Photo-realistic Neural Domain Randomization

Sergey Zakharov*, Rares Ambrus*, Vitor Guizilini1, Wadim Kehl2, and Adrien Gaidon1

1 Toyota Research Institute, Los Altos, CA
2 Woven Planet, Tokyo, Japan

Abstract. Synthetic data is a scalable alternative to manual supervision, but it requires overcoming the sim-to-real domain gap. This discrepancy between virtual and real worlds is addressed by two seemingly opposed approaches: improving the realism of simulation or foregoing realism entirely via domain randomization. In this paper, we show that the recent progress in neural rendering enables a new unified approach we call Photo-realistic Neural Domain Randomization (PNDR). We propose to learn a composition of neural networks that acts as a physics-based ray tracer generating high-quality renderings from scene geometry alone. Our approach is modular, composed of different neural networks for materials, lighting, and rendering, thus enabling randomization of different key image generation components in a differentiable pipeline. Once trained, our method can be combined with other methods and used to generate photo-realistic image augmentations online and significantly more efficiently than via traditional ray-tracing. We demonstrate the usefulness of PNDR through two downstream tasks: 6D object detection and monocular depth estimation. Our experiments show that training with PNDR enables generalization to novel scenes and significantly outperforms the state of the art in terms of real-world transfer.

1 Introduction

Collecting labelled data for various machine learning tasks is an expensive, error-prone process that does not scale. Instead, simulators hold the promise of unlimited, perfectly annotated data without any human intervention but often introduce a domain gap that affects real-world performance. Effectively using simulated data requires overcoming the sim-to-real domain gap which arises due to differences in content or appearance. Domain adaptation methods rely on target data (i.e., real-world data) to bridge that gap [53, 56, 40, 62, 64, 33, 16]. A separate paradigm that requires no target data is that of Domain Randomization [51, 52], which forgoes expensive, photo-realistic rendering in favor of random scene augmentations. In the context of object detection, CAD models are typically assumed known [18, 60, 42] and a subset of lighting, textures, materials, and object poses are randomized. Although typically inefficient, sample efficiency can be improved via differentiable guided augmentations [59], while content [25, 10] and appearance [44, 35] gaps can also be addressed by leveraging real data. However, a significant gap remains in terms of the photo-realism
of the images generated. As an alternative, recent work [1, 16] has shown that
downstream task performance can be improved by increasing the quality of syn-
thetic data. However, generating high-quality photo-realistic synthetic data is
an expensive process that requires access to detailed assets and environments,
as well as modeling light sources and materials inside complex graphics pipelines
which are typically not differentiable.

We propose a novel method that brings together these two separate paradigms
by generating high-quality synthetic data in a domain randomization framework.
We combine intermediate geometry buffers ("G-buffers") generated by modern
simulators and game engines together with recent advances in neural render-
ing [45, 36, 2], and build a neural physics-based ray tracer that models scene
materials and light positions for photo-realistic rendering. Our Photo-realistic
Neural Domain Randomization (PNDR) pipeline learns to map scene geometry
to high quality renderings and is trained on a small amount of high-quality photo-
realistic synthetic data generated by a traditional ray-tracing simulator. Thanks
to its geometric input, PNDR generalizes to novel scenes and novel object con-
figurations. Once trained, PNDR can be integrated in various downstream task
training pipelines and used online to generate photo-realistic augmentations.
This alleviates the need to resort to expensive simulators to generate additional
high-quality image data when training the downstream task. Our method is more
efficient in terms of time (PNDR renderings are generated 3 orders of magni-
tude faster than traditional simulators), space (PNDR renderings are generated
on-the-fly during training and therefore do not need storage space) and leads
to better generalization. Although our proposed pipeline is generic in nature,
we quantify the usefulness of our synthetic training for the specific tasks of 6D
object detection and monocular depth estimation in a zero-shot setting (i.e.,
without using any real-world data), and demonstrate that our method presents
a distinct improvement over current SoTA approaches.

In summary, our contributions are:

- We unify photo-realistic rendering and domain randomization for synthetic
data generation using neural rendering;
- Our learned deferred renderer, RenderNet, allows flexible randomization of
physical parameters while being 1, 600× faster than comparable ray-tracers;
- Our Photo-realistic Neural Domain Randomization (PNDR) approach yields
state-of-the-art zero-shot sim-to-real transfer for 6D object detection and
monocular depth estimation, almost closing the domain gap;
- We show that realistic physics-based randomization, especially for lighting,
is key for out-of-domain generalization.

