D^{2}NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video

Tianhao Wu
University of Cambridge

Fangcheng Zhong
University of Cambridge

Andrea Tagliasacchi
Google Research
Simon Fraser University

Forrester Cole
Google Research

Cengiz Oztireli
Google Research
University of Cambridge

Abstract

Given a monocular video, segmenting and decoupling dynamic objects while recovering the static environment is a widely studied problem in machine intelligence. Existing solutions usually approach this problem in the image domain, limiting their performance and understanding of the environment. We introduce Decoupled Dynamic Neural Radiance Field (D^{2}NeRF), a self-supervised approach that takes a monocular video and learns a 3D scene representation which decouples moving objects, including their shadows, from the static background. Our method represents the moving objects and the static background by two separate neural radiance fields with only one allowing for temporal changes. A naive implementation of this approach leads to the dynamic component taking over the static one as the representation of the former is inherently more general and prone to overfitting. To this end, we propose a novel loss to promote correct separation of phenomena. We further propose a shadow field network to detect and decouple dynamically moving shadows. We introduce a new dataset containing various dynamic objects and shadows and demonstrate that our method can achieve better performance than state-of-the-art approaches in decoupling dynamic and static 3D objects, occlusion and shadow removal, and image segmentation for moving objects. Project page: d2nerf.github.io

1 Introduction

Reasoning about motion is a fundamental task in machine vision which facilitates intelligent interactions with the 3D environment for applications such as robotics and autonomous driving. Given a monocular RGB video captured from a moving casual camera, we consider the problem of disentangling the camera from object motion, and simultaneously recovering a 3D model of the static environment.

While decomposition of scenes in the image domain has been addressed in the literature, the use of 2D priors and inpainting technique lacks 3D understanding, leading to sub-optimal results. We approach this problem in 3D, aiming to reconstruct a decoupled 3D scene representation that allows for synthesizing the dynamic and static objects *separately* in a free-view and time-varying fashion.

Compared to the task of *static* scene reconstruction [32], modeling a scene with time-dependent effects is a severely ill-posed problem. Existing works seek for robust solutions by incorporating additional supervision such as multi-view capture [25], optical flow [10], or depth [60], but they treat

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

every part of the scene as time-dependent, leading to a poor reconstruction of background details due to limited network capacity.

In this paper, we adapt Neural Radiance Fields [32] (NeRF) and its extension HyperNeRF [40] to time-varying scenes by decoupling the dynamic and static components of the scene into separate radiance fields. Previous techniques that decouple dynamic and static scenes either rely on pre-trained object detection/segmentation modules [19, 26, 12, 23], or are limited to a single rigid object [68] or semi-static objects [54]. Our method learns dynamic and static components separately in a self-supervised fashion, using a novel skewed-entropy loss to encourage a clean separation of static and dynamic objects.

A crucial issue in creating a clean separation is *properly handling shadows*, as dynamic objects cast shadows that cause the radiance of the shadow receiver to vary with time. When the shadow receiver is part of the static component, this time-varying change in radiance cannot be directly modeled. Our solution is to relax the static component with a time-varying *shadow field* that modulates the radiance, allowing the shadows cast by moving objects to be captured while constraining density and color to be static.

Our method enables 3D scene decoupling and reconstruction from a monocular video captured from casual equipment such as a mobile phone, and can be readily extended to multi-view videos. By separately modeling the time-varying and time-independent targets in the video, our method can remove the dynamic occluders and their shadows, and synthesize a clean background from novel views.

We demonstrate the effectiveness of our method in *two aspects*: (i) the quality of novel view synthesis of the decoupled static background for monocular videos where the dynamic objects and shadows heavily occlude the scene, and (ii) the correctness of segmentation of dynamic objects and shadows on 2D images.

We introduce a *new dataset* with rigid and non-rigid dynamic objects, rapid camera motion and various moving shadows in both the synthetic and real-world settings to evaluate these two aspects, and show that our method achieves better performance than state-of-the-art approaches.

2 Related Work

As our method learns a decoupled neural 3D representation of the dynamic and static scenes, we start this section with a review of scene representations, and then focus on methods for object motion decoupling. We also review prior works for 2D segmentation of moving objects.

Scene Representations A 3D scene representation is a data structure that encodes the geometry and appearance of a 3D scene, upon which many algorithms and applications are developed. Recently, there has been a surge of methods that combine deep learning methods with traditional 3D representations: point clouds [17, 41, 11, 1], meshes [7, 33, 3, 55], voxels [6, 59], implicit surfaces [38, 63, 15, 14, 51], light fields [50, 52, 2, 48] and volumetric fields [28, 32]. Among neural representations, NeRF [32] has attracted substantial attention due to its photo-realistic performance in novel view synthesis for scenes with complex geometry, lighting, and materials. Via differentiable volume rendering and inputs of multiple views of the scene, NeRF applies an MLP to learn a 5D radiance field of the scene modeling the spatially and view-dependent radiance. Various extensions of NeRF have been developed to improve its performance and generality such as training with only one or few views [67, 18, 22, 35, 45], allowing for input images with inconsistent lighting and object locations [31, 69], learning large-scale scenes with street or satellite views [47, 61], speeding up rendering to reduce training and inference time [29, 8, 44, 46, 66, 13, 34, 27, 53, 58, 65], capture of dynamic effects within the scene [25, 43, 10, 60, 39, 40, 21, 36], and decomposing self-occluded shadows to recover correct albedo [9]. We further extend NeRF to *decouple* dynamic from static effects.

Motion Decoupling Prior works to acquire a decoupled 3D representation of dynamic and static scenes can be divided into either supervised or self-supervised approaches. Among the supervised approaches, STNeRF [19] learns individual NeRFs with deformation fields for each human in a dynamic scene through pre-trained human segmentation networks. Similarly, NSFF [26] and DynNeRF [12] rely on pre-trained semantic and motion segmentation methods to obtain masks for

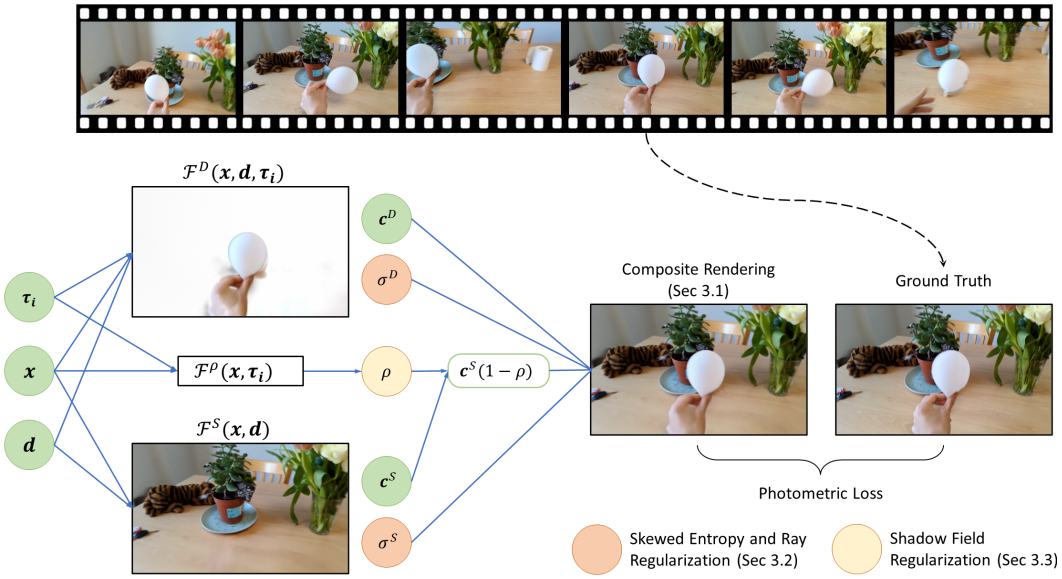

Figure 1: **Overview** – Given the ground truth view, camera pose and the time frame, our method reconstructs the underlying scene as a composite radiance field. Dynamic objects are represented by $\mathcal{F}^D$, while the static scene is represented by $\mathcal{F}^S$. The shadow-field $\mathcal{F}^\rho$ models non-static shadows within the input video.

moving objects in a monocular video, and explicitly guide the training of separate NeRF networks to decouple the scene based on motion. Among the self-supervised approaches, SIMONe [20] incorporates a transformer encoder and variational autodecoder to simultaneously recover novel views, object segmentation masks and dynamic object trajectories, but they do not allow for a synthesis of dynamic or static objects alone. NeRF-W [31] employs per-frame embeddings to model non-photometric consistent effects in unconstrained photo collections, but their design was not intended for a clean separation between moving objects and the static scene. STaR [68] reconstructs and decomposes a rigid dynamic object and the static background simultaneously by optimizing two NeRFs and a set of time-varying object poses in a self-supervised way, but it is only suitable for scenes with a single rigid dynamic object and requires multi-view videos. Conversely, our approach works with more complex scenes involving multiple non-rigid and topologically varying objects, and our method can be directly applied from monocular video. NeuralDiff [54] incorporates three NeRF-based streamlines to decompose background, object and actor from an egocentric video, and it is the most similar to ours within the literature. However, its use of a naive time-varying NeRF architecture leads to blurry results and therefore heavily limits its performance on both scene decomposition and reconstruction.

