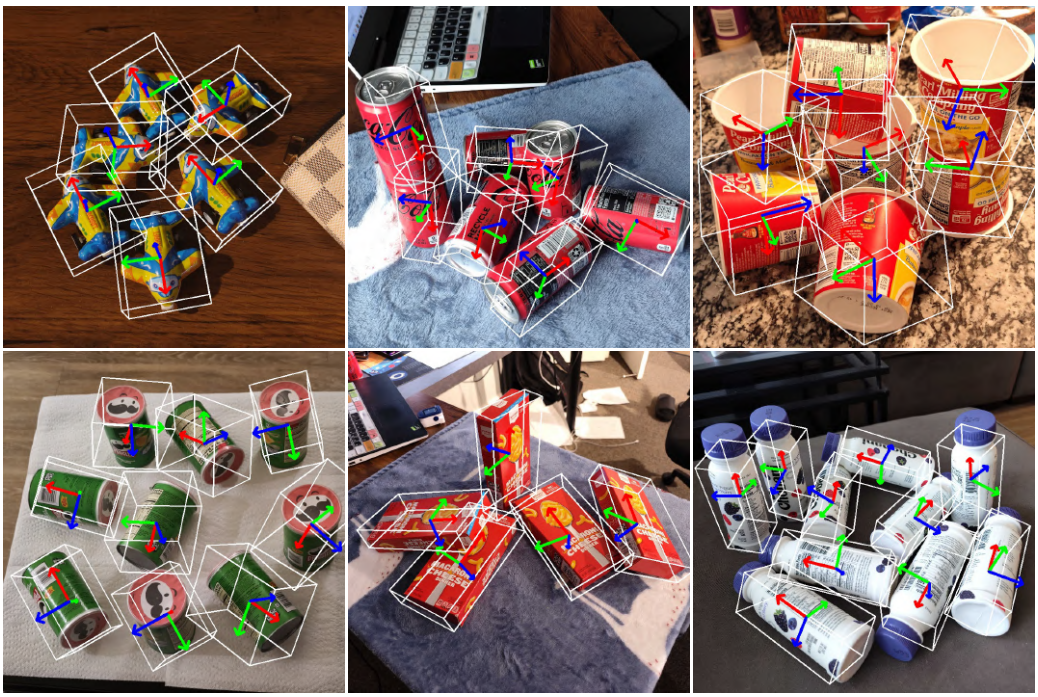


Figure 5: **The bounding box of real-world dataset.** The bounding boxes does not fully overlapped with each object because the bounding boxes are plotted according to the SfM points clouds, which does not fully cover object's surface.

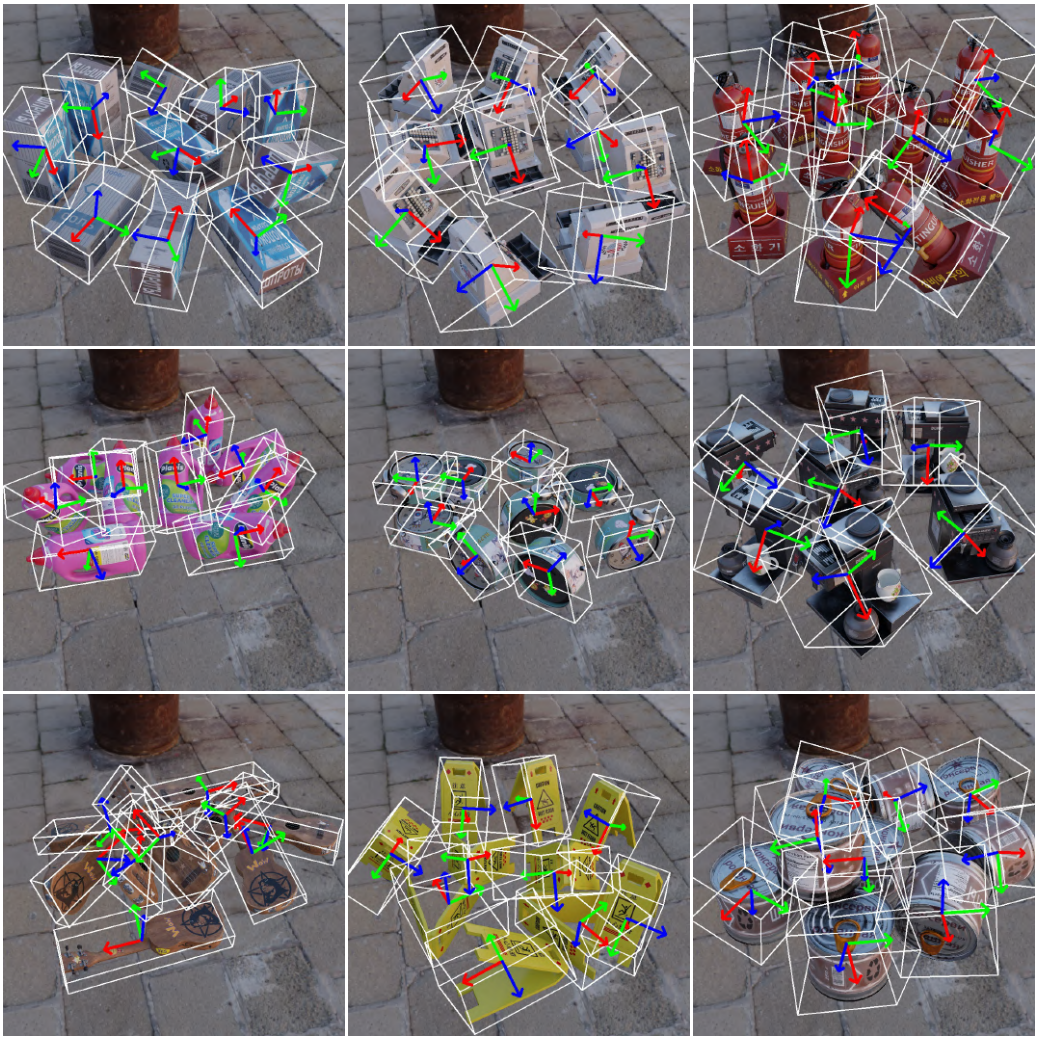


Figure 6: **The bounding box of real-world dataset.** The bounding boxes does not fully overlapped with each object because the bounding boxes are plotted according to the SfM points clouds, which does not fully cover object's surface.

Figure 7: Without considering the lighting effect and occlusion, a single-view image with duplicated objects (left) is equivalent to use multi-view camera to observe a single object (right).

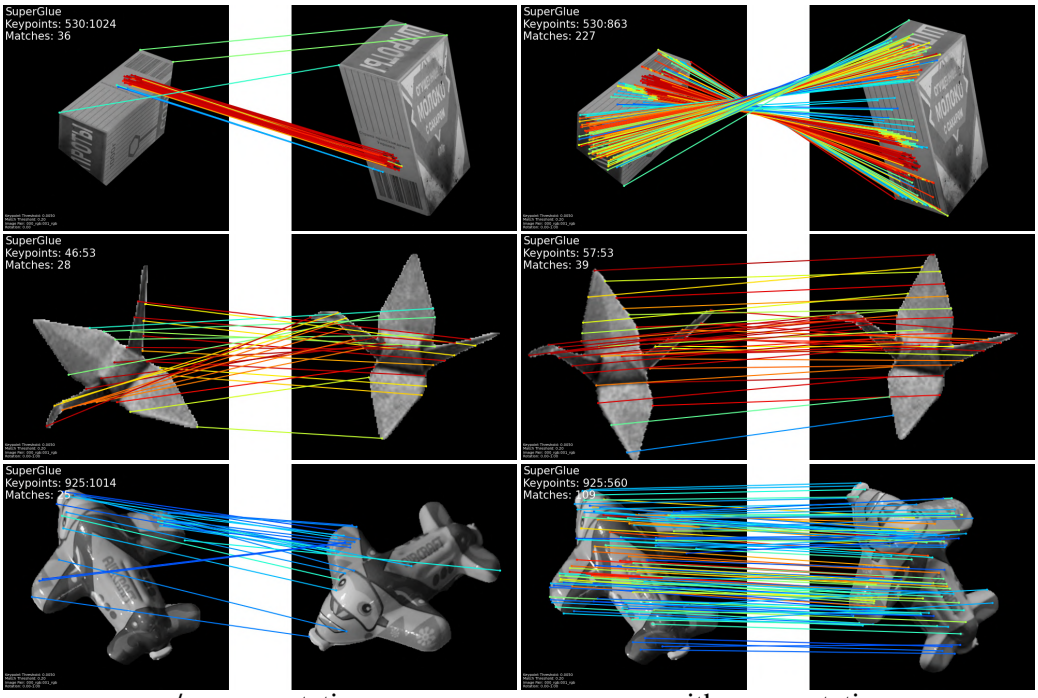


Figure 8: After in-plane augmentation, the pre-trained Super-point[1] and Super-glue[7] model can generate more matching points between two instances.

Figure 9: **Clean segmentation map(Left) and noisy segmentation map(right).** The right segmentation map is generated by pre-trained model and without post-processing.