2 Related Work

Domain Adaptation. Due to the domain gap, models trained on synthetic
data suffer performance drops when applied on statistically different unlabelled
target datasets. Domain Adaptation is an active area of research [7] with the
Fig. 1: **PNDR Architecture.** The main component of our domain randomization method is the ray tracer approximator (RenderNet). It takes a G-buffer as well as random material maps and light maps produced by corresponding samplers and generates intermediate light outputs. These outputs are then combined using a tone mapper to generate a final rendering. The lower-right row shows different material and light samples (e.g., roughness, specularity, light position).

The aim of minimizing the *sim-to-real* gap. Common approaches rely on adversarial learning for feature or pixel adaptation [6, 53, 13], paired [56] or unpaired [66, 40, 30] image translation, style transfer [62], refining pseudo-labels [67, 64, 33], or unsupervised geometric guidance [16].

**Domain Randomization.** A different approach to closing the sim-to-real gap relies on generating augmentations of the input data through random perturbations of the environment (e.g., lights, materials, background) [51, 52, 18]. The aim is to learn more discriminative features that generalize to other domains. While simple and inexpensive, this method is sample inefficient because the randomization is essentially unguided with many superfluous (or even harmful) augmentations, and it rarely captures the complexity and distribution of real scenes. Differently, procedurally generating synthetic scenes [43] can preserve the context of real scenes while minimizing the gaps in content [25, 10, 20] and appearance [44, 60, 42]. While some of these methods require expensive, bespoke simulators [25, 10], pixel-based augmentations can be generated differentiably and combined with the task network to generate adversarial augmentations [59]. Similarly to [59] our pipeline is also differentiable, however while [59] is limited to handcrafted image augmentations where respective parameters are sampled from artificial distributions, our method approximates a material-based ray tracer simulating the physical process of light scattering and global illumination, enabling effects such as shadows and diffuse interreflection. Our augmentations are solely based on light and material changes, thus reducing the randomization set to physically plausible augmentations. Moreover, as opposed to [59], we assume no color information of the objects of interest, making our method more practical for real-world applications.

**Photo-Realistic Data Generation.** Although expensive to generate, high-quality synthetic data (i.e., photo-realistic) can increase model generalization
The task of view synthesis allows the rendering of novel data given a set of input images \[49\]. Neural Radiance Fields \[34\] overfit to specific scenes and can generate novel data with very high levels of fidelity, while also accounting for materials and lights \[4, 48, 5\]. Alternative methods use point-based differentiable rendering \[46, 3\] and can optimize over scene geometry, camera model, and various image formation properties. While these methods overfit to specific scenes, recent self-supervised approaches learn generative models of specific objects \[35\] and can render novel and controllable complex scenes by exploiting compositionality \[37\]. While neural volume rendering and point based techniques can yield impressive results, other methods aim to explicitly model various parts of traditional graphics pipelines \[45, 36, 2, 24, 50\]. Our work is similar to \[45\] in that we also use intermediate simulation buffers to generate photo-realistic scenes. However, while \[45\] relies on real data and minimizes a perceptual loss in an adversarial framework, we focus on the task of 6D object detection in a zero-shot setting using only object CAD model information and no real images.

**6D Object Detection.** Correspondence-based methods \[61, 32, 23, 19, 39, 41\] tend to show superior generalization performance in terms of adapting to different pose distributions. However, they use PnP and RANSAC to estimate poses from correspondences, which makes them non-differentiable. Additionally, they are very reliant on the quality of these correspondences, and errors can result in unreasonable estimates (e.g., behind the camera, or very far away). Conversely, regression-based methods \[65, 12, 28\] show superior performance for in-domain pose estimation. However, they do not generalize very well to out-of-domain settings. To validate our method, we implement a correspondence-based object detector, which allows us to also evaluate instance segmentation and object correspondences in addition to the object pose regressed.

# 3 Photo-realistic Neural Domain Randomization

Our photo-realistic neural domain randomization (PNDR) approach consists of two main components: a neural ray tracer approximator (RenderNet), and sampling blocks for material and light. To increase perceptual quality and realism, the network outputs are passed through a non-linear tone-mapping function which yields the final rendering. We now describe the main two components of PNDR. All other implementation and training details are provided in the supplementary.

## 3.1 Geometric Scene Representation

As a first step, we define a geometric room representation outlining our synthetic environment. We place 3D objects inside an empty room ensuring no collisions. Next, we assign random materials to both objects and room walls and position a point light source to illuminate the scene (see Fig 2). Resulting output buffers, consisting of G-Buffer (scene coordinates in camera space \(X\), surface normals
Fig. 2: Geometric scene representation. Visualization of RenderNet’s input consisting of G-Buffer (scene coordinates in camera space $X$, surface normals map $N$), material properties (albedo $A$, roughness $R$, specularity $S$), and lighting (light direction map $L_{dir}$, and light distance map $L_{dist}$).

