Rotation-Equivariant Conditional Spherical Neural Fields for Learning a Natural Illumination Prior

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inverse rendering is an ill-posed problem. Previous work has sought to resolve this by focussing on priors for object or scene shape or appearance. In this work, we instead focus on a prior for natural illuminations. Current methods rely on spherical harmonic lighting or other generic representations and, at best, a simplistic prior on the parameters. We propose a rotation-equivariant, high dynamic range (HDR) neural illumination model that is compact and able to express complex, high frequency features of natural environment maps. We propose a conditional neural field representation based on a variational auto-decoder with a SIREN network and, extending Vector Neurons, build equivariance directly into the network. We curate a dataset of 1.6K HDR environment maps of natural scenes on which we train our model. We compare against traditional representations, demonstrate its applicability for an inverse rendering task and show environment map completion from partial observations. A PyTorch implementation, our dataset and trained models will be made available.

1 Introduction

Compact but expressive lighting representations play an essential role in graphics, enabling realistic lighting effects at real-time frame rates [38, 52, 53, 34, 22] and in computer vision enabling scene relighting [57, 58], face relighting [54, 43, 17, 42] and object insertion [55, 47, 31]. Real-world illumination is highly complex and variable, with a very high dynamic range, and is therefore inherently challenging to represent. However, real-world illumination does contain statistical regularities [16], particularly for outdoor, naturally lit scenes. For example, lighting usually comes predominantly from above, often with the strongest illumination coming from primary light sources in a few directions. Also, the sky and sunlight produce only a limited range of colours. In addition, illumination environments have a canonical up direction (vertical axis aligns with gravity) but arbitrary horizontal rotation (any rotation about the vertical is equally likely). These regularities and geometric symmetries can significantly restrict the space of possible illuminations to constrain inverse problems or enable the synthesis of realistic lighting.

Lighting representations An illumination environment is a spherical signal. A relatively small set of alternatives are used for their representation within vision and graphics. A widely used representation in graphics is an environment map [39, 38, 52, 53], which is a regularly sampled 2D image representing a flattening of the sphere, usually via an equirectangular projection. However, the projection introduces distortions leading to irregular sampling on the sphere, it is not compact, introduces boundaries and provides no constraint. Nevertheless, environment map representations have been used in inverse settings where every pixel in the map is optimised independently [41].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

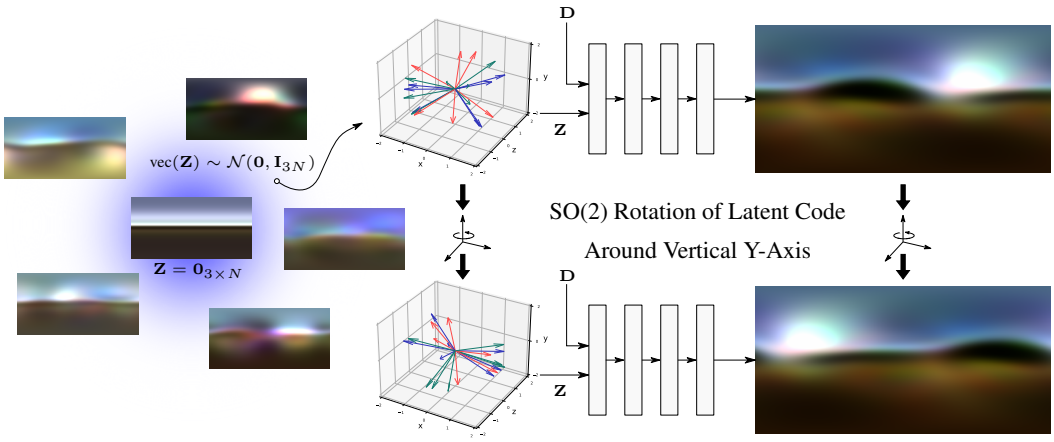

Figure 1: On the left, we visualize environment maps derived from random latent samples of RENI, our natural illumination prior, as well as the average illumination in the centre. RENI is rotation-equivariant to rotations of the latent codes around the vertical y-axis (right). Plots are shown for a 3×20 latent code at two rotations, 160 degrees apart, and the resulting output of the RENI network equally rotated.

Spherical harmonic (SH) lighting [3, 38] is a compact lighting representation commonly used in real-time computer graphics [38, 22, 46] and inverse rendering [57, 52, 31, 40, 17]. While SHs can be used to represent the illumination environment directly, more commonly they represent pre-integrated lighting, i.e. the illumination environment convolved with a bidirectional reflectance distribution function (BRDF). When the BRDF is low frequency, as it is for Lambertian diffuse reflectance, then the convolution is also low frequency making the approximation with SHs very accurate [3].

An alternative, growing in popularity, is the Spherical Gaussian (SG) representation [53, 52, 60]. SGs represent a lighting environment as a collection of Gaussian lobes on the sphere, each of which has 6 degrees of freedom (three for RGB amplitude, two for spherical direction and one for sharpness). While this allows the reconstruction of localised high-frequency features, it still requires many lobes to approximate complex illumination environments. [31] compares SH and SG for object re-lighting, finding SG was able to recover higher frequency lighting using a similar number of parameters as SH, though both still required a large number of parameters to approximate ground truth.

Both SHs and SGs are *rotation equivariant*. A rotation of the illumination environment corresponds directly to a rotation of the SH basis or the SG lobe directions. Equivalently, they can represent any rotation of a given environment with equal accuracy. However, they provide no prior over the space of possible illuminations. SGs or SHs can represent any colour of light coming from any direction.

Illumination Priors When humans solve inverse rendering tasks, they rely on strong priors over the space of possible illuminations. For example, they resolve convex/concave shading ambiguities with the lighting-from-above assumption [26, 51]. Given this, it is surprising that statistical illumination priors have been almost completely ignored in computer vision, with the vast majority of inverse rendering techniques allowing arbitrary illumination within their chosen representation space.