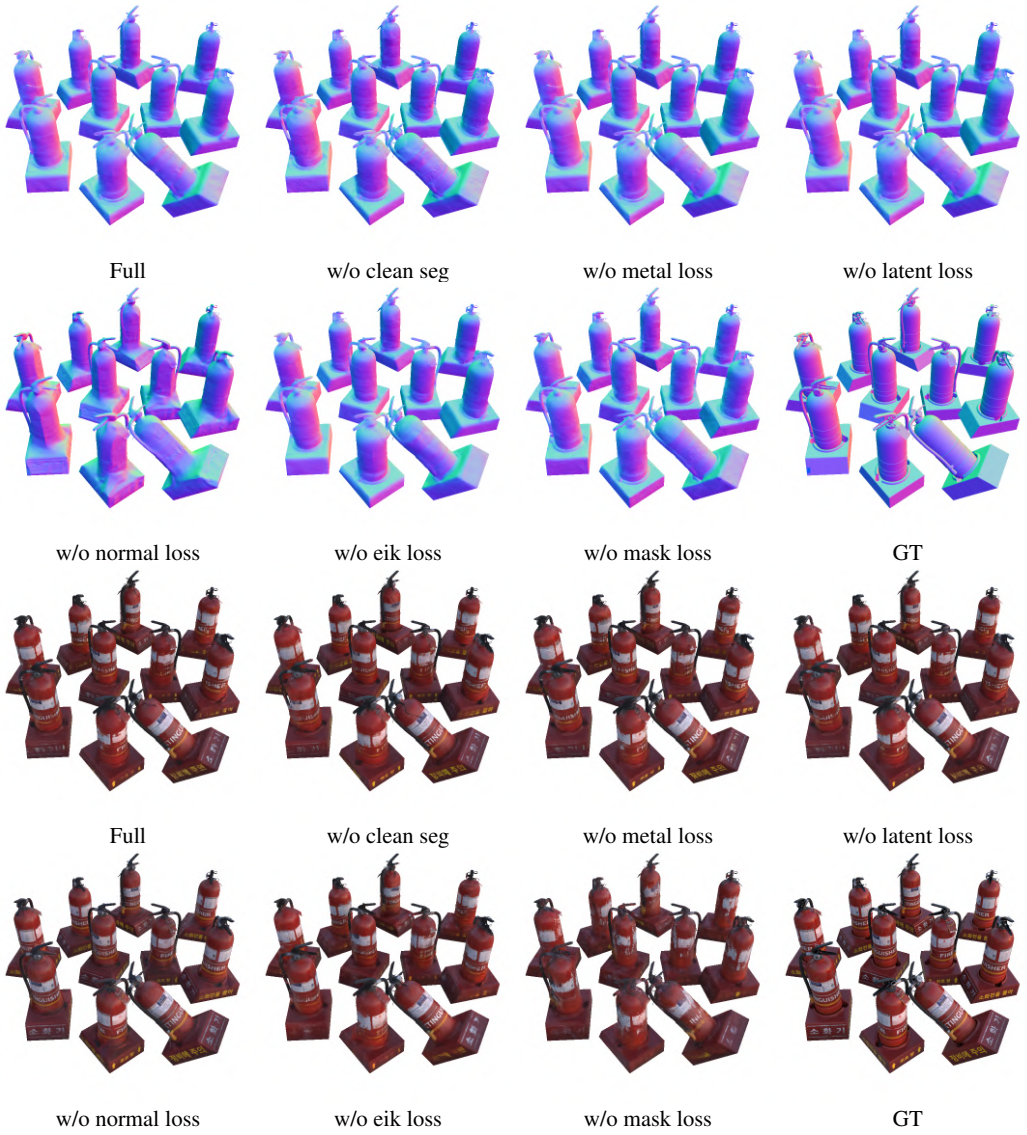


Figure 10: **Ablation for different loss term.** We evaluate the contribution of each loss term or input noise to our model. Our full most reaches the best result on most metrics

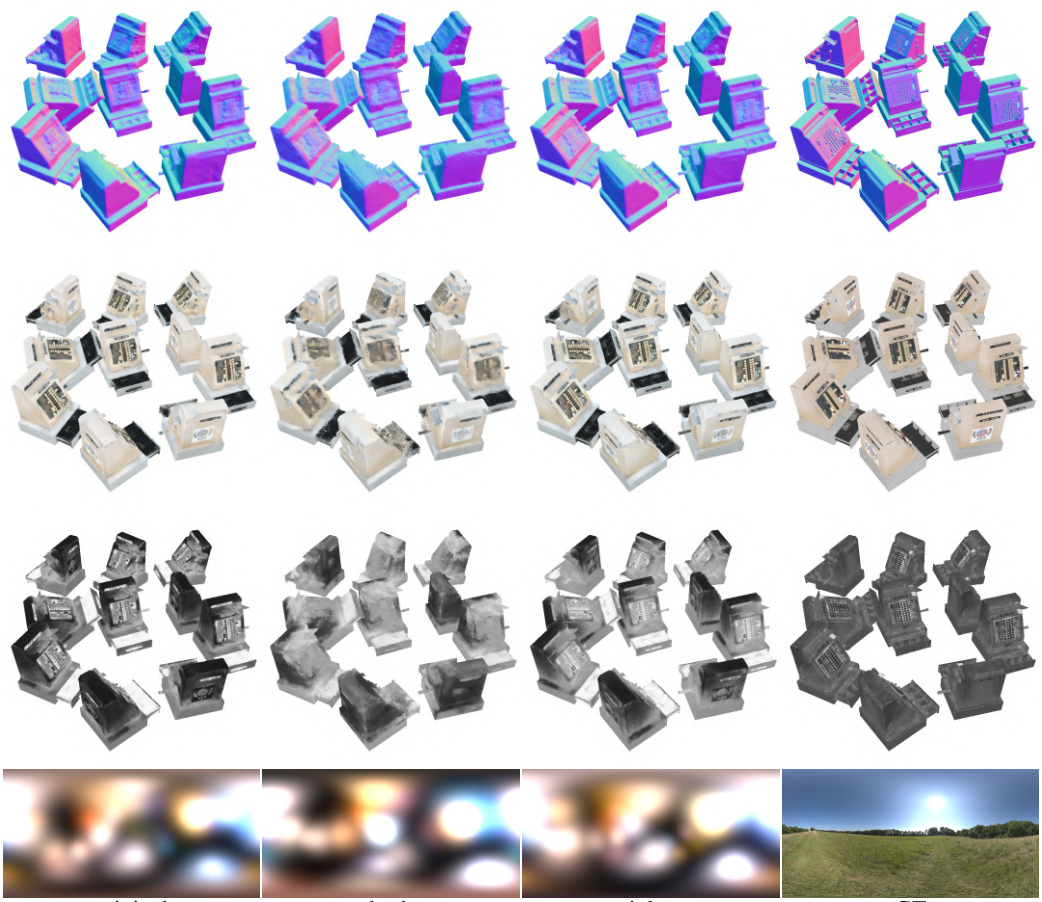


Figure 11: **Different neural representation.** The triplane representation (adapted from [10]) has better performance than our naive MLP representation, while hash position encoding (adapted from [3]) has worse performance due to overfitting.

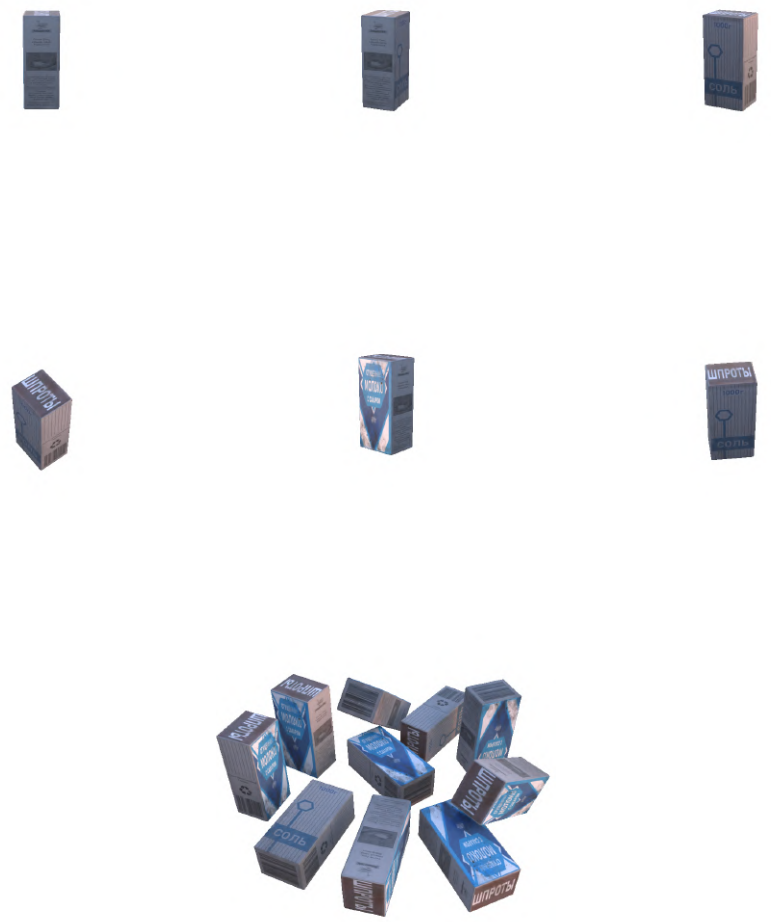


Figure 12: **First two rows**: Training images for multi-view single object (M-S), there are 10 in total, only show 6 here. **Last row**: Training image for single-view multi-object (S-M).