3.2 Neural Ray Tracer Approximator

Our neural ray tracer RenderNet $f_R$ is an encoder-decoder CNN taking G-buffer, material properties, and lighting as input, and generating a final high-fidelity rendering (see Fig. 1). This is akin to deferred rendering, a common practice in computer graphics [8]. Instead of outputting a final rendering directly, we split the output into direct and indirect light outputs and colors which can be easily combined to form a final, shaded image. This allows not only for a much more explainable representation, but also for better control over the complexity of the rendering. As a result, our RenderNet $f_R$ is capable of generating photo-realistic images, generalizes well to novel material and light distributions, and even novel scenes, objects, and poses.

Light Modelling Lighting in ray tracers can often be decomposed into (1) direct lighting as coming from lamps, emitting surfaces, the background, or ambient occlusion after a single reflection or transmission from a surface; and (2) indirect lighting that comes from lamps, emitting surfaces or the background after more than one reflection or transmission. Simulating indirect lighting approximates realistic energy transfer much closer and produces better images, but comes at much higher computational cost. To be computationally reasonable, we render all scenes with a single point light source.

Light Sampler. Our light sampler is a uniform random 3D coordinate generator. We limit the light pose space to the upper hemisphere and normalize the position to be at a distance of 1.5m from the scene center as defined in our training data. The resulting light source position in scene coordinates is then
brought into the camera space given a fixed transform. Next, we parametrize
the scene lighting by composing two light maps: $L_{\text{dir}}$ defines the direction to the
light source from each visible coordinate and $L_{\text{dist}}$ defines the metric distance
to the light source. Since RenderNet $f_R$ is fully differentiable, we can also use
it to recover scene parameters in terms of lighting, particularly when combined
with a correspondence-based object detector (see Sec. 4 with qualitative results
in 6.6). In this case we define the light sampling network $f_L$ as a SIREN-based
MLP [47] conditioned on the scene ID, which allows us to optimize for the light
position given an input image.

**Material Modelling** For both direct and indirect lighting our RenderNet $f_R$
outputs two separate images representing diffuse and glossy bidirectional scatter-
ing distribution functions (BSDF). The diffuse BSDF is used to add Lambertian
[29] and Oren-Nayar [38] diffuse reflection, whereas the glossy BSDF adds a
GGX microfacet distribution [54] that models metallic and mirror-like materials.

**Material Sampler.** Similarly to the light sampler, the material sampler is a
uniform random value generator. It samples five values per object: RGB values
for albedo $A$, roughness $R$ and specularity $S$ values. We query the material
sampler for all objects in the scene including the background and, given ground
truth instance masks, compose final 2D maps for each output property. The
RGB albedo values are then multiplied by the GT decolorized albedo to form the
final coloring map. Roughness and specular values are assigned to corresponding
object masks to form full 2D maps.

Following a similar architecture to LightNet, we introduce an object material
sampling network $f_M$, outputting material properties for each of the objects
present in the dataset as well as the background environment. As shown in Fig. 1,
it takes an object ID and scene ID values as input, and as before, produces the
same object-specific material properties, i.e., albedo $A$, roughness $R$, and
specularity $S$. Similarly to the uniform sampler, we query MaterialNet for all
scene objects and compose final 2D maps for each output property.

**Image Compositing and Tone Mapping** Supplied with the G-buffer that
provides us with scene coordinates in camera space $X$ and surface normals map
$N$, we can form the final input to the RenderNet $f_R$ by concatenating all inter-
mediate results and passing them through the encoder-decoder structure:

$$f_R(X, N, A, S, R, L_{\text{dir}}, L_{\text{dist}}) = [D_{\text{dir}}, D_{\text{ind}}, G_{\text{dir}}, G_{\text{ind}}].$$  \hspace{1cm} (1)

Here, $A, S, R, L_{\text{dir}}, L_{\text{dist}}$ are the outputs of the material and light submodules,
as previously explained, whereas $D_{\text{dir}}, D_{\text{ind}}$ and $G_{\text{dir}}, G_{\text{ind}}$ are the diffuse and
glossy BRDF outputs for direct and indirect lighting, respectively. During train-
ing we supervise the 4 outputs of RenderNet using corresponding ground truth
quantities through an L1 loss. As outlined in Figure 1, the final HDR image is
a combination of the light and BSDF outputs. In particular:

$$I_{\text{HDR}} = (D_{\text{dir}} + D_{\text{ind}}) * D_{\text{col}} + (G_{\text{dir}} + G_{\text{ind}}) * G_{\text{col}},$$  \hspace{1cm} (2)
where $D_{col}$ represents object albedo and $G_{col}$ represents the probability that light is reflected for each wavelength. All compositing computations are performed in linear color space, which corresponds closer to nature and results in a more physically accurate output, i.e., there is a linear relationship between color intensity and the number of incident photons. However, these values do not directly correspond to human color perception and display devices.