There are a small number of exceptions. Both Egger et al. [17] and Yu and Smith [57] learn a linear statistical model with Gaussian prior in the space of SH coefficients. While providing a useful constraint to avoid unrealistic illumination environments, this approach inherits the weakness of SHs in being unable to reproduce high-frequency lighting effects while also losing the rotation equivariance. Yu and Smith [57] seek to overcome this by rotation augmentation at training time, but this brute force approach makes no guarantee of rotation equivariance. Sztrajman et al. [49] separate environment maps into HDR and LDR components. Using a CNN-based auto-encoder for estimations of LDR components of lighting alongside a low dimensional SG model for the HDR lighting provided by the sun. They too require data augmentation in the form of rotations, can make no equivariance guarantees and, due to the low dimensionality of the SG model, struggle to represent environments with multiple HDR light sources.

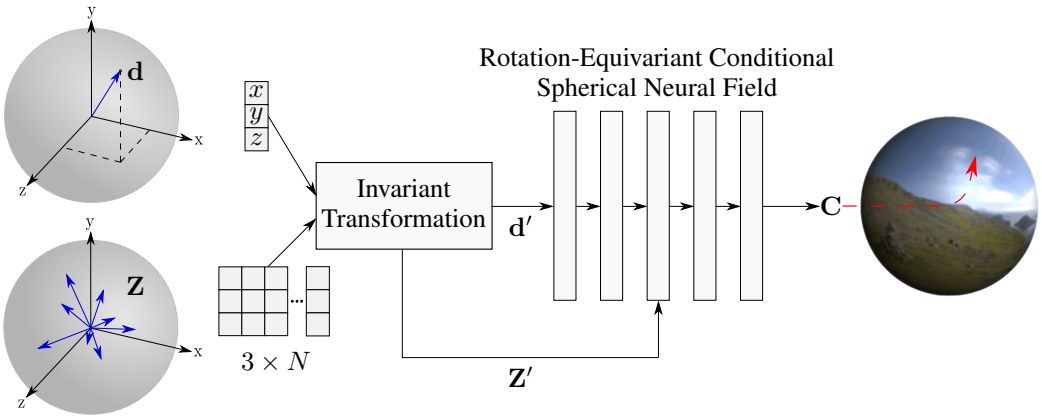

Figure 2: We propose to represent a space of spherical signals via a rotation-equivariant conditional spherical neural field. The signal in a direction $\mathbf{d}$ can be queried by evaluation of the network and rotating the Vector Neuron conditioning latent code $\mathbf{Z}$, corresponds to rotating the spherical signal.

Neural Fields Neural fields [56] have provided impressive results in a range of applications including representations of objects and scenes [44, 2, 33, 13, 12, 23, 36, 50], in inverse rendering [7, 5, 40, 48, 8, 60] and robotics [30, 11, 35]. The work most closely related to ours is Neural-PIL [8]. They use a FiLM SIREN similar to that proposed in Pi-GAN [10], and like us use a direction query vector and auto-decoder architecture. However they do not have rotation-equivariance and apply no natural light prior. During training they interpolate between embeddings and enforce a set of losses to ensure a smooth, interpretable latent space whilst using an auto-decoder architecture.

Rotation Invariance/Equivariance Two important symmetries in computer vision are invariance and equivariance to the rotation group [9]. Some works attempt to achieve these properties via data augmentation [57, 37, 49], which still results in missed cases for continuous rotations. A solution alleviating the need for extensive data augmentation is the Vector Neuron [15], which offers a framework for designing SO(3)-Equivariant networks via a latent matrix representation rather than a latent vector. This allows a direct mapping of rotations applied to the network's input to its output. All possible rotations now being equally well represented implicitly in rotations of the latent codes.

Contribution We desire a representation of natural illumination environments that offers the following features. *Generative:* A generative model that captures the statistical regularities of natural illumination with a well-behaved latent space within which we can optimise to solve inverse problems. *Compact:* Reduces the dimensionality of inverse problems while preserving high-frequency lighting effects that are important for non-Lambertian appearance. *Rotation-Equivariant:* Respect the canonical orientation, i.e. any rotation of an environment about the vertical should be equally likely and equally well represented. *Statistical Prior:* Provides a prior to regularise inverse problems or that can be sampled from for synthesis, only generating plausible illumination environments. *HDR:* Correctly handle HDR quantities essential for realistic rendering and the reproduction of natural light.

We introduce RENI - A Rotation-Equivariant Natural Illumination model. Our key contributions are:

- An extension of Vector Neurons to a rotation-equivariant neural field representation for spherical images, optionally restricted to rotations about the vertical axis.
- A variational autodecoder architecture for a generative model of spherical signals.
- The first natural, outdoor HDR illumination model.
- Evaluated in an inverse rendering task showing significant performance improvements over other lighting representations.

We choose to model scene radiance, i.e. environment lighting, directly as opposed to pre-integrated lighting with a particular BRDF. This makes our model more general since it can be used with arbitrary BRDFs at inference time or even for tasks other than rendering such as to constrain shape from specular flow.

2 Rotation-Equivariant Conditional Spherical Neural Fields

We wish to construct a generative model of spherical signals that is rotation equivariant with respect to the latent representation of the signal. That is to say, a rotation of the latent representation corresponds to a rotation of a spherical signal and a signal can be reconstructed with exactly the same accuracy in any rotation. We begin by proposing a variant of Vector Neurons [15] for $SO(3)$ equivariant representation of spherical signals.

To achieve this, we use an ordered list of 3D vectors for our latent representation. Our model takes the form of a conditional spherical neural field, $f : S^2 \times \mathbb{R}^{3\times N} \to \mathbb{R}^M$, such that $f(\mathbf{d}, \mathbf{Z})$ computes the value of the signal represented by latent code $\mathbf{Z} \in \mathbb{R}^{3\times N}$ in direction $\mathbf{d} \in \mathbb{R}^3$, with $\|\mathbf{d}\| = 1$. For colour images, $M = 3$ and f outputs an RGB colour. By using a spherical neural field, we are agnostic to how the signals are sampled on the sphere. We can generate any sampling simply by choosing the grid of directions as appropriate. We also avoid boundary effects since our domain is continuous.