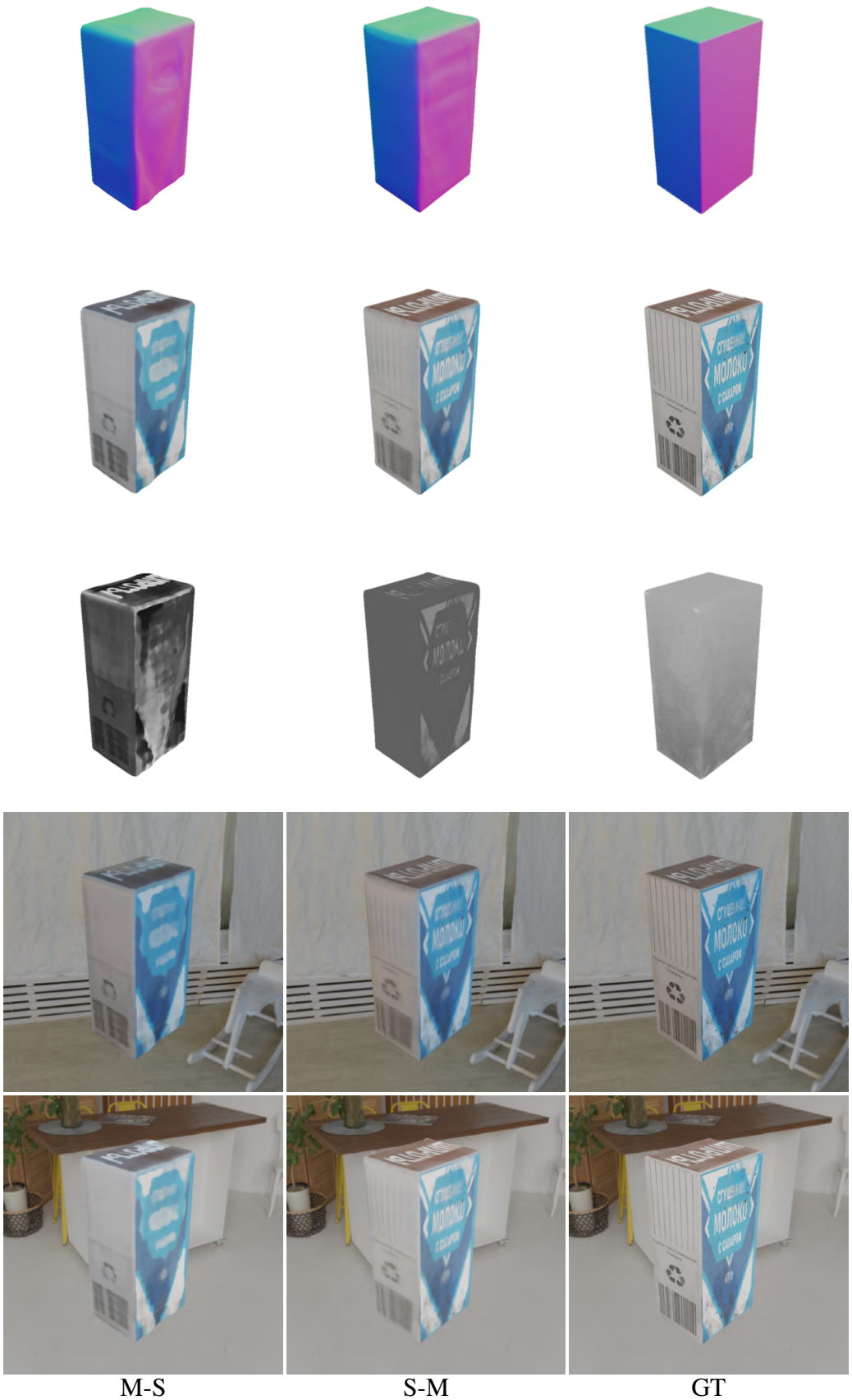


Figure 13: **M-S vs S-M** Visual result for multi-view single object (M-S) and single-view multiple objects (S-M).

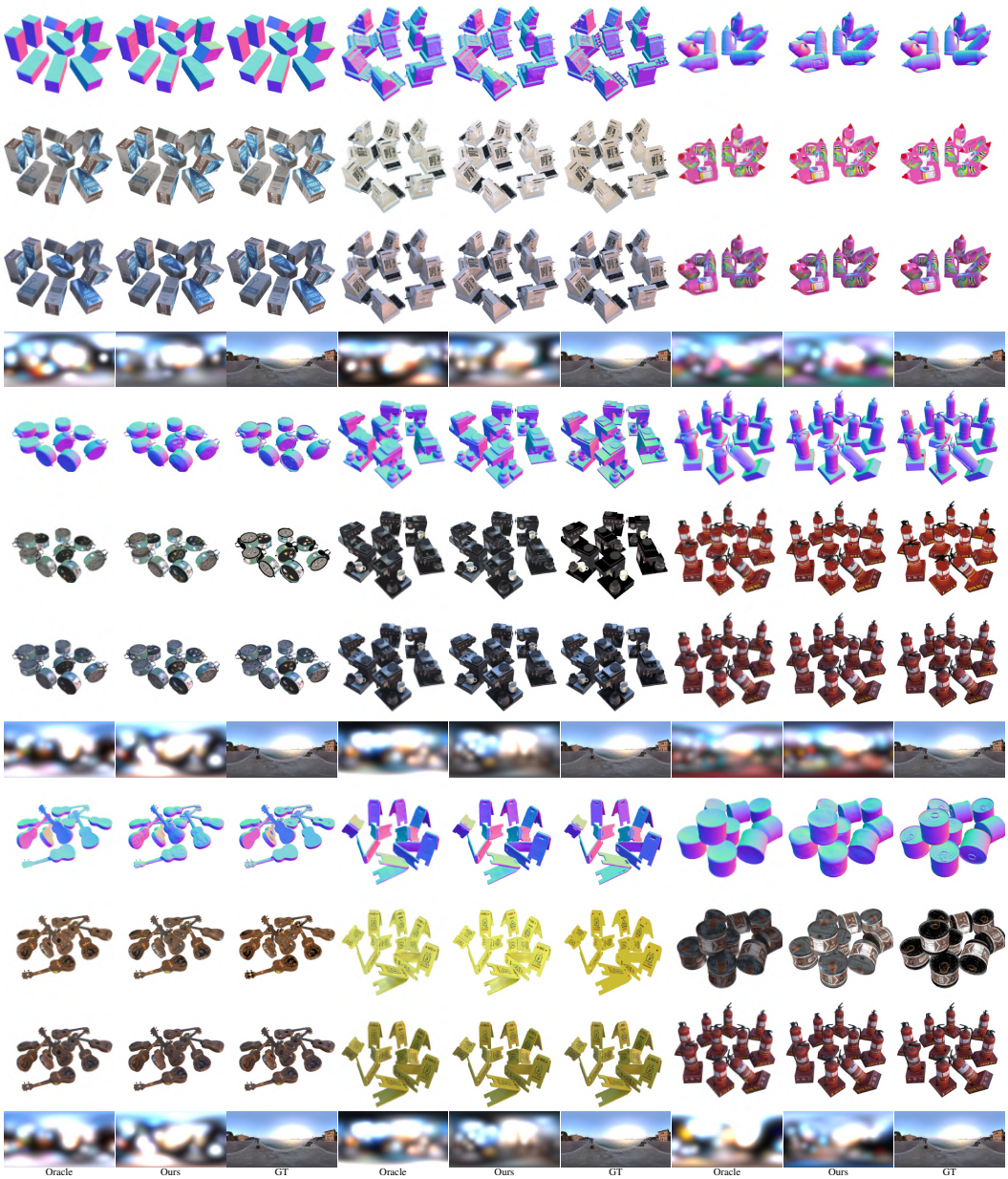


Figure 14: **Use ground-truth pose instead of SfM-derived pose.** Our model get similar results as orcale model.