Image Segmentation of Moving Objects Orthogonal to the reconstruction and disentanglement in 3D, there have also been extensive researches in self-supervised and template-free segmentation at the image level (i.e. 2D). The majority relies on motion-clues to segment objects with different optical flow patterns [4, 64, 37, 62]. Some techniques incorporate a transformer style slot-based attention scheme to learn consistent object segmentation over a sequence of optical flow images [64], while others learn alpha-matting from a single video with smooth camera movement and homographic background and extend the segmentation target to correlated effects such as shadow or reflectance [30]. Those approaches come with obvious shortcomings, as they focus on *image* level segmentation and incorporate no 3D understanding, they cannot handle large scale camera motion, complicated static background and cannot recover 3D geometry, or perform novel view synthesis.

3 Method

Given a *monocular* video captured from a freely *moving* hand-held camera, our method reconstructs a neural scene representation that decouples moving objects from the static environment, assuming a

constant illumination and known camera poses (e.g. calibrated with COLMAP [49]). As illustrated in Figure 1, our method achieves this by learning *separate* radiance fields for static and dynamic portions of the scene, and doing so in a fully self-supervised fashion. We describe our architecture (Section 3.1), detail of our self-supervised losses (Section 3.2), and describe how, while shadows are not explicitly modeled by NeRFs, a simple technique for their effective removal is attainable (Section 3.3).

3.1 Composite Neural Radiance Field

The static component builds upon NeRF [32], which represents the scene as continuous spatial-dependent density σ and spatial-view-dependent radiance $\mathbf{c}$ using an multi-layer perceptron $\mathcal{F}^S$:

$$\left.\begin{array}{l}\sigma^S(\mathbf{x}) \in \mathbb{R} \\ \mathbf{c}^S(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^3\end{array}\right\} = \mathcal{F}^S(\mathbf{x}, \mathbf{d}) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ is the spatial coordinate, and $\mathbf{d} \in \mathbb{R}^3, \|\mathbf{d}\| = 1$ is the view direction. To model the dynamic component of a scene, we adapt HyperNeRF [40], which accurately captures scenes with *non-rigid* motion as well as *topological changes* by introducing additional degree of freedom and network capacity. For convenience, we denote it as a neural function $\mathcal{F}^D$:

$$\left.\begin{array}{l}\sigma^D(\mathbf{x}, \boldsymbol{\tau}_i) \in \mathbb{R} \\ \mathbf{c}^D(\mathbf{x}, \mathbf{d}, \boldsymbol{\tau}_i) \in \mathbb{R}^3\end{array}\right\} = \mathcal{F}^D(\mathbf{x}, \mathbf{d}, \boldsymbol{\tau}_i) \tag{2}$$

where $\boldsymbol{\tau}_i \in \mathbb{R}^m$ is the per-frame time latent code. Given a camera ray $\mathbf{r} = \mathbf{o} + t\mathbf{d}$ originating from $\mathbf{o}$ and with direction $\mathbf{d}$, the two models are then composited to calculate the color $\hat{C}$ of the camera ray by integrating the radiance according to volumetric rendering within a pre-defined depth range $[t_n, t_f]$:

$$\hat{C}(\mathbf{r}, \boldsymbol{\tau}_i) = \int_{t_n}^{t_f} T(t) \left(\sigma^S(t) \cdot \mathbf{c}^S(t) + \sigma^D(t, \boldsymbol{\tau}_i) \cdot \mathbf{c}^D(t, \boldsymbol{\tau}_i)\right) dt \tag{3}$$

$$T(t) = exp\left(-\int_{t_n}^{t} (\sigma^S(s) + \sigma^D(s, \boldsymbol{\tau}_i))\, ds\right) \tag{4}$$

where we simplify our notation as $\sigma(t) \equiv \sigma(\mathbf{r}(t))$ and $\mathbf{c}(t) \equiv \mathbf{c}(\mathbf{r}(t), \mathbf{d})$. Note that, with such an additive decomposition, samples from either fields are capable of terminating the camera ray and occluding the other.

3.2 Supervision Losses

To find the parameters of the static (Eq. 1) and dynamic (Eq. 2) NeRF networks, a photometric loss is applied to ensure that the output image sequences of the composite NeRF (Eq. 3) align with the input video frames:

$$\mathcal{L}_p(\mathbf{r}, \boldsymbol{\tau}_i) = \|\hat{C}(\mathbf{r}, \boldsymbol{\tau}_i) - C(\mathbf{r}, \boldsymbol{\tau}_i)\|_2^2 \tag{5}$$

where $C(\mathbf{r}, \boldsymbol{\tau}_i)$ indicates the true color of camera ray $\mathbf{r}$ obtained from the i-th input video frame. However, note the dynamic component can naturally take over the static counterpart by incorrectly assigning occupancy of static objects to dynamic NeRF, and the photometric loss alone also does not guarantee a correct separation. In what follows, we design a collection of regularizers that promote such decoupling in a self-supervised fashion.

Dynamic vs. Static Factorization As physical objects cannot co-exist at the same spatial location, a physically realistic solution should have any position in space *either* occupied by a the static scene or by a dynamic object, but *not both*. To enforce this behavior we denote the spatial ratio of dynamic vs. static density as:

$$w(\mathbf{x}, \boldsymbol{\tau}_i) = \frac{\sigma^D(\mathbf{x}, \boldsymbol{\tau}_i)}{\sigma^D(\mathbf{x}, \boldsymbol{\tau}_i) + \sigma^S(\mathbf{x})} \in [0, 1] \tag{6}$$

and then penalize its deviation from a categorical $\{0, 1\}$ distribution via a binary entropy loss [68]:

$$\mathcal{L}_b(\mathbf{r}, \boldsymbol{\tau}_i) = \int_{t_n}^{t_f} H_b(w(\mathbf{r}(t), \boldsymbol{\tau}_i))\, dt \tag{7}$$

$$H_b(x) = -(x \cdot log(x) + (1 - x) \cdot log(1 - x)) \tag{8}$$

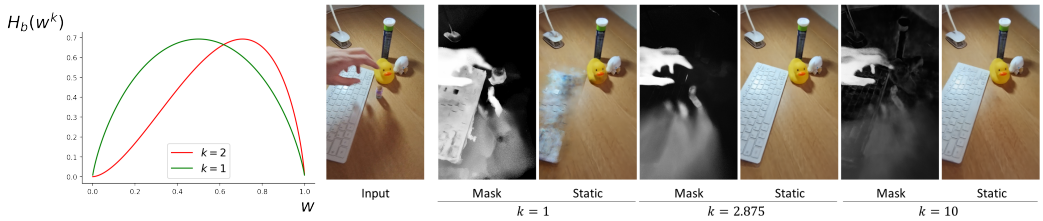

Figure 2: **Skewed entropy –** (left) the skewed ($k > 1$) and classical ($k = 1$) entropy losses. A skewed entropy encourages a wider range of w to decrease and has a larger gradient on values around 0.5, but its gradient vanishes when w approaches 0. (right) The decoupled alpha masks and static components when original, properly-skewed and over-skewed binary entropy losses are applied.