We construct the neural field such that it is *invariant* to a rotation of both $\mathbf{d}$ and $\mathbf{Z}$ simultaneously (i.e. $f(\mathbf{Rd}, \mathbf{RZ}) = f(\mathbf{d}, \mathbf{Z})$ with $\mathbf{R} \in SO(3)$). This entails that the neural field is *equivariant* with respect to a rotation of $\mathbf{Z}$ only (i.e. rotating $\mathbf{Z}$ corresponds to rotating the spherical signal such that $f(\mathbf{d}, \mathbf{RZ}) = f(\mathbf{R}^\top \mathbf{d}, \mathbf{Z})$).

Equivariant Transformation Key to our approach is a transformation of the inputs to the neural field, $(\mathbf{d}, \mathbf{Z})$, such that they are rotation invariant. These divide into two parts: 1. $\mathbf{d}'$ - the directional input to the spherical neural field and 2. $\mathbf{Z}'$ - the latent code on which the neural field is conditioned.

The direction in which we wish to evaluate the spherical neural field must be encoded relative to the latent code in the particular rotation in which we encounter it. This is satisfied by using the inner product $\langle \mathbf{d}, \mathbf{Z} \rangle$, i.e. the matrix-vector product $\mathbf{d}' = \mathbf{Z}^\top \mathbf{d} \in \mathbb{R}^N$. Unlike Vector Neurons, our input is a direction not a position. Hence, the rotation invariant feature $\|\mathbf{d}\|$ conveys no information and we do not use it.

For the latent code, we follow the Vector Neurons invariant layer: $\mathbf{Z}' = \text{VN-In}(\mathbf{Z})$. However, we use the full Gram matrix $\mathbf{Z}^\top \mathbf{Z}$ since we expect the dimensionality of the latent space, N, to remain moderate for our data. If this $O(N^2)$ size becomes problematic, we can use the same scalable solution used by Vector Neurons [15].

Since our neural field is equivariant **we do not need to augment our training data over the space of rotations**. Observing a spherical signal once means we can reconstruct it with the same accuracy in any rotation.

Variational Auto-decoder We train our conditional spherical neural field as a decoder-only architecture, i.e. an auto-decoder [36]. This means that we optimise the network weights simultaneously with the latent codes for each training sample. This avoids the need to design a rotation-equivariant encoder while, for inverse tasks, only the decoder is needed so we avoid the redundancy of also training an encoder. However, training with no regularisation on the learnt latent space does not lead to a space that is smooth or that follows a known distribution. This means it cannot be sampled from, does not produce meaningful interpolations and provides no prior for inverse problems. For this reason, we use a *variational auto-decoder* [59] architecture. Each training sample is represented by a mean, $\boldsymbol{\mu}_i \in \mathbb{R}^{3N}$, and standard deviation, $\boldsymbol{\sigma}_i \in \mathbb{R}^{3N}$, that provide the parameters of a normal distribution from which the flattened latent code for that training sample is drawn: $\text{vec}(\mathbf{Z}_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\Sigma}_i = \text{diag}(\sigma_{i,1}^2, \ldots, \sigma_{i,3N}^2)$ is the diagonal covariance matrix. Using the reparameterisation trick [28], we can generate a latent code $\text{vec}(\mathbf{Z}_i) = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}$ where the noise is sampled as $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{3N})$. During training we optimise $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ for each training sample and use the Kullback–Leibler divergence as a loss to regularise the distribution of each latent code toward the standard normal distribution:

$$\mathcal{L}_{\text{KLD}} = -\frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{3N} (1 + \log(\sigma_{i,j}^2) - \mu_{i,j}^2 - \sigma_{i,j}^2), \tag{1}$$

where K is the number of training samples.

3 RENI: A Statistical Model of Natural Illumination

We now describe how to construct a statistical model of natural illumination environments as a restricted version of a rotation-equivariant conditional spherical neural field. We call our model RENI (Rotation-Equivariant Natural Illumination). In contrast to the full $SO(3)$-equivariant formulation in Section 2, we propose a variant with only $SO(2)$ equivariance and explain in detail our training data, losses and implementation.

SO(2) equivariance Natural environments have a canonical "up" direction (defined by gravity) but arbitrary rotation about this vertical axis. For this reason, we do not want arbitrary $SO(3)$ rotation equivariance. This would have the undesirable effect of permitting unnatural environment orientations (such as with the sky at the bottom), providing a less useful prior when solving inverse problems. So, we propose a restricted transformation of the neural field inputs that are invariant only to rotations, $\mathbf{R}_y$, about the vertical (y) axis.

In order to construct the invariant features, we use two selection matrices: $\mathbf{S}_{xz}$ selects the components that are orthogonal to the axis of rotation (i.e. the x and z components) and are thus affected by a y-axis rotation, $\mathbf{s}_y$ selects the unaffected y component: $\mathbf{S}_{xz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{s}_y = [0 \quad 1 \quad 0]$. The directional part of the invariant input now contains three components: $\mathbf{d}' = (\mathbf{s}_y\mathbf{d}, \langle \mathbf{S}_{xz}\mathbf{d}, \mathbf{S}_{xz}\mathbf{Z} \rangle, \|\mathbf{S}_{xz}\mathbf{d}\|)$. The first is the invariant component of $\mathbf{d}$. The second encodes $\mathbf{d}$ relative to $\mathbf{Z}$ in the x-z plane. The third measures the norm of $\mathbf{d}$ projected into the x-z plane which is unchanged by rotations about y.

The conditioning part of the invariant input now contains two components: $\text{VN-In}(\mathbf{Z}) = \mathbf{Z}' = (\mathbf{s}_y\mathbf{Z}, (\mathbf{S}_{xz}\mathbf{Z})^\top \mathbf{S}_{xz}\mathbf{Z})$. The first is simply the invariant component of each column of $\mathbf{Z}$. The second is the invariant transformation (Gram matrix) of latent vectors projected into the x-z plane.

Reconstruction Loss and HDR HDR imaging is essential for realistic rendering and enables the accurate representation of the full dynamic range of natural light. Therefore our model must learn an HDR representation of natural illumination. Computing a reconstruction loss in linear HDR space is dominated by large values and leads to poor reconstruction of most of the environment. Therefore, similar to [18], we train our network to output $\log(\text{HDR})$ values and compute the reconstruction loss in log space. We further normalise the $\log(\text{HDR})$ values to the range $[-1, 1]$ by scaling and shifting using the maximum and minimum values across the whole training set.