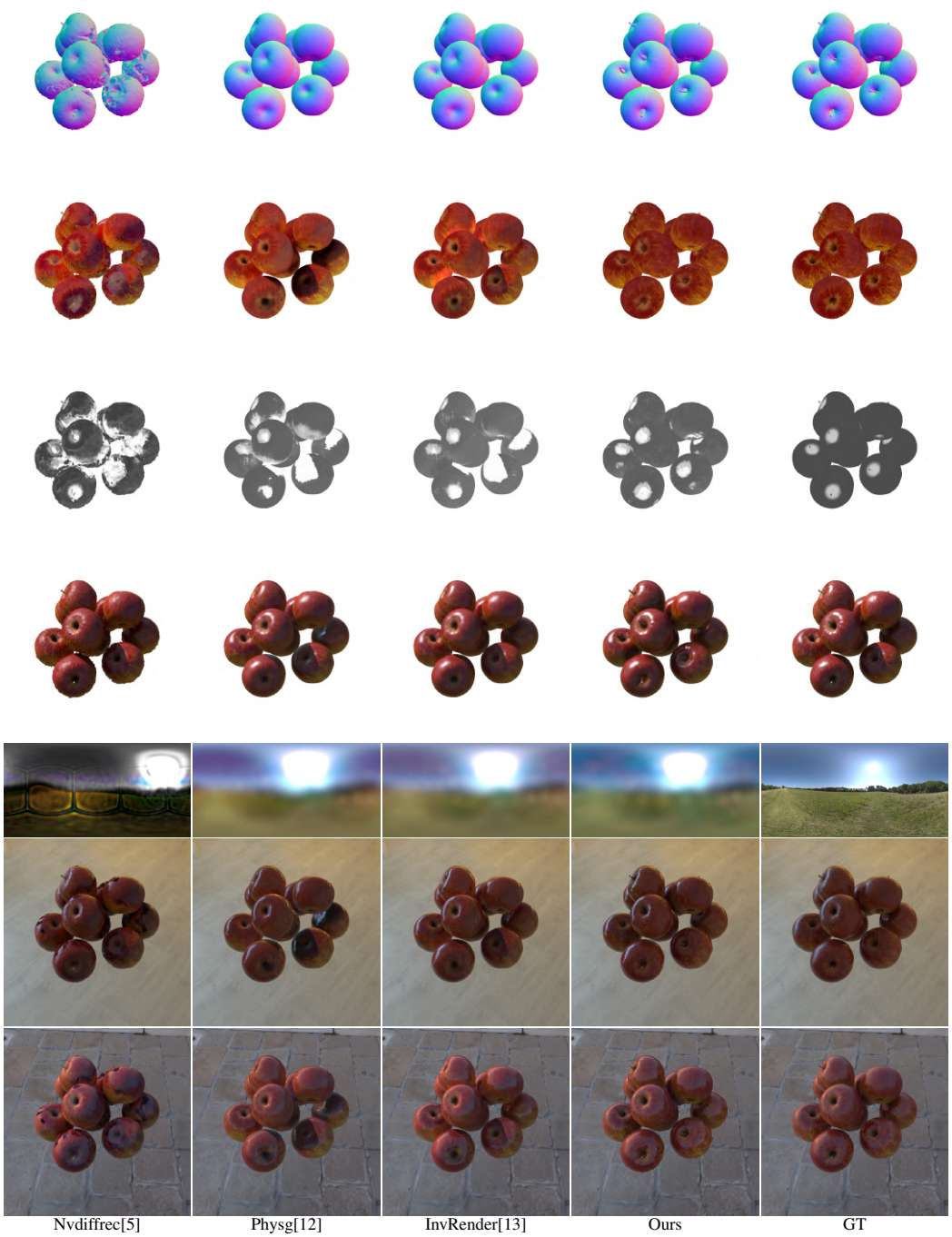


Figure 15: **Multi-view synthetic. Apple.**

Figure 16: **Multi-view synthetic. Box.**

Figure 17: **Multi-view synthetic. can.**

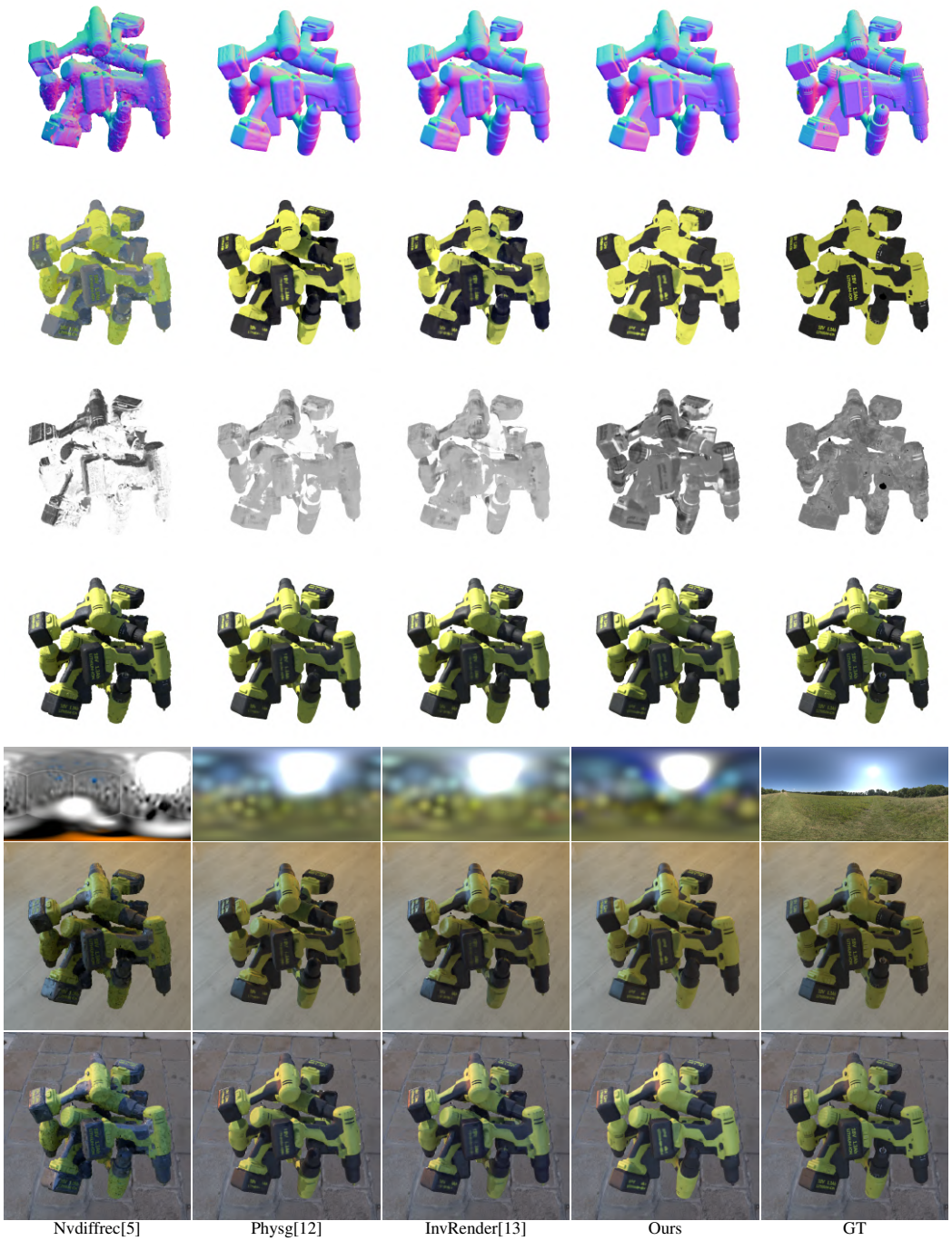
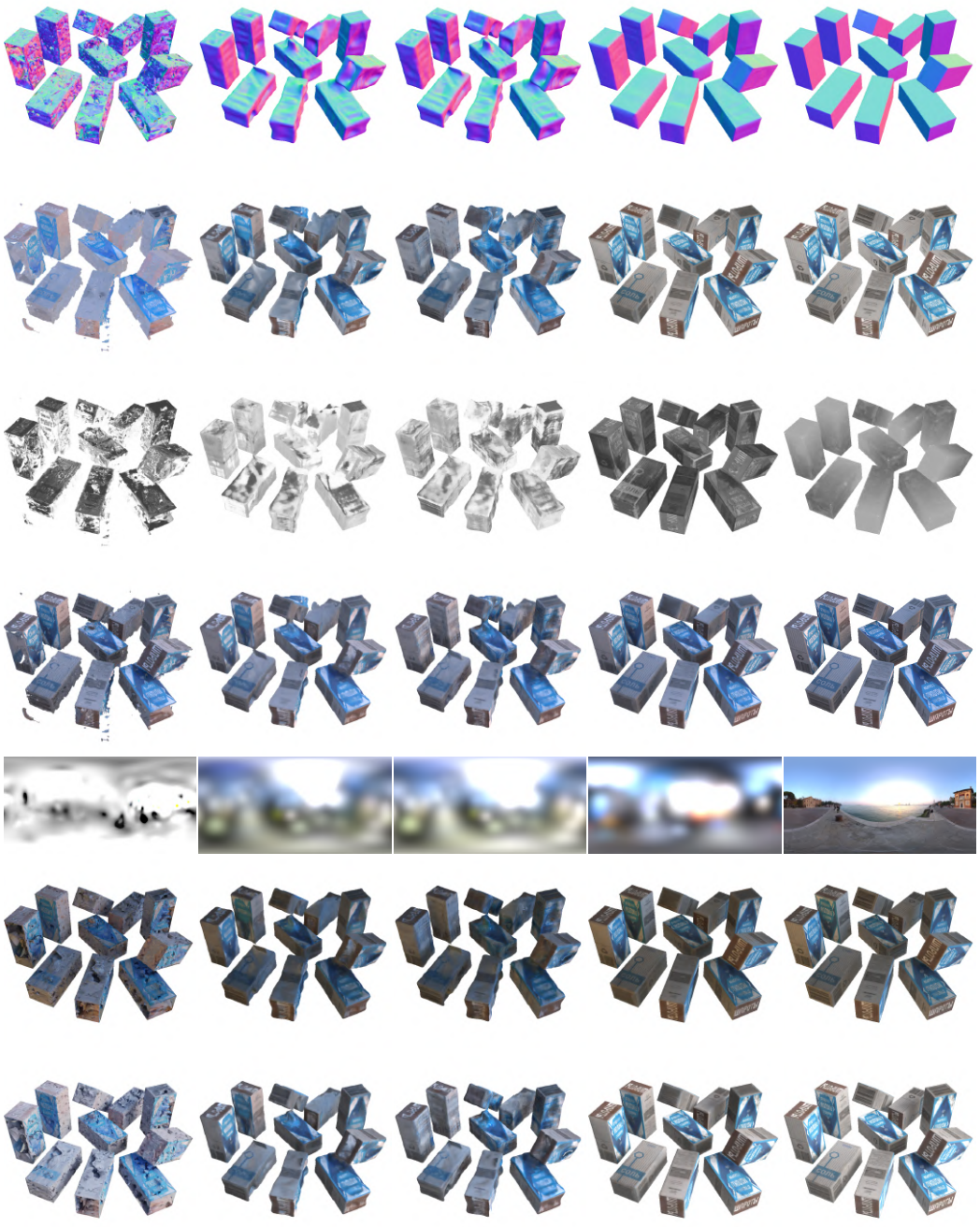


Figure 18: **Multi-view synthetic. drill.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours GT