To address the limited color spectrum and brightness of displays, we apply a non-linear tone mapping to fit the device gamut. Although there are many families of mappings to choose from, we picked the commonly-used operator proposed by Jim Hejl and Richard Burgess-Dawson [21]. At its core, it is a rational function that mimics the response curve of a Kodak film commonly used in cinematography:

$$I_{final} = \frac{I_s \ast (6.2 \ast I_s + .5)}{I_s \ast (6.2 \ast I_s + 1.7) + 0.06},$$

where $I_s = \max(0, I_{HDR} - 0.004)$.

### 4 Downstream Tasks

We combine PNDR with two downstream task methods. At each training step, PNDR is used to generate photo-realistic augmentations on the fly, which are fed to the downstream task network.

**Correspondence-Based 6D Object Detection.** Following related work [61, 23, 39, 32, 19], our Correspondence-Based 6D Object Detector (CBOD) operates on RGB images and outputs the probability of each pixel belonging to a certain local object coordinate. Estimated 2D-3D correspondences are then fed into a PnP+RANSAC solver together with camera parameters to estimate final poses. We use non-uniform Normalized Object Coordinates Space (NOCS) [55, 58] maps that maximize the volume inside a unit cube, and we train using a cross-entropy loss. To disambiguate objects, our detector also outputs an instance segmentation mask. We then define object regions relying on instance mask probabilities and use respective correspondences to compute final poses. In addition to achieving competitive results, this simple yet effective architecture allows us to analyze the benefit of PNDR not just for a single task of 6D pose estimation, but also for instance mask estimation and geometric correspondence accuracy, which are

![Fig. 3: Downstream Tasks Coupled with PNDR. During training, both downstream tasks (detection and depth estimation) take PNDR renderings generated online, providing new realistic augmentations at each iteration.](image-url)
crucial components of general scene understanding. The structure of our detector is shown in Fig. 3, while the network architecture details are provided in the supplementary material.

**Monocular Depth Estimation.** We aim to learn a function $f_D : I \rightarrow D$ that recovers the depth $\hat{D} = f_D(I(p))$ for every pixel $p \in I$. We operate in the supervised setting where we have access to the ground truth depth map, and we train the monocular depth network using the SILog loss [11, 31] defined between the predicted and the ground truth depth maps. We evaluate the effect of PNDR using two network architectures: monodepth2 [14] and packnet-sfm [15].

## 5 Experiments

We designed a number of experiments aimed at exploring how PNDR-generated data compares to real data as well as expensive, ray-tracer based simulation data in terms of downstream task performance.

### 5.1 Evaluation Metrics

**6D Object Detection.** Following related work [61, 27], we use ADD [17] as the metric to evaluate object detection. ADD is defined as the average Euclidean distance between the model vertices transformed with ground truth and predicted poses:

$$m = \text{avg}_{x \in \mathcal{M}} \left\| (Rx + t) - (\hat{R}x + \hat{t}) \right\|_2,$$

where $\mathcal{M}$ is a set of vertices of a 3D model, $(R, t)$ and $(\hat{R}, \hat{t})$ are ground truth and predicted rotation and translation, respectively. Most commonly, a predicted pose is considered to be correct if ADD calculated with this pose is less than 10\% of a model diameter. However, this is a very strict metric especially for objects with a small diameter since it can completely disregard good pose estimates and estimates that could be refined. To be able to better analyze pose quality, we instead compute ADD under multiple thresholds (from 5 to 50 with a step of 5) and then estimate the area under the curve (AUC).

**Instance Segmentation.** To evaluate the quality of the instance segmentation we use a standard Intersection over Union (IoU) metric, which quantifies the percent overlap between the target mask and our prediction output.

**Object Correspondences.** To evaluate the quality of estimated object correspondences, we compare per-point metric distances in object’s coordinate space between the GT partial shape and the predicted one. To do that we first use GT masks to recover partial object shapes and compute their absolute scale given provided model information. Then, we measure one-to-one distances between the GT shape and predicted shape in millimeters.

**Depth Estimation.** We evaluate the performance of our depth networks using the standard metrics found in the literature: AbsRel, RMSE and $\delta_1$, which are defined in detail in the supplementary.
Table 1: **HB Dynamic Lighting Benchmark**: All methods are trained on the training set of $HB_5$ and evaluated on the $HB_5$ test set and on $HB_{10}$. † indicates that [22, 56] are trained with synthetic and real image pairs, while ‡ indicates unpaired synthetic and real images for [66, 40]. Training on photo-realistic synthetic data is competitive with real data training and generalizes better to new domains. By training on PNDR images we further close the gap to training on real data in $HB_5$ and increase generalization performance to the novel lighting setting of $HB_{10}$.