Each training sample comprises P pairs of directions and corresponding normalised $\log(\text{HDR})$ RGB colours that we store in the matrices $\mathbf{D}_i = [\mathbf{d}_{i1}, \ldots, \mathbf{d}_{iP}] \in \mathbb{R}^{3\times P}$ and $\mathbf{C}_i = [\mathbf{c}_{i1}, \ldots, \mathbf{c}_{iP}] \in \mathbb{R}^{3\times P}$ respectively. RENI is agnostic to the resolution and sampling of the spherical signal. The neural field can be queried for any direction. In practice, our dataset contains spherical images in an equirectangular sampling containing $P = 2H^2$ pixels where H is the height of the images. Since equirectangular images are irregularly sampled, we weight the mean squared reconstruction loss by the sine of the polar angle, $\theta(\mathbf{d})$:

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{K} \frac{1}{P} \sum_{j=1}^{P} \sin(\theta(\mathbf{d}_{ij})) \left\| f(\mathbf{d}_{ij}) - \mathbf{c}_{ij} \right\|^2 . \tag{2}$$

Neural Field Our neural field, f, is implemented as a SIREN [44] - an MLP with periodic activation function. This architecture has proven highly effective at representing a wide range of complex natural signals. In order to condition the SIREN on the invariant latent code $\mathbf{Z}'$, we use *conditioning-by-concatenation* [44, 56], i.e. we simply input both $\mathbf{d}'$ and $\mathbf{Z}'$ to the network. We also tested a FiLM-conditioned SIREN [10] but could not get significant performance improvements and, perhaps due to the small size of our dataset, FiLM conditioning resulted in a non-smooth latent space that could drastically affect performance when fitting to unseen images.

Training Data We have curated a dataset of 1,694 HDR equirectangular images of outdoor, natural illumination environment obtained with either CC0 1.0 Universal Public Domain Dedication license [24, 19, 27] or with written permission to redistribute a low-resolution version of their dataset [25, 1, 45]. All images were then checked to ensure they did not contain any personally identifiable information or offensive content and any images that contained predominantly unnatural light sources

were removed. 21 images were also selected and held back for optimising only the latent codes at test time, resulting in a training dataset of 1,673 HDR images.

Training We use Adam [29] to optimise the sum of the reconstruction and KLD losses:

$$\mathcal{L}_{\text{Train}} = \mathcal{L}_{\text{recon}} + \frac{\beta}{D}\mathcal{L}_{\text{KLD}} \tag{3}$$

where β is a hyperparameter to weight the KLD loss and $D = 3N$ is the dimensionality of the latent space, used to normalise for the choice of N. We randomly initialise the mean latent code for each image, $\boldsymbol{\mu}_i$, from a standard normal distribution. In order to ensure the positivity of the variances, $\sigma^2_{i,j}$, we optimise $\log(\sigma^2_{i,j})$ which we initialise randomly from a normal distribution with mean -5 and variance 1. We use all the pixels from a single training image as a mini-batch and use batch size 1.

In order to speed up training, we employ a progressive strategy. We start by training with low resolution ($H = 16$) images and double resolution every 800 epochs until we reach $H = 128$ resolution for a total of 2,400 epochs. This allows the network to quickly learn low-frequency features, gradually obtaining higher frequency details later in the training. Unlike with a convolutional architecture, the spatially-continuous nature of the neural field means we can implement multi-resolution training using a single network. We simply upsample the grid of directions, $\mathbf{D}_i$, and increase the resolution of the target equirectangular images to match this resolution.

We used Weights and Biases [6] for experiment tracking, visualisations and hyper-parameter grid search. This yielded the best performance using a variational auto-decoder, SIREN with 5 layers each with 128 hidden features. We use an exponentially decaying learning rate starting at 10^{-5} and decreasing to 10^{-7} by the end of training and weight the KLD loss $\beta = 10^{-4}$ to be of similar magnitude to the reconstruction loss. Training RENI with a latent code dimension of $D = 27$, took around 15 hours on local Nvidia A40 48GB GPUs.

Model fitting At test time, the SIREN network is held static, and only latent codes are optimised in order to fit to an unseen image. We initialise the latent code to zeros, corresponding to the mean environment map (see Figure 1, left). This provides an unbiased initialisation in the absence of any prior information about the environment. We found performance on test images was improved by including a cosine similarity loss $\mathcal{L}_{Cosine}$ on the RGB colour vectors [32] weighted using the same $\sin(\theta(\mathbf{d}))$ weight as used in $\mathcal{L}_{\text{recon}}$. We also include a prior loss on the latent vector $\mathcal{L}_{\text{prior}} = \|\mathbf{Z}\|^2_{\text{Fro}}$. Our test time loss is therefore:

$$\mathcal{L}_{\text{Test}} = \mathcal{L}_{\text{recon}} + \rho\mathcal{L}_{\text{Cosine}} + \gamma\mathcal{L}_{\text{Prior}}. \tag{4}$$

We again use multi-resolution fitting with the same resolutions and number of epochs as used during training. A hyper-parameter grid search found the best performance when using $\rho = 10^{-2}$, $\gamma = 10^{-7}$, and an exponentially decaying learning rate starting at 10^{-2} and decreasing to 10^{-4}. Trained models and a PyTorch implementation will be made available.

4 Evaluation

Generalisation We begin by evaluating the generalisation performance of RENI when approximating unseen environments. We compare against SH and SG and explore how generalisation performance varies as a function of latent code dimension. We consider RGB spherical harmonics of order 2, 5, 6 and 9 equal to latent code dimension $D = 3 \times N$ for $N = 9, 36, 49, 100$. Since SG requires a dimensionality that is an multiple of 12, where D is not a multiple of 12, we bias in SGs favour by using the next multiple, i.e. $\lceil D/12 \rceil \cdot 12$. We used an open source implementation of per-pixel environment map fitting provided by [31] to fit our SG models. Both SH and SG were fit to the same sine-weighted and scaled $\log(\text{HDR})$ equirectangular images as

Table 1: The mean PSNR when fitting to the test set for increasing latent dimensions. Where an exact dimensionality comparison is not possible for SG the dimensionality used is shown in brackets.