Figure 19: **Single-view synthetic. box.**

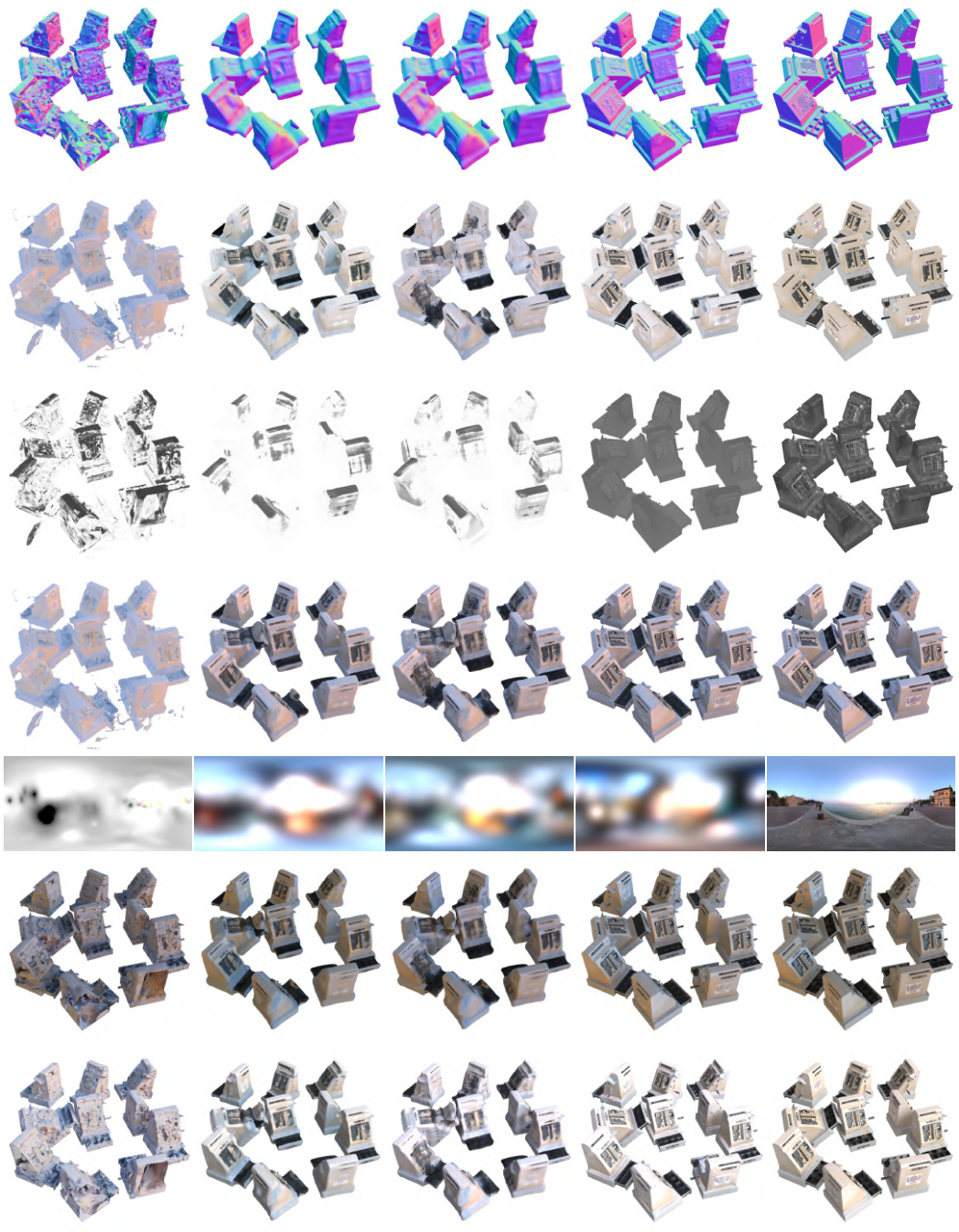


Figure 20: **Single-view synthetic. cash.**

Nvdiffrec[5] Physg[12] InvRender[13] Ours GT

Figure 21: **Single-view synthetic. cleaner.**

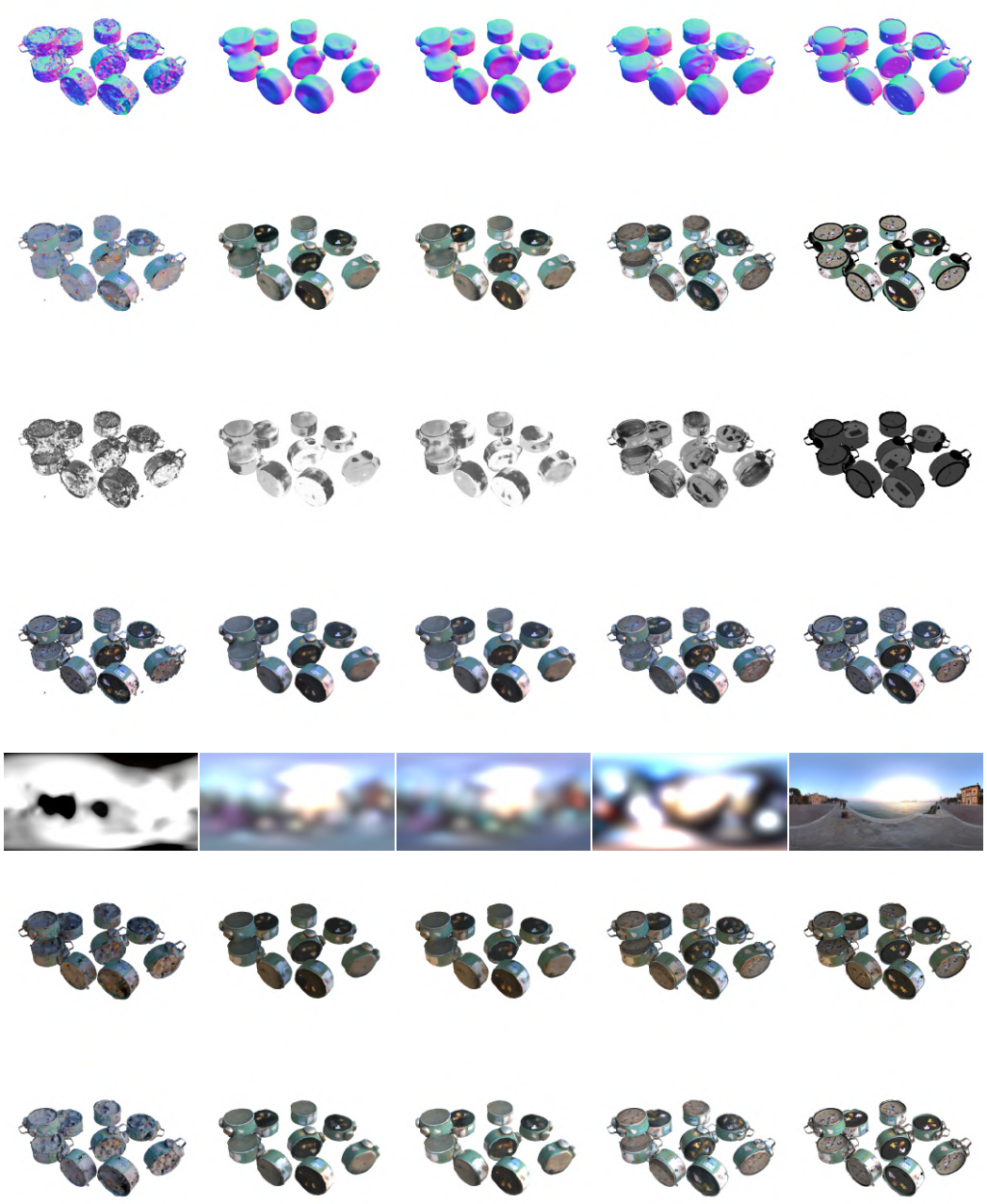
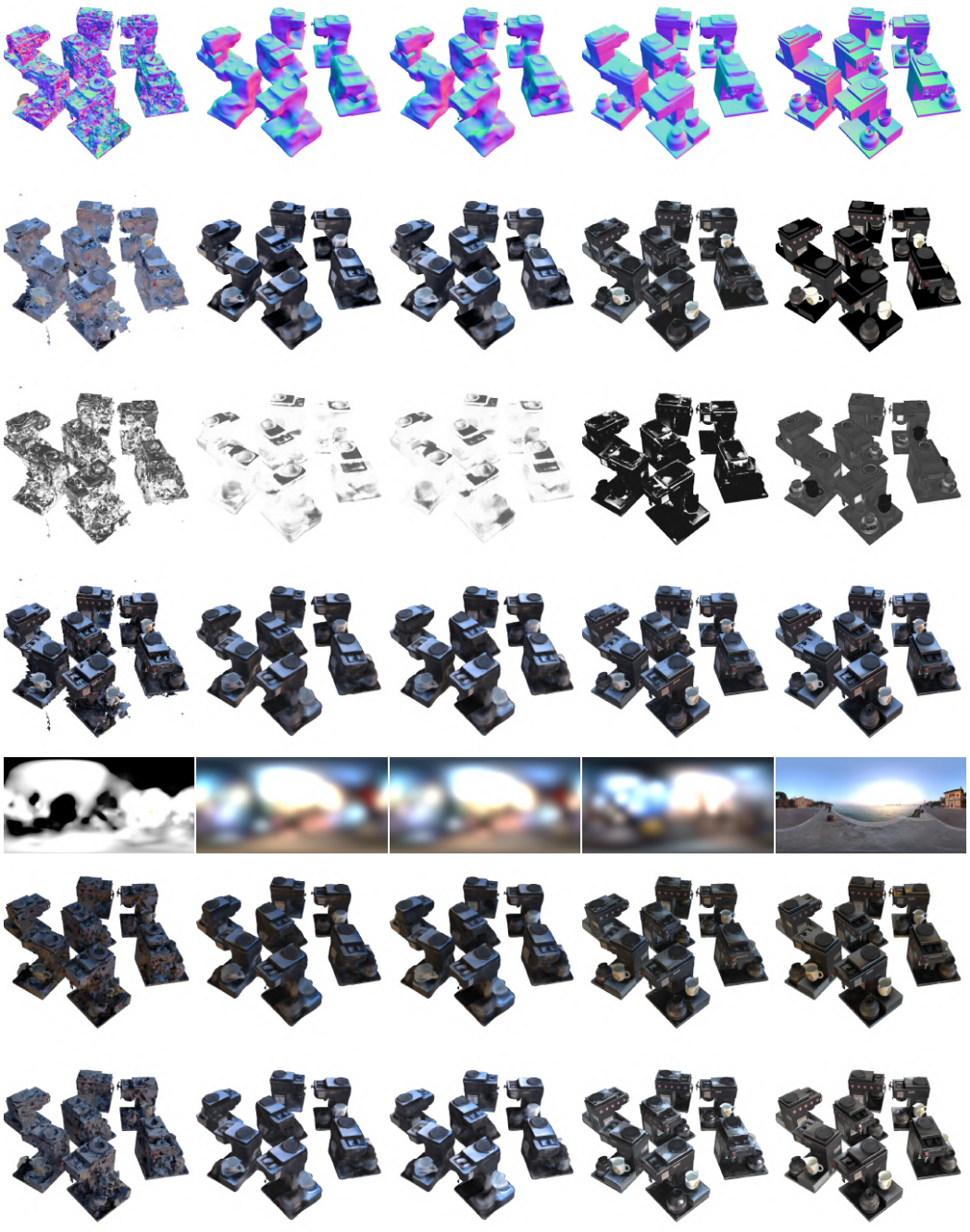


Figure 22: **Single-view synthetic. clock.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours GT