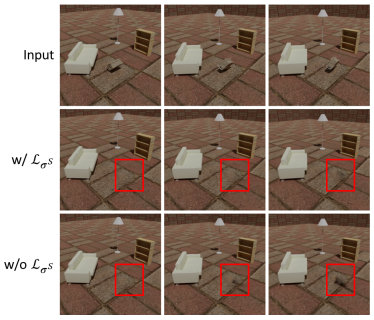

Figure 3: **Static regularization –** By encouraging a more concentrated density distribution along each camera ray in static component, the recovered background contains less view-dependent artifacts.

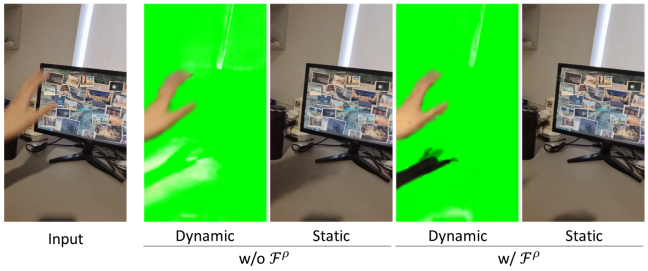

Figure 4: **Shadow ambiguity –** When shadows occur frequently in the input data, the average shadow gets integrated in the static component, and the dynamic component incorrectly learns the differential with respect to this average and appears as a brighter surface. This can be avoided by a more direct modeling of shadow effects as a dynamic darkening of static regions (i.e. the shadow field).

However, due to the strong expressive power of the dynamic networks (Eq. 2), optimizing the loss (Eq. 7) leads to the technique modeling parts of the scene as dynamic, regardless of whether they are dynamic or static; see Figure 2 (right). To overcome this issue, we propose a *skewed* entropy loss to bias our loss to *slightly favor* static explanations of the scene with skewness hyper-parameter k, that, as illustrated in Figure 2 (left, k>1), attains the desired behavior:

$$\mathcal{L}_s(\mathbf{r}, \boldsymbol{\tau}_i) = \int_{t_n}^{t_f} H_b(w(\mathbf{r}(t), \boldsymbol{\tau}_i)^k)\, dt \tag{9}$$

Ray Regularization Choosing a large value of skewness k causes the appearance of fuzzy floaters (low-density particles) in the static portion of the scene; see Figure 2 (right, k=10). As it can be intuitively understood from Figure 2 (left), this is caused by the small gradients of $H_b(x^k)$ as x approaches zero. To mitigate this effect, and reduce fuzziness in the reconstruction, we penalize the maximum of w along each camera ray:

$$\mathcal{L}_r(\mathbf{r}, \boldsymbol{\tau}_i) = \max_{t \in [t_n, t_f]} w(\mathbf{r}(t), \boldsymbol{\tau}_i) \tag{10}$$

Such loss can be intuitively interpreted as constraining the dynamic component to occupy as few pixels as possible while keeping minimal impact on the overall loss for all samples. Note that $\mathcal{L}_r$ only removes density floaters that sit along camera rays that *do not intersect* with any dynamic objects.

Static Regularization We empirically found that static component may abuse the camera pose as the hint for the current time frame and learn dynamic effects as sparse clouds that lead to high-frequency appearance changes; see Figure 3. The ambiguity comes from the fact that we are using monocular casual videos where the camera almost never visits the exact same position twice during the capture. That is, there exists a one-to-one mapping between camera pose and time variable. We

solve this issue by imposing a prior on the distribution of density along a ray, penalizing density distributions that would cause cloud-like artifacts [22, 45]:

$$\mathcal{L}_{\sigma^S}(\mathbf{r}) = -\int_{t_n}^{t_f} p(t) \cdot log\, p(t)\, dt \quad \text{where} \quad p(t) = \frac{\sigma^S(\mathbf{r}(t))}{\int_{t_n}^{t_f} \sigma^S(\mathbf{r}(s))\, ds} \tag{11}$$

3.3 Shadow Fields

Neural radiance fields cannot faithfully model standalone shadows without significant changes to its architecture necessary to modeling materials and illumination; see NeRFactor [71]. In simple cases where shadows of the dynamic objects move rapidly, they could alternatively be learned by the dynamic radiance field as semi-transparent layers on top of the static surface. However, this tends to fail for shadows that do not move much, or that are highly correlated with the camera motion. As shadows are texture-less, understanding their movement is ambiguous, and representing them as a semi-transparent layer causes difficulties in the optimization; see Figure 4. To overcome this issue, and under the assumption of a direct illumination model (i.e. negligible global illumination effects), we make the observation that a cast shadow can be represented as a *pointwise reduction* in the radiance of the the static scene, and incorporate this within Eq. 3 as:

$$\hat{C}(\mathbf{r}, \boldsymbol{\tau}_i) = \int_{t_n}^{t_f} T(t)((1 - \underbrace{\rho(\mathbf{r}(t), \boldsymbol{\tau}_i)}) \cdot \sigma^S(t) \cdot \mathbf{c}^S(t) + \sigma^D(t, \boldsymbol{\tau}_i) \cdot \mathbf{c}^D(t, \boldsymbol{\tau}_i))\, dt \tag{12}$$

$$\rho(\mathbf{x}, \boldsymbol{\tau}_i) \in [0, 1] = \mathcal{F}^\rho(\mathbf{x}, \boldsymbol{\tau}_i) \tag{13}$$

where $\rho(\mathbf{x}, \boldsymbol{\tau}_i)$ is a *shadow ratio* that scales-down the radiance of the static scene to incorporate the shadow. To avoid the shadow-ratio from over-explaining dark regions of the scene, we penalize its average squared magnitude along a ray:

$$\mathcal{L}_\rho(\mathbf{r}, \boldsymbol{\tau}_i) = \tfrac{1}{t_f - t_n} \int_{t_n}^{t_f} \rho(\mathbf{r}(t), \boldsymbol{\tau}_i)^2\, dt \tag{14}$$

Finally, note that shadows cast from dynamic objects onto other dynamic objects are *already* expressed from the radiance term of the dynamic branch, and do not need explicit modeling.

4 Experiments

4.1 Implementation details

Our method is easily reproducible, as we intend to release code and datasets upon publication to facilitate future research. We adopt the HyperNeRF [40] architecture as the dynamic component, which has a NeRF MLP network of 8 layers, each with 256 channels, and our static NeRF component has the same architecture. Similar to NeRF [32], we apply a hierarchical volume sampling with 64 coarse and 64 fine samples. The optimization takes $100k$ iterations with batch size 1024 and an exponentially decayed learning rate from 10^{-3} to 10^{-5}. This training procedure spans approximately two hours on four NVIDIA A100-SXM-80GB GPUs. The overall loss of our method is:

$$\mathcal{L}(\mathbf{r}, \boldsymbol{\tau}_i) = \mathcal{L}_p(\mathbf{r}, \boldsymbol{\tau}_i) + \lambda_s \mathcal{L}_s(\mathbf{r}, \boldsymbol{\tau}_i) + \lambda_r \mathcal{L}_r(\mathbf{r}, \boldsymbol{\tau}_i) + \lambda_{\sigma^S} \mathcal{L}_{\sigma^S}(\mathbf{r}) + \lambda_\rho \mathcal{L}_\rho(\mathbf{r}, \boldsymbol{\tau}_i) \tag{15}$$

where λ_s, λ_r, λ_{σ^S}, λ_ρ are the weights of the regularization terms respectively. For scenes with a mixture of dynamic objects and shadows, we apply shadow decay and set λ_ρ=0.1. We set λ_ρ=0.001 for scenes featuring view-correlated dynamic shadows only. We experimentally found that the optimal choice of the hyperparameters, especially λ_b, λ_r and the skewness k, are strongly influenced by the level of object motion, camera motion, and video length. Therefore, we performed a grid search on our synthetic and held-out real-world scenes, and some scenes from DAVIS [42], to establish a set of hyperparameters applicable to a variety of scenarios; details about hyperparameters can be found in the supplementary. We do not apply shadow field for evaluations on our synthetic scenes, as we empirically found that shadow field is not needed to learn correct shadows. We also disable the view direction input for synthetic scenes as they do not contain strong view-dependent effects.