D	RENI	SH	SG	
27	**26.97**	22.80	24.75	$(N = 36)$
108	**28.21**	22.70	26.36	
147	**28.42**	23.00	26.07	$(N = 156)$
300	**28.72**	22.95	27.72	

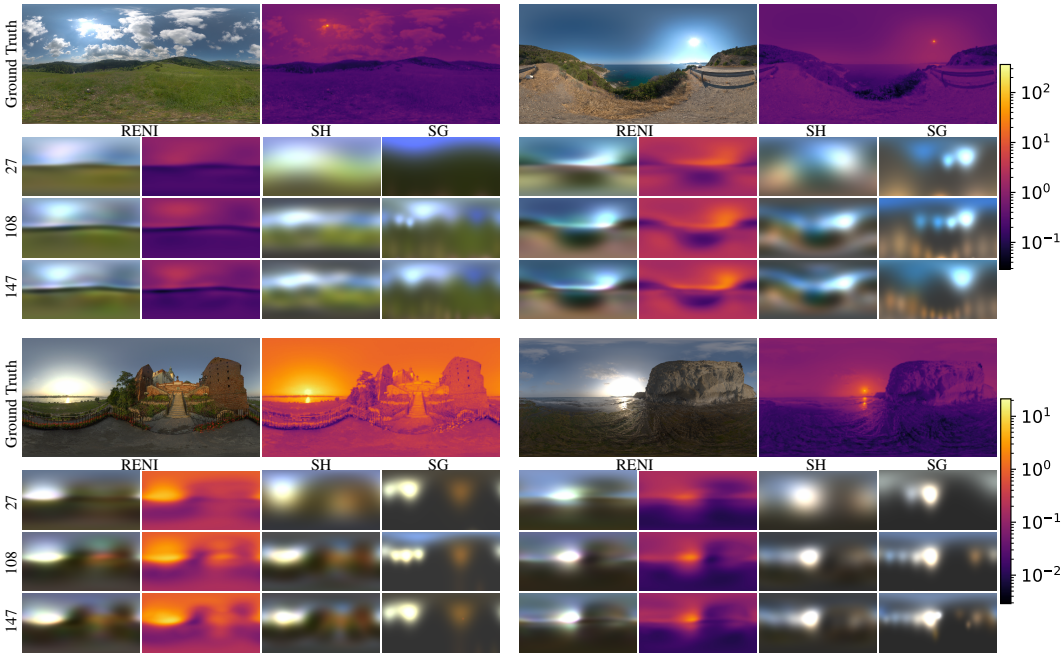

Figure 3: Generalisation to unseen images with latent code dimensions, $D = 3N$ for $N = 9, 36, 49$ and for SH of equal dimensionality (orders 2, 5, and 6). SG results are with dimensionality $D = 36, 108, 156$. Heat maps with log-scale colour bars for ground truth and RENI are also shown.

RENI. As shown in Figure 3, RENI can capture higher frequency detail than both SH and SG, is less dominated by high value pixels and can reproduce accurate HDR values. A comparison of the mean PSNR across the test set images for increasing latent code dimensionality is shown in Table 1.

Latent Space Interpolation As shown in Figure 4, linear interpolations between codes result in smooth transitions between images and plausible natural environments for all intermediate latent codes showing how RENI encodes a meaningful internal representation of natural illumination.

Environment Completion Any picture of a natural scene contains cues about the surrounding environment that the scene was captured in, such as likely sun locations and possible environmental content. If RENI is only provided with a small portion of the complete environment map in its loss at test time, RENI can hallucinate plausible completions of the environment. As shown in Figure 5, RENI makes sensible estimations about the possible colours and shapes of land and sky and often predicts quite accurate sun locations despite the sun being outside the image crop.

Inverse Rendering To test the performance of RENI in an inverse rendering pipeline, we implemented a normalised Blinn-Phong environment map shader in PyTorch3D, enabling fully differentiable rendering. We render a 3D object with fixed geometry, pose, camera and material parameters such that only lighting in the scene is unknown. Optimising only latent codes, we minimise the same losses as used in $\mathcal{L}_{Test}$, without the $\sin(\theta(\mathbf{d}))$ weighting needed for equirectangular images, between a rendering using a ground truth environment map and a one using the output of RENI. This was tested for incremental increases in weighting of the Blinn-Phong specular term (K_s), from $K_s = 0$ to $K_s = 1.0$ in steps of 0.2. A normalisation factor $\alpha = (n+2)/(4\pi(2 - e(\frac{-n}{2})))$ [21], was applied to K_s to get a steady transition from diffuse to specular. We achieved the best performance using $\alpha = 10^3$, $\gamma = 10^{-4}$, and an exponentially decaying learning rate starting at 10^{-2} and decreasing to 10^{-4} over 2,400 epochs. We use an environment map resolution of $H = 64$ and render the object with a resolution of 128^2. In Figure 6 we compare against SH environment maps computed in closed form using linear least squares for increasing specularity. As expected, SH performs well at producing accurate renders of diffuse objects. However, the environment maps produced are unnatural, and as the specular term increases in weight, SH degrades in performance significantly compared to RENI.

Figure 4: Interpolation results for RENI with latent code dimension of $D = 108$. Rows 1 and 2 show interpolations between two random latent codes, and row 3 shows an interpolation between two training images with the ground-truth images shown.