Figure 23: **Single-view synthetic. coffee.**

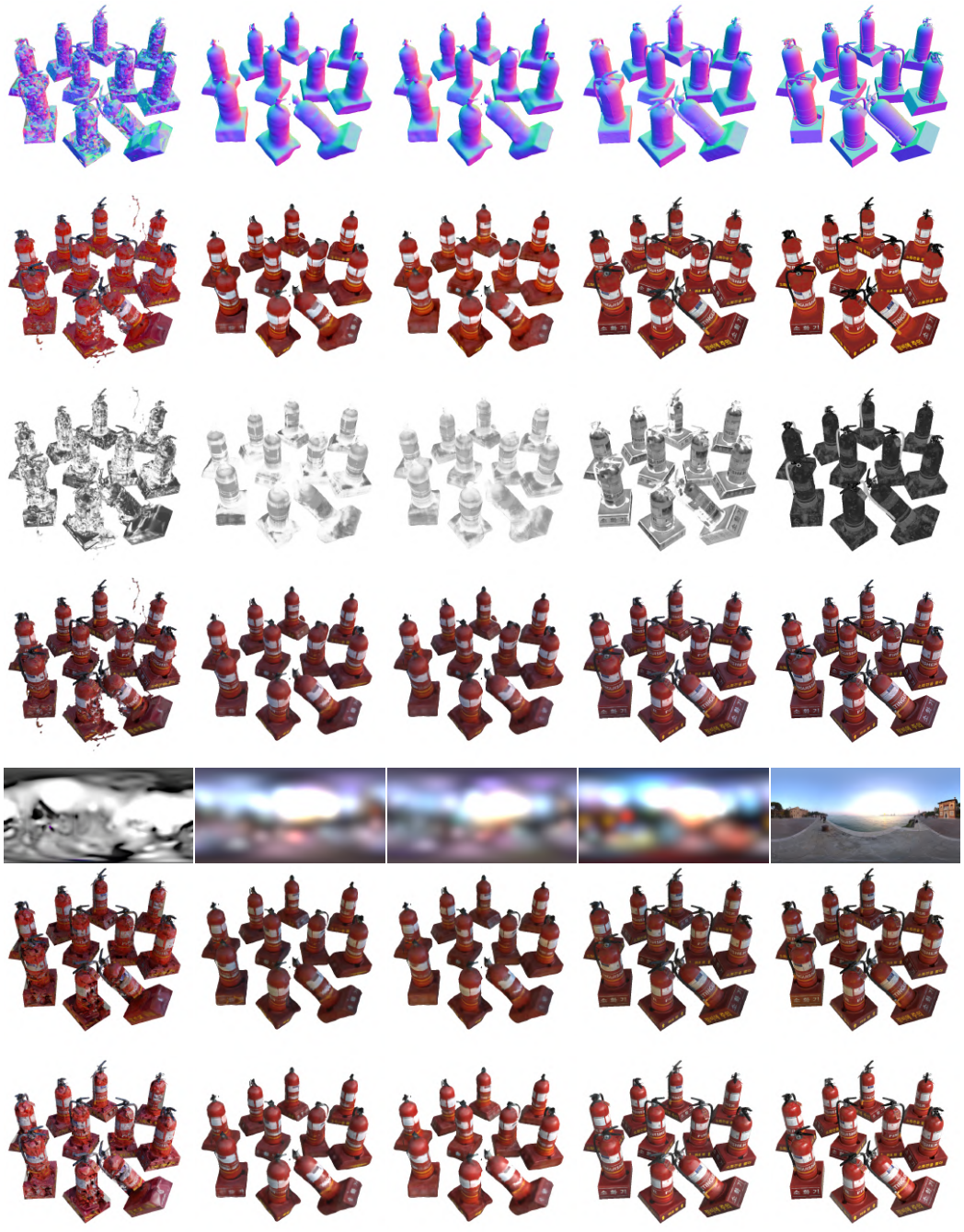


Figure 24: **Single-view synthetic. fire.**

Figure 25: **Single-view synthetic. guitar.**

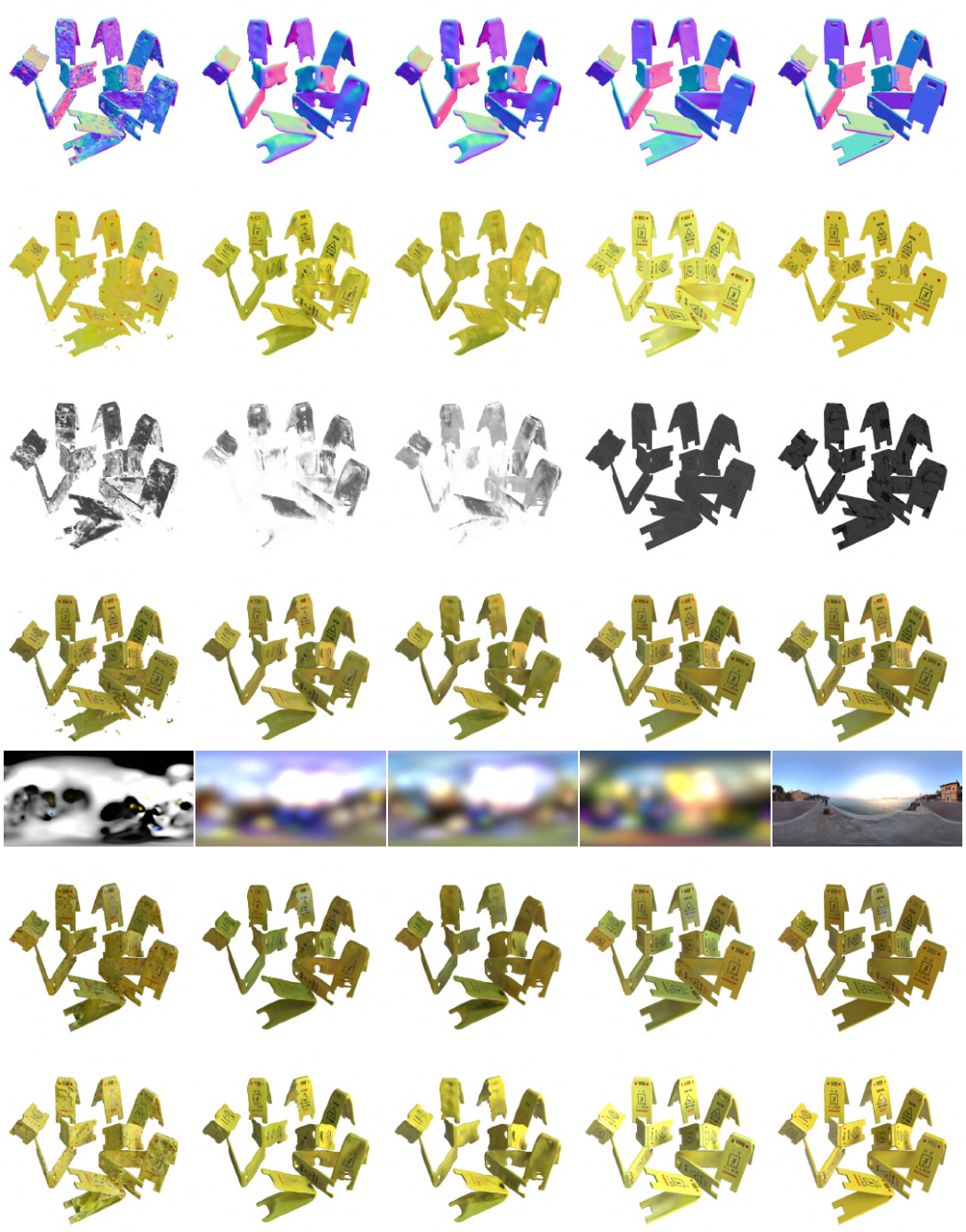
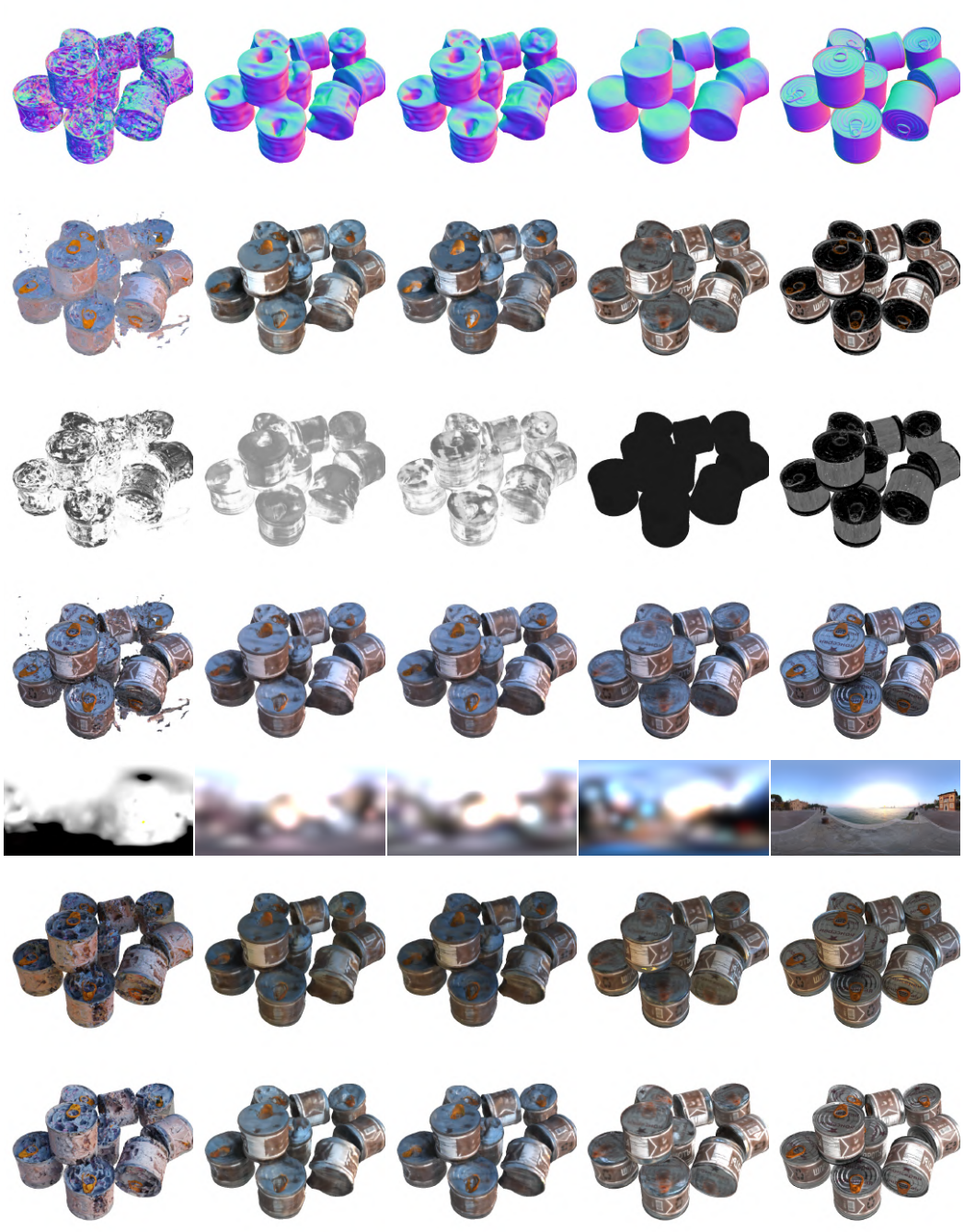