	Pick2	Duck	Balloon	Water	Cookie	Mean
NeuralDiff [54]	**.208**	.222	.167	.172	.159	**.186**
HN [40]	.486	.253	.187	.361	.162	.290
Ours	.253	**.214**	**.153**	**.153**	**.156**	**.186**

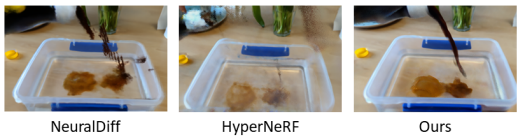

Figure 5: **Novel view synthesis** – LPIPS↓

	Car	Cars	Bag	Chairs	Pillow	Mean
MG [64]	.603	.363	.629	.484	.044	.424
NeuralDiff [54]	.806	.508	.080	.368	.097	.372
Ours	**.848**	**.790**	**.703**	**.551**	**.693**	**.717**

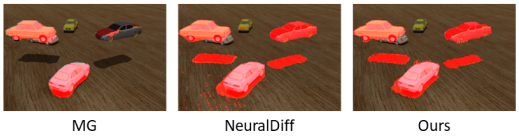

Figure 6: **Video segmentation** – $\mathcal{J}\uparrow$

4.2 Evaluation

We demonstrate the performance of our method both quantitatively and qualitatively on three tasks. We focus our attention to our main objective of decoupling and removing dynamic objects, including their shadows, with a 3D reconstruction of the static environment. We only include a summary of our results for novel view synthesis (in Figure 5) and video segmentation (in Figure 6), and refer the reader to the Supplementary E and D for a more in-depth discussion. We strongly encourage the readers to watch videos on the project page (https://d2nerf.github.io/) to better appreciate our results.

4.3 Datasets

In addition to the data obtained from HyperNeRF [40] and Nerfies [39], we acquire more complex datasets in the real-world, as well as design a synthetic dataset to enable quantitative comparisons.

Synthetic dataset We generate a synthetic dataset with ground-truth masks for moving objects and their shadows with Kubric [16]. This dataset consists of five scenes containing one or multiple dynamic objects from ShapeNet [5] with rigid or non-rigid motion, and the corresponding Kubric worker script is provided in our supplementary material. We move the virtual camera over 10 keyframes randomly sampled from azimuth $[2, 2 + \pi/4]$ and altitude $[1, 1.2]$ to generate a 200-frame video sequence for training. We also rotate the virtual camera around the center of all keyframes to generate 100 validation views with only the static background being visible. We additionally generate masks for both the dynamic objects and their shadows, allowing us to quantitatively study the performance of our algorithm. Note that shadows are usually absent in existing moving-objects segmentation benchmarks.

Real-world dataset We also capture ten video sequences of real scenes to showcase our performance. Compared to HyperNeRF's, our dataset contains more challenging scenarios with rapid motion and non-trivial dynamic shadows. Note however that we *cannot* perform quantitative analysis for these datasets due to the absence of ground-truth views of a static scene or ground-truth masks. Five of real scenes are captured with a similar setting as Nerfies, where we use a dual-hold phone rig and synchronize the capture based on audio. We use the images captured by one of the two phones as validation views for novel-view synthesis, which are discussed in the Supplementary E. To demonstrate the ability of fully self-supervised scene decoupling, we *do not* apply any masks when registering real-world images using COLMAP [49].

4.4 Scene Decoupling – Table 1, Figure 7

We report the evaluation of our method on its ability to decouple dynamic objects and their shadows, while recovering the static background. We evaluate our performance against NeRF-W [31], NSFF [26] and NeuralDiff [54]. For NeRF-W, we used only transient embedding and disabled the appearance embedding that models variable lighting for evaluation on synthetic scenes, as they have constant illumination. For NSFF, we disabled the hard mining initialization via 2D masks input. For NeuralDiff, we disabled the actor component (as our input videos are not egocentric) and only use the transient component.

	Car			Cars			Bag			Chairs			Pillow			Mean		
	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑	LPIPS↓	MS-SSIM↑	PSNR↑
NeRF-W [31]	.218	.814	24.23	.243	.873	24.51	.139	.791	20.65	.150	.681	23.77	.088	.935	28.24	.167	.819	24.28
NSFF [26]	.200	.806	24.90	.620	.376	10.29	.108	.892	25.62	.682	.284	12.82	.782	.343	4.55	.478	.540	15.64
NeuralDiff [54]	.065	.952	31.89	.098	.921	25.93	.117	.910	29.02	.112	**.722**	24.42	.565	.652	20.09	.191	.831	26.27
Ours	**.062**	**.975**	**34.27**	**.090**	**.953**	**26.27**	**.076**	**.979**	**34.14**	**.095**	.707	**24.63**	**.076**	**.979**	**36.58**	**.080**	**.919**	**31.18**

Table 1: **Scene decoupling (quantitative) –** We train on each scene with 200 frames, decouple the dynamic objects and shadows, and render the static component from 100 novel views to compare with ground truth. Note these are computed on the *synthetic* dataset, for which ground truth is available.

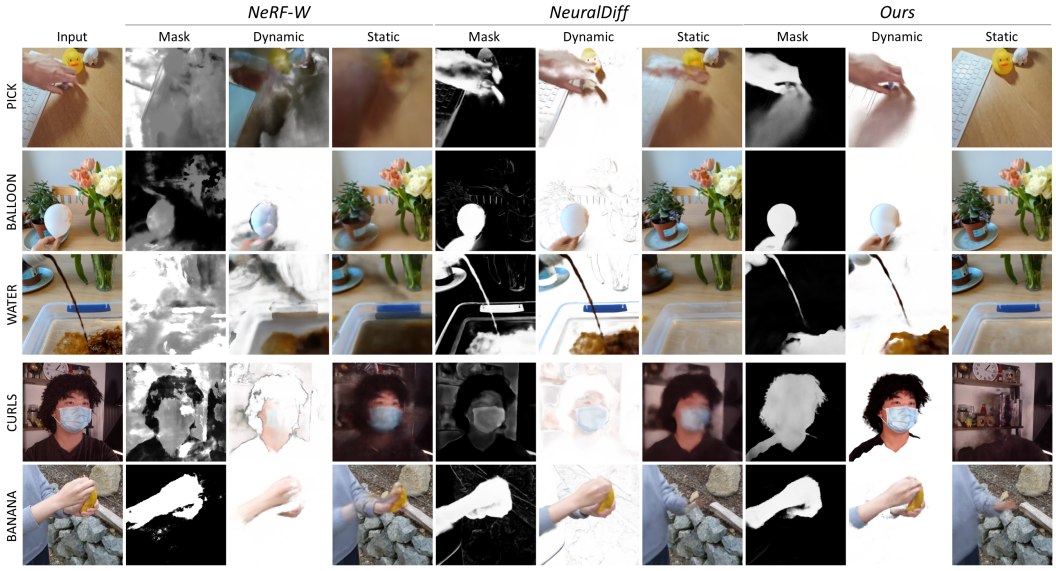

Figure 7: **Scene decoupling (qualitative)** – We visualize results on (top) our new real scenes, and on (bottom) scenes from HyperNeRF [40] and Nerfies [39]. To better illustrate the decoupled object and shadow, we render the dynamic component with a white background. Note that since our method only decouples dynamic targets, it does not include parts of objects that remain still throughout the capture, such as the body in "curls" and "banana" scenes.

In the evaluation, we used each method to synthesize the static background from multiple validation views with the moving objects and their shadows removed; see Figure 7 for qualitative results on real data, as well as Supplementary C for qualitative results on synthetic data. We compare the results with the ground truth on the synthetic data and report LPIPS [70], Multi-Scale SSIM [57], and PSNR as the metrics for novel view synthesis of the decoupled static background; see Table 1. Note that although NSFF similarly learns separate dynamic and static NeRFs, it is not designed to maximize the performance of scene decoupling and does not incorporate regularization on the dynamic NeRF. Therefore, it is extremely unstable and can fail completely by learning everything as dynamic, leaving static component as an empty scene.