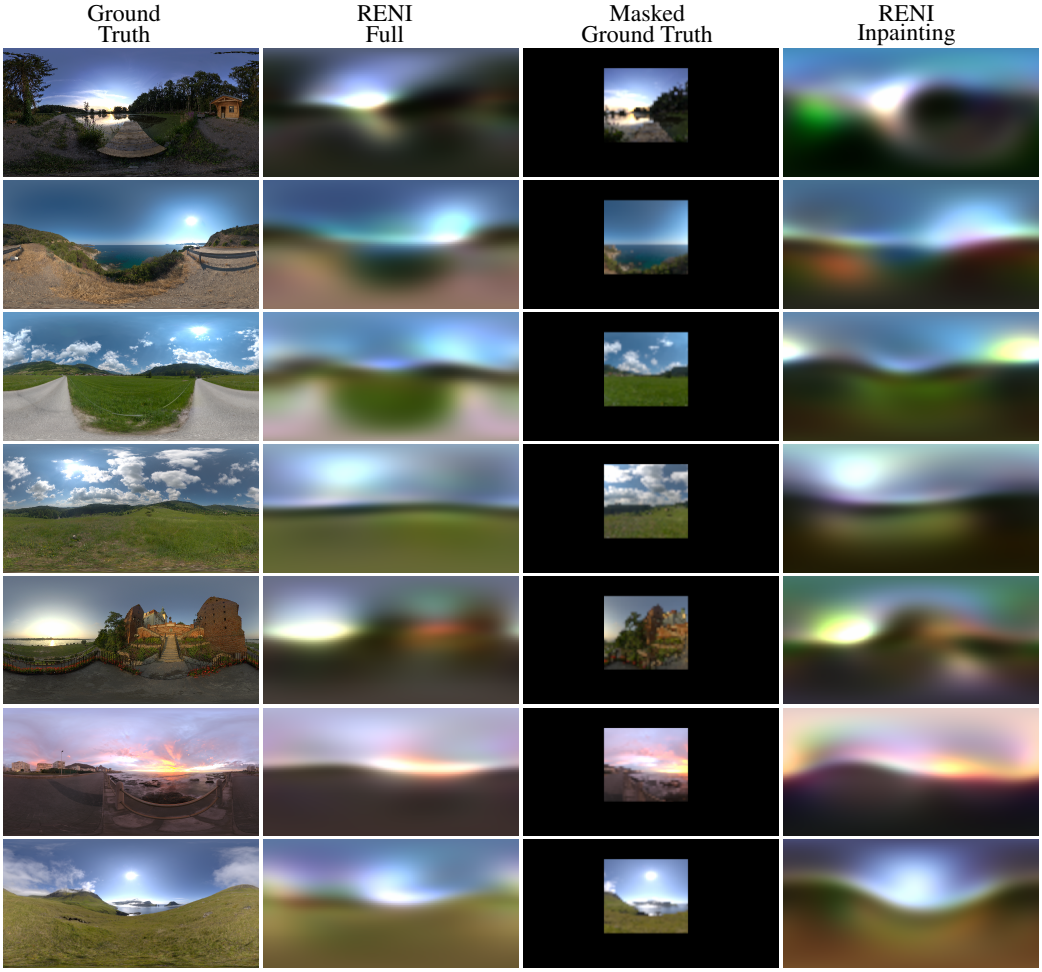

Figure 5: Col. 2 shows the output of optimising latent codes on full ground-truth (Col. 1). When trained on a masked ground-truth (Col. 3) RENI predicts plausible continuations for the environment and make accurate estimations of sun locations (Col. 4). Results from a $D = 108$ model.

5 Discussion and Conclusion

We introduced rotation-equivariant spherical neural fields and used them to create RENI, a natural illumination prior. Demonstrating how random samples from RENI always produce plausible illumination maps and its use for environment completion and inverse rendering. There are many exciting avenues for future research, for example, implementing RENI in larger inverse rendering pipelines where it could be a simple drop-in replacement for SH and using RENI for LDR to HDR

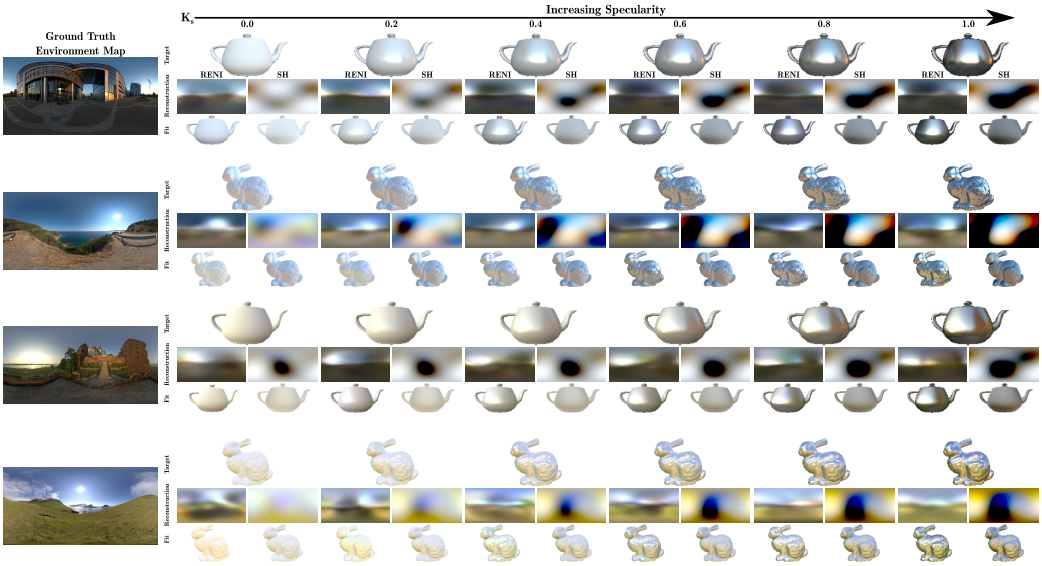

Figure 6: Reconstruction results in an inverse rendering task. The specular Blinn-Phong term K_s increases from left to right in steps of 0.2. Both RENI and SH have a dimensionality of $D = 27$. SH performs well in the purely diffuse case, but the resulting environment map reconstruction is unnatural, and as K_s increases RENI performs significantly better.

image reconstruction. Furthermore, our spherical neural field is the initial step towards a Generative Adversarial Network [20] (GAN) for spherical signals and could be used as the generator alongside a Spherical CNN [14] for the discriminator in a GAN framework.

Limitations Using the Gram matrix for the invariant representation, which is $O(n^2)$, limits the size of the latent code you can realistically use. This could be addressed through the use of the invariant layer proposed by [15] to reduce this to $O(n)$. The very highest frequency detail is still not resolved even when using larger latent code sizes; scaling up with a larger dataset and network size is likely to address this.