**Perceptual Quality.** To evaluate generated RenderNet images, we use the standard image quality metrics PSNR and SSIM [57] for all evaluations. Moreover, we include LPIPS [63], more accurately reflecting human perception.

### 6 Results

#### 6.1 HB Dynamic Lighting Benchmark

In this first experiment we aim to isolate the effects of training and testing under significantly different illumination while keeping the scene contents constant. We use scenes 5 and 10 of the HomeBrewedDB (HB) dataset [26] ($HB_5$ and $HB_{10}$ for short), as shown in Fig. 4. Both scenes contain the same objects in the same environment and consist of 340 images with associated depth maps and object annotations (CAD models and poses). $HB_5$ and $HB_{10}$ are captured with drastically different lighting conditions, allowing us to isolate the effect...
simulated data has on overcoming this perceptual domain gap. We split HB$_5$ into a training and in-domain testing subsets consisting of 272 and 68 frames respectively; HB$_{10}$ is used entirely for testing.

We present the benchmark results in Table 1 (qualitative results are shown in Fig. 6). Our first baseline consists of training CBOD directly on the HB$_5$ real images, and we record good in-domain performance (90.93) and poor transfer to different light configurations (22.24). Our second baseline uses entirely synthetic photo-realistic images of increasing sizes. Using the object CAD models and associated poses corresponding to the different training frames (i.e. we have a total of 272 different object configurations), we generate Domain Randomized synthetic photo-realistic images with BlenderProc [9]. Specifically, for each training configuration we vary object materials and light positions. For backgrounds we randomly select from 5 different asset classes (Bricks, Wood, Carpet, Tile, Marble) and also randomize their materials. For each training configuration we generate an increasing number of augmentations using this technique, leading to larger synthetic datasets with very high perceptual quality at the expense of rendering time and storage space. We train CBOD on the synthetic images, and, as expected, downstream task performance improves as more high-quality synthetic data is available (i.e., with 4352 synthetic images we achieve 37.35 generalization performance). We compare this with the proposed PNDR method as follows: using the high-quality synthetic data along with the corresponding G-buffer information, we train PNDR, and use it in the training pipeline of CBOD to generate new, high-quality augmentations on the fly, saving rendering time and storage space. As shown in our experiments, 1088 synthetic images are enough to train PNDR, and we almost match the performance of training on real data and increase generalization to scenes with significant light variation by 82% (40.44 vs 22.24). We note that as CBOD is trained over 400 epochs, it would require ~30h and 600GB storage space to generate as many images with the raytracer as were generated by PNDR.

---

3 https://ambientcg.com
Table 2: HB-LM Cross-Domain Adaptation Benchmark: All methods are trained on the training set of HB2 and evaluated on HB2 test set as well as on the LM2, LM8, and LM15 scenes. Training on real data generalizes poorly to novel object categories. Training on photo-realistic synthetic data is competitive with real data training when evaluated on the same object categories, and greatly increases generalization performance for novel object categories. † indicates that [22, 56] are trained with paired synthetic and real image pairs, while ‡ indicates unpaired images for [66, 40].

Additionally we train state-of-the-art image-translation methods on the same images we train PNDR on, and we use the corresponding real images as translation targets. Specifically, we compare against (i) pix2pix [22] and pix2pixHD [56] using BlenderProc and real image pairs; and (ii) cycleGAN [66] and CUT [40] using unpaired BlenderProc and real images. Although our method does not require any real images, adversarial image translation methods are representative of the state-of-the-art in domain adaptation and serve as good baselines. The training details of all the baselines are provided in the supplementary. The paired GAN baselines [22, 56] increase generalization performance, although we note that having access to synthetic and image pairs is an unrealistic scenario in practice and this serves as an upper-bound, at least for in-domain evaluation. The more realistic case of unpaired translation [66, 40] performs much worse, as expected. This is easily explained by the rather large domain shift induced from the different scene setups and cameras.

6.2 HB-LM Cross-Domain Adaptation Benchmark

Our HB-LM cross-domain benchmark (see Fig. 5) is represented by HB2 covering three objects of the LineMOD (LM) [17] dataset (benchvise, driller, and phone). Additionally, we use scenes 2, 8 and 15 from the LM dataset for testing: these scenes contain the same objects as HB2 but with significantly different poses and in a different setting. This setting allows us to evaluate the generalization performance of PNDR to new scenes and new object poses. As before, we partition the HB2 into a training and test split consisting of 272 and 68 images, respectively, and we use BlenderProc [9] to generate the same synthetic photo-realistic renderings and G-buffer information. In addition to the HB2 data, we also generate 1000 photo-realistic images using BlenderProc while randomizing
both the camera and the poses of the 3 objects from $HB_2$. As we show in the experiments, using this extra simulated data allows us to generalize much better to the $LM$ scenes where the object pose distribution is significantly different. We train $PNDR$ as before and use its output to train $CBOD$, which is evaluated both in domain, i.e., on the test split of $HB_2$ as well as out of domain on $LM_2$, $LM_8$ and $LM_{15}$.