Figure 26: **Single-view synthetic. sign.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours GT

Figure 27: **Single-view synthetic. tin.**

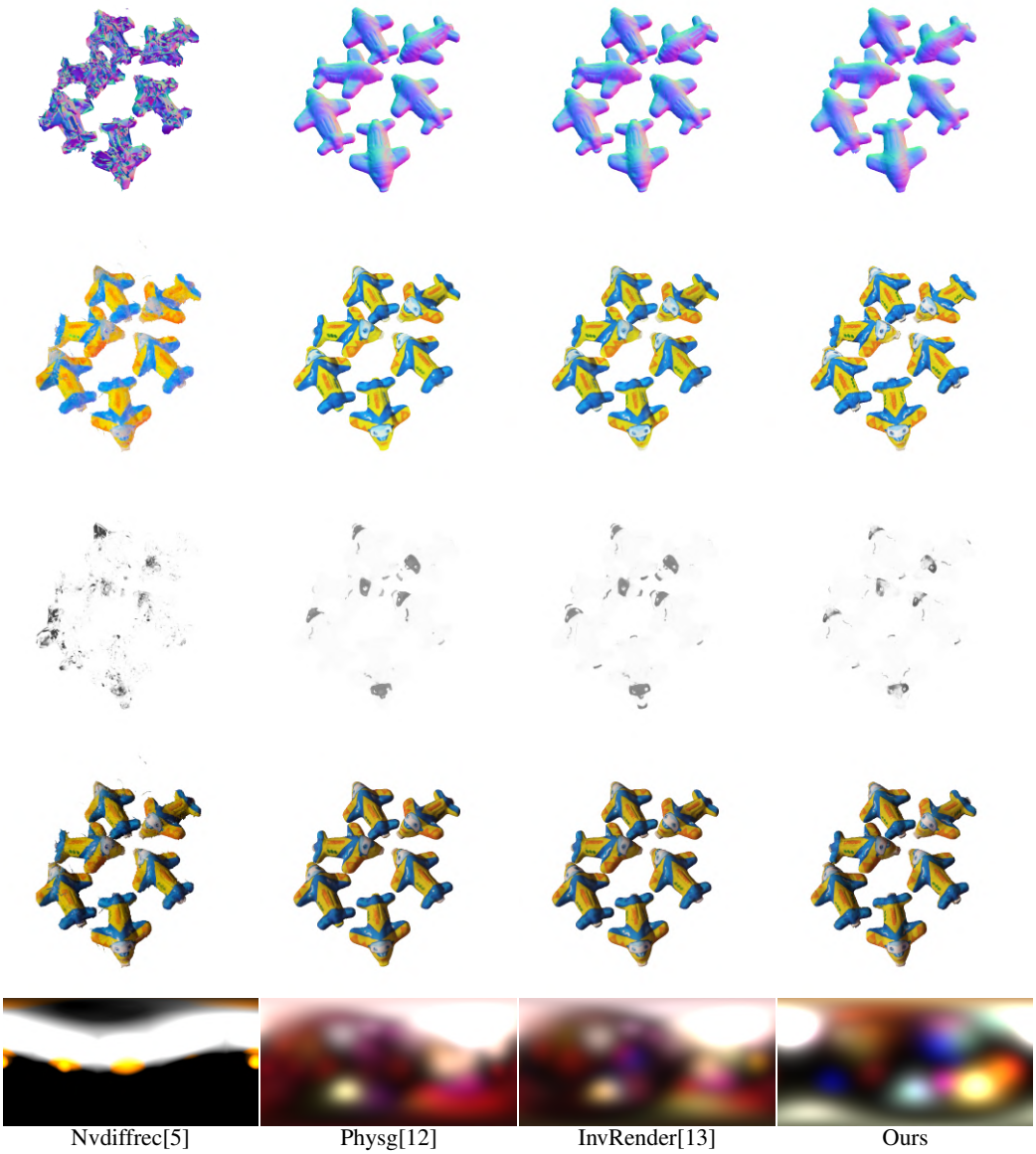
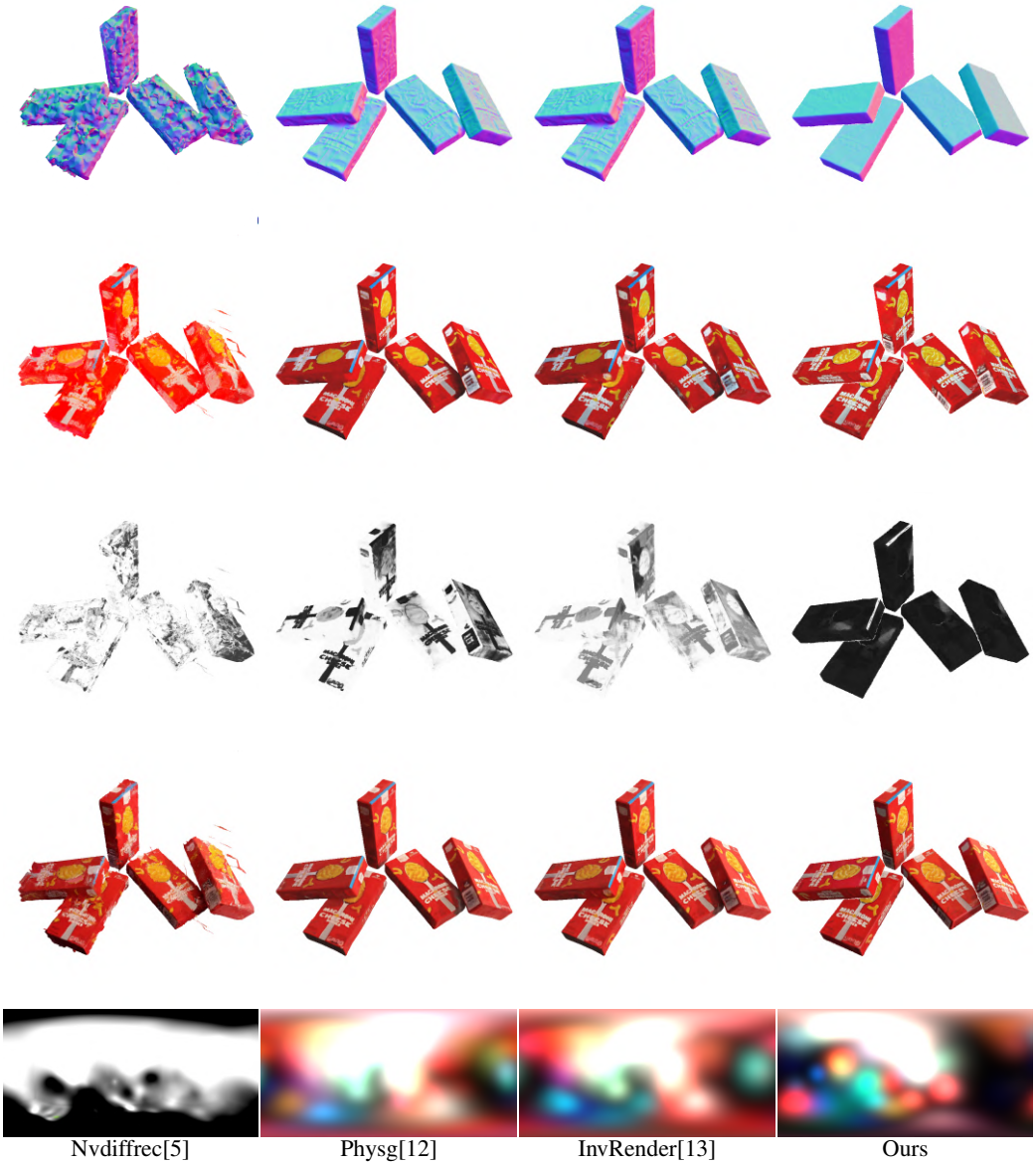


Figure 28: **Single-view realworld. airplane.**

Nvdiffrec[5] Physg[12] InvRender[13] Ours

Figure 29: **Single-view realworld. cake.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours

Figure 30: **Single-view realworld. cheese.**

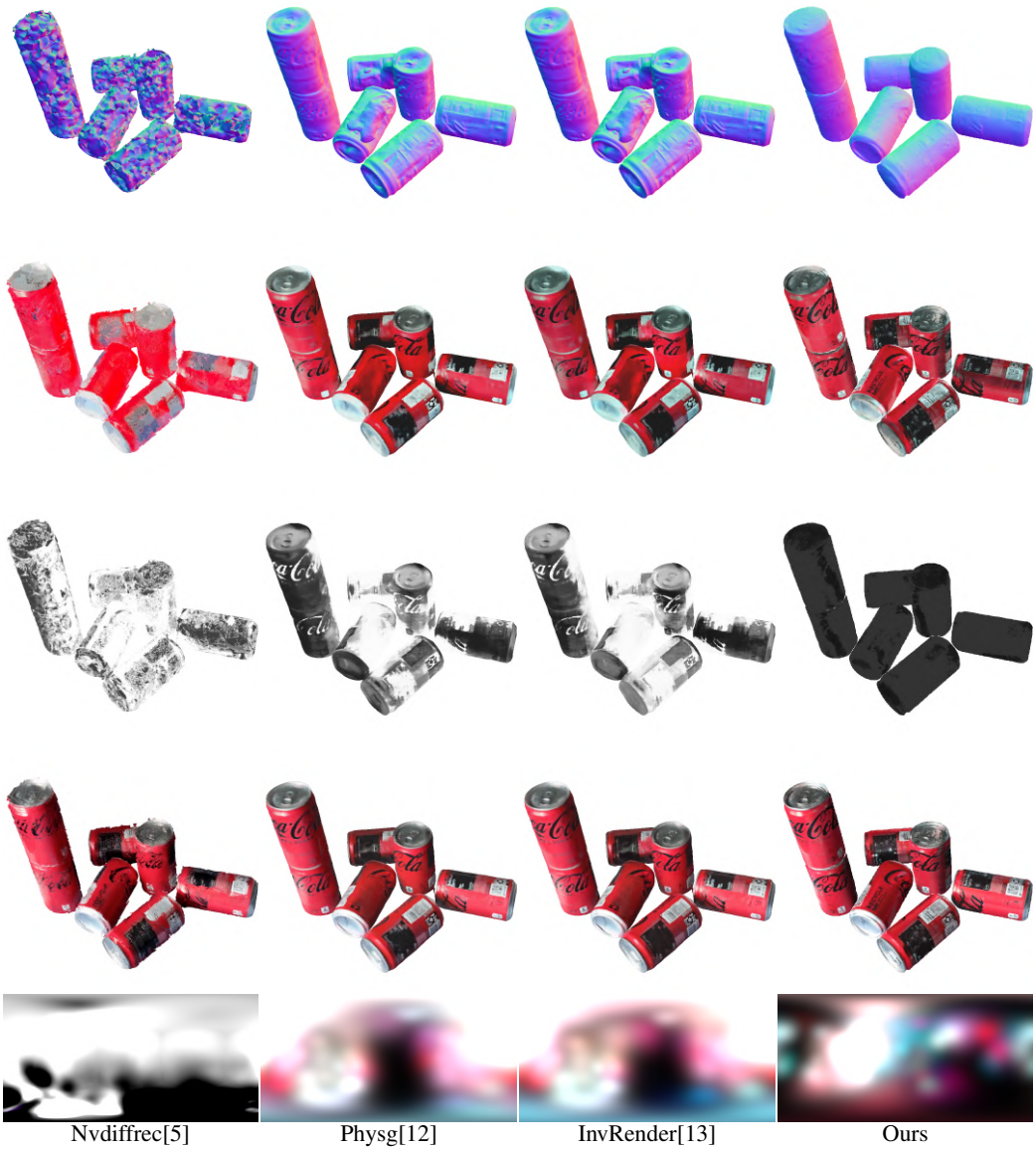
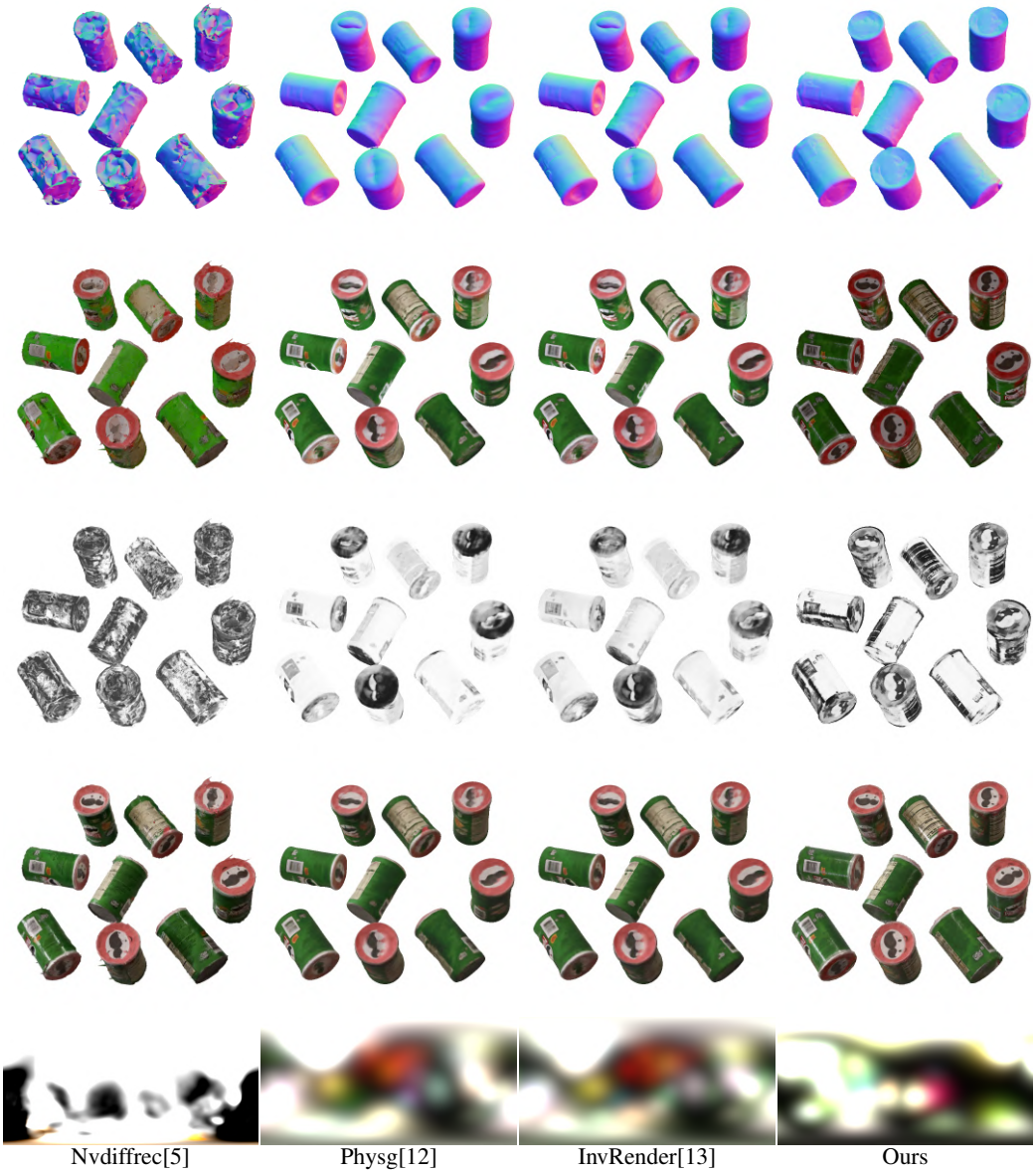
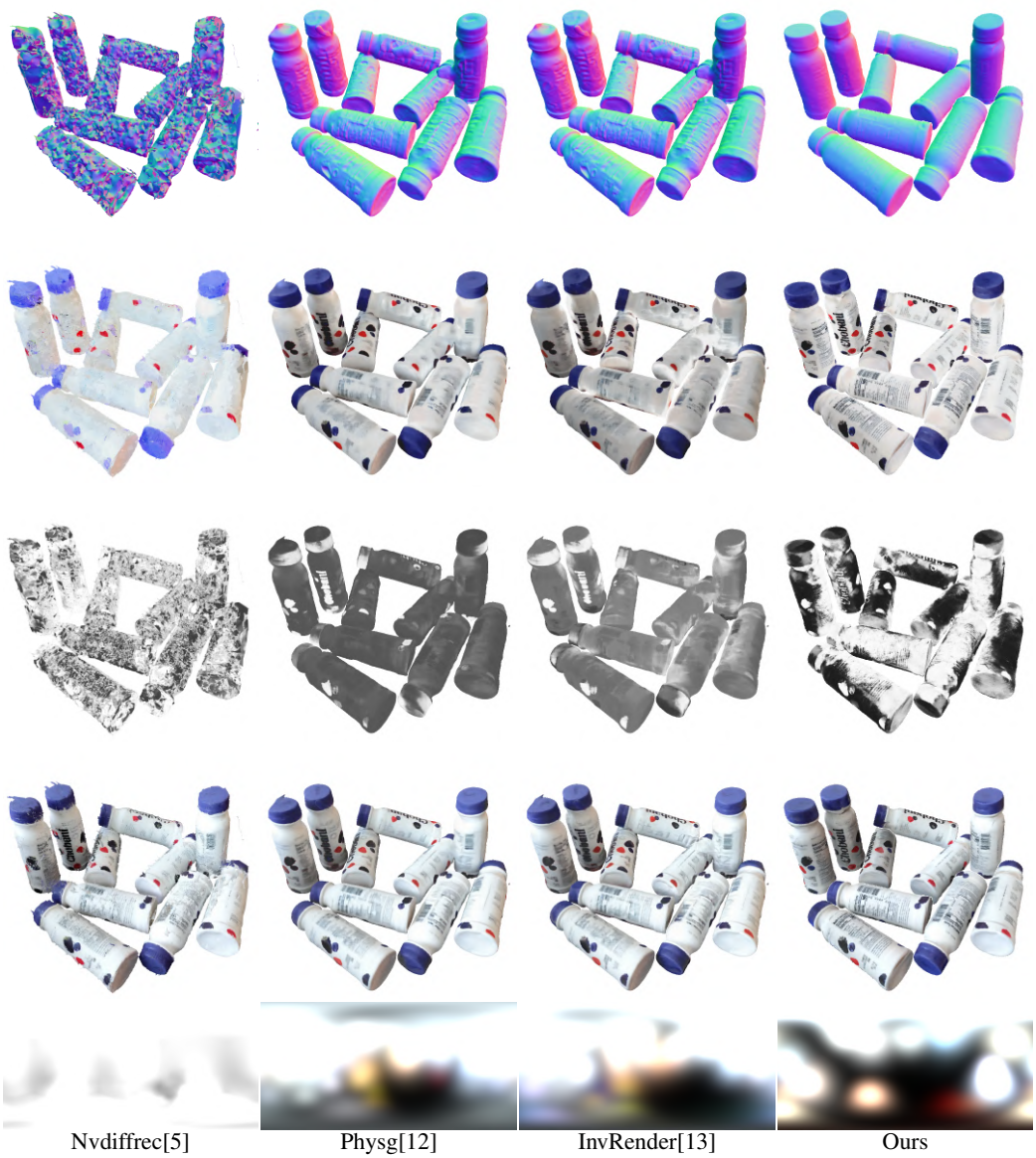


Figure 31: **Single-view realworld. cola.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours

Figure 32: **Single-view realworld. potato.**



Nvdiffrec[5] Physg[12] InvRender[13] Ours

Figure 33: **Single-view realworld. yogurt.**