Because RENI has a prior for natural illumination, unlike SH, RENI's performance decreases when fit to indoor scenes. Interestingly, RENI still generalises to indoor scenes quite well; see supplementary material for examples of RENI fitting to indoor scenes.

There are complex interactions between illumination, geometry and texture priors in human vision. For example, the Hollow Face Illusion [26] arises from face geometry priors overriding the lighting from above illumination prior; while in the Bas relief ambiguity [4], geometric priors cause incorrect lighting estimation. Our model discounts these interactions, learning an illumination prior independently but not its interactions with other cues.

We strictly allow only $SO(2)$ equivariance. However, considering typical camera coordinate systems, the up axis will sometimes not align with gravity when the camera is pointed up or down. For inverse problems, this would mean the gravity vector would need to be explicitly estimated (an accelerometer would resolve this). Alternatively, we could build our model with full $SO(3)$ equivariance but then learn a prior over the space of camera poses relative to gravity.

Broader Impact The illumination prior presented here has the potential to improve the performance of inverse rendering pipelines, of which there are both positive and negative downstream use cases that should be considered; for example, the generation of highly realistic 3D models of a person's likeness without their consent. Advances in inverse and neural rendering could potentially lead to employment loss in the generation of 3D assets. However, we are optimistic that the democratisation of 3D model generation will be a value-generating technology for the large majority. The illumination prior might be biased since most of the available and used HDR images were captured in Europe.

References

[1] Whitemagus 3D. *HDRIs • Whitemagus 3D Models*. en-US. URL: http://www.whitemagus3dmodels.com (visited on 05/17/2022).

[2] Matan Atzmon and Yaron Lipman. *SAL: Sign Agnostic Learning of Shapes from Raw Data*. _eprint: 1911.10414. 2020.

[3] R. Basri and D.W. Jacobs. "Lambertian reflectance and linear subspaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.2 (2003), pp. 218–233. DOI: 10.1109/TPAMI.2003.1177153.

[4] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. "The Bas-Relief Ambiguity". In: *International Journal of Computer Vision* 35.1 (Nov. 1999), pp. 33–44. ISSN: 1573-1405. DOI: 10.1023/A:1008154927611. URL: https://doi.org/10.1023/A:1008154927611.

[5] Sai Bi et al. *Neural Reflectance Fields for Appearance Acquisition*. _eprint: 2008.03824. 2020.

[6] Lukas Biewald. *Experiment Tracking with Weights and Biases*. 2020. URL: https://www.wandb.com/.

[7] Mark Boss et al. *NeRD: Neural Reflectance Decomposition from Image Collections*. _eprint: 2012.03918. 2021.

[8] Mark Boss et al. *Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition*. _eprint: 2110.14373. 2021.

[9] Michael M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. _eprint: 2104.13478. 2021.

[10] Eric R. Chan et al. *pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis*. _eprint: 2012.00926. 2021.

[11] Boyuan Chen et al. *Full-Body Visual Self-Modeling of Robot Morphologies*. _eprint: 2111.06389. 2021.

[12] Zhiqin Chen and Hao Zhang. *Learning Implicit Fields for Generative Shape Modeling*. _eprint: 1812.02822. 2019.

[13] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. *Neural Unsigned Distance Fields for Implicit Function Learning*. _eprint: 2010.13938. 2020.

[14] Taco S. Cohen et al. "Spherical CNNs". In: *arXiv:1801.10130 [cs, stat]* (Feb. 2018). arXiv: 1801.10130. URL: http://arxiv.org/abs/1801.10130 (visited on 01/05/2022).

[15] Congyue Deng et al. "Vector Neurons: A General Framework for SO(3)-Equivariant Networks". In: (Apr. 2021). _eprint: 2104.12229.

[16] Ron O. Dror, Alan S. Willsky, and Edward H. Adelson. "Statistical characterization of real-world illumination". In: *Journal of Vision* 4.9 (Sept. 2004). _eprint: https://arvojournals.org/arvo/content_public/journal/jov/933508/jov-4-9-11.pdf, pp. 11–11. ISSN: 1534-7362. DOI: 10.1167/4.9.11. URL: https://doi.org/10.1167/4.9.11.

[17] Bernhard Egger et al. "Occlusion-Aware 3D Morphable Models and an Illumination Prior for Face Image Analysis". In: *International Journal of Computer Vision* 126 (2018), pp. 1269–1287.

[18] G Eilertsen, J Kronander, G Denes, et al. "HDR image reconstruction from a single exposure using deep CNNs". In: *ACM transactions on* (2017). Publisher: dl.acm.org.

[19] GiantCowFilms. *HDRIs • GiantCowFilms*. en-US. URL: https://giantcowfilms.com/category/hdris/ (visited on 05/17/2022).

[20] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661.

[21] Yoshiharu Gotanda. "Physically Based Shading Models in Film and Game Production: Practical Implementation at tri-Ace". In: 2010.

[22] Robin Green. "Spherical harmonic lighting: The gritty details". In: *Archives of the game developers conference*. Vol. 56. 2003, p. 4.

[23] Amos Gropp et al. *Implicit Geometric Regularization for Learning Shapes*. _eprint: 2002.10099. 2020.

[24] Poly Haven. *HDRIs • Poly Haven*. URL: https://polyhaven.com/hdris/ (visited on 05/17/2022).

[25] hdrmaps. *HDRIs • hdrmaps*. en-US. URL: https://hdrmaps.com/ (visited on 05/17/2022).

[26] Harold Hill and Vicki Bruce. "Independent Effects of Lighting, Orientation, and Stereopsis on the Hollow-Face Illusion". In: *Perception* 22.8 (1993). _eprint: https://doi.org/10.1068/p220887, pp. 887–897. DOI: 10.1068/p220887. URL: https://doi.org/10.1068/p220887.

[27] iHDRi. *HDRIs • iHDRi*. URL: https://www.ihdri.com/hdri-skies-outdoor/ (visited on 05/17/2022).

[28] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[29] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[30] Yunzhu Li et al. *3D Neural Scene Representations for Visuomotor Control*. _eprint: 2107.04004. 2021.