We analyze how well we generalize to completely different scenes with different lighting conditions, environment, camera setup and object poses; our results are summarized in Table 2. As before, we report the best in-domain results when training on real data, with a slight performance drop when training directly on the photo-realistic synthetic BlenderProc data. We note that by using $PNDR$ we significantly increase performance. As before, the paired translation GAN baselines compare quite well, and we record a similar performance drop when doing unpaired image translation. Both the GAN and the baseline trained on real data generalize poorly to the LM scenes, reflecting the challenging nature of this benchmark. Interestingly, the unpaired image translation baselines generalize better in this setting - we provide qualitative examples in the supplementary. By training on the synthetic data which contains additional renderings with randomized object poses we significantly improve performance. As before, using $PNDR$ as part of the training pipeline further improves performance, achieving 27.33 on the LM scenes.

6.3 HB Generalization Benchmark

Here we aim to evaluate how well $PNDR$ can generate novel photo-realistic images in and out of domain. We train $PNDR$ on $HB_2$ and evaluate on the test set of $HB_2$ as well as $HB_5$. Note that these two scenes contain different objects, allowing us to investigate if our learned ray-tracing network generalizes to novel scene geometries. For completeness, we also perform the same experiment by training on $HB_5$ and evaluating on $HB_2$.

We quantify the generalization capabilities of $PNDR$ when applied to novel scenes, object arrangements, material properties and lighting. We train $PNDR$ on

![Real image](image1) ![Recovered materials / lighting](image2)

Fig. 7: Recovering scene properties via our RenderNet. Given the recovered G-Buffer we optimize over $MaterialNet$ and $LightNet$ to find the best fit explaining the input image.
Table 3: PNDR Generalization: We achieve strong performance not only when applied to a test image of the same scene containing the same objects with different material properties and different poses, but also when applied on a completely different scene.

| Train | Test | PSNR↑ | SSIM↑ | LPIPS↓ |
|-------|------|-------|-------|--------|
| HB2 - Train | HB2 - Train | 30.36 | 0.96 | 0.03 |
|       | HB2 - Test  | 24.14 | 0.92 | 0.06 |
| HB5 - Train | HB5 - Train | 30.39 | 0.96 | 0.03 |
|       | HB5 - Test  | 26.40 | 0.94 | 0.06 |
| HB2 - Test  | HB5 - Test  | 23.72 | 0.92 | 0.07 |

Table 4: Ablation: We analyze the effect of different augmentations on downstream task performance. All methods are trained on the HB5 train set and evaluated on the HB5 test set.

| Modes | ADD AUC↑ | mIoU↑ | Corr (mm)↓ |
|-------|---------|-------|-----------|
| Material | A + S + R | 3% | 3% | 11% |
| Light | Fixed | 73.04 | 72.24 | 76.94 |
|       | Dynamic | 21% | 19% | 55% |
| Render | HBdir, Gdir | 87.09 | 85.55 | 38.42 |
|        | HBind, Gind | 1% | 0% | 11% |
| Full   |         | 88.18 | 85.97 | 34.32 |

6.4 Ablation Study

We analyze how different physically-based augmentations affect the downstream task performance, and consider: material randomization, light randomization, and rendering complexity. PNDR is conditioned on material properties of the objects, i.e., albedo $A$, specularity $S$, and roughness $R$, with $A$ being the most important property. In Table 4 we see that training with just albedo randomization results in very good performance already. Additionally simulating $S$ and $R$ brings a relative gain of 2% with respect to ADD AUC, 3% to mIoU, and 11% to correspondence quality. Furthermore, we note that lighting is by far the most important randomization parameter. Going from fixed to dynamic lighting significantly improves the results: 19% ADD AUC gain, 19% mIoU gain, and 55% correspondence quality gain. Finally, we note that simulating computationally expensive indirect lighting only helps improve correspondence quality but is negligible for the other metrics. Since we use an outlier-robust PnP+RANSAC solver, small deviations in correspondence quality do not significantly affect the pose quality as evaluated by ADD AUC.