4.5 Ablations – Table 2, Figure 8, Figure 9

We quantitatively ablate our method on our synthetic dataset; see Table 2: where "skew" means skewness is applied in the binary entropy regularization, and "$\mathcal{L}_r$" stands for the density ratio ray regularization. We also qualitatively ablate it on a real scene; see Figure 8. We also qualitatively illustrate the ablation on shadow field network, which is necessary for decoupling shadows with

skew	$\mathcal{L}_r$	Car			Cars			Bag			Chairs			Pillow			Mean		
		LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑
☐	☐	.214	.834	26.26	.119	.943	26.10	.254	.666	19.96	.104	.698	24.42	.385	.671	14.24	.215	.762	22.20
☑	☐	.182	.865	25.89	.260	.803	22.47	.189	.893	28.38	.107	.693	24.44	.311	.770	15.27	.210	.805	23.29
☐	☑	.067	.973	34.06	.104	.948	26.19	.091	.955	31.55	.151	.653	22.92	.118	.940	28.17	.106	.894	28.58
☑	☑	**.062**	**.975**	**34.27**	**.090**	**.953**	**26.27**	**.076**	**.979**	**34.14**	**.095**	.707	**24.63**	**.076**	**.979**	**36.58**	**.080**	**.919**	**31.18**

Table 2: **Ablations (quantitative)** – We train on each scene with 200 frames, decouple the dynamic objects and shadows, and render the static component from 100 novel views for metric evaluations.

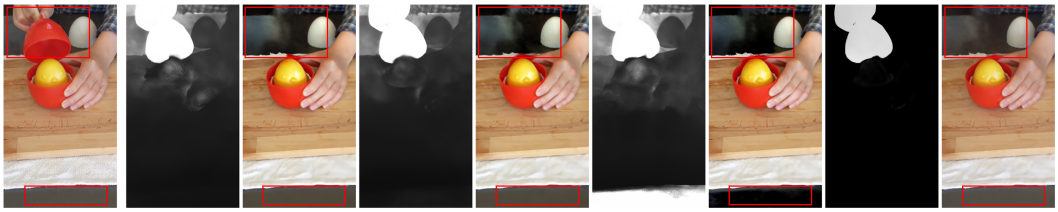

Figure 8: **Ablations (qualitative) –** For scenes with slow motion or strong view-dependent reflectance, $\mathcal{L}_r$ is used together with the skewed entropy to prevent the dynamic component from incorrectly decoupling the scene. In the scene above this appears as a slightly darkened color on the table.

Figure 9: **Ablations (shadows) –** Our method is able to remove large area of shadows, even if they are strongly correlated with the view direction (e.g., shadow cast by camera or the photographer). Note that the appearance embedding from NeRF-W [31] is not sufficient to remove shadow that is present throughout the capture; see the our website for additional qualitative results.

large area, slow or repetitive motion, or shadows that are highly correlated with the camera view; see Figure 9.

5 Conclusions

We presented D^{2}NeRF, a method for self-supervised 3D scene decoupling and reconstruction from casual monocular videos. Our method decouples occluders and correlated shadows, recovers clean background representations, and enables high quality novel views synthesis. Our novel skewed entropy regularizer is critical to separate dynamic from static components of the scene, while our shadow-field allows for the removal of dynamic shadows without having to explicitly model the interaction between light and geometry. We demonstrate superior results for multiple tasks on existing datasets, as well as on two datasets that we introduce alongside our technique.

Limitations Similar to many NeRF-based methods, our approach relies on accurate camera registration to achieve success decoupling and reconstruction. Our approach also suffers from high frequency view-dependent radiance change, such as those caused by the presence of reflective surface within the scene. Due to the monocular moving camera setting, those effects could be misinterpreted as dynamic effects, resulting in incorrect decoupling. Removing texture-less target that repeatedly moves within a very small range is also difficult, as the motion clues are extremely ambiguous in this case; see Figure 10. We also note that the skewed binary entropy is not the only loss that suits the purpose of separating two NeRF components while suppressing one of them. Many losses such as the beta distribution from [28] and Laplacian distribution from [45] can behave similarly by

Figure 10: **Limitations –** Because the hand stays around the same position for the majority of the time during the video, our method is unable to fully decouple and remove the texture-less shadow.

incorporating an exponential coefficient. The main difference between our skewed entropy loss and beta or Laplacian distributions is that the former has a vanishing gradient when approaching 0, while the gradients of the latter become infinity. Choosing an appropriate form of loss could reduce the complexity of hyperparameters without affecting the decoupling accuracy.

Acknowledgements

This work was supported by a UKRI Future Leaders Fellowship [grant number G104084]. We thank Ziwen Cui and Ningding Wang for their help with data collection.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds, 2018.

[2] Benjamin Attal, Jia-Bin Huang, Michael Zollhoefer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks, 2021.

[3] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling facial geometry using compositional vaes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018.

[4] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos, 2016.

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[6] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, 2016.

[7] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free, 2021.

[9] Dawa Derksen and Dario Izzo. Shadow neural radiance fields for multi-view satellite photogrammetry, 2021.

[10] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image, 2016.

[12] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[13] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps, 2021.

[14] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape, 2019.

[15] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions, 2019.

[16] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[18] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis, 2021.

[19] Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. Editable free-viewpoint video using a layered neural representation. In *ACM SIGGRAPH*, 2021.

[20] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition, 2021.

[21] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[22] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. *arXiv.org*, 2021.

[23] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[25] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video, 2021.

[26] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[27] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering, 2020.

[28] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019.

[29] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering, 2021.

[30] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimatte: Associating objects and their effects in video. In *CVPR*, 2021.

[31] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[32] 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, 2020.

[33] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International Conference on Machine Learning*, pages 7220–7229. PMLR, 2020.

[34] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum*, 40(4), 2021.

[35] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, 2021.

[36] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2856–2865, June 2021.

[37] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *2013 IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[39] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[40] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

[41] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34, 2021.

[42] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[43] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.

[44] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[45] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[46] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps, 2021.

[47] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields, 2021.

[48] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

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

[50] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021.

[51] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations, 2020.

[52] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering, 2021.

[53] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction, 2021.

[54] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. NeuralDiff: Segmenting 3D objects that move in egocentric videos. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2021.

[55] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images, 2018.

[56] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 06 2014.

[57] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

[58] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[59] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes, 2015.

[60] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9421–9431, 2021.

[61] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Citynerf: Building nerf at city scale, 2021.

[62] Christopher Xie, Yu Xiang, Zaid Harchaoui, and Dieter Fox. Object discovery in videos as foreground motion clustering, 2018.

[63] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 492–502. Curran Associates, Inc., 2019.

[64] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[65] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks, 2021.

[66] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields, 2021.

[67] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2020.

[68] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13144–13152, 2021.

[69] Jason Y. Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *Conference on Neural Information Processing Systems*, 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 *CVPR*, 2018.

[71] 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.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary A
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.2
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Repeated evaluations not applicable due to the limitation in amount of computational resources required for training
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.2
   (b) Did you mention the license of the assets? [Yes] See Section 4.2
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Supplementary A
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Consent obtained from license
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Only personally identifiable information would be the "curls" dataset, which is obtained from Nerfies [39] and used under their license.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

D^{2}NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video

Supplementary Material

A Code and Data

All coda and data, as well as additional video results can be found on our project page: d2nerf.github.io. We also include a static copy of the code and website as zip files in the supplementary.

B Hyperparameters

As we do not have large TPUs available for training, we incorporate a light-weighted HyperNeRF [40] as the dynamic component to reduce training time. Compared to the original hyperparameters described in the paper, we reduce the number of samples per ray (64 vs. 128 in [40]), batch size (1024 vs. 6144 in [40]), and number of iterations (100k vs. 250k in [40]). We also do not apply the background regularization, as it requires a set of known background 3D points, which rely on accurate dynamic masks during COLMAP registration.

We experimentally established a set of hyperparameters applicable for a variety of scenes. In total, we used five sets of hyperparameters for the evaluation on the real-world dataset, and four on the synthetic dataset. To ensure the various scenes are fully separated into different components, we increase λ_s during training, where $\rightarrow$ indicates the value is linearly increased and $\Rightarrow$ indicates it is exponentially increased. - entry for λ_ρ indicates that the shadow field is not applied.

Table 3: **Hyperparameters –** Row 1-4 specify hyperparameters for real-world scenes containing a mixture of dynamic objects and shadows, whereas row 5 is suitable for real-world scenes with dynamic shadows only. Row 6-9 contain hyperparameters for synthetic scenes.

	k	λ_s	λ_r	λ_{σ^S}	λ_ρ	Dataset
1	1.75	$1e{-4} \rightarrow 1e{-2}$	$1e{-3}$	0	$1e{-1}$	Broom, Chicken, Curls, Pick2, Duck, Balloon, Cookie, Hand, Shark, Toy
2	3	$1e{-4} \Rightarrow 1$	$1e{-3}$	0	$1e{-1}$	Banana (novel view)
3	2.75	$1e{-5} \Rightarrow 1$	$1e{-3}$	0	-	Water, Banana (decoupling)
4	2.875	$5e{-4} \Rightarrow 1$	0	0	-	Pick
5	1.5	$1e{-3} \Rightarrow 1$	$1e{-1}$	$1e{-2}$	$1e{-2}$	Camera Shadow, Shadow Car
6	2	$1e{-5} \Rightarrow 1$	$1e{-5}$	$1e{-4}$	-	Cars, Soft
7	1.75	$1e{-5} \Rightarrow 0.1$	$1e{-4}$	0	-	Car
8	2.5	$1e{-5} \Rightarrow 1$	$1e{-5}$	$1e{-3}$	-	Chairs
9	2.75	$1e{-4} \Rightarrow 1$	$2e{-4}$	$1e{-4}$	-	Bag

C Scene Decoupling – Figure 11, Figure 13, Figure 14

We demonstrate additional qualitative results on scene decoupling task on both real-world and synthetic scenes; see Figure 11, 12, 13, and 14.

D Video Segmentation – Table 4, Figure 15

As our method learns a density distribution of the dynamic objects in the scene, we can further produce an alpha mask of the objects. We therefore also evaluate the correctness of object segmentation at the image level. Existing benchmarks on video segmentation [42, 56, 24] either contain too few video frames for reliable SfM reconstruction, or do not have correct ground truth masks for all of the dynamic objects and effects in the scene. Similarly, the dataset from NeuralDiff [54] focuses on egocentric videos and the over-exaggerated difference between frames in the videos is not suitable for HyperNeRF which we use as the dynamic component. Hence, to highlight the ability of our method to decouple dynamic objects and shadows from video sequences with large viewpoint shifts,

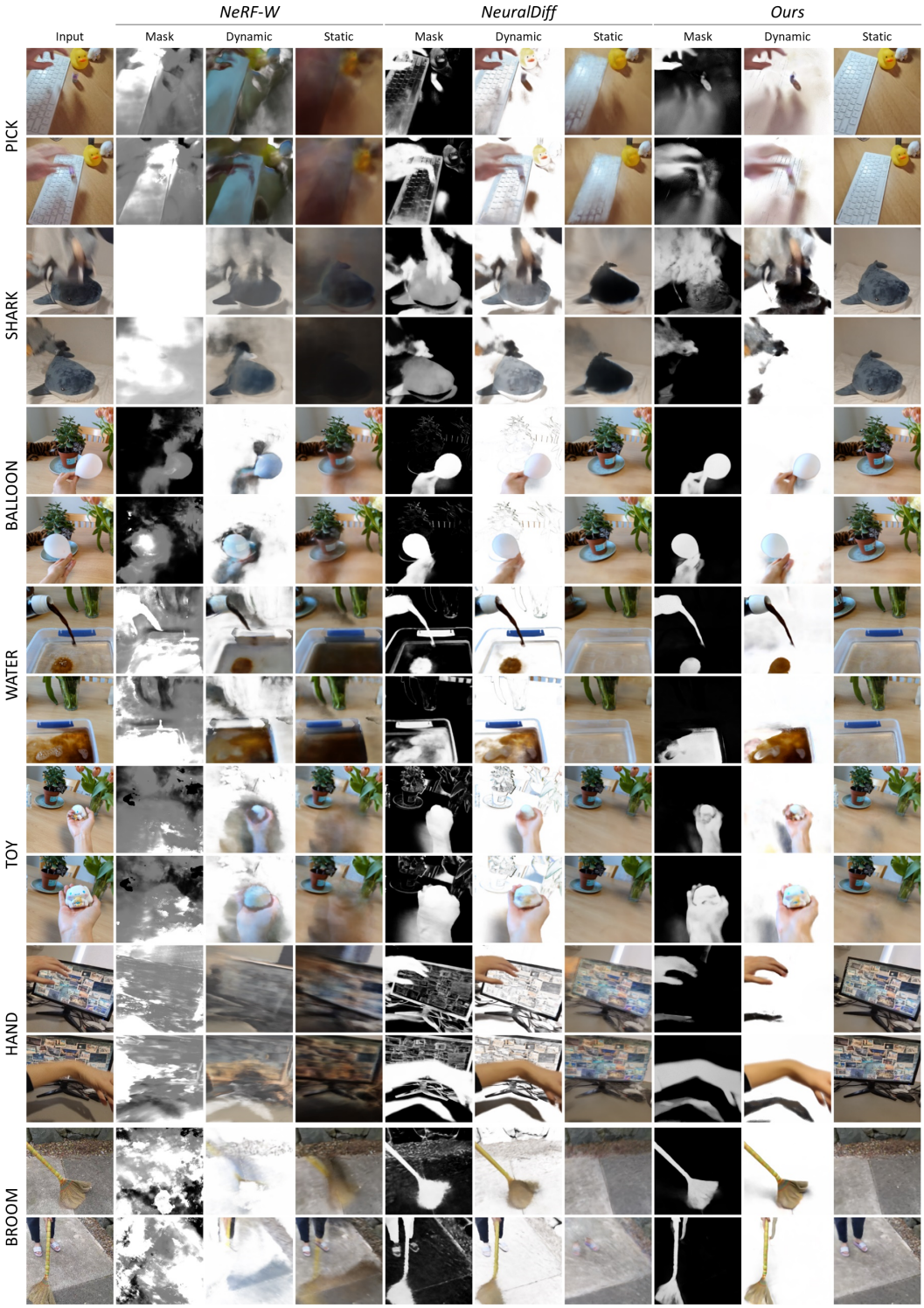

Figure 11: **Additional results on real-world scene decoupling and segmentation –** Similar to Figure 7, we show the dynamic alpha mask, dynamic part and static background respectively. Last two rows at bottom show the "broom" scene from HyperNeRF [40].

we evaluate on our synthetic dataset. In addition to the NeRF-like baselines, we also compare with Motion Grouping (MG) [64], a motion-based 2D image segmentation method. We fine-tuned the

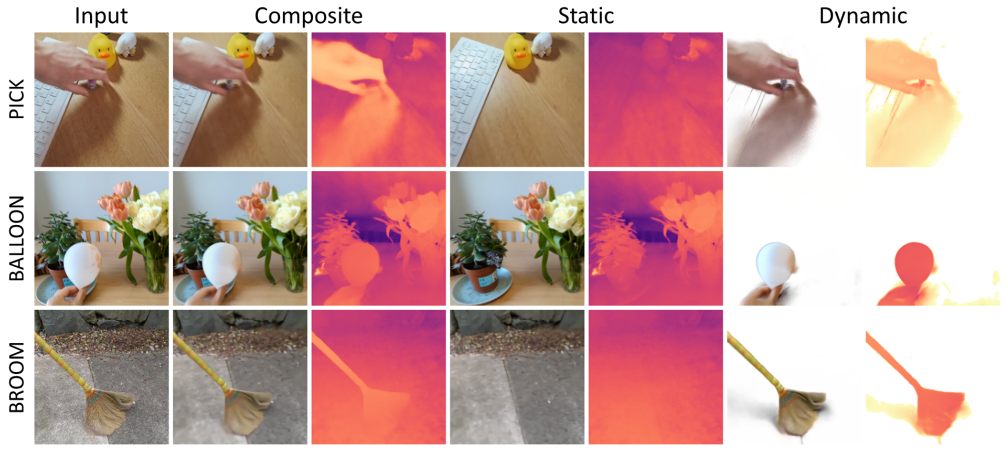

Figure 12: **Decoupled depth –** We show the disentangled geometry as depth maps for dynamic and static components respectively.

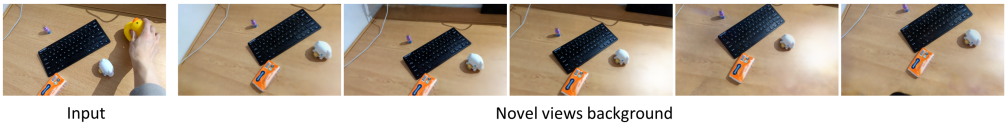

Figure 13: **Background novel view –** Our method learns the decoupled static background and can render it from unseen views, with the dynamic occluders cleanly removed.

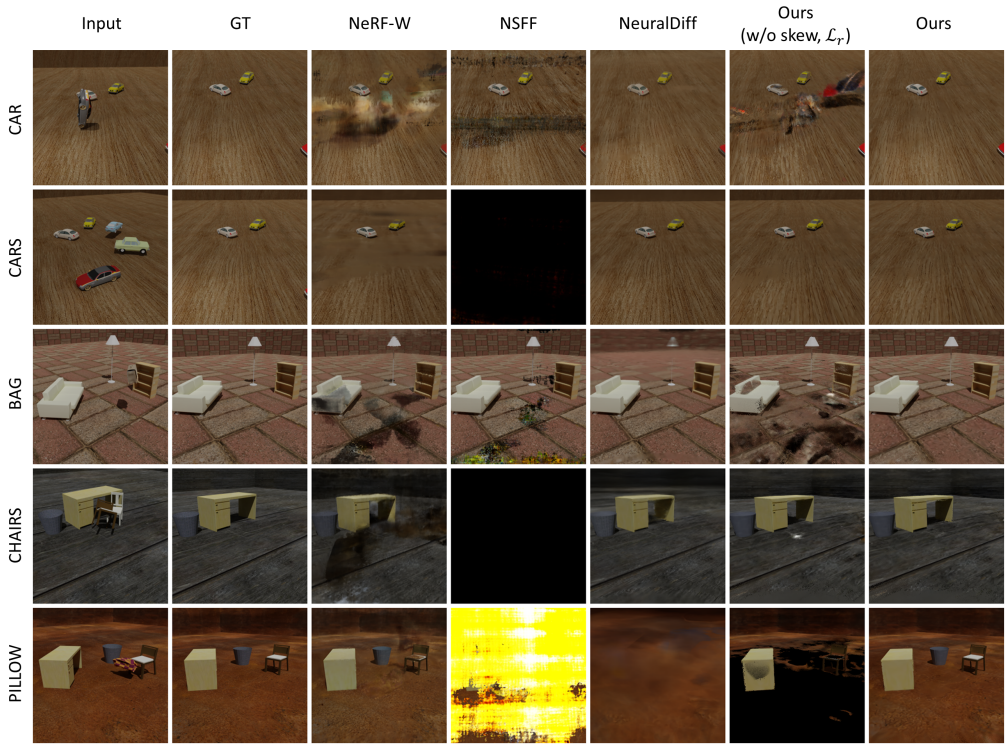

Figure 14: **Scene decoupling and novel view background recovery on synthetic scenes –** We train each method with videos containing various dynamic occluders and shadows, decouple the scene and render the background from unseen views to compare with the ground truth. Quantitative evaluation on corresponding scenes can be found at Table 1.

	Car		Cars		Bag		Chairs		Pillow		Mean	
	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$
MG [64]	.603	.743	.363	.474	.629	.738	.484	.613	.044	.080	.424	.529
NeRF-W [31]	.072	.132	.098	.162	.027	.052	.154	.254	.194	.314	.109	.183
NSFF [26]	.083	.152	.058	.104	.102	.182	.046	.087	.104	.188	.079	.143
NeuralDiff [54]	.806	.891	.508	.578	.080	.144	.368	.513	.097	.177	.372	.461
Ours (w/o skew)	.814	.896	**.807**	**.883**	.342	.483	.114	.198	.347	.511	.485	.594
Ours (w/o L_r)	.076	.139	.174	.261	.048	.089	.237	.367	.040	.078	.115	.187
Ours (w/o skew, L_r)	.077	.142	.376	.464	.043	.081	.315	.453	.027	.053	.168	.238
Ours	**.848**	**.917**	.790	.874	**.703**	**.818**	**.551**	**.687**	**.693**	**.818**	**.717**	**.822**

Table 4: **Video segmentation –** We report Jaccard index $\mathcal{J}$ and boundary measure $\mathcal{F}$ on training views. Our method performs well on "car" and "cars" scenes without the use of skewed entropy because the background is clearly distinguishable from the moving object.

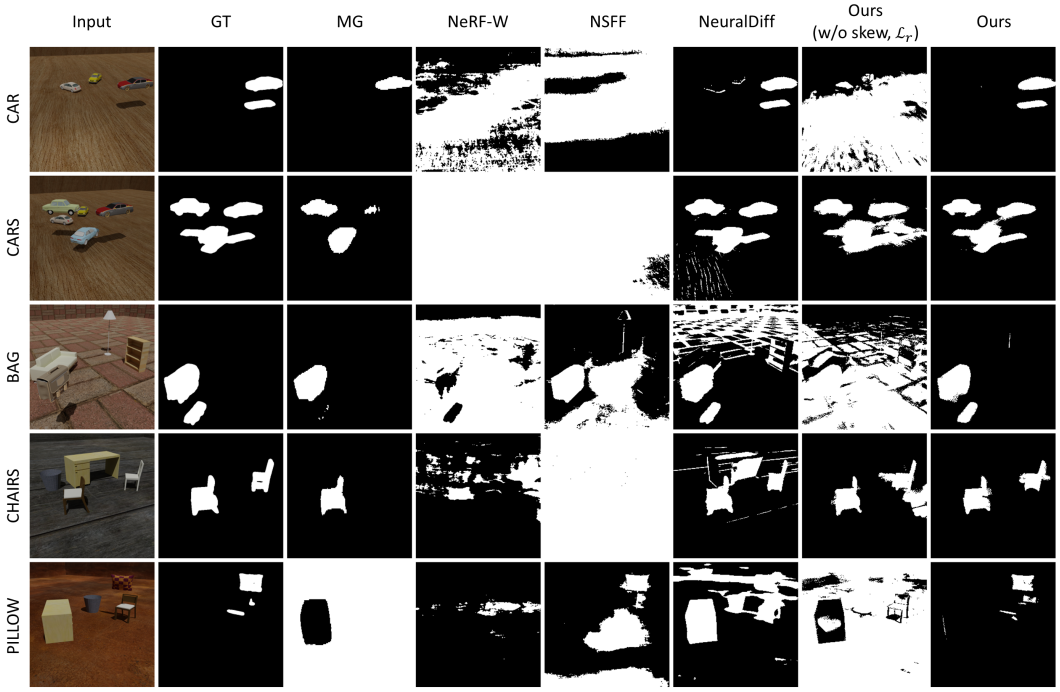

Figure 15: **Video segmentation (qualitative) –** MG fails to identify the moving pillow and segments everything out except for the table due to its Hungarian setting. NeRF-W learns the transient component with severe cloud-like effects. While our method achieves best segmentation in all the scenes.

pre-trained MG model on each scene for 5k iterations. For other NeRF-based methods, we used the same settings as in Section 4.4 and produced the alpha masks as the normalized radiance weights of the time-varying component, and then applied a threshold of 0.1 to obtain the binary masks. See Table 4, Figure 15 for the results.

E Novel View Synthesis – Table 5, Figure 16

Although the aim of our method is not to improve the quality of time-varying scene reconstruction, as a by-product, we find that by introducing a static component to fully utilize the network capacity, our method achieves more robust reconstruction for both the dynamic objects and background. We therefore also evaluate our method on the ability to synthesize the whole scene from novel views. We compare several approaches for dynamic scene reconstruction, including NeuralDiff [54] and a baseline version of HyperNeRF [40], which has the same architecture as our dynamic component. Since our method uses an additional static component and naturally has more network parameters and capacity, we also compare with a fair version of HyperNeRF with roughly the same number of total parameters by extending the NeRF MLP width to 375. Unlike novel view experiments in [40], we do not interleave between two cameras, but use only the right camera as training view and the left

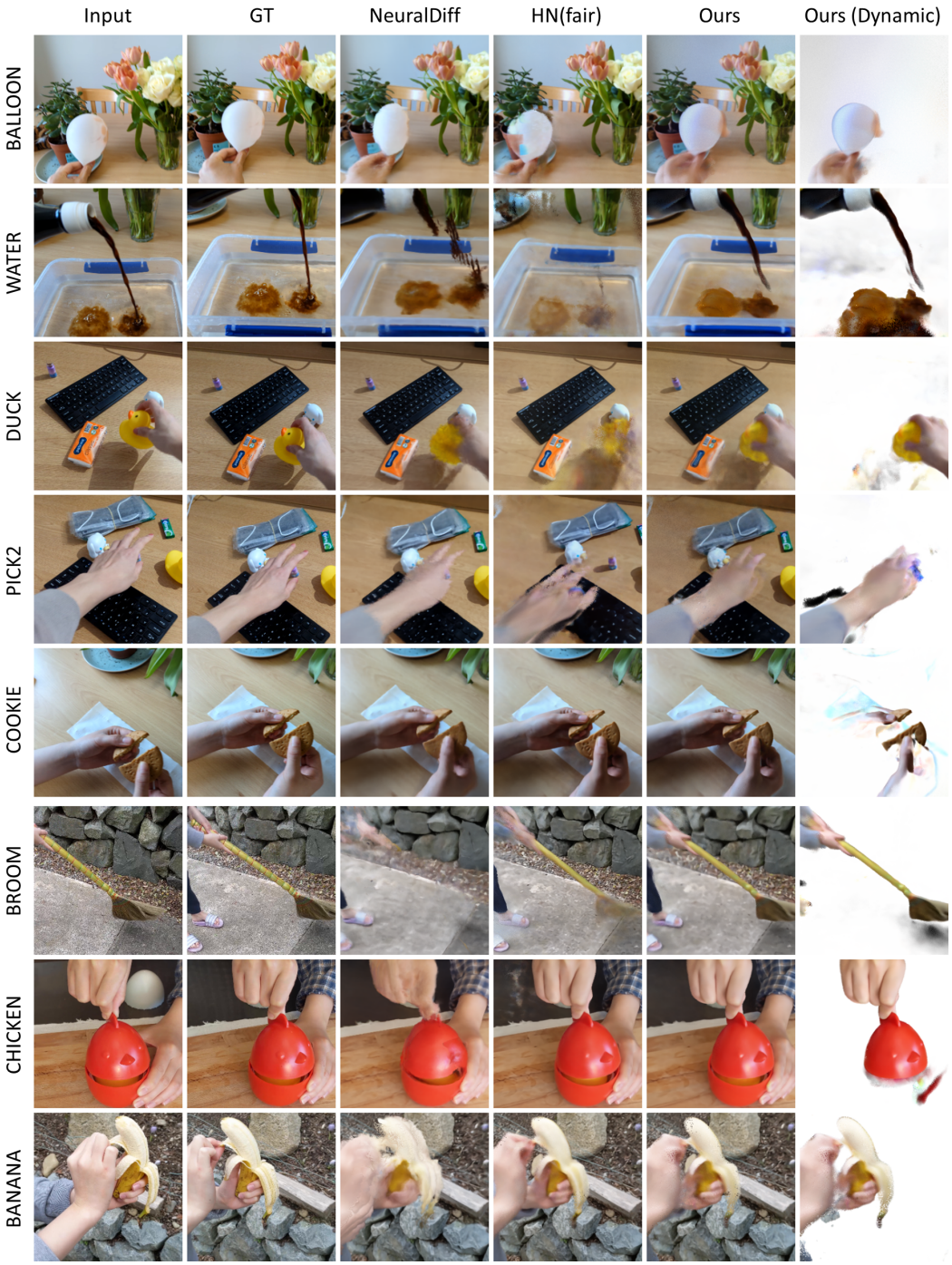

Figure 16: **Novel view synthesis (qualitative) –** For challenging scenes such as "water" and "duck" where the dynamic object moves rapidly or training/validation views differ largely, HyperNeRF [40] fails to reconstruct a reasonable shape for the dynamic object, while ours might potentially predict a shifted object pose, but can still render the view with high fidelity. We additionally show the decoupled dynamic object from our method. The quality is slightly degraded compared to the decoupling results in Figure 7 and 11 as we are rendering from the more challenging novel views.

camera as validation view. This presents greater challenges to all the methods. See Table 5, Figure 16 for the results.

	Pick2			Duck			Balloon			Water			Cookie		
	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑
NeuralDiff	**.208**	**.853**	**21.81**	.222	**.862**	21.92	.167	.836	20.13	.172	.811	18.36	.159	.875	**20.53**
HN (base)	.496	.413	13.06	.251	.830	20.64	.195	.803	17.81	.360	.483	15.06	.161	.836	19.93
HN (fair)	.486	.409	13.14	.253	.818	20.32	.187	.804	17.92	.361	.465	14.80	.162	.801	19.75
Ours	.253	.825	20.32	**.214**	.856	**22.07**	**.153**	**.858**	**20.92**	**.153**	**.849**	**21.63**	**.156**	**.877**	19.93

	Broom			Chicken			Banana			Mean		
	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑	LPIPS↓	MS-SSIM↑	PNSR↑
NeuralDiff	.631	.468	17.75	.249	.822	21.17	.303	.748	19.43	.264	.784	20.14
HN (base)	.524	.636	19.65	.222	.878	23.94	.223	.818	21.20	.304	.712	18.91
HN (fair)	**.503**	.624	19.38	**.180**	.881	23.68	**.194**	**.832**	**21.52**	.291	.704	18.82
Ours	.565	**.712**	**20.66**	.204	**.890**	**24.27**	.260	.820	21.35	**.245**	**.836**	**21.39**

Table 5: **Novel view synthesis (quantitative) –** We compare with NeuralDiff [54], a baseline version of HyperNeRF [40], denoted HN (base), and a fair version with extended network width to match the total number of parameters in our method, denoted HN (fair). Three scenes displayed in the bottom row are from HyperNeRF[40]

F Ambiguity between Dynamic Component and Shadow

The shadow field network represents the density-less shadows in a more physically realistic way, and resolves the ambiguity in their motion. However, the aim of our method is to achieve decoupling of dynamic occluders from the static environment, and we do not over-extend to consider further decoupling between objects and shadows, which would require more priors related to environmental lighting conditions and background texture.

We empirically found that shadow field is not necessary for scenes with strong and fast-moving shadows, where they can be directly learned by the dynamic component as thin layers on top of the static geometry; see Figure 17. On the other hand, there exists unsolvable ambiguity between the shadow and dynamic object, especially for which with a similar or darker color to the background, and hence could be potentially explained as a moving shadow instead of an actual 3D shape due to our monocular camera setting; see Figure 18. As the later case causes failed dynamic geometry reconstruction, leading to a severe decrease in novel view synthesis performance, we deliberately choose a large value of λ_ρ to suppress the shadow field for scenes containing a mixture of dynamic objects and shadows. Although this potentially favors the former case and causes more shadows to be interpreted as thin-layers, we found that such setting has minimal impact on performance of both scene decoupling and reconstruction, and is still sufficient to achieve correct shadow decoupling, as the shadow field resolves the ambiguity in shadow in very early stage of the training.

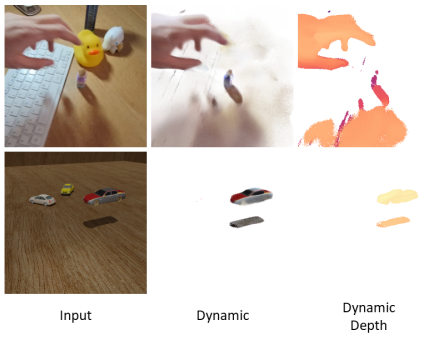

Figure 17: **Shadow as thin layer –** Although such representation could potentially represent more than just shadows, it still tends to exclude unnecessary texture from static background and learns only the darkening effects.

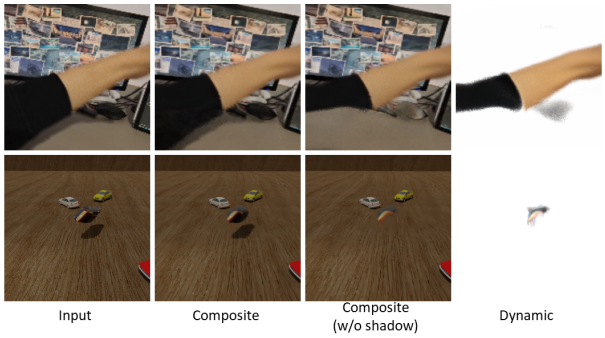

Figure 18: **Incorrect shadow –** The shadow field is incorrectly used to explain the black sleeve as well as the gray top of the moving car.