[31] Zhengqin Li et al. "Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image". en. In: *arXiv:1905.02722 [cs]* (May 2019). arXiv: 1905.02722. URL: http://arxiv.org/abs/1905.02722 (visited on 02/16/2022).

[32] Demetris Marnerides et al. "ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content". In: *Computer Graphics Forum* 37 (2018).

[33] Lars Mescheder et al. *Occupancy Networks: Learning 3D Reconstruction in Function Space*. _eprint: 1812.03828. 2019.

[34] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. "All-Frequency Shadows Using Non-Linear Wavelet Lighting Approximation". In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. event-place: San Diego, California. New York, NY, USA: Association for Computing Machinery, 2003, pp. 376–381. ISBN: 1-58113-709-5. DOI: 10.1145/1201775.882280. URL: https://doi.org/10.1145/1201775.882280.

[35] Joseph Ortiz et al. "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception". en. In: *arXiv:2204.02296 [cs]* (May 2022). arXiv: 2204.02296. URL: http://arxiv.org/abs/2204.02296 (visited on 05/10/2022).

[36] Jeong Joon Park et al. *DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation*. _eprint: 1901.05103. 2019.

[37] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[38] Ravi Ramamoorthi and Pat Hanrahan. "An efficient representation for irradiance environment maps". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, Aug. 2001, pp. 497–500.

[39] Ravi Ramamoorthi and Pat Hanrahan. "Frequency Space Environment Map Rendering". In: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '02. event-place: San Antonio, Texas. New York, NY, USA: Association for Computing Machinery, 2002, pp. 517–526. ISBN: 1-58113-521-1. DOI: 10.1145/566570.566611. URL: https://doi.org/10.1145/566570.566611.

[40] Viktor Rudnev et al. *Neural Radiance Fields for Outdoor Scene Relighting*. _eprint: 2112.05140. 2021.

[41] Soumyadip Sengupta et al. *Neural Inverse Rendering of an Indoor Scene from a Single Image*. 2019. DOI: 10.48550/ARXIV.1901.02453. URL: https://arxiv.org/abs/1901.02453.

[42] Soumyadip Sengupta et al. *SfSNet: Learning Shape, Reflectance and Illuminance of Faces in the Wild*. _eprint: 1712.01261. 2018.

[43] Zhixin Shu et al. *Neural Face Editing with Intrinsic Image Disentangling*. _eprint: 1704.04131. 2017.

[44] Vincent Sitzmann et al. "Implicit neural representations with periodic activation functions". In: *Adv. Neural Inf. Process. Syst.* 33 (2020), pp. 7462–7473.

[45] HDRI Skies. *HDRIs • HDRI Skies*. en-US. URL: https://hdri-skies.com/ (visited on 05/17/2022).

[46] Peter-Pike Sloan, Jan Kautz, and John Snyder. "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments". en. In: (), p. 10.

[47] Shuran Song and Thomas Funkhouser. *Neural Illumination: Lighting Prediction for Indoor Environments*. _eprint: 1906.07370. 2019.

[48] Pratul P. Srinivasan et al. *NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis*. _eprint: 2012.03927. 2020.

[49] Alejandro Sztrajman et al. "High-Dynamic-Range Lighting Estimation From Face Portraits". In: *2020 International Conference on 3D Vision (3DV)*. 2020, pp. 355–363. DOI: 10.1109/3DV50981.2020.00045.

[50] Matthew Tancik et al. *Block-NeRF: Scalable Large Scene Neural View Synthesis*. _eprint: 2202.05263. 2022.

[51] Rhiannon Thomas, Marko Nardini, and Denis Mareschal. "Interactions between "light-from-above" and convexity priors in visual development". In: *Journal of Vision* 10.8 (July 2010). _eprint: https://arvojournals.org/arvo/content_public/journal/jov/933477/jov-10-8-6.pdf, pp. 6–6. ISSN: 1534-7362. DOI: 10.1167/10.8.6. URL: https://doi.org/10.1167/10.8.6.

[52] Yu-Ting Tsai and Zen-Chung Shih. "All-Frequency Precomputed Radiance Transfer using Spherical Radial Basis Functions and Clustered Tensor Approximation". en. In: (), p. 10.

[53] Jiaping Wang et al. "All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance". en. In: (), p. 10.

[54] Yang Wang et al. "Face Relighting from a Single Image under Arbitrary Unknown Lighting Conditions". In: *IEEE transactions on pattern analysis and machine intelligence* 31 (Nov. 2009), pp. 1968–84. DOI: 10.1109/TPAMI.2008.244.

[55] Zian Wang et al. "Learning Indoor Inverse Rendering with 3D Spatially-Varying Lighting". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2021.

[56] Yiheng Xie et al. "Neural Fields in Visual Computing and Beyond". In: (Nov. 2021). _eprint: 2111.11426.

[57] Ye Yu and William A. P. Smith. *Outdoor inverse rendering from a single image using multiview self-supervision*. _eprint: 2102.06591. 2021.

[58] Ye Yu et al. "Self-supervised Outdoor Scene Relighting". In: *European Conference on Computer Vision (ECCV)*. 2020.

[59] Amir Zadeh et al. *Variational Auto-Decoder: A Method for Neural Generative Modeling from Incomplete Data*. _eprint: 1903.00840. 2021.

[60] Kai Zhang et al. "PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting". In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 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? [Yes] See Section Broader Impact

(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] Please see supplementary material for the code which includes test set images and a README explaining how to setup and run a trained model. This is not yet release ready and will be refactored prior to being publicly released.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See page 6, paragraph 3 'Training Details' and page 7, paragraph 4 'Inverse Rendering'.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] No error bars due to deterministic results.

(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 page 6, paragraph 3 'Training Details'.

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 page 5, paragraph 3 'Training Data' and page 6, paragraph 4 'Generalisation'

(b) Did you mention the license of the assets? [Yes] See page 5, paragraph 3 'Training Data'

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] No original assets were created for this research.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See page 5, paragraph 3 'Training Data'

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See page 5, paragraph 3 'Training Data'

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] No crowdsourcing or human subjects were used in this research.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No crowdsourcing or human subjects were used in this research.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No crowdsourcing or human subjects were used in this research.