6.5 Monocular Depth Estimation

We quantify the impact of PNDR when applied for the task of monocular depth estimation. We use the same data as for the Dynamic Lighting Benchmark.
Table 5: PNDR vs Raytraced - monocular depth results.

| Method          | Training | HB5   | HB10  |
|-----------------|----------|-------|-------|
|                 | AbsRel↓ | RMSE↓ | a1↑   |
| Monodepth2      | 0.082   | 0.09  | 0.951 |
| Raytraced       |          |       |       |
| PNDR            | 0.075   | 0.083 | 0.966 |
| PackNet-SfM     | 0.11    | 0.111 | 0.884 |
| Raytraced       | 0.141   | 0.14  | 0.833 |
| PNDR            | 0.082   | 0.087 | 0.977 |
|                 |          |       |       |
|                 | 0.135   | 0.131 | 0.852 |

(see 6.1). For both monodepth2 and packnet-sfm we compare performance when training directly on the 1088 raytraced images with performance when PNDR is integrated in the training pipeline and generates novel augmentations on the fly. As before, we note that when training with PNDR we achieve better in-domain and better generalization performance (see Table 5).

6.6 Object Material and Light Recovery

The fact that RenderNet is fully differentiable allows us to optimize over scene parameters. In particular, given an initial scene prediction provided by CBOD we can recover material properties of the objects and scene light (see Fig. 7). First, we construct a G-buffer by estimating a depth map using estimated poses and object models, which is in turn used to estimate scene coordinates $X$ and surface normals $N$. LightNet $f_L$ and MaterialNet $f_M$ conditioned on the scene and object IDs output the remaining maps $A$, $R$, $S$, $L_{dir}$, and $L_{dist}$ required for the RenderNet. Finally we generate a rendering that is then compared to the GT RGB image to find the best fit. Estimated scene parameters can be used to learn a distribution of material and light configurations across the entire dataset. This information might not only be useful for analysis, but also for domain-specific data generation using our RenderNet, especially where the same object instances exist in multiple material variations.

7 Conclusion

We have presented a novel approach towards sim-to-real adaptation by means of a neural ray tracer approximator with randomizable material and light modules that we named PNDR. We have demonstrated that applying our photo-realistic randomized output to the problem of zero-shot 6D object detection significantly outperforms other established DA approaches, and even comes close to training on real data. We have identified lighting as the most crucial component, but it remains an open question what kind of additional randomization could further benefit the domain transfer. One possible future research avenue would be randomized sampling of low-level camera sensor artifacts, or the coupling of randomization and downstream task optimization in a common framework.
References

1. Alghonaim, R., Johns, E.: Benchmarking domain randomisation for visual sim-to-real transfer. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 12802–12808. IEEE (2021)
2. Alhaija, H.A., Mustikovela, S.K., Geiger, A., Rother, C.: Geometric image synthesis. In: Asian Conference on Computer Vision. pp. 85–100. Springer (2018)
3. Aliev, K.A., Sevastopolsky, A., Kolos, M., Ulyanov, D., Lempitsky, V.: Neural point-based graphics. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 696–712. Springer (2020)
4. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12684–12694 (2021)
5. Boss, M., Jampani, V., Braun, R., Liu, C., Barron, J.T., Lensch, H.: Neural-pil: Neural pre-integrated lighting for reflectance decomposition. arXiv preprint arXiv:2110.14373 (2021)
6. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3722–3731 (2017)
7. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374 (2017)
8. Deering, M., Winner, S., Schediwy, B., Duffy, C., Hunt, N.: The triangle processor and normal vector shader: a vlsi system for high performance graphics. Acm siggraph computer graphics 22(4), 21–30 (1988)
9. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
10. Devaranjan, J., Kar, A., Fidler, S.: Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In: European Conference on Computer Vision. pp. 715–733. Springer (2020)
11. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS (2014)
12. Engelmann, F., Rematas, K., Leibe, B., Ferrari, V.: From points to multi-object 3d reconstruction. In: CVPR. pp. 4588–4597 (2021)
13. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR (2016)
14. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3828–3838 (2019)
15. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2485–2494 (2020)
16. Guizilini, V., Li, J., Ambrus, R., Gaidon, A.: Geometric unsupervised domain adaptation for semantic segmentation. arXiv preprint arXiv:2103.16694 (2021)
17. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: ACCV (2012)
18. Hinterstoisser, S., Pauly, O., Heibel, H., Martina, M., Bokeloh, M.: An annotation saved is an annotation earned: Using fully synthetic training for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)

19. Hodan, T., Barath, D., Matas, J.: Epos: Estimating 6d pose of objects with symmetries. In: CVPR. pp. 11703–11712 (2020)

20. Hodań, T., Vineet, V., Gal, R., Shalev, E., Hanzelka, J., Connell, T., Urbina, P., Sinha, S.N., Guenter, B.: Photorealistic image synthesis for object instance detection. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 66–70. IEEE (2019)

21. Hoffman, N.: Crafting physically motivated shading models for game development, part of “Physically Based Shading Models in Film and Game Production,” SIGGRAPH (2010)

22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)

23. Jafari, O.H., Mustikovela, S.K., Pertsch, K., Brachmann, E., Rother, C.: iPose: instance-aware 6d pose estimation of partly occluded objects. In: ACCV. pp. 477–492. Springer (2018)

24. Janner, M., Wu, J., Kulkarni, T.D., Yildirim, I., Tenenbaum, J.B.: Self-supervised intrinsic image decomposition. arXiv preprint arXiv:1711.03678 (2017)

25. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Feidler, S.: Meta-sim: Learning to generate synthetic datasets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4551–4560 (2019)

26. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: Homebreweddb:Rgb-d dataset for 6d pose estimation of 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)

27. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE international conference on computer vision. pp. 1521–1529 (2017)

28. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: ECCV. pp. 574–591. Springer (2020)

29. Lambert, J.H.: Photometria sive de mensura et gradibus luminis, colorum et umbrae. sumptibus viduae E. Klett, typis CP Detleffsen (1760)

30. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: ECCV (2018)

31. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation (2019)

32. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: ICCV. pp. 7678–7687 (2019)

33. Liu, X., Guo, Z., Li, S., Xing, F., You, J., Kuo, C.C.J., El Fakhri, G., Woo, J.: Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10367–10376 (2021)

34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)

35. Mustikovela, S.K., De Mello, S., Prakash, A., Iqbal, U., Liu, S., Nguyen-Phuoc, T., Rother, C., Kautz, J.: Self-supervised object detection via generative image syn-
thesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8609–8618 (2021)
36. Nalbach, O., Arabadzhyska, E., Mehta, D., Seidel, H.P., Ritschel, T.: Deep shading: convolutional neural networks for screen space shading. In: Computer graphics forum. vol. 36, pp. 65–78. Wiley Online Library (2017)
37. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11453–11464 (2021)
38. Oren, M., Nayar, S.K.: Generalization of lambert’s reflectance model. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. pp. 239–246 (1994)
39. Park, K., Pattan, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: ICCV. pp. 7668–7677 (2019)
40. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision. pp. 319–345. Springer (2020)
41. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnnet: Pixel-wise voting network for 6dof pose estimation. In: CVPR. pp. 4561–4570 (2019)
42. Planche, B., Zakharov, S., Wu, Z., Hutter, A., Kosch, H., Ilic, S.: Seeing beyond appearance-mapping real images into geometrical domains for unsupervised cad-based recognition. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2579–2586. IEEE (2019)
43. Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., Birchfield, S.: Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7249–7255. IEEE (2019)
44. Prakash, A., Debnath, S., Lafleche, J.F., Cameracci, E., Birchfield, S., Law, M.T., et al.: Self-supervised real-to-sim scene generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16044–16054 (2021)
45. Richter, S.R., AlHajja, H.A., Koltun, V.: Enhancing photorealism enhancement. arXiv preprint arXiv:2105.04619 (2021)
46. Rückert, D., Franke, L., Stamminger, M.: Adop: Approximate differentiable one-pixel point rendering. arXiv preprint arXiv:2110.06635 (2021)
47. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. NeurIPS (2020)
48. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021)
49. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. arXiv preprint arXiv:2111.05849 (2021)
50. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics (TOG) 38(4), 1–12 (2019)
51. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. IROS (2017)
52. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Proceedings of the
53. Volpi, R., Moreiro, P., Savarese, S., Murino, V.: Adversarial feature augmentation for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5495–5504 (2018)
54. Walter, B., Marschner, S.R., Li, H., Torrance, K.E.: Microfacet models for refraction through rough surfaces. Rendering techniques 2007. 18th (2007)
55. Wang, H., Sridhar, S., Huang, J., Valentín, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)
56. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
57. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4), 600–612 (2004)
58. Zakharov, S., Ambrus, R.A., Park, D., Guizilini, V.C., Kehl, W., Durand, F., Tenenbaum, J.B., Sitzmann, V., Wu, J., Gaidon, A.: Single-shot scene reconstruction. In: 5th Annual Conference on Robot Learning (2021)
59. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 532–541 (2019)
60. Zakharov, S., Planche, B., Wu, Z., Hutter, A., Kosch, H., Ilic, S.: Keep it unreal: Bridging the realism gap for 2.5d recognition with geometry priors only. 3DV (2018)
61. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: ICCV (2019)
62. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)
63. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
64. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. International Journal of Computer Vision 129(4), 1106–1120 (2021)
65. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
66. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)
67. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5982–5991 (